

XML API for Mitel 69xx Minet Phones Firmware 1.8.0

DECEMBER 2021

58016588-00

DEVELOPMENT GUIDE



The information conveyed in this document is confidential and proprietary to Mitel® and is intended solely for Mitel employees and members of Mitel's reseller channel who specifically have a need to know this information. If you are not a Mitel employee or a Mitel authorized PARTNER, you are not the intended recipient of this information. Please delete or return any related material. Mitel will enforce its right to protect its confidential and proprietary information and failure to comply with the foregoing may result in legal action against you or your company.

TABLE OF CONTENTS

1	Introduction	12
1.1	<i>MITEL XML API.....</i>	12
1.2	<i>Revision History</i>	12
1.2.1	VERSION 1.5.1 (6920 / 6930 / 6940).....	12
1.2.2	VERSION 1.8.0 (6920 / 6930 / 6940).....	13
2	XML and the Mitel 69xx Phones	14
2.1	<i>What is XML?</i>	14
2.2	<i>Functionality</i>	14
2.3	<i>How does it work?</i>	15
2.3.1	Phone initiated application	15
2.3.2	Server initiated application.....	15
2.4	<i>System Architecture</i>	15
2.4.1	Corporate applications	16
2.4.2	Telephony applications	16
2.5	<i>Development environment.....</i>	17
2.5.1	Typical software architecture	17
2.5.2	Web server	17
2.5.3	Scripts/Languages	17
2.5.4	XML Validation tools	18
2.6	<i>XML format.....</i>	18
2.7	<i>HTTP format.....</i>	18
2.8	<i>XML display control and keys.....</i>	19
2.8.1	Mitel 6920	19
2.8.2	Mitel 6930	21
2.8.3	Mitel 6940	21
3	Mitel IP Phone XML Objects	23
3.1	<i>TextMenu Object</i>	24
3.1.1	Implementation (6920/6930)	24
3.1.2	Implementation (6940)	24
3.1.3	XML Description.....	26
3.1.4	Dialing from a TextMenu	30
3.1.5	Examples	31
3.2	<i>IconMenu Object</i>	33
3.2.1	Implementation (6920)	34
3.2.2	Implementation (6930)	35

3.2.3	Implementation (6940)	36
3.2.4	XML Description	37
3.2.5	Examples	41
3.3	<i>TextScreen Object</i>	45
3.3.1	Implementation (6920/6930)	45
3.3.2	Implementation (6940)	46
3.3.3	XML Description	47
3.3.4	Examples	50
3.4	<i>FormattedTextScreen Object</i>	52
3.4.1	Implementation (6920/6930)	52
3.4.2	Implementation (6940)	53
3.4.3	XML Description	55
3.4.4	Examples	59
3.5	<i>ImageScreen Object</i>	62
3.5.1	Implementation (6920/6930)	62
3.5.2	Implementation (6940)	63
3.5.3	XML Description	64
3.5.4	Examples	67
3.6	<i>InputScreen Object – Single Input field</i>	68
3.6.1	Implementation (6920/6930)	68
3.6.2	Implementation (6940)	69
3.6.3	XML Description	70
3.6.4	Input Type: IP	73
3.6.5	Input Type: Number	75
3.6.6	Input Type: String or StringN	77
3.6.7	Input Type: timeUS	79
3.6.8	Input Type: timeInt	81
3.6.9	Input Type: dateUS	83
3.6.10	Input Type: dateInt	85
3.7	<i>InputScreen Object – Multiple Input fields</i>	88
3.7.1	Implementation (6920/6930)	88
3.7.2	Implementation (6940)	89
3.7.3	XML Description	90
3.7.4	EXAMPLES	95
3.8	<i>PhoneExecute Object (all models)</i>	97
3.8.1	Implementation	97
3.8.2	XML Description	97
3.8.3	Examples	98
3.9	<i>PHONESOFTKEY OBJECT (6920/6930/6940)</i>	99

3.9.1	IMPLEMENTATION	99
3.9.2	XML DESCRIPTION	100
3.9.3	EXAMPLES	101
3.10	IMAGEMENU OBJECT (6920/6930/6940)	103
3.10.1	IMPLEMENTATION (6920)	103
3.10.2	IMPLEMENTATION (6930)	104
3.10.3	IMPLEMENTATION (6940)	105
3.10.4	XML DESCRIPTION	106
3.10.5	EXAMPLES	110
3.11	PHONESTATUS OBJECT	112
3.11.1	IMPLEMENTATION (6920/6930/6940)	112
3.11.2	XML DESCRIPTION	113
3.11.3	EXAMPLES	116
4	XML extensions	119
4.1	Customizable Softkeys ((6920/6930/6940))	119
4.2	Graphics (6920/6930/6940)	121
4.2.1	Images	121
4.2.2	Icons	122
4.2.3	Colors	126
4.3	Special attributes	127
4.3.1	Beep	127
4.3.2	Timeout	127
4.3.3	LockIn	127
4.3.4	triggerDestroyOnExit	127
4.3.5	CALLPROTECTION	127
4.4	TextMenu or IconMenu user selection (6920, 6930, 6940)	128
4.5	"Select" and "Dial" in the same TextMenu or IconMenu object	128
4.6	"Select", "Dial" and "Dial2" behavior in a TextMenu or IconMenu object	129
4.7	Refresh of an XML page	129
4.8	HTTP headers format for a phone HTTP GET	130
4.8.1	User-Agent	130
4.8.2	Accept-Language	131
4.9	Dial And DIALLINE URIS	132
4.10	LED CONTROL	132
4.10.1	IMPLEMENTATION (6920/6930)	133
4.10.2	IMPLEMENTATION (6940)	134
4.10.3	XML EXAMPLES	135
4.11	KEYPRESS EMULATION	137
4.12	CRASH AND CONFIGURATION FILEs RETRIEVAL	140

4.13	<i>Some development guidelines</i>	140
5	URL Format and Variables	141
5.1	<i>URL format</i>	141
5.2	<i>URL Variables</i>	141
5.3	<i>HTTPS</i>	141
5.4	<i>XML Objects Pushed to the Phone</i>	142
6	XML Configuration	144
6.1	<i>Configuring a Programmable Key using the MiVB ESM</i>	144
6.2	<i>Action URIs</i>	144
6.2.1	CONFIGURATION	145
6.2.2	ACTION URI DETAILED BEHAVIOR.....	145
6.2.3	APPLICATIONS	147
6.3	<i>Configuring ACTION URIs using the MiVB ESM</i>	148
7	Troubleshooting XML Applications	150
7.1	<i>Introduction</i>	150
7.2	<i>Troubleshooting tools</i>	151
8	Why XML Applications for an IP Phone?	152
8.1	<i>Media and information</i>	152
8.2	<i>Vertical applications</i>	152
8.2.1	Human Resources.....	152
8.2.2	Travel/Hotel	153
8.2.3	Health Care	153
8.2.4	Education	153
8.2.5	Law Enforcement	153
9	Sample XML applications	154
9.1	<i>Access from the Internet</i>	154
9.2	<i>Local server using XAMPP</i>	154
9.2.1	XAMPP installation.....	155
9.2.2	XML scripts installation.....	155
9.2.3	XAMPP start and stop	155
9.2.4	Test your installation	157
9.2.5	Troubleshooting Apache	157
9.3	<i>Applications</i>	157
9.3.1	Area code lookup (US)	157
9.3.2	Biorhythms	158
9.3.3	CNN News	158
9.3.4	Currency Converter	159
9.3.5	ESPN News	160

9.3.6	FOX News	161
9.3.7	Horoscope	161
9.3.8	IP Geolocation	162
9.3.9	Movies	162
9.3.10	Netflix.....	163
9.3.11	Stock Quotes	163
9.3.12	Today... ..	164
9.3.13	Local Weather.....	164
9.3.14	Yahtzee.....	165
9.3.15	Global menu	165
10	Appendix A: XSL Model	167
11	Appendix B: Object Oriented PHP Classes.....	168
11.1	<i>AastraIPPhoneExecute()</i>	168
11.2	<i>AastraIPPhoneFormattedTextScreen()</i>	168
11.3	<i>AastraIPPhoneIconMenu()</i>	170
11.4	<i>AastraIPPhoneImageScreen()</i>	172
11.5	<i>AastraIPPhoneInputScreen()</i> – Single Input field	174
11.6	<i>AastraIPPhoneInputScreen()</i> – Multiple Input fields	176
11.7	<i>AastraIPPhoneTextMenu()</i>	179
11.8	<i>AastraIPPhoneTextScreen()</i>	181
11.9	<i>AastraIPPhoneIMAGEMenu()</i>	183
11.10	<i>AastraIPPhonesoftkey()</i>	187
11.11	<i>AastraIPPhonesTATUS()</i>	188
11.12	<i>Examples provided with the PHP API</i>	188
12	Appendix D: Localized input character set	190
12.1	<i>English.....</i>	190
12.2	<i>French/Français</i>	190
12.3	<i>Spanish/Español</i>	190
12.4	<i>German/Deutsch</i>	191
12.5	<i>Italian/Italiano</i>	191
12.6	<i>Portuguese/Português.....</i>	192
13	APPENDIX D: CSV BASED DIRECTORY	193
13.1	<i>INTRODUCTION.....</i>	193
13.2	<i>PHONE COMPATIBILITY</i>	193
13.3	<i>INSTALLATION.....</i>	193
13.4	<i>XML KEY CONFIGURATION Directory</i>	193
13.5	<i>CSV FILE FORMAT</i>	194
13.6	<i>CONFIGURATION FILES</i>	194

13.6.1	SERVER.CONF	194
13.6.2	DIRECTORY.CONF	195
13.7	APPLICATION FILES	196
14	APPENDIX E: LDAP DIRECTORY	198
14.1	INTRODUCTION	198
14.2	INSTALLATION	199
14.3	NETWORK REQUIREMENTS.....	199
14.3.1	PHONE -> XML PROXY SERVER.....	199
14.3.2	XML PROXY SERVER -> LDAP-SERVER	200
14.4	ACTIVE DIRECTORY / LDAP-SERVER REQUIREMENTS.....	200
14.5	CONFIGURATION FILES.....	200
14.5.1	SERVER.CONF	200
14.5.2	LDAP_DIRECTORY.CONF	200
15	LAST PAGE OF THE DOCUMENT	204

TABLE OF FIGURES

Figure 1: Basic XML document.....	14
Figure 2: Mitel IP Phone acting as a client.....	15
Figure 3: Mitel IP Phone acting as a server	15
Figure 4: Access to an internal application	16
Figure 5: Access to an Internet application	16
Figure 6: Access to a telephony application	17
Figure 7: Typical software architecture of an XML application.....	17
Figure 8: XML conversion table	18
Figure 9: Mitel 6920 XML display and keys	20
Figure 10: Mitel 6930 XML display and keys	21
Figure 11: Mitel 6940 XML display.....	22
Figure 12: TextMenu Implementation on 6940	25
Figure 13: TextMenu touchLaunch tag usage	26
Figure 14: fontMono tag usage.....	26
Figure 15: icon positions in MenuItem	26
Figure 16: TextMenu Example 1 (6920)	31
Figure 17: TextMenu Example 2 (6940)	32
Figure 18: IconMenu cell layout.....	33
Figure 19: IconMenu cell rendering in portrait mode.....	33
Figure 20: IconMenu cell rendering in landscape mode	34
Figure 21: IconMenu screen modes “regular” and “extended”.....	34
Figure 22: 6920 - Layout 1 and 2 mode “regular”	34
Figure 23: 6930 - Layout 1 and 2 in “regular” mode.....	35
Figure 24: 6940 - Layout 1 and 2 in “regular” mode.....	36
Figure 25: IconMenu Example 1 (6930).....	41
Figure 26: IconMenu Example 2 (6930).....	43
Figure 27: IconMenu Example 2 (6940).....	44
Figure 28: TextScreen implementation on 6920	46
Figure 29: TextScreen implementation on 6940	47
Figure 30: TextScreen Example (6930).....	50
Figure 31: TextScreen Example (6930).....	51
Figure 32: FormattedTextScreen layout	52
Figure 33: FormattedTextScreen implementation on 6920.....	53
Figure 34: FormattedTextScreen layout on 6940	54
Figure 35: FormattedTextScreen implementation on 6940.....	54
Figure 36: FormattedTextScreen Example 1 (6930).....	60

Figure 37: FormattedTextScreen Example 2 (6930)	61
Figure 38: ImageScreen 6920 implementation (mode regular/extended and full screen)	63
Figure 39: ImageScreen 6940 implementation (mode regular/extended and full screen)	64
Figure 40: ImageScreen Example 1 (6920)	68
Figure 41: ImageScreen Example 2 (6940)	68
Figure 42: InputScreen implementation on 6930.....	69
Figure 43: InputScreen implementation on 6940.....	70
Figure 44: InputScreen “IP” Example (6930).....	74
Figure 45: InputScreen “IP” Example (6940).....	75
Figure 46: InputScreen “Number” Example (6930)	76
Figure 47: InputScreen “Number” Example (6940)	76
Figure 48: InputScreen “String” Example (6930).....	78
Figure 49: InputScreen “String” Example (6940).....	78
Figure 50: InputScreen “TimeUS” Example (6930)	80
Figure 51: InputScreen “TimeUS” Example (6940)	80
Figure 52: InputScreen “TimeInt” Example (6930)	82
Figure 53: InputScreen “TimeInt” Example (6940)	83
Figure 54: InputScreen “DateUS” Example (6930).....	84
Figure 55: InputScreen “DateUS” Example (6940).....	85
Figure 56: InputScreen “DateInt” Example (6930).....	86
Figure 57: InputScreen “DateInt” Example (6940).....	87
Figure 58: InputScreen implementation on 6930 “normal” mode	88
Figure 59: InputScreen implementation on 6930 “condensed” mode.....	89
Figure 60: InputScreen implementation on 6940 “normal” mode	90
Figure 61: InputScreen implementation on 6940 “condensed” mode.....	90
Figure 62: Multiple Input Fields Example (6930).....	96
Figure 63: Multiple Input Fields Example (6940).....	96
Figure 64: PhoneSoftkey Example (6940)	102
Figure 65: ImageMenu 6920 implementation (mode regular/extended and full screen).....	104
Figure 66: ImageMenu 6930 implementation (mode regular/extended and full screen).....	105
Figure 67: ImageMenu 6940 implementation (mode regular/extended and full screen).....	106
Figure 68: ImageMenu Example (6930).....	111
Figure 69: PhoneStatus Object (6940).....	112
Figure 70: PhoneStatus icons example.....	116
Figure 71: PhoneStatus icons example.....	117
Figure 72: PhoneStatus toaster example	118
Figure 73: XML colors (6920/6930/6940).....	127
Figure 74: XML LEDs on 6940	136
Figure 75: Figure 74: XML LEDs on 6930.....	137

Figure 76:HTTP flow for XML applications150

1 INTRODUCTION

1.1 MITEL XML API

Mitel 69xx phones support an XML API in firmware release 1.2.0. This XML API allows external applications to control the display of the phone.

The list of potential XML applications is endless, see chapter 8 for some examples of potential XML applications.

This document details the XML objects supported by the Mitel 69xx phones using firmware version 1.8.0 and how to implement them.

1.2 REVISION HISTORY

1.2.1 VERSION 1.5.1 (6920 / 6930 / 6940)

- New background color tag ("bgColor") for all UI XML objects.
- New "CallProtection" tag for all UI XML objects to avoid the XML screen to be destroyed on telephony events.
- New root tag attribute allowDTMF in some interactive UI objects to allow DTMF passthrough when the phone is in the connected mode and in the idle mode as well.
- New PhoneExecute commands-
 - Any supported uri
 - DialLine
 - Led control
 - Simulate keypress on the phone
 - Phone Reboot
 - Phone Fast Reboot
 - Delete the local configuration and reboot
 - Clear the callers list and missed calls indicator
 - Clear local Directory
 - Clear the redial list
 - Retrieve crash and configuration file
 - Launch the directory application
 - Launch the callers list application
 - Launch the redial list application
- New XML objects-
 - AastralIPPhoneStatus
 - AastralIPPhoneSoftkey
 - AastralIPPhoneImageMenu
- New Action URIs-

○ Successful registration	action uri registered
○ On-hook	action uri onhook
○ Off-hook	action uri offhook
○ Incoming call	action uri incoming
○ Outgoing call	action uri outgoing
○ Connect	action uri connected
○ Disconnect	action uri disconnected

1.2.2 VERSION 1.8.0 (6920 / 6930 / 6940)

Added support for \$\$REMOTENUMBER\$\$ URL variable in XML Action URIs.

2 XML AND THE MITEL 69XX PHONES

2.1 WHAT IS XML?

XML stands for **eXtensible Markup Language**. It is a markup language much like HTML. HTML was designed to display data and to focus on how data looks. XML was designed to describe data and to focus on what data is.

The following are characteristics of XML:

- XML tags are not predefined. You must define your own tags
- XML uses a Document Type Definition (DTD) or an XML Schema to describe the data
- XML with a DTD or XML Schema is designed to be self-descriptive
- XML is a W3C Standard Recommendation

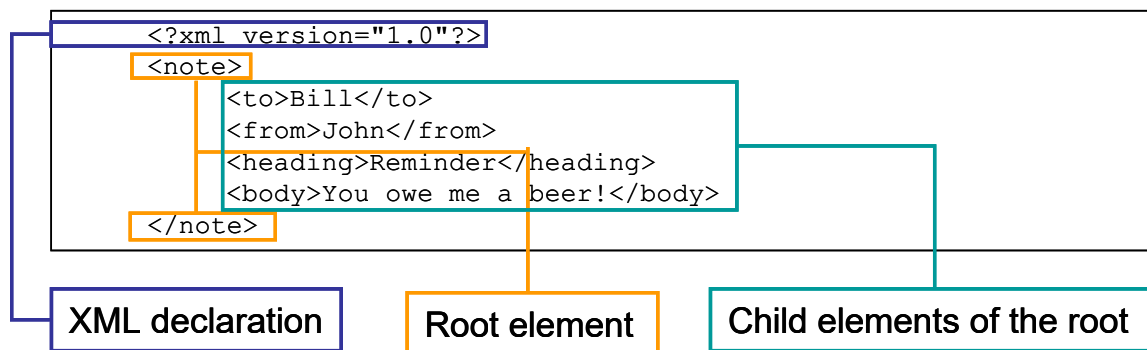


Figure 1: Basic XML document

More information available at <http://www.xml.com>

2.2 FUNCTIONALITY

The XML browser in Mitel 69xx IP phones allows developers to create custom services that they can use via the phone's keypad and display. These services include things like weather and traffic reports, contact information, company info, stock quotes, or custom call scripts.

With firmware release 1.8.0, the Mitel 69xx XML API supports 6 proprietary objects that allow the creation of powerful XML applications.

There are 2 types of XML objects:

- UI objects, XML objects which will use the display of the phone when they are received.
- Non UI objects, XML objects which have no direct impact on the current display.

The supported objects are:

- TextMenu object (UI)
- IconMenu (UI)
- TextScreen object (UI)
- FormattedTextScreen (UI)
- InputScreen object (UI)
- ImageScreen (UI)
- Execute object

Some of these objects also support customizable softkeys that are declared as an independent object.

The following sections describe the process of creating XML objects for the Mitel 69xx IP phones.

2.3 HOW DOES IT WORK?

Leveraging on the IP infrastructure, Mitel has decided to develop the browser capability on the phone using the HTTP transport protocol but as a direct support of HTML would not be suitable for the phone horsepower and limited display, the choice has been made to support only XML objects in the browser.

The Mitel 69xx IP phones support two types of applications:

- Phone-initiated
- Server-initiated

2.3.1 PHONE INITIATED APPLICATION

The phone issues an HTTP (or HTTPS) GET command to the Web server, waits for the answer, decodes and displays this answer as any Web browser such as Microsoft Internet Explorer or Firefox would do as a Web client.

This can be done through a phone programmable key and from the list of custom features (see chapter 6 for more details).

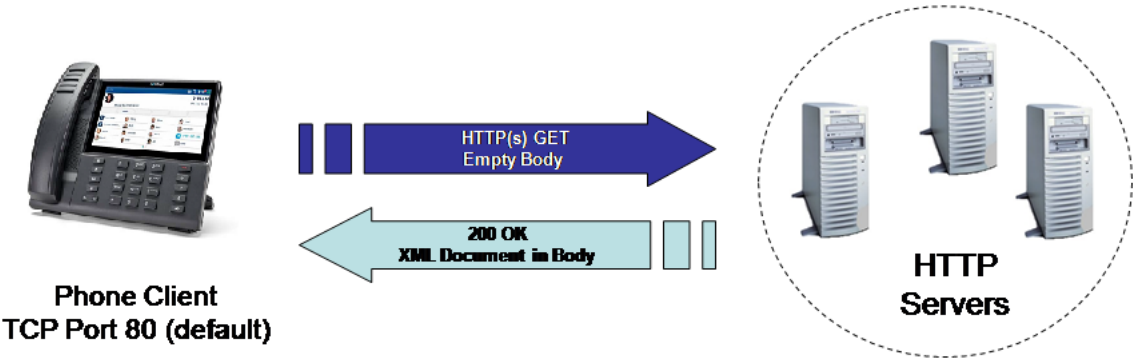


Figure 2: Mitel IP Phone acting as a client

When the phone performs an HTTP GET, it is a non blocking operation (phone will keep processing other events) as long as the GET is not requested by an action uri.

2.3.2 SERVER INITIATED APPLICATION

The other type of application would be more used for alerting as an application is pushing an XML object to the phone. The phone is now acting as a limited Web Server.

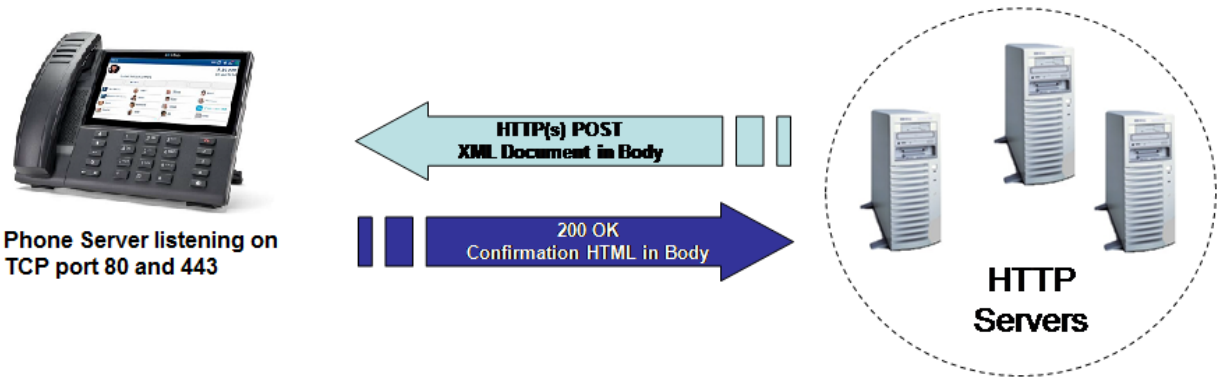


Figure 3: Mitel IP Phone acting as a server

2.4 SYSTEM ARCHITECTURE

The XML applications are hosted by one or multiple Web servers which will serve as a proxy to either other applications or to Internet Web servers.

2.4.1 CORPORATE APPLICATIONS

The following figure details the architecture to allow Mitel IP Phones to access an internal application. The application hosted by the Web server translates the phone requests to a protocol specific to the target application and formats the answer as an XML object to be displayed on the phone.

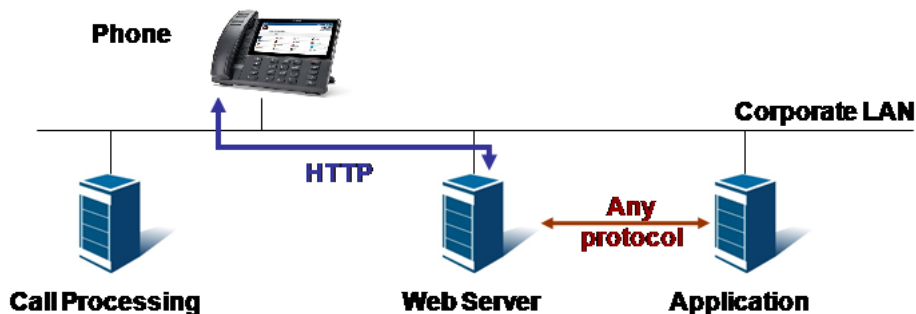


Figure 4: Access to an internal application

The following figure details the architecture of an XML application that would retrieve data from the internet such as a real-time stock-quote service.



Note: for certain Web applications that are not real time, the Internet content can be cached on the XML web server for a faster access.

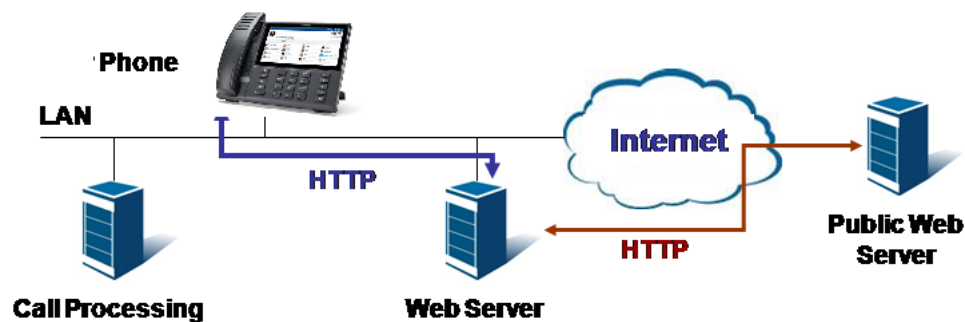


Figure 5: Access to an Internet application

2.4.2 TELEPHONY APPLICATIONS

The following figure details the architecture of an XML application that would provide more telephony features at the phone level. The application could for instance

- control a conference from the phone

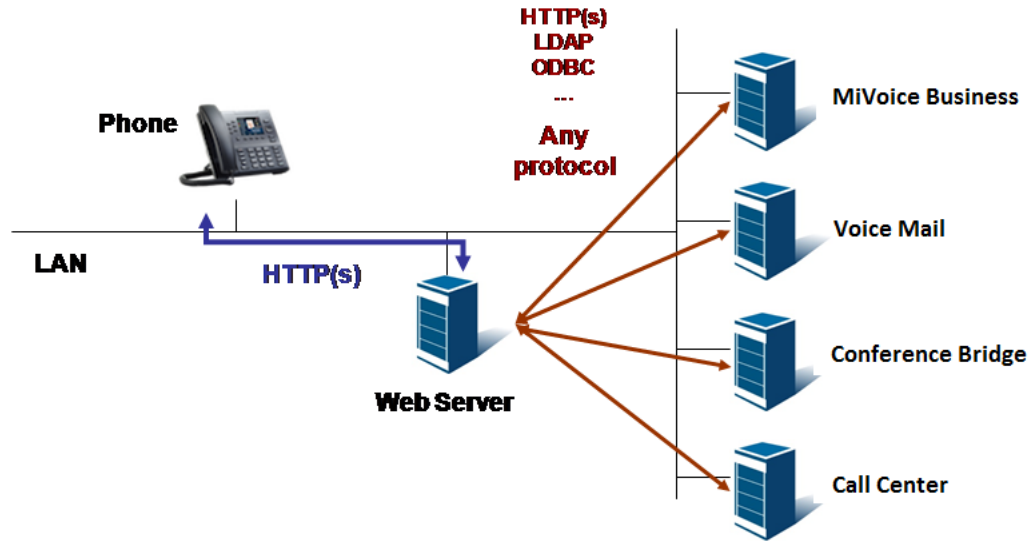


Figure 6: Access to a telephony application

2.5 DEVELOPMENT ENVIRONMENT

2.5.1 TYPICAL SOFTWARE ARCHITECTURE

The following diagram details the typical architecture of an XML application.

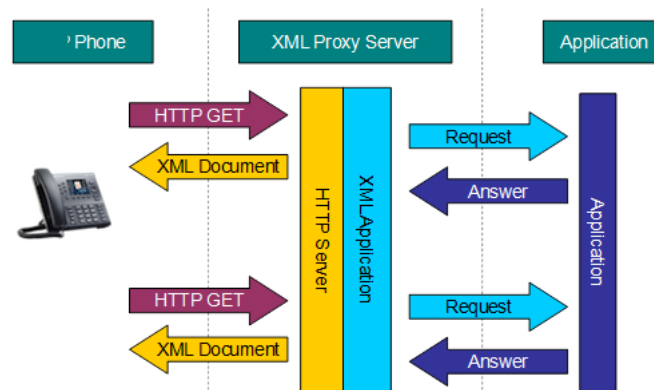


Figure 7: Typical software architecture of an XML application

The XML application is in fact translating requests from the phone to the protocol used by the external application and formats the answer into an XML document that the Mitel 69xx phone can interpret.

2.5.2 WEB SERVER

There is no constraint in the choice of the Web Server software to be used for Mitel XML applications; the choice is more based on the tools needed for the development such as the script language, the corporate policy or even the cost of the platform.

The most common Web server applications supported are:

- Apache (<http://www.apache.org>) for Microsoft and Linux Operating systems
- Microsoft IIS for Microsoft Operating systems

2.5.3 SCRIPTS/LANGUAGES

As for the Web Server, there is no specific constraint on the tools to develop the applications. All the languages supported to develop a Web application are supported to develop XML applications. The most common are:

- Compiled languages: C, C++, C#...
- Scripting languages: VBscript, Perl, Python, PHP, asp...

2.5.4 XML VALIDATION TOOLS

A large number of tools are available to validate the XML document you will be sending to the phone, these tools use the XSD schema (provided at chapter 9) to check the syntax of the generated XML document.

Example of a Web based tool

- <http://tools.decisionsoft.com/schemaValidate/>

2.6 XML FORMAT

The text in the Mitel XML objects must be compliant with XML recommendations and special characters must be escape encoded:

Character	Name	Escape Sequence
&	Ampersand	&
"	Quote	"
'	Apostrophe	'
<	Left angle bracket	<
>	Right angle bracket	>

Figure 8: XML conversion table

To respect XML recommendations, the following header can be set at the beginning of the XML document,

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

Or

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

By default UTF-8 is used by the phone.

2.7 HTTP FORMAT

The HTTP message sent to the Mitel 69xx Phone must respect HTTP/1.1 and must include the following parameters in the header:

- Content-Length
- Content-Type

The other parameters of the HTTP header are optional such as charset.

Example

```

HTTP/1.1 200 OK
Date: Tue, 15 May 2007 14:24:33 GMT
Server: Apache/2.0.52 (CentOS)
X-Powered-By: PHP/4.3.11
Content-Length: 564
Connection: close
Content-type: text/xml; charset=ISO-8859-1

<?xml version="1.0" encoding="ISO-8859-1"?>
<AstralPPhoneInputScreen type="string">
<Title>Title</Title>
<Prompt>Enter value</Prompt>
<URL>http://myserver.com/script.php</URL>
<Parameter>value</Parameter>
<Default></Default>
</AstralPPhoneInputScreen>

```

HTTP Header

HTTP body

2.8 XML DISPLAY CONTROL AND KEYS

This chapter describes the available part of the display for each Mitel 69xx phones as well as the keys that are controlled by the XML objects.

2.8.1 MITEL 6920

The display and keys available for XML applications on a Mitel 6920 are:

- 4 softkeys (6 logical softkeys for XML applications)
- A graphical zone for the XML objects
- Left/Right arrow navigation keys
- Up/Down arrow navigation keys
- Select key

See chapter 3 for more detailed information on each XML object.

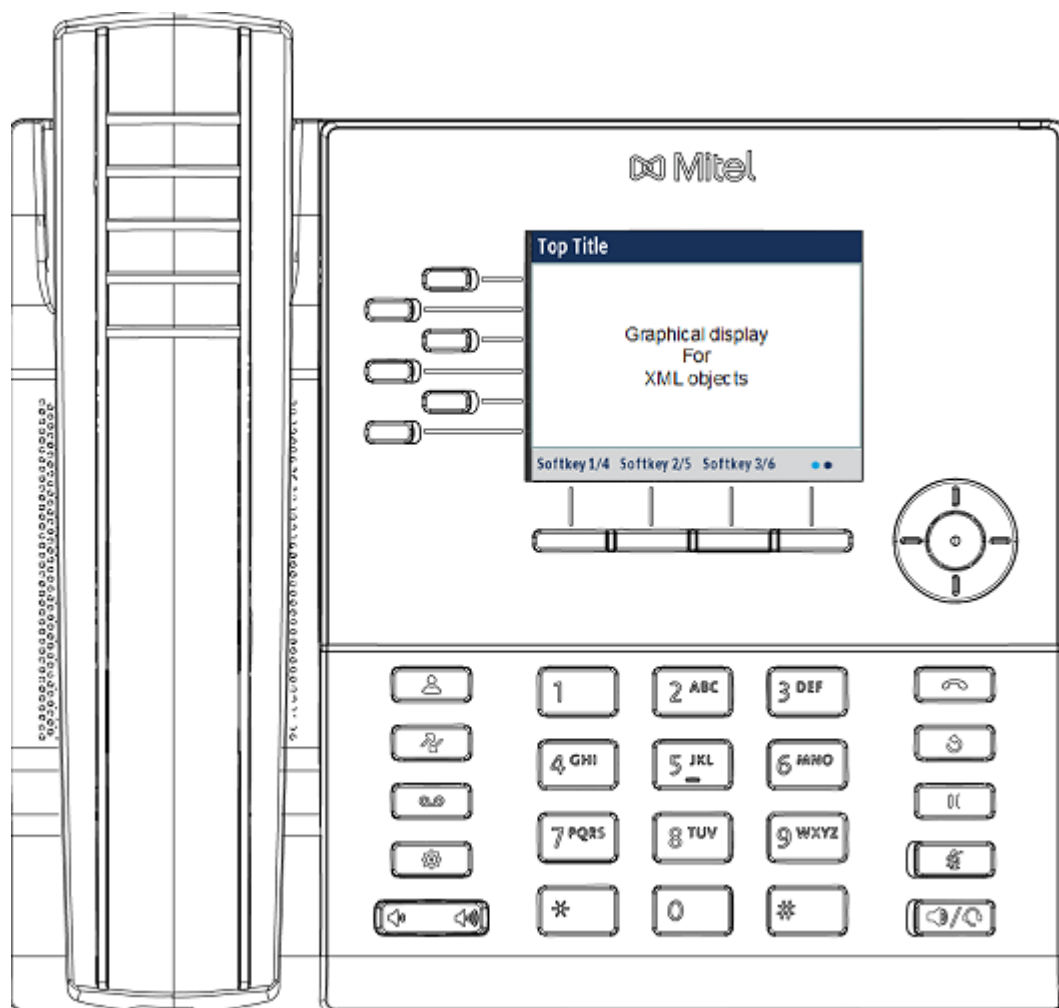


Figure 9: Mitel 6920 XML display and keys

2.8.2 MITEL 6930

The display and keys available for XML applications on a Mitel 6930 are:

- 5 softkeys (8 logical softkeys for XML applications)
- A graphical zone for the XML objects
- Left/Right arrow navigation keys
- Up/Down arrow navigation keys
- Select key

See chapter 3 for more detailed information on each XML object.

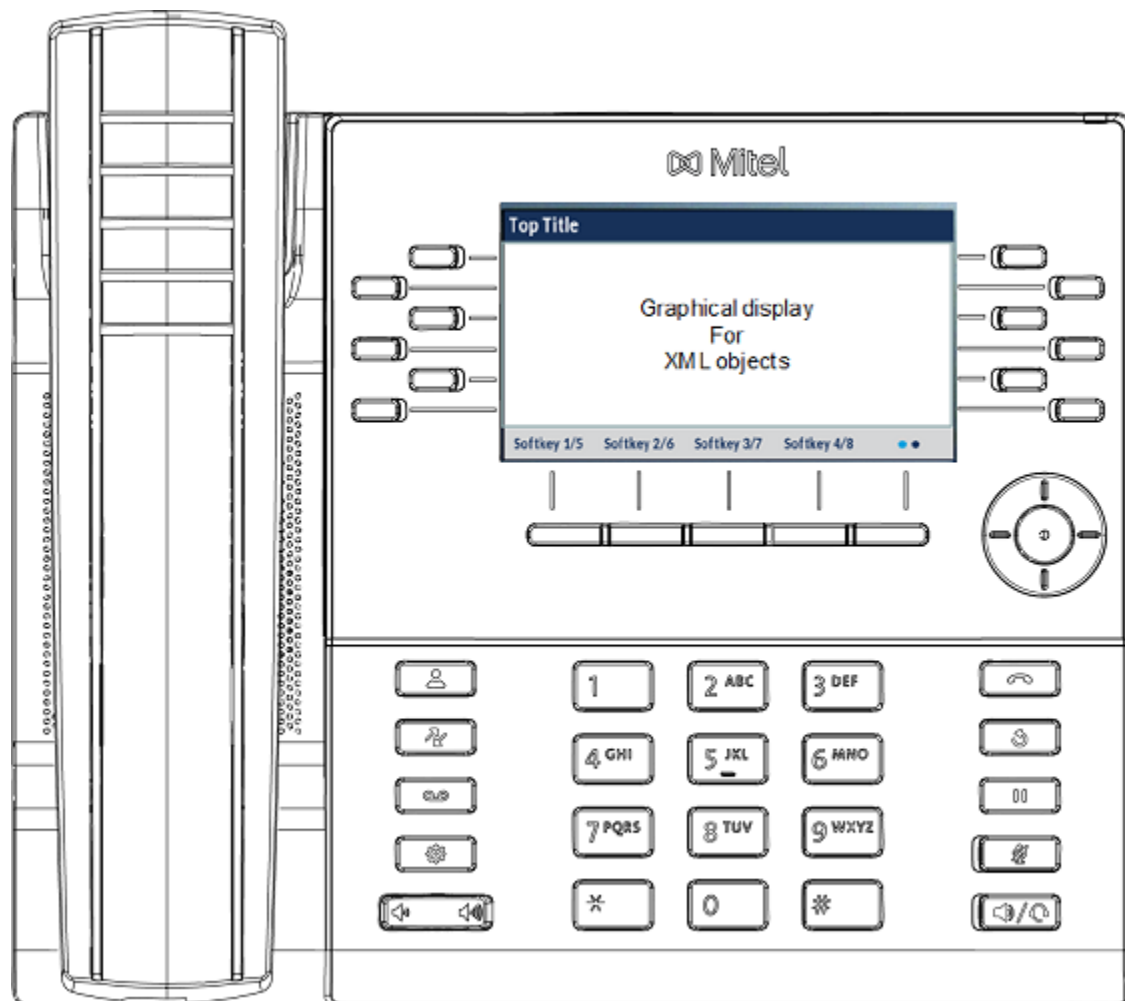


Figure 10: Mitel 6930 XML display and keys

2.8.3 MITEL 6940

The display and keys available for XML applications on a Mitel 6940 are:

- 6 softkeys (10 logical softkeys for XML applications)
- A graphical zone for the XML objects
- No navigation key as the 6940 is touch screen phone

See chapter 3 for more detailed information on each XML object.

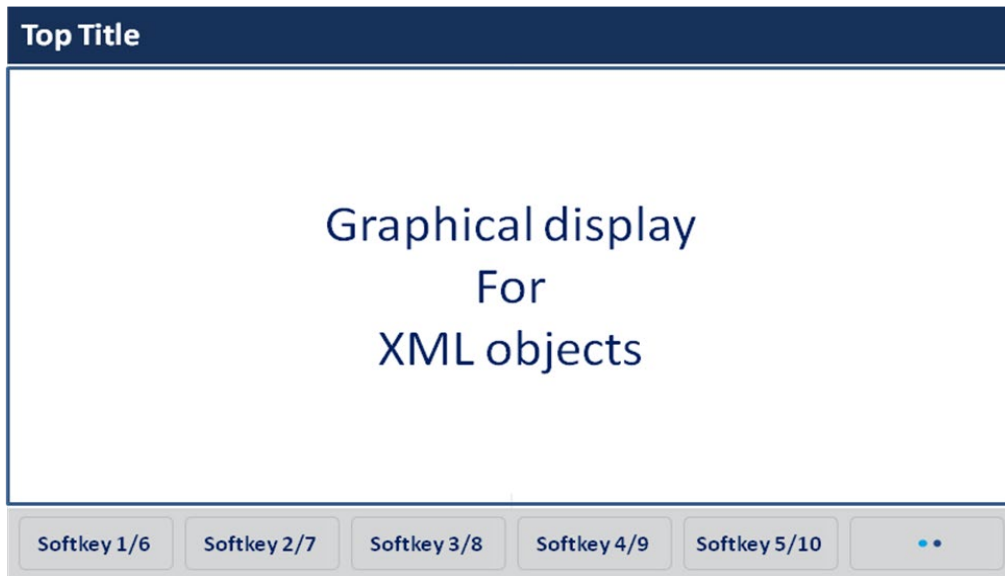


Figure 11: Mitel 6940 XML display

3 MITEL IP PHONE XML OBJECTS

This chapter details all the XML objects supported by the Mitel 69xx phones

In this chapter:

Softkey phones are Mitel phones with softkeys:

- Mitel 6920/6930 (graphical screen)

Touchscreen phones are Mitel phone(s) with a color graphical touch screen

- Mitel 6940

Notes:



- the size of an XML object cannot exceed 10000 bytes (10 kb).
 - per XML specifications, only one XML object is supported in the XML document sent to the phone.
-

3.1 TEXTMENU OBJECT

The TextMenu object allows developers to create a list of menu items on the IP phones. Go-to line support, arrow indicator, and scroll key support are built into these objects, along with the “**Select**” and “**Done**” softkeys for the phones supporting softkeys. The TextMenu object allows users to navigate the application, by linking HTTP requests to menu items.

For more details on the TextMenu behavior using the Selection tag or the Dial tag, please refer to section 4.5 and 4.6.

3.1.1 IMPLEMENTATION (6920/6930)

Object native interaction

- **Select** Executes the content of the URI field assigned to the selected MenuItem
- **Exit** Redisplays the previous XML object present in the phone browser.

The object is displayed on up to 6 lines, the title stays on the top of the display. The Up and Down arrow keys allow the user to browse the list up and down.

Line selected	Keys	
Item	Up and Down Arrow	Browse up or down
	Right Arrow	Select
	Left Arrow	Exit

Notes:



- the Left Arrow key interaction is disabled if the LockIn tag is set to “yes”.
- the Left Arrow key interaction can be modified using the cancelAction tag.
- If the LockIn tag is set to “yes” and the cancelAction tag is configured, pressing the Left Arrow key triggers the configured cancelAction.

Object default Softkeys (softkey phones)

Position	Label	Interaction	URIs
1	Select	Select: Executes the content of the URI field assigned to the selected MenuItem;	SoftKey:Select
6	Done	Exit: Redisplays the previous XML object present in the phone browser.	SoftKey:Exit

3.1.2 IMPLEMENTATION (6940)

The title is displayed at the top of the XML area and uses up to 2 lines. The menu items are displayed after the title area and up to 6 items (7 if no title) can be displayed at a time.

The item prompts are automatically wrapped on 2 lines if needed.

The scrolling is done by swiping the finger on the elements, pressing the left button of an item selects it.

The teal background on the button indicates that the item is selected.

The ➤ icon located on the right side of each item is the “Select” button and triggers the default URI of the object.

Object default Softkeys

Ten customizable softkeys are available for this object.

Position	Label	Interaction	URIs
1	Select	Select: Executes the content of the URI field assigned to the selected MenuItem;	SoftKey:Select
6	Done	Exit: Redisplays the previous XML object present in the phone browser.	SoftKey:Exit

The following figure details how the AstralPPhoneTextMenu is implemented on the Mitel 6940.

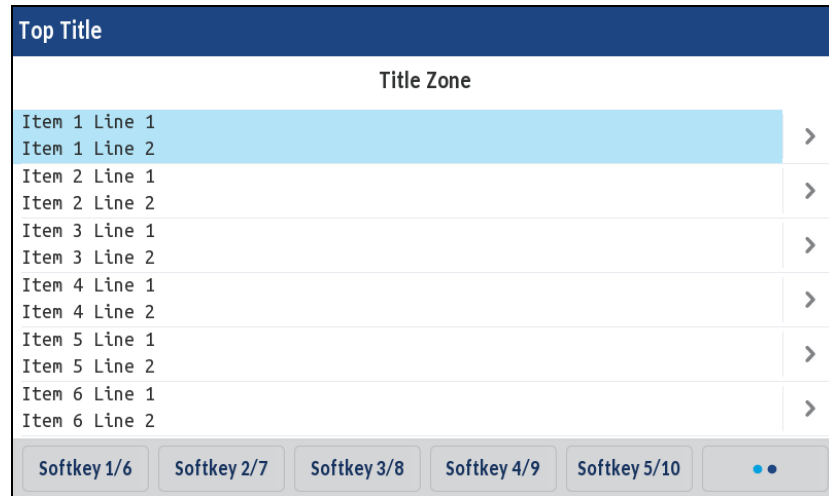


Figure 12: TextMenu Implementation on 6940

touchLaunch tag usage

The “touchLaunch” tag specific to the 6940 changes the overall behavior of the AstralPPhoneTextMenu object, when set to “yes”:

- The ➤ button is no longer available
- User can select and launch the uri configured in the item by just pressing anywhere on the item
- Selection background is no longer displayed
- The default “Select” softkey is no longer displayed as well as custom softkeys using “Softkey:Select”, “Softkey:Dial” and “Softkey:Dial2”.

This allows creating quick touch menus.



Figure 13: TextMenu touchLaunch tag usage

fontMono tag usage

The “fontMono” tag specific to the 6920, 6930 and 6940 changes the rendering of the TextMenu object, by allowing the usage of a proportional font instead of the default monotype font.

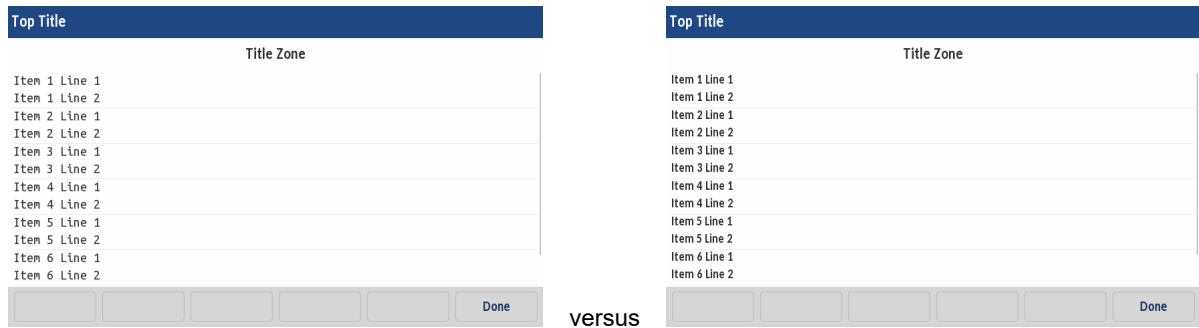


Figure 14: fontMono tag usage

icon tag usage

The “icon” tag allows to display an icon to the left of the MenuItem Label

The “iconr1” to “iconr4” tags specific to the 6920, 6930 and 6940 allow to display up to 4 icons to the right side of the MenuItem. If one or more right icons are configured in one item of the list, the phone clips the label for all the MenuItem in the list to the size of the MenuItem with the higher right icon.



Figure 15: icon positions in MenuItem

3.1.3 XML DESCRIPTION

“Red” tags indicate that the tag does not apply to all phones, when not supported the tags are just ignored.

```
<AstralPPhoneTextMenu
  defaultIndex = "some integer"
  destroyOnExit = "yes/no"
  Beep = "yes/no"
  Timeout = "some integer"
  bgColor = "white/black..."
  LockIn = "yes/no"
  CallProtection = "yes/no/notif"
  GoodbyeLockInURI = "some URI"
  cancelAction = "some URI"
  scrollConstrain = "yes/no"
  scrollUp = "Some URI"
  scrollDown = "Some URI"
  numberLaunch = "yes/no"
  touchLaunch = "yes/no"
  fontMono = "yes/no"
>
  <Title Color="white/black..."
  >Menu Title</Title>
```

```

<TopTitle      icon="icon index"
                Color="white/black..."
>Top Title</TopTitle>
<MenuItem base = "http://base/"
  icon = "icon index"
  iconr1 = "icon index"
  iconr2 = "icon index"
  iconr3 = "icon index"
  iconr4 = "icon index"
>
  <Prompt Color="white/black..."
    split="integer"
  >First Choice</Prompt>
  <URI>http://somepage.xml</URI>
  <Dial> Number to dial</Dial>
  <Selection>Selection</Selection>
</MenuItem>
<!--Additional Menu Items may be added (up to 30)-->
<IconList>
  <Icon index = "int">Icon:Iconname or HEX string</Icon>
  <!--As many as different icons used in the object -->
  <!--Up to 22 icons, index can be 1 to 21 -->
</IconList>
<!--Additional Softkey Items may be added (softkey phones)-->
</AstralIPPhoneTextMenu>

```

Notes:



- The number of items in a TextMenu object is limited to 30.
- You must declare at least one item in a TextMenu or the phone will generate a parsing error.

XML Document Objects

Document Object	Position	Type	Comments
AstralIPPhoneTextMenu	Root tag	Mandatory	Root object
defaultIndex	Root tag	Optional	Position of the cursor when the XML object is open. If not specified, the selection indicator is positioned on the first menu item.
destroyOnExit	Root tag	Optional	"yes/no" indicates if the object is kept or not in the phone browser after exit. If not specified, the object is kept in the browser.
cancelAction	Root tag	Optional	Defines the URI to be called when the user cancels the XML object.
Beep	Root tag	Optional	"yes" or "no" to indicate if a notification beep must be generated by the phone.
Timeout	Root tag	Optional	Overrides the default 45 seconds timeout of the UI XML object. A Timeout set to "0" will disable the timeout feature. See section 4.3.2 for more details

Document Object	Position	Type	Comments
bgColor	Root tag	Optional	Set the background color of the object. the possible values are detailed in chapter 4.2.3. If not specified, the default value is "white".
LockIn	Root tag	Optional	If set to "yes", the phone ignores all events that would cause the screen to exit without using the keys defined by the object. Default value is "no". See section 4.3.3 for more details.
CallProtection	Root tag	Optional	If set to "yes", the phone will not destroy the XML object being displayed on an incoming call. If set to "notif", the behavior is the same as "yes" but a notification is displayed on the screen providing the caller ID details.
GoodbyeLockInURI	Root tag	Optional	Valid only if LockIn="yes", this tag defines a URI to be called when the "Goodbye" key is pressed during a locked XML session. This URI overrides the native behavior of the "Goodbye" key which is to destroy the current XML object displayed.
scrollConstrain	Root tag	Optional	If set to "yes", the phone will not "wrap" the list: scrolling down the last item does not move the cursor to the first item. Default value is "no". <i>Not supported on 6940</i>
numberLaunch	Root tag	Optional	If set to "yes", the phone will allow the user to "launch" an item URI using the keypad (items 1-9 only). Default value is "no".
touchLaunch	Root tag	Optional	If set to "yes", selection is no longer needed user can press anywhere on the item to launch the uri. Default value is "no". <i>6940 only</i>
fontMono	Root tag	Optional	If set to "no", a proportional font will be used to display the object instead of a monotype font. Default value is "yes".
scrollUp	Root tag	Optional	This tag allows overriding the default behavior of the Up arrow key once the scrolling reaches an end.
scrollDown	Root tag	Optional	This tag allows overriding the default behavior of the Down arrow key once

Document Object	Position	Type	Comments
			the scrolling reaches an end.
Title	Body	Optional	Text to be used as title for the object
Color	Title tag	Optional	Label color, the possible values are detailed in chapter 4.2.3. If not specified, the default value is "white".
TopTitle	Body	Optional	Text to be used as top title for the object
Icon	TopTitle tag	Optional	Index of the icon to be used for the top title
Color	TopTitle tag	Optional	Label color, the possible values are detailed in chapter 4.2.3. If not specified, the default value is "white".
MenuItem	Body	Mandatory	Choice Item (up to 30 instances, minimum is 1 instance)
Base	MenuItem tag	Optional	The value of this attribute is prepended to the value in the URI tags.
icon	MenuItem tag	Optional	Index of the icon to be used for this menu entry.
iconr1	MenuItem tag	Optional	Index of the icon to be used for the first right icon.
iconr2	MenuItem tag	Optional	Index of the icon to be used for the second right icon
iconr3	MenuItem tag	Optional	Index of the icon to be used for the third right icon
iconr4	MenuItem tag	Optional	Index of the icon to be used for the fourth right icon
Prompt	MenuItem body	Mandatory	Label of the item
Color	Prompt tag	Optional	Label color, the possible values are detailed in chapter 4.2.3. If not specified, the default value is "white".
split	Prompt tag	Optional	This attribute tells where to split the prompt on 2 lines giving the position of the first character of the second line.
URI	MenuItem body	Mandatory	URI to be used if the user press "Select" with the cursor on this item

Document Object	Position	Type	Comments
Dial	Menuitem body	Optional	Defines what number will be dialed when an offhook action is performed on the phone or if the “Dial2” custom softkey is pressed
Selection	Menuitem body	Optional	This tag must be used in conjunction with custom softkeys. See Section 4.4 for details
IconList	Body	Optional	List of icon definitions
Icon	IconList body	Optional	Icon value, it can be “Icon:Iconname”, or the URL to a png file . See section 4.2.2 for more details.
Index	Icon tag	Optional	Index of the icon must be consistent with the ‘Icon’ used in the Menuitems. Possible values are 1 to 21.
SoftKey	Body	Optional	See section 4.1 for details



Note: the numberLaunch tag works for all TextMenu styles.

3.1.4 DIALING FROM A TEXTMENU

User has 4 ways to dial from a TextMenu using custom softkeys

- “Dial: PhoneNumber” in the menuitem URI tag, triggered by the “Select” softkey,
- “PhoneNumber” in the menuitem URI tag, triggered by the “Dial” custom softkey,
- “PhoneNumber” in the menuitem Dial tag, triggered by the “Dial2” custom softkey.
- “PhoneNumber” in the menuitem Selection tag, triggered by a custom softkey pointing to a script parsing the selection parameter and dialing “PhoneNumber” via a PhoneExecute command.

3.1.5 EXAMPLES

XML Example 1

```
<AstralPPhoneTextMenu>
  <Title>Phone Services</Title>
  <MenuItem base = "http://10.50.10.53/">
    <Prompt>Traffic Reports</Prompt>
    <URI> rss_to_xml.pl</URI>
  </MenuItem>
  <MenuItem>
    <Prompt>Employee List</Prompt>
    <URI>employees.xml</URI>
  </MenuItem>
  <MenuItem base = "">
    <Prompt>Weather</Prompt>
    <URI>http://10.50.10.52/weather.pl</URI>
  </MenuItem>
</AstralPPhoneTextMenu>
```

Resulting Screen

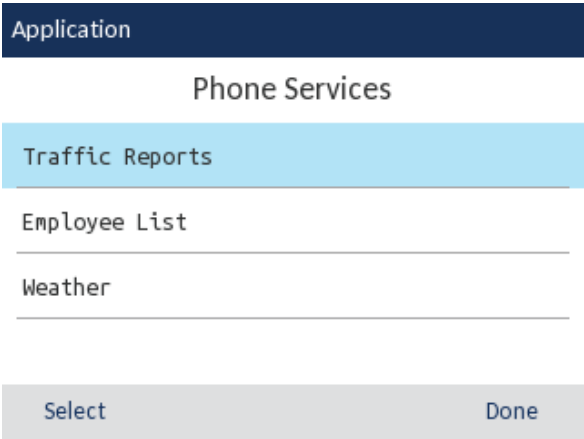


Figure 16: TextMenu Example 1 (6920)

XML Example 2 (touchscreen phone)

```
<AstralPPhoneTextMenu>
  <Title>Directory John Doe</Title>
  <MenuItem>
    <Prompt>Work Number</Prompt>
    <URI>http://myserver/myscript.php?a=zoom&n=1234</URI>
    <Dial>1234</Dial>
    <Selection>1234</Selection>
  </MenuItem>
  <MenuItem>
    <Prompt>Cell Number</Prompt>
    <URI>http://myserver/myscript.php?a=zoom&n=2345</URI>
    <Dial>2345</Dial>
    <Selection>2345</Selection>
  </MenuItem>
  <MenuItem>
    <Prompt>Home Number</Prompt>
    <URI>http://myserver/myscript.php?a=zoom&n=3456</URI>
    <Dial>3456</Dial>
    <Selection>3456</Selection>
  </MenuItem>
  <SoftKey index = "1">
    <Label>Delete</Label>
    <URI>http://myserver/myscript.php?a=delete</URI>
  </SoftKey>
  <SoftKey index = "2">
    <Label>Edit</Label>
    <URI>http://myserver/myscript.php?a=edit</URI>
  </SoftKey>
</AstralPPhoneTextMenu>
```

In this example, with item 1 selected:

- Pressing the > or on the item, will call <http://myserver/myscript.php?a=zoom&n=1234>
- Pressing the “Delete” softkey, will call <http://myserver/myscript.php?a=delete&selection=1234>
- Pressing the “Edit” softkey, will call <http://myserver/myscript.php?a=edit&selection=1234>

Resulting Screen

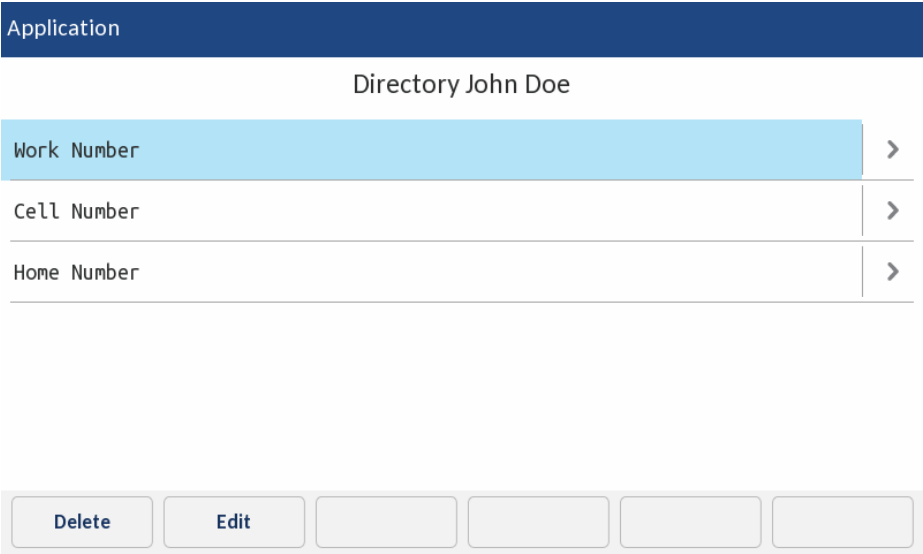


Figure 17: TextMenu Example 2 (6940)

3.2 ICONMENU OBJECT

The IconMenu object allows developers to create a list of menu items on the IP phones just like TextMenu but in this case menus are represented by an icon or a sequence of text lines.

The IconMenu object allows users to navigate the application, by linking HTTP requests to menu items.

For more details on the IconMenu behavior using the Selection tag or the Dial tag, please refer to section 4.5 and 4.6.

This object allows to configure up to 24 menu items which are represented as cells on the screen, the user can navigate to select the cell and then launch the attached URI or action.

The number, format and size of the visible cells depend on the selected layout and screen mode. If the total number of menu items is less than the number of cells per row, the cells are centered on the screen.

Icons can be scaled to fit the placeholder size, which allows a picture to be displayed in a cell.

Layouts

Two layouts are available for this object

- Layout “1” screen is split in 2 rows with either 4 (6920 - grid of 8 portrait cells) or 6 (6930/6940 - grid of 12 portrait cells) columns respectively.
- Layout “2” screen is split in 2 rows with 2 columns, forming a grid of 4 landscape cells.

Each cell can be composed of an icon and/or lines of text.

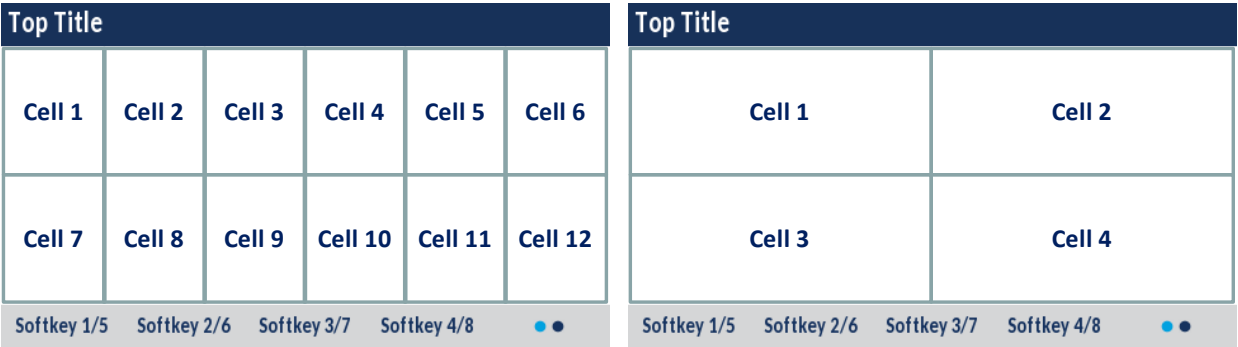


Figure 18: IconMenu cell layout

Portrait mode (layout=1)

If an icon is configured it is positioned at the top of the cell and any respective cell text is then positioned below it. However, if there is no icon configured, any respective cell text will be positioned at the top of the cell in its

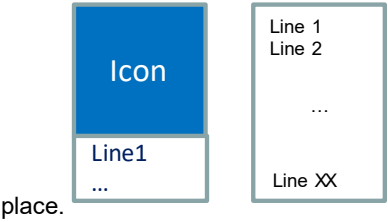


Figure 19: IconMenu cell rendering in portrait mode

Landscape mode (layout=2)

If an icon is configured it is positioned on the left hand side of the cell and any respective cell text is then positioned to the right of the icon. However, if there is no icon configured, any respective cell text will be positioned on the left hand side of the cell in its place.



Figure 20: IconMenu cell rendering in landscape mode

Screen modes

Two screen modes are available for this object:

- Regular Mode: Full screen, where both the top bar and bottom softkeys are displayed.
- Extended Mode: Same as regular mode but without the top bar displayed.

The screen mode does not affect the number of cells in the grid just their size.

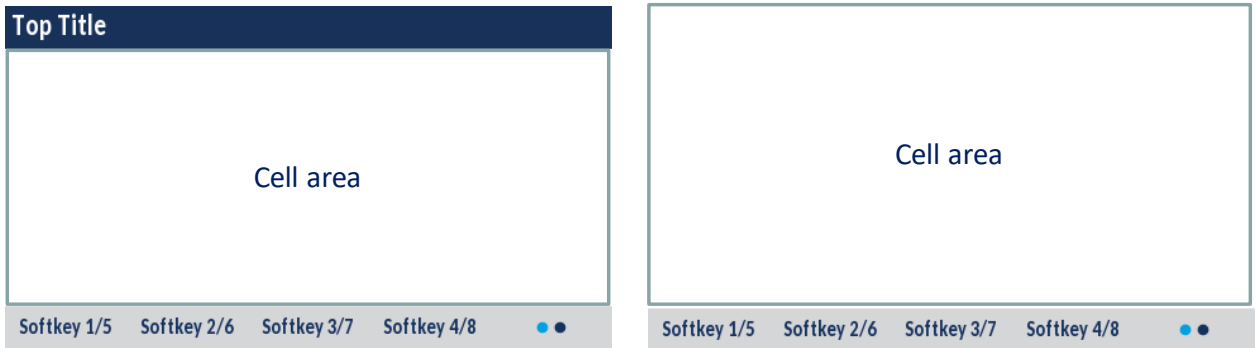


Figure 21: IconMenu screen modes “regular” and “extended”

Object native interaction

- **Select** Executes the content of the URI field assigned to the selected MenuItem
- **Exit** Redisplays the previous XML object present in the phone browser.

3.2.1 IMPLEMENTATION (6920)

The 6920 supports the 2 grid layouts.

- Layout 1 screen has 2 rows of 4 cells
- Layout 2 screen has 2 rows of 2 cells

Up to 24 cells can be created and a maximum of 8 cells per screen can be displayed at once. Any additional cells are placed underneath on the next virtual screen. The up and down scroll keys can be used to highlight/select and navigate up/down over the current screen icons or to navigate further to the next screen(s). Likewise, the left and right arrow keys can also be used to highlight/select and navigate left/right over the current screen icons.

Selected cell is highlighted.

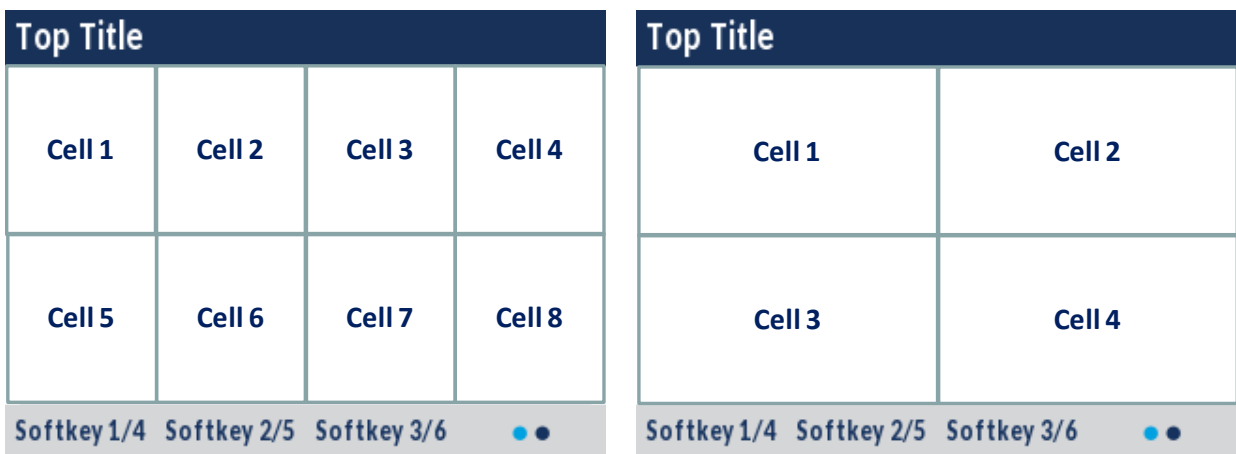


Figure 22: 6920 - Layout 1 and 2 mode “regular”

Navigation

- Up moves a row up and selects cell
- Down moves a row down and selects cell, if no cell configured below the cell selection moves to the last cell of the row
- Left moves a cell to the left and selects cell
- Right moves a cell to the right and selects cell
- OK is equivalent to select softkey

Component size

Layout	Mode	Icon size	Number of text lines
1	regular	60x60	1
1	extended	60x60	1
2	regular	66x66	3
2	extended	81x81	4

Object default Softkeys

Six customizable softkeys are available for this object.

Position	Label	Interaction	URIs
1	Select	Select: Executes the content of the URI field assigned to the selected MenuItem;	SoftKey:Select
4	Done	Exit: Redisplays the previous XML object present in the phone browser.	SoftKey:Exit

3.2.2 IMPLEMENTATION (6930)

The 6930 supports the 2 grid layouts.

- Layout 1 screen has 2 rows of 6 cells
- Layout 2 screen has 2 rows of 2 cells

Up to 24 cells can be created and a maximum of 12 cells per screen can be displayed at once. Any additional cells are placed underneath on the next virtual screen. The up and down scroll keys can be used to highlight/select and navigate up/down over the current screen icons or to navigate further to the next screen(s). Likewise, the left and right arrow keys can also be used to highlight/select and navigate left/right over the current screen icons.

Selected cell is highlighted.

Top Title						Top Title			
Cell 1	Cell 2	Cell 3	Cell 4	Cell 5	Cell 6	Cell 1	Cell 2		
Cell 7	Cell 8	Cell 9	Cell 10	Cell 11	Cell 12	Cell 3	Cell 4		
Softkey 1/5 Softkey 2/6 Softkey 3/7 Softkey 4/8 ● ●						Softkey 1/5 Softkey 2/6 Softkey 3/7 Softkey 4/8 ● ●			

Figure 23: 6930 - Layout 1 and 2 in “regular” mode

Navigation

- Up moves a row up and selects cell
- Down moves a row down and selects cell, if no cell configured below the cell selection moves to the last cell of the row
- Left moves a cell to the left and selects cell
- Right moves a cell to the right and selects cell
- OK is equivalent to select softkey

Component size

Layout	Mode	Icon size	Number of text lines
1	regular	61x61	1
1	extended	61x61	2
2	regular	78x78	4
2	extended	95x95	5

Object default Softkeys

Eight customizable softkeys are available for this object.

Position	Label	Interaction	URIs
1	Select	Select: Executes the content of the URI field assigned to the selected MenuItem;	SoftKey:Select
5	Done	Exit: Redisplays the previous XML object present in the phone browser.	SoftKey:Exit

3.2.3 IMPLEMENTATION (6940)

The 6940 supports the 2 grid layouts.

- Layout 1 screen has 2 rows of 6 cells
- Layout 2 has 2 rows of 2 cells

Up to 24 cells can be created and a maximum of 12 cells per screen can be displayed at once. Any additional cells are placed on the next virtual screen. Swipe left or right to access the next screen.

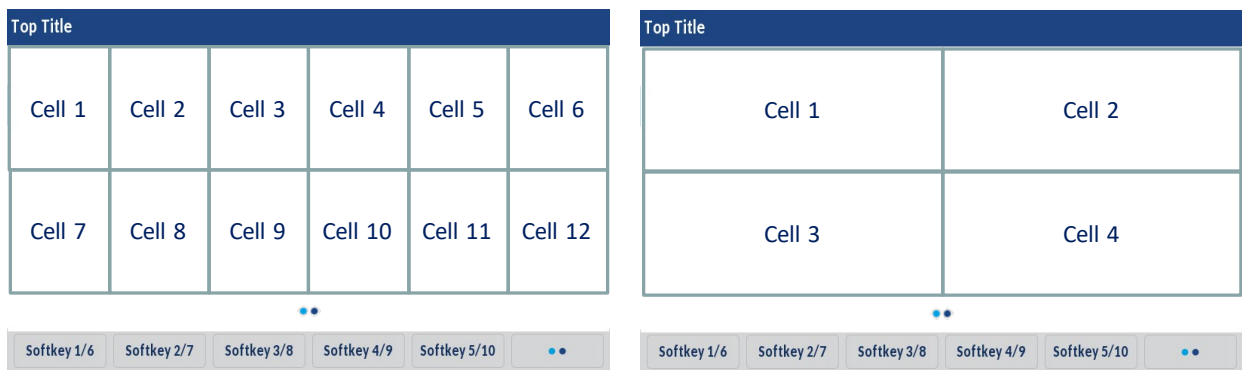


Figure 24: 6940 - Layout 1 and 2 in “regular” mode

Navigation

- Touch screen, clicking on a cell selects it and launches the associated URI
- Swipe changes page.

User cannot select and then launch the URI on the 6940, it is all in one operation, if user clicks on a cell, the URI attached to it is launched.

Custom softkeys using “Softkey:Select” or “Softkey:Dial” or “Softkey:Dial2” are not displayed on the 6940 as user cannot select a cell, it is comparable to the TextMenu with touchLaunch tag enabled.

Component size

Layout	Mode	Icon size	Number of text lines
1	regular	114x114	1
1	extended	114x114	2
2	regular	147x147	6
2	extended	171x171	7

Object default Softkeys

Ten customizable softkeys are available for this object.

Position	Label	Interaction	URIs
6	Done	Exit: Redisplays the previous XML object present in the phone browser.	SoftKey:Exit

3.2.4 XML DESCRIPTION

“Red” tags indicate that the tag does not apply to all phones, when not supported the tags are just ignored.

```
<AstralIPPhoneIconMenu
  defaultIndex = “some integer”
  destroyOnExit = “yes/no”
  Beep = “yes/no”
    Timeout = “some integer”
  bgColor=“white/black...”
  LockIn = “yes/no”
  CallProtection = “yes/no/notif”
  GoodbyeLockInURI = “some URI”
  fontMono = “yes/no”
  layout = “integer 1 or 2”
  mode = “regular/extended”
>
  <TopTitle      icon=“icon index”
                  Color=“white/black...”
  >Top Title</TopTitle>
  <Menuitem fontMono = “yes/no”>
    <iconName scaled=“yes/no”>IconName or location</iconName>
    <URI>http://somepage.xml</URI>
    <Dial>Number to dial</Dial>
    <Selection>Selection</Selection>
  <Line Align=“left/center/right” Color=“white/black...”
  >A line of static text</Line>
  <!--Additional Lines may be added -->
```

```

</MenuItem>
<!--Additional Menu Items may be added (up to 24)-->
<IconList>
    <Icon index = "int">Icon:Iconname or icon location</Icon>
    <!--As many as different icons used in the object -->
    <!--Up to 22 icons, index can be 1 to 21 -->
</IconList>
<!--Additional Softkey Items may be added -->
<AstralPPhoneIconMenu>

```

Notes:

- The number of items in a IconMenu object is limited to 24.
- You must declare at least one item in a IconMenu or the phone will generate a parsing error.
- "cancelAction" tag is not supported by this object as the left navigation key is used for screen navigation.
- It is not recommended to scale an icon up.

XML Document Objects

Document Object	Position	Type	Comments
AstralPPhoneIconMenu	Root tag	Mandatory	Root object
defaultIndex	Root tag	Optional	Position of the selected item when the XML object is open. If not specified, the arrow is positioned on the first menu item. On the 6940 the object will open at the page containing the selected item.
destroyOnExit	Root tag	Optional	"yes/no" indicates if the object is kept or not in the phone browser after exit. If not specified, the object is kept in the browser.
Beep	Root tag	Optional	"yes" or "no" to indicate if a notification beep must be generated by the phone.
Timeout	Root tag	Optional	Overrides the default 45 seconds timeout of the UI XML object. A Timeout set to "0" will disable the timeout feature. See section 4.3.2 for more details
bgColor	Root tag	Optional	Set the background color of the object. the possible values are detailed in chapter 4.2.3. If not specified, the default value is "white".
LockIn	Root tag	Optional	If set to "yes", the phone ignores all events that would cause the screen to exit without using the keys defined by

Document Object	Position	Type	Comments
			the object. Default value is “no”. See section 4.3.3 for more details.
CallProtection	Root tag	Optional	If set to “yes”, the phone will not destroy the XML object being displayed on an incoming call. If set to “notif”, the behavior is the same as “yes” but a notification is displayed on the screen providing the caller ID details.
GoodbyeLockInURI	Root tag	Optional	Valid only if LockIn=“yes”, this tag defines a URI to be called when the “Goodbye” key is pressed during a locked XML session. This URI overrides the native behavior of the “Goodbye” key which is to destroy the current XML object displayed.
fontMono	Root tag	Optional	If set to “no”, a proportional font will be used to display the object instead of a monotype font. Default value is “no”.
layout	Root tag	Optional	Object layout, “1” or “2”. Default value is “1”.
mode	Root tag	Optional	Object mode “regular” or “extended”. Default value is “regular”.
TopTitle	Body	Optional	Text to be used as top title for the object.
Icon	TopTitle tag	Optional	Index of the icon to be used for the top title.
Color	TopTitle tag	Optional	Label color, the possible values are detailed in chapter 4.2.3. If not specified, the default value is “white”.
MenuItem	Body	Mandatory	Choice Item (up to 24 instances, minimum is 1 instance)
fontMono	MenuItem tag	Optional	Indicates if a monotype font must be used in this menu item. If set, it overrides the value defined in the root tag.
iconName	MenuItem body	Optional	Name or location of the icon to be used for this menu entry
scaled	iconName tag	Optional	Indicates if the icon needs to be scaled to the size of the placeholder or not. Values can be “yes” or “no”. If not specified, the icon is not scaled.

Document Object	Position	Type	Comments
URI	Menuitem body	Mandatory	URI to be used if the user press "Select" with the cursor on this item
Dial	Menuitem body	Optional	Defines what number will be dialed when an offhook action is performed on the phone or if the "Dial2" custom softkey is pressed. <i>Ignored on 6940 as custom softkeys based on item selection are not supported.</i>
Line	Menuitem body	Optional	Defines the text to display
Align	Line tag	Optional	Text alignment, the possible values are "left", "right" and "center". If not specified, the default value is "center".
Color	Line tag	Optional	Text color, the possible values are detailed in chapter 4.2.3. If not specified, the default value is "white".
Selection	Menuitem body	Optional	This tag must be used in conjunction with custom softkeys. See Section 4.4 for details <i>Ignored on 6940 as custom softkeys based on selection not supported.</i>
IconList	Body	Optional	List of icon definitions
Icon	IconList body	Optional	Icon value, it can be "Icon:Iconname" or the URL to a png file. See section 4.2.2 for more details.
Index	Icon tag	Optional	Index of the icon must be consistent with the 'Icon' used in the TopTitle Possible values are 1 to 21.
SoftKey	Body	Optional	See section 4.1 for details <i>Softkeys using "Softkey:Select" or "Softkey:Dial" or "Softkey:Dial2" are ignored on the 6940</i>

3.2.5 EXAMPLES

XML Example 1

```
<AstralPPhoneIconMenu
destroyOnExit="yes"
defaultIndex = "2"
layout = "1"
mode="regular"
>
<TopTitle>IconMenu</TopTitle>
<MenuItem>
<Line>Menu 1</Line>
<iconName>http://192.168.0.135/6930-icon1-1.png</iconName>
<URI>http://myserver/myscript.php?menu=1</URI>
</MenuItem>
<MenuItem>
<Line>Menu 2</Line>
<iconName>http://192.168.0.135/6930-icon2-1.png</iconName>
<URI>http://myserver/myscript.php?menu=2</URI>
</MenuItem>
<MenuItem>
<Line>Menu 3</Line>
<iconName>http://192.168.0.135/6930-icon3-1.png</iconName>
<URI>http://myserver/myscript.php?menu=3</URI>
</MenuItem>
</AstralPPhoneIconMenu>
```

Resulting Screen (6930)

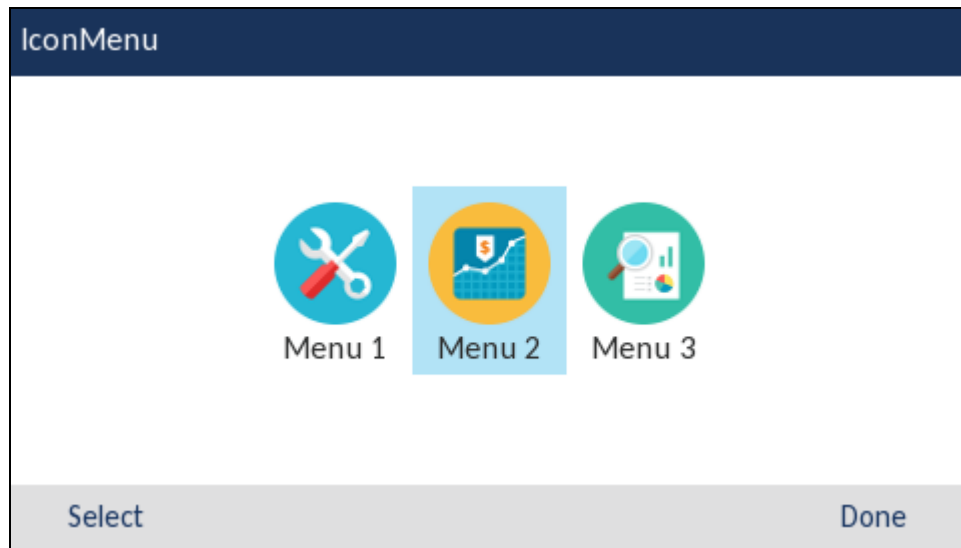


Figure 25: IconMenu Example 1 (6930)

XML Example 2

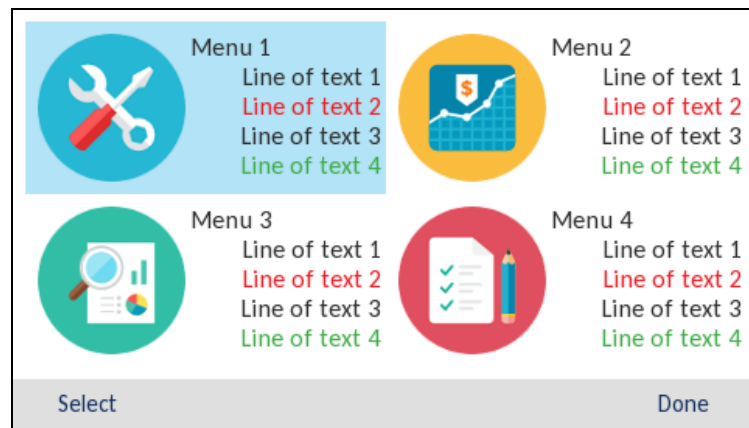
```
<AstralPPhoneIconMenu
destroyOnExit="yes"
defaultIndex = "1"
layout = "2"
mode="extended"
>
<TopTitle>IconMenu</TopTitle>
<MenuItem>
<Line Align="left">Menu 1</Line>
<Line Align="right">Line of text 1</Line>
<Line Align="right" Color="red">Line of text 2</Line>
<Line Align="right">Line of text 3</Line>
<Line Align="right" Color="green">Line of text 4</Line>
<Line Align="right">Line of text 5</Line>
<Line Align="right">Line of text 6</Line>
<Line Align="right">Line of text 7</Line>
<iconName>http://192.168.0.135/6930-icon1-2.png</iconName>
<URI>http://myserver/myscript.php?menu=1</URI>
</MenuItem>
<MenuItem>
<Line Align="left">Menu 2</Line>
<Line Align="right">Line of text 1</Line>
<Line Align="right" Color="red">Line of text 2</Line>
<Line Align="right">Line of text 3</Line>
<Line Align="right" Color="green">Line of text 4</Line>
<Line Align="right">Line of text 5</Line>
<Line Align="right">Line of text 6</Line>
<Line Align="right">Line of text 7</Line>
<iconName>http://192.168.0.135/6930-icon2-2.png</iconName>
<URI>http://myserver/myscript.php?menu=2</URI>
</MenuItem>
<MenuItem>
<Line Align="left">Menu 3</Line>
<Line Align="right">Line of text 1</Line>
<Line Align="right" Color="red">Line of text 2</Line>
<Line Align="right">Line of text 3</Line>
<Line Align="right" Color="green">Line of text 4</Line>
<Line Align="right">Line of text 5</Line>
<Line Align="right">Line of text 6</Line>
<Line Align="right">Line of text 7</Line>
<iconName>http://192.168.0.135/6930-icon3-2.png</iconName>
<URI>http://myserver/myscript.php?menu=3</URI>
</MenuItem>
<MenuItem>
<Line Align="left">Menu 4</Line>
<Line Align="right">Line of text 1</Line>
<Line Align="right" Color="red">Line of text 2</Line>
<Line Align="right">Line of text 3</Line>
<Line Align="right" Color="green">Line of text 4</Line>
<Line Align="right">Line of text 5</Line>
<Line Align="right">Line of text 6</Line>
<Line Align="right">Line of text 7</Line>
<iconName>http://192.168.0.135/6930-icon4-2.png</iconName>
<URI>http://myserver/myscript.php?menu=4</URI>
</MenuItem>
<MenuItem>
<Line Align="left">Menu 5</Line>
<Line Align="right">Line of text 1</Line>
<Line Align="right" Color="red">Line of text 2</Line>
<Line Align="right">Line of text 3</Line>
```

```

<Line Align="right" Color="green">Line of text 4</Line>
<Line Align="right">Line of text 5</Line>
<Line Align="right">Line of text 6</Line>
<Line Align="right">Line of text 7</Line>
<iconName>http://192.168.0.135/6930-icon5-2.png</iconName>
<URI>http://myserver/myscript.php?menu=5</URI>
</MenuItem>
<MenuItem>
<Line Align="left">Menu 6</Line>
<Line Align="right">Line of text 1</Line>
<Line Align="right" Color="red">Line of text 2</Line>
<Line Align="right">Line of text 3</Line>
<Line Align="right" Color="green">Line of text 4</Line>
<Line Align="right">Line of text 5</Line>
<Line Align="right">Line of text 6</Line>
<Line Align="right">Line of text 7</Line>
<iconName>http://192.168.0.135/6930-icon6-2.png</iconName>
<URI>http://myserver/myscript.php?menu=6</URI>
</MenuItem>
</AastraIPPhoneIconMenu>

```

Resulting Screens (6930)



Scroll down twice to display the rest of the items

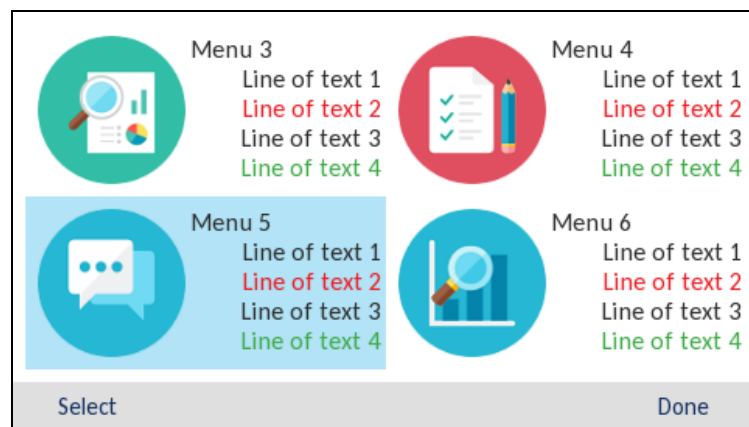
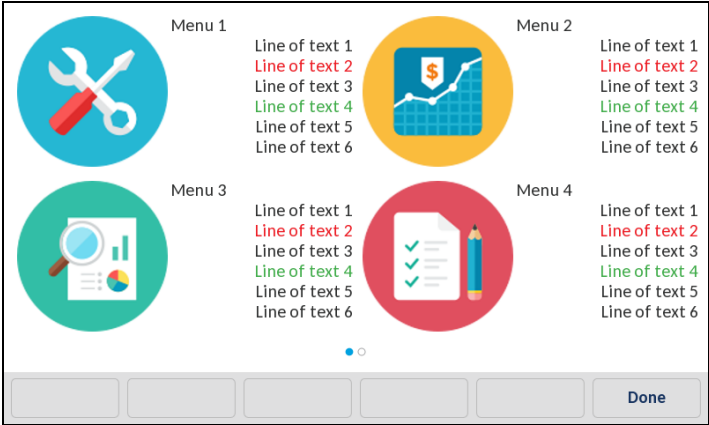


Figure 26: IconMenu Example 2 (6930)

Resulting Screens (6940)



Swipe to the left to reach the second page

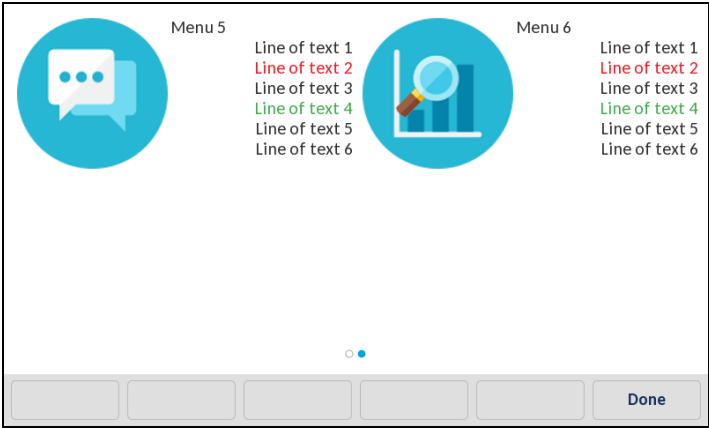


Figure 27: IconMenu Example 2 (6940)

3.3 TEXTSCREEN OBJECT

The TextScreen object can be used to display some text. The screen word-wraps appropriately and user can scroll to display a message longer then the physical display.

3.3.1 IMPLEMENTATION (6920/6930)

The title is displayed at the top of the XML area and uses up to 2 lines. The text is displayed after the title area and up to 7 lines can be displayed (9 lines if no title).



Note: The Text and Title area use the same font (size and type) but the Title is displayed in bold.

The **Left** key is mapped to the cancelAction tag, if not specified the XML object is destroyed.

The **Up/Down** keys allow up and down scrolling in order to display the complete text.

The **Right** and **Select** keys are mapped to the doneAction tag, if not specified the previous XML object in the stack is displayed.

Six (eight) customizable softkeys are available for this object. If more than 4 (5) softkeys are configured the softkeys are displayed on 2 pages.



Note: the **Left** key interaction is disabled if the LockIn tag is set to “yes”.

Object default Softkeys (6920)

Six customizable softkeys are available for this object.

Position	Label	Interaction	URIs
4	Done	Exit: Redisplays the previous XML object present in the phone browser.	SoftKey:Exit

Object default Softkeys (6930)

Eight customizable softkeys are available for this object.

Position	Label	Interaction	URIs
5	Done	Exit: Redisplays the previous XML object present in the phone browser.	SoftKey:Exit

The following figures detail how the AstralIPPhoneTextScreen is implemented on the Mitel 6920. The size of the text zone depends on the presence of the optional title.

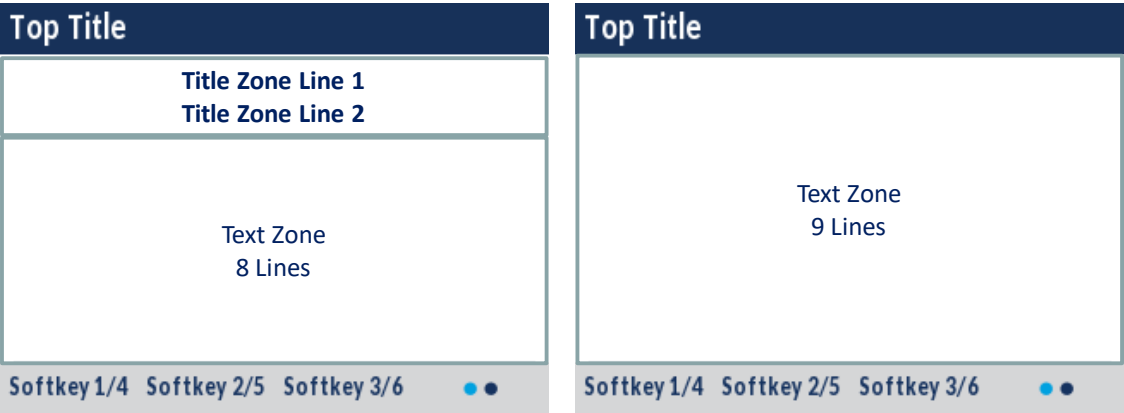


Figure 28: TextScreen implementation on 6920

Native Interaction

- “SoftKey:Exit”

3.3.2 IMPLEMENTATION (6940)

The title is displayed at the top of the XML area and uses up to 2 lines. The text is displayed after the title area and up to 13 lines can be displayed (15 lines if no title).

 **Note:** The Text and Title area use the same font (size and type) but the Title is displayed in bold.

As the 6940 have no navigation keys, scrolling the text is done by swiping the screen up and down.

Ten customizable softkeys are available for this object. If more than 6 softkeys are configured the softkeys are displayed on 2 pages.

Object default Softkeys

Ten customizable softkeys are available for this object.

Position	Label	Interaction	URIs
6	Done	Exit: Redisplays the previous XML object present in the phone browser.	SoftKey:Exit

The following figures detail how the AastralPPhoneTextScreen is implemented on the Mitel 6940. The size of the text zone depends on the presence of the optional title.



Figure 29: TextScreen implementation on 6940

Native Interaction

- “SoftKey:Exit”

3.3.3 XML DESCRIPTION

“Red” tags indicate that the tag does not apply to all phones, when not supported the tags are just ignored.

```
<AstralPPhoneTextScreen
  destroyOnExit = “yes/no”
  cancelAction = “some URI”
  doneAction = “some URI”
  Beep = “yes/no”
  Timeout = “some integer”
  bgColor=“white/black...”
  LockIn = “yes/no”
  CallProtection = “yes/no/notif”
  GoodbyeLockInURI = “some URI”
  allowDTMF = “yes/no”
  scrollUp = “some URI”
  scrollDown = “some URI”
  scrollLeft = “some URI”
  scrollRight = “some URI”
>
  <Title   wrap=“yes/no”
           Color=“white/black...”
  >Screen Title</Title>
  <TopTitle   icon=“icon index”
              Color=“white/black...”
  >Top Title</TopTitle>
  <Text   Color=“white/black...”
  >The screen text goes here</Text>
  <Dial>Number to dial</Dial>
  <!--Additional Softkey Items may be added-->
</AstralPPhoneTextScreen>
```

XML Document Objects

Document Object	Position	Type	Comments
AstralPPhoneTextScreen	Root tag	Mandatory	Root object
destroyOnExit	Root tag	Optional	“yes/no” indicates if the object is kept or not in the phone browser after exit. If not specified, the object is kept in the browser.
cancelAction	Root tag	Optional	Defines the URI to be called when the user cancels the XML object.
doneAction	Root tag	Optional	Defines the URI to be called when the user selects the “Done” softkey or ‘SoftKey:Exit’.
Beep	Root tag	Optional	“yes” or “no” to indicate if a notification beep must be generated by the phone.
Timeout	Root tag	Optional	Overrides the default 45 seconds timeout of the UI XML object. A Timeout set to “0” will disable the timeout feature. See section 4.3.2 for more details

Document Object	Position	Type	Comments
bgColor	Root tag	Optional	Set the background color of the object. the possible values are detailed in chapter 4.2.3. If not specified, the default value is "white".
LockIn	Root tag	Optional	If set to "yes", the phone ignores all events that would cause the screen to exit without using the keys defined by the object. Default value is "no". See section 4.3.3 for more details.
CallProtection	Root tag	Optional	If set to "yes", the phone will not destroy the XML object being displayed on an incoming call. If set to "notif", the behavior is the same as "yes" but a notification is displayed on the screen providing the caller ID details.
GoodbyeLockInURI	Root tag	Optional	Valid only if LockIn="yes", this tag defines a URI to be called when the "Goodbye" key is pressed during a locked XML session. This URI overrides the native behavior of the "Goodbye" key which is to destroy the current XML object displayed.
allowDTMF	Root tag	Optional	This tag allows letting keypad strokes as DTMF when the phone is in the connected status. Allow the DTMF keys for Idle state if CallProtection is enabled and allowDTMF is set to "yes"
scrollUp	Root tag	Optional	This tag allows overriding the default behavior of the Up arrow key once the scrolling reaches an end. <i>Not available on 6940.</i>
scrollDown	Root tag	Optional	This tag allows overriding the default behavior of the Down arrow key once the scrolling reaches an end. <i>Not available on 6940.</i>
scrollLeft	Root tag	Optional	This tag allows overriding the default behavior of the Left arrow key once the scrolling reaches an end. <i>Not available on 6940.</i>
scrollRight	Root tag	Optional	This tag allows overriding the default behavior of the Right arrow key once the scrolling reaches an end. <i>Not available on 6940.</i>
Title	Body	Optional	Label to be used as title for the object
Color	Title tag	Optional	Label color, the possible values are detailed in chapter 4.2.3. If not specified, the default value is

Document Object	Position	Type	Comments
			"white".
TopTitle	Body	Optional	Text to be used as top title for the object
Icon	TopTitle tag	Optional	Index of the icon to be used for the top title
Color	TopTitle tag	Optional	Label color, the possible values are detailed in chapter 4.2.3. If not specified, the default value is "white".
Text	Body	Mandatory	Text to be displayed.
Color	Text tag	Optional	Text color, the possible values are detailed in chapter 4.2.3. If not specified, the default value is "darkgray".
Dial	Body	Optional	Defines what number will be dialed when an offhook action is performed on the phone or if the "Dial2" custom softkey is pressed



Note: the URI configured via the scrollUp and scrollDown tags is triggered only when the scrolling reaches an end. If no scrolling is needed, the URI are immediately triggered otherwise, they are triggered after the scroll reaches its beginning or its end.

3.3.4 EXAMPLES

XML Example 1

```
<AstralPPhoneTextScreen>  
  <Title>Screen Object</Title>  
  <Text>The screen object can be implemented similar to the firmware info screen. Note that white space is preserved in XML  
so the display should word-wrap appropriately. Only three lines can display at a time.</Text>  
</AstralPPhoneTextScreen>
```

Resulting Screen (6930)

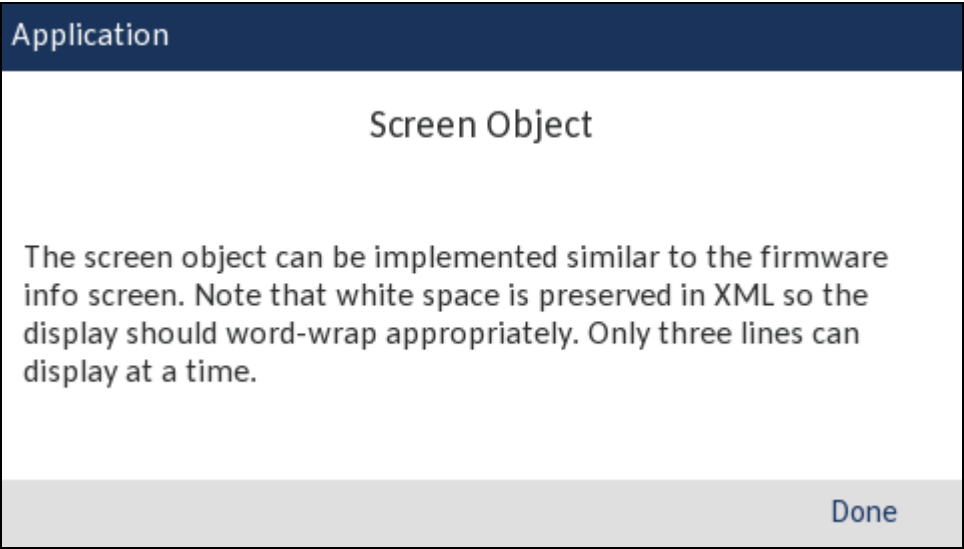


Figure 30: TextScreen Example (6930)

XML Example 2

```
<AstralPPhoneTextScreen>
  <Title>This is a very long title which should be on two lines</Title>
  <Text>Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's
standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen
book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was
popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop
publishing software like Aldus PageMaker including versions of Lorem Ipsum.</Text>
</AstralPPhoneTextScreen>
```

Resulting Screen (6930)

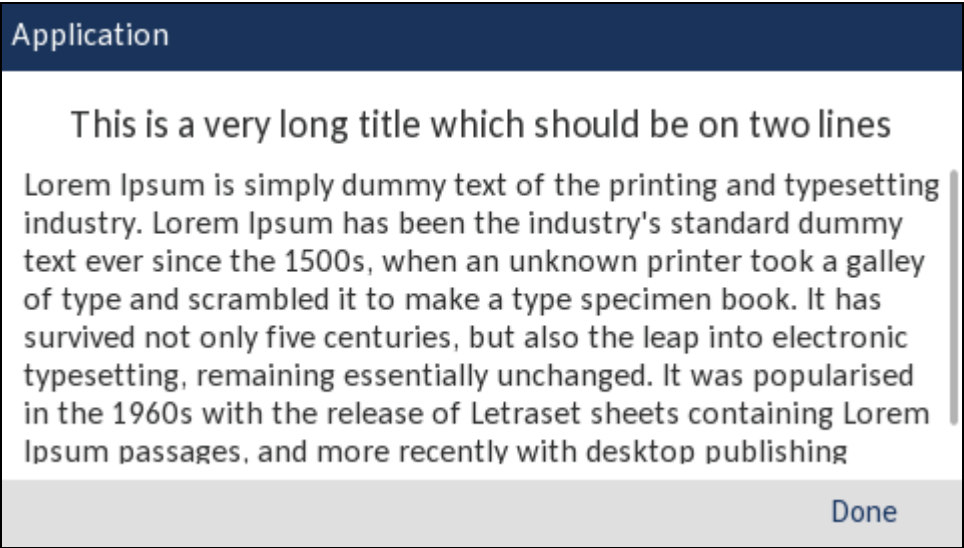


Figure 31: TextScreen Example (6930)

3.4 FORMATTEDTEXTSCREEN OBJECT

The FormattedTextScreen object allows the XML designer to display formatted (alignment, size, color and scrolling) text.

3.4.1 IMPLEMENTATION (6920/6930)

The text zone is divided into 3 distinct blocks, any of which can be empty.

The first block (header zone) is displayed at the top of the display and contains static text. This block takes up as many lines as the XML object specifies and can range from 0 up to the size of the physical screen.

The next block (scrolling zone), displayed below the first block, displays scrolling text and takes up as many lines as available, the size of the scrolling zone is automatically determined by using what's left of the display between the header zone and the footer zone.

The final block (footer zone) contains static text and will take up whatever lines are left on the screen starting from the bottom.



Figure 32: FormattedTextScreen layout

The *Left* key is mapped to the cancelAction tag, if not specified the XML object is destroyed.

The *Up/Down* keys allow up and down scrolling in order to display the complete text if needed.

 **Note:** the *Left* key interaction is disabled if the LockIn tag is set to “yes”.

Object default Softkeys (6920)

Six customizable softkeys are available for this object.

Position	Label	Interaction	URIs
4	Done	Exit: Redisplays the previous XML object present in the phone browser.	SoftKey:Exit

Object default Softkeys (6930)

Eight customizable softkeys are available for this object.

Position	Label	Interaction	URIs
5	Done	Exit: Redisplays the previous XML object present in the phone browser.	SoftKey:Exit

The following figure details how the AastralPPhoneFormattedTextScreen is implemented on the Mitel 6920.

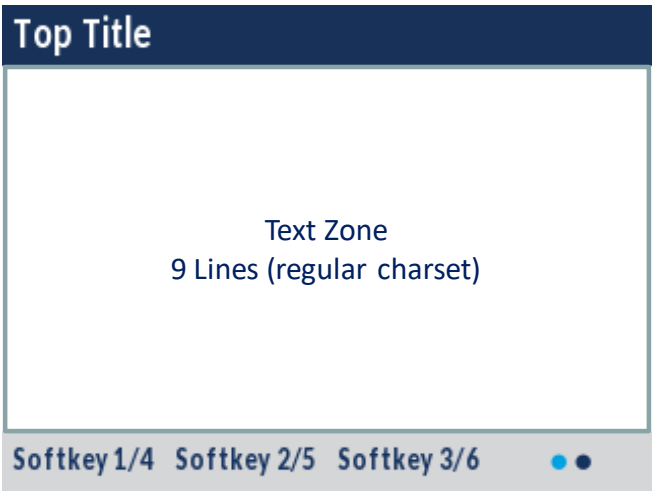


Figure 33: FormattedTextScreen implementation on 6920

Native Interaction

- “SoftKey:Exit”

3.4.2 IMPLEMENTATION (6940)

The text zone is divided into 3 distinct blocks, any of which can be empty.

The first block (header zone) is displayed at the top of the display and contains static text. This block takes up as many lines as the XML object specifies and can range from 0 up to the size of the physical screen.

The next block (scrolling zone), displayed below the first block, displays scrolling text and takes up as many lines as available, the size of the scrolling zone is automatically determined by using what’s left of the display between the header zone and the footer zone.

The final block (footer zone) contains static text and will take up whatever lines are left on the screen starting from the bottom.



Figure 34: FormattedTextScreen layout on 6940

As the 6940 does not have navigation keys, the scrolling is performed by swiping the scrollable zone up and down.

Object default Softkeys

Ten customizable softkeys are available for this object.

Position	Label	Interaction	URIs
6	Done	Exit: Redisplays the previous XML object present in the phone browser.	SoftKey:Exit

The following figure details how the AstralPPhoneFormattedTextScreen is implemented on the Mitel 6940.

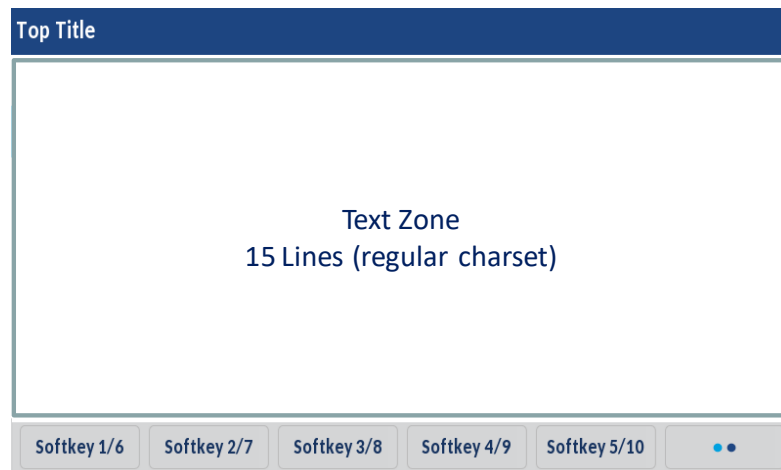


Figure 35: FormattedTextScreen implementation on 6940

Native Interaction

- “SoftKey:Exit”

3.4.3 XML DESCRIPTION

“Red” tags indicate that the tag does not apply to all phones, when not supported the tags are just ignored.

```
<AstralIPPhoneFormattedTextScreen
  destroyOnExit = "yes/no"
  cancelAction = "some URI"
  doneAction = "some URI"
  Beep = "yes/no"
  Timeout = "some integer"
  bgColor="white/black..."
  LockIn = "yes/no"
  CallProtection = "yes/no/notif"
  GoodbyeLockInURI = "some URI"
  allowDTMF = "yes/no"
  scrollUp = "some URI"
  scrollDown = "some URI"
  scrollLeft = "some URI"
  scrollRight = "some URI"
  fontMono = "yes/no"
>
  <TopTitle      icon="icon index"
                  Color="white/black..."
  >Top Title</TopTitle>
  <Line          wrap="yes/no"
    blink="no/slow/fast"
    Size="small/normal/double/large"
    Align="left/center/right"
    Color="white/black..."
    >A line of static text</Line>
    <!--Additional Lines may be added -->
  <Scroll Height="integer">
    <Line wrap="yes/no"
      blink="no/slow/fast"
      Size="small/normal/double/large"
      Align="left/center/right"
      Color="white/black..."
      >Scrolling text</Line>
      <!--Additional Lines may be added -->
  </Scroll>
  <Line          wrap="yes/no"
    blink="no/slow/fast"
    Size="small/normal/double/large"
    Align="left/center/right"
    Color="white/black..."
    >Some static footer text</Line>
  <!--Additional Lines may be added -->
  <Dial>Number to dial</Dial>
  <IconList>
    <Icon index = "int">Icon:Iconname or HEX string</Icon>
    <!--As many as different icons used in the object -->
    <!--Up to 22 icons, index can be 1 to 21 -->
  </IconList>
  <!--Additional Softkey Items may be added -->
</AstralIPPhoneFormattedTextScreen>
```



Notes:

- Any lines that would display past the bottom of the screen will be ignored.

- Text will not display with the top or bottom cut off.
- Text that extends past the edge of the screen will be cropped to the last fully displayed word.

XML Document Objects

Document Object	Position	Type	Comments
AastrapPhoneFormattedTextScreen	Root tag	Mandatory	Root object
destroyOnExit	Root tag	Optional	“yes/no” indicates if the object is kept or not in the phone browser after exit. If not specified, the object is kept in the browser.
cancelAction	Root tag	Optional	Defines the URI to be called when the user cancels the XML object.
doneAction	Root tag	Optional	Defines the URI to be called when the user selects the “Done” softkey or ‘SoftKey:Exit’.
Beep	Root tag	Optional	“yes” or “no” to indicate if a notification beep must be generated by the phone.
Timeout	Root tag	Optional	Overrides the default 45 seconds timeout of the UI XML object. A Timeout set to “0” will disable the timeout feature. See section 4.3.2 for more details
bgColor	Root tag	Optional	Set the background color of the object. the possible values are detailed in chapter 4.2.3. If not specified, the default value is “white”.
LockIn	Root tag	Optional	If set to “yes”, the phone ignores all events that would cause the screen to exit without using the keys defined by the object. Default value is “no”. See section 4.3.3 for more details.
CallProtection	Root tag	Optional	If set to “yes”, the phone will not destroy the XML object being displayed on an incoming call. If set to “notif”, the behavior is the same as “yes” but a notification is displayed on the screen providing the caller ID details.
GoodbyeLockInURI	Root tag	Optional	Valid only if LockIn=“yes”, this tag defines a URI to be called when the “Goodbye” key is

Document Object	Position	Type	Comments
			pressed during a locked XML session. This URI overrides the native behavior of the “Goodbye” key which is to destroy the current XML object displayed.
allowDTMF	Root tag	Optional	This tag allows letting keypad strokes as DTMF when the phone is in the connected status. Allow the DTMF keys for Idle state if CallProtection is enabled and allowDTMF is set to “yes”
scrollUp	Root tag	Optional	This tag allows overriding the default behavior of the Up arrow key once the scrolling reaches an end. <i>Not available on 6940</i>
scrollDown	Root tag	Optional	This tag allows overriding the default behavior of the Down arrow key once the scrolling reaches an end. <i>Not available on 6940</i>
scrollLeft	Root tag	Optional	This tag allows overriding the default behavior of the Left arrow key once the scrolling reaches an end. <i>Not available on 6940</i>
scrollRight	Root tag	Optional	This tag allows overriding the default behavior of the Right arrow key once the scrolling reaches an end. <i>Not available on 6940</i>
fontMono	Root tag	Optional	If set to “no”, a proportional font will be used to display the object instead of a monotype font. Default value is “yes”.
TopTitle	Body	Optional	Text to be used as top title for the object
Icon	TopTitle tag	Optional	Index of the icon to be used for the top title
Color	TopTitle tag	Optional	Label color, the possible values are detailed in chapter 4.2.3. If not specified, the default value is “white”.
Line	Body	Optional	Text to be displayed on the line. If the text is larger than the display, line is cropped to the last word.
wrap	Line tag	Optional	Indicates if the text for the line

Document Object	Position	Type	Comments
			is wrapped on multiple lines or limited to a single line (default).
blink	Line tag	Optional	Indicates if the blink mode of the line “slow”, “fast” or “no” (no by default).
Size	Line tag	Optional	Size of the font for the line: “small” for the small font “normal” for the regular font, “double” for the larger font. “large” for the largest font. “normal” is the default value if not specified.
Align	Line tag	Optional	Alignment of the line, “left”, “right” or “center”. If not specified, the default value is “left”.
Color	Line tag	Optional	Color of the line, the possible values are detailed in chapter 4.2.3. If not specified, the default value is “white”.
Scroll	Body	Optional	Defines the scrolling section of the display.
Height	Scroll tag	Optional	Specifies the height of the scroll zone. If not specified, the phone uses the remaining lines of the physical screen and does not allow footer lines.
Line	Scroll Body	Optional	Text to be displayed on the line in the scrolled zone. If the text is larger than the display, line is cropped to the last word.
wrap	Line tag	Optional	Indicates if the text for the line is wrapped on multiple lines or limited to a single line (default).
blink	Line tag	Optional	Indicates if the blink mode of the line “slow”, “fast” or “no” (no by default).
Size	Line tag	Optional	Size of the font for the line: “small” for the small font “normal” for the regular font, “double” for the larger font. “large” for the largest font. “normal” is the default value if not specified.

Document Object	Position	Type	Comments
Align	Line tag	Optional	Alignment of the scrolled line, "left", "right" or "center". If not specified the default value is "left".
Color	Line tag	Optional	Color of the line, the possible values are detailed in chapter 4.2.3. If not specified, the default value is "white".
Dial	Body	Optional	Defines what number will be dialed when an offhook action is performed on the phone or if the "Dial2" custom softkey is pressed
IconList	Body	Optional	List of icon definitions
Icon	IconList body	Optional	Icon value, it can be "Icon:Iconname", or the URL to a png file . See section 4.2.2 for more details.
Index	Icon tag	Optional	Index of the icon must be consistent with the 'Icon' used in the MenuItems. Possible values are 1 to 21.
SoftKey	Body	Optional	See section 4.1 for details Not available on non softkey phones

Limitations on the non softkey phones

- Custom Softkeys are not supported
- Only 2 lines are available to display the object.
- A double size line on line 1 does not allow any scrolling
- The scroll height cannot be more than 2.

3.4.4 EXAMPLES

XML Example 1

In this example, there is no need for scrolling.

```

<AstralPPhoneFormattedTextScreen destroyOnExit="yes">
  <Line Size="large" Align="center" Color="red">Header Line</Line>
  <Scroll>
    <Line>Scrolled Line 1</Line>
    <Line>Scrolled Line 2</Line>
    <Line>Scrolled Line 3</Line>
    <Line>Scrolled Line 4</Line>
    <Line>Scrolled Line 5</Line>
  </Scroll>
  <Line Size="double" Align="center" Color="blue">Footer Line</Line>
</AstralPPhoneFormattedTextScreen>

```

Resulting Screen

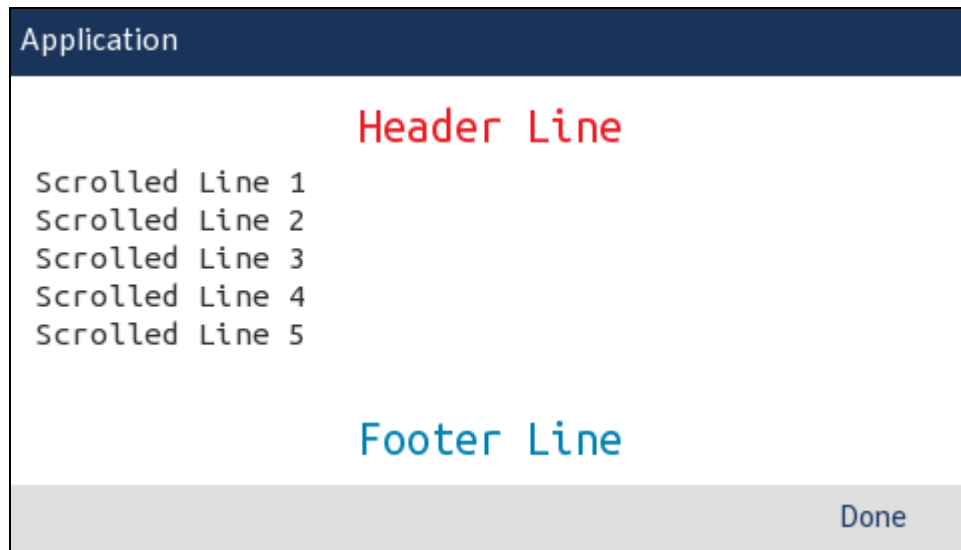


Figure 36: FormattedTextScreen Example 1 (6930)

XML Example 2

In this example, scrolling is needed.

```

<AstralPPhoneFormattedTextScreen destroyOnExit="yes" fontMono="no">
  <TopTitle Color="yellow">FormattedTextScreen</TopTitle>
  <Line Size="double" Align="center" Color="red">Header Line</Line>
  <Scroll>
    <Line wrap="yes">Scrolled Line 1 - This is an example where the line can be wrapped like a paragraph.</Line>
    <Line>Scrolled Line 2 - This is an example where the line is not wrapped like a paragraph.</Line>
    <Line>Scrolled Line 3</Line>
    <Line>Scrolled Line 4</Line>
    <Line>Scrolled Line 5</Line>
    <Line>Scrolled Line 6</Line>
    <Line>Scrolled Line 7</Line>
    <Line>Scrolled Line 8</Line>
    <Line>Scrolled Line 9</Line>
    <Line>Scrolled Line 10</Line>
  </Scroll>
  <Line Size="double" Align="center" Color="blue">Footer Line</Line>
</AstralPPhoneFormattedTextScreen>

```

Resulting Screen

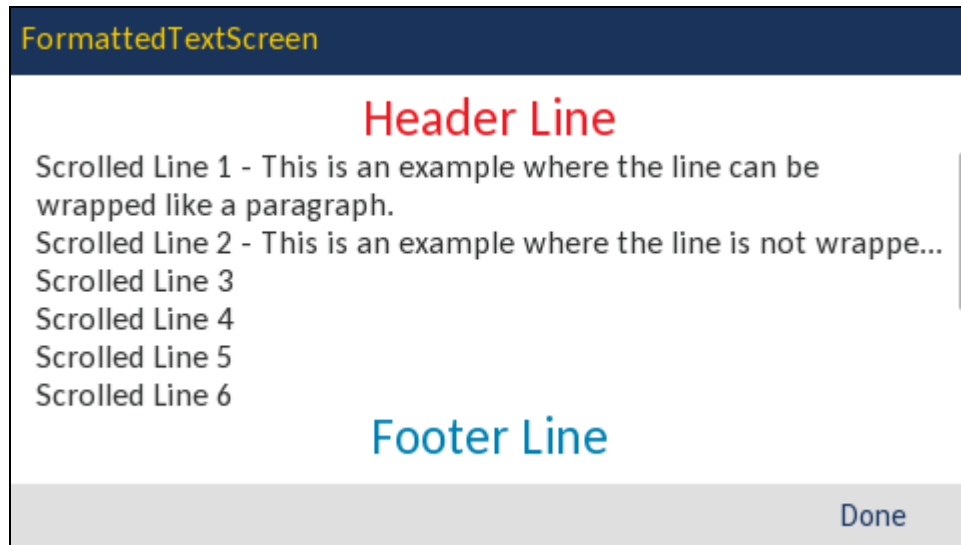


Figure 37: FormattedTextScreen Example 2 (6930)

3.5 IMAGESCREEN OBJECT

The ImageScreen object can be used to display single image (jpeg or png file only)

The user can specify where the image should be placed by setting horizontal and vertical alignment of the upper left hand corner, along with the height and width of the image.

3.5.1 IMPLEMENTATION (6920/6930)

For the 6920 and 6930, the image is either a 24/32 bit depth “png” or a “jpeg” file located on a server and which can be downloaded by the phone using TFTP, HTTP or HTTPS. See chapter 4.2.1.1 for more details.

Two image sizes are supported:

6920

- Up to 320x180 pixels (mode=regular or extended).
- Up to 320x240 pixels (mode=fullscreen). In that case, the softkeys are not displayed.

6930

- Up to 480x204 pixels (mode=regular or extended).
- Up to 480x272 pixels (mode=fullscreen). In that case, the softkeys are not displayed

If the image is bigger than the size supported by the requested mode, the image is clipped based on the requested alignment.

If the image is smaller than the size supported by the requested mode, the image is displayed based on the requested alignment

The **Left** key is mapped to the cancelAction tag, if not specified the XML object is destroyed.

Note:



- the Left Arrow key default interaction is disabled if the LockIn tag is set to “yes”.
- If the LockIn tag is set to “yes” and the cancelAction tag is configured, pressing the Left Arrow key triggers the configured cancelAction.
- The up, down, left and right arrow key default interactions can be overridden using the scrollUp, scrollDown, scrollLeft and scrollRight tags.

Object default Softkeys (6920)

Six (four physical) customizable softkeys are available for this object.

Position	Label	Interaction	URIs
4	Done	Exit: Redisplays the previous XML object present in the phone browser.	SoftKey:Exit

Object default Softkeys (6930)

Eight (five physical) customizable softkeys are available for this object.

Position	Label	Interaction	URIs
5	Done	Exit: Redisplays the previous XML object present in the phone browser.	SoftKey:Exit

The following figure details how the AastralPPhoneImageScreen is implemented on the Mitel 6920.



Figure 38: ImageScreen 6920 implementation (mode regular/extended and full screen)

Also a URI can be programmed to be called when the user presses the **Select** key (imageAction root tag). If this tag is empty, phone will use the URI configured by the doneAction tag (SoftKey:Exit by default).

So by default pressing the **Select** key destroys the current XML object unless an imageAction or a doneAction is configured.

Native Interaction

- “SoftKey:Exit”

3.5.2 IMPLEMENTATION (6940)

For the 6940, the image is either a 24/32 bit depth “png” or a “jpeg” file located on a server and which can be downloaded by the phone using TFTP, HTTP or HTTPS. See chapter 4.2.1.1 for more details.

Two image sizes are supported:

- Up to 800x372 pixels (mode=regular or extended).
- Up to 800x480 pixels (mode=fullscreen). In that case, the softkeys are not displayed.

If the image is bigger than the size supported by the requested mode, the image is clipped following the requested alignment.

If the image is smaller than the size supported by the requested mode, the image is displayed following the requested alignment

Note: as the 6940 do not have navigation keys, the scroll actions are mapped to screen swipes



- scrollUp, by swiping from top to bottom
- scrollDown, by swiping from bottom to top
- scrollLeft, by swiping from left to right
- scrollRight, by swiping from right to left

Object default Softkeys

Ten (six physical) customizable softkeys are available for this object.

Position	Label	Interaction	URIs
6	Done	Exit: Redisplays the previous XML object present in the phone browser.	SoftKey:Exit

The following figure details how the AstralPPhoneImageScreen is implemented on the Mitel 6940.

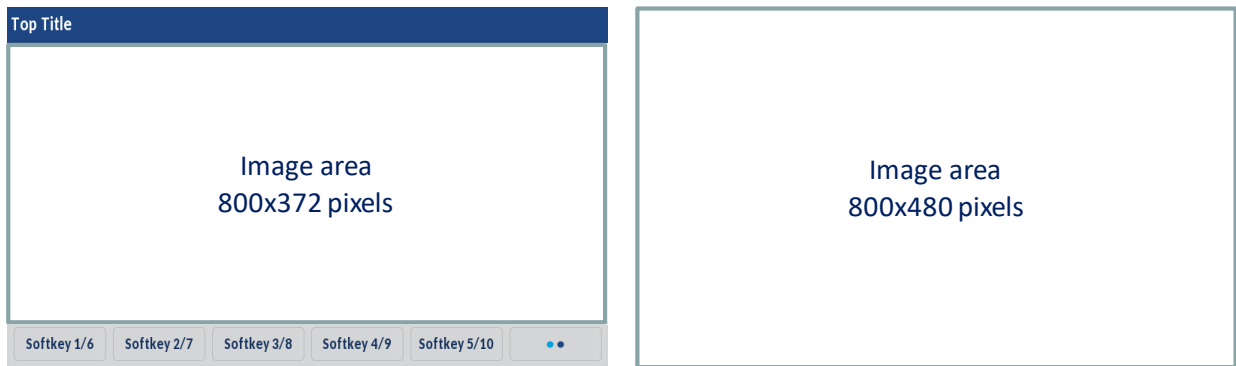


Figure 39: ImageScreen 6940 implementation (mode regular/extended and full screen)

Also a URI can be programmed to be called when the user presses on the displayed image (imageAction root tag). If this tag is empty, phone will use the URI configured by the doneAction tag (SoftKey:Exit by default).

So, by default, pressing on the image destroys the current XML object unless an imageAction or a doneAction is configured.

Native Interaction

- “SoftKey:Exit”

3.5.3 XML DESCRIPTION

“Red” tags indicate that the tag does not apply to all phones, when not supported the tags are just ignored.

```
<AstralPPhoneImageScreen
  destroyOnExit = “yes/no”
  cancelAction = “some URI”
  doneAction = “some URI”
  imageAction = “some URI”
  Beep = “yes/no”
  Timeout = “some integer”
  bgColor = “white,black...”
  LockIn = “yes/no”
  CallProtection = “yes/no/notif”
  GoodbyeLockInURI = “some URI”
  allowDTMF = “yes/no”
  scrollUp = “some URI”
  scrollDown = “some URI”
  scrollLeft = “some URI”
  scrollRight = “some URI”
  mode = “regular/extended/fullscreen”
>
  <TopTitle      icon=”icon index”
                  Color=”white/black...”
  >Top Title</TopTitle>
  <Image
    verticalAlign = “top,middle,bottom”
    horizontalAlign = “left,middle,right”
  >Image as hexadecimal characters or URL</Image>
  <!--Additional Softkey Items may be added -->
</AstralPPhoneImageScreen>
```

XML Document Objects

Document Object	Position	Type	Comments
AstralPPhoneImageScreen	Root tag	Mandatory	Root object
destroyOnExit	Root tag	Optional	“yes/no” indicates if the object is kept or not in the phone browser after exit. If not specified, the object is kept in the browser.
cancelAction	Root tag	Optional	Defines the URI to be called when the user cancels the XML object.
doneAction	Root tag	Optional	Defines the URI to be called when the user selects the “Done” softkey or ‘SoftKey:Exit’.
imageAction	Root tag	Optional	Defines the URI to be called when the user presses on the image (6940) or presses Select on 6920 and 6930.
Beep	Root tag	Optional	“yes” or “no” to indicate if a notification beep must be generated by the phone.
Timeout	Root tag	Optional	Overrides the default 45 seconds timeout of the UI XML object. A Timeout set to “0” will disable the timeout feature. See section 4.3.2 for more details
bgColor	Root tag	Optional	Set the background color of the object. the possible values are detailed in chapter 4.2.3. If not specified, the default value is “white”.
LockIn	Root tag	Optional	If set to “yes”, the phone ignores all events that would cause the screen to exit without using the keys defined by the object. Default value is “no”. See section 4.3.3 for more details.
CallProtection	Root tag	Optional	If set to “yes”, the phone will not destroy the XML object being displayed on an incoming call. If set to “notif”, the behavior is the same as “yes” but a

Document Object	Position	Type	Comments
			notification is displayed on the screen providing the caller ID details.
GoodbyeLockInURI	Root tag	Optional	Valid only if LockIn="yes", this tag defines a URI to be called when the "Goodbye" key is pressed during a locked XML session. This URI overrides the native behavior of the "Goodbye" key which is to destroy the current XML object displayed.
allowDTMF	Root tag	Optional	This tag allows letting keypad strokes as DTMF when the phone is in the connected status. Allow the DTMF keys for Idle state if CallProtection is enabled and allowDTMF is set to "yes"
scrollUp	Root tag	Optional	This tag allows overriding the default behavior of the Up arrow key.
scrollDown	Root tag	Optional	This tag allows overriding the default behavior of the Down arrow key.
scrollLeft	Root tag	Optional	This tag allows overriding the default behavior of the Left arrow key.
scrollRight	Root tag	Optional	This tag allows overriding the default behavior of the Right arrow key.
mode	Root tag	Optional	Configures the display mode, "regular", "extended" or "fullscreen". "extended" mode is equivalent to "regular" and kept for compatibility reasons.
TopTitle	Body	Optional	Text to be used as top title for the object
Icon	TopTitle tag	Optional	Index of the icon to be used for the top title
Color	TopTitle tag	Optional	Label color, the possible

Document Object	Position	Type	Comments
			values are detailed in chapter 4.2.3. If not specified, the default value is "white".
Image	Body	Mandatory	Image to be displayed as a predefined icon name or URL to get the png file. See section 4.2.1 for more details.
verticalAlign	Image tag	Optional	Vertical position of the image ("top", "middle" or "bottom"). If the tag is not specified, the object will use "middle" as a default value.
horizontalAlign	Image tag	Optional	Horizontal position of the image ("left", "middle" or "right"). If the tag is not specified, the object will use "middle" as a default value.
SoftKey	Body	Optional	See section 4.1 for details

3.5.4 EXAMPLES

XML Example 1

```
<AstralPPhoneImageScreen>
  <TopTitle>Top Title</TopTitle>
  <Image
    verticalAlign = "middle"
    horizontalAlign = "middle"
    >http://10.112.95.119/image.png</Image>
  <!--Additional Softkey Items may be added -->
</AstralPPhoneImageScreen>
```

Resulting Screen



Figure 40: ImageScreen Example 1 (6920)

XML Example 2

```
<AstralPPhoneImageScreen
mode = "extended">
  <Image
    verticalAlign = "bottom"
    height="200" width="150"
    >http://10.112.95.119/image.png</Image>
  <!--Additional Softkey Items may be added -->
</AstralPPhoneImageScreen>
```

Resulting Screen

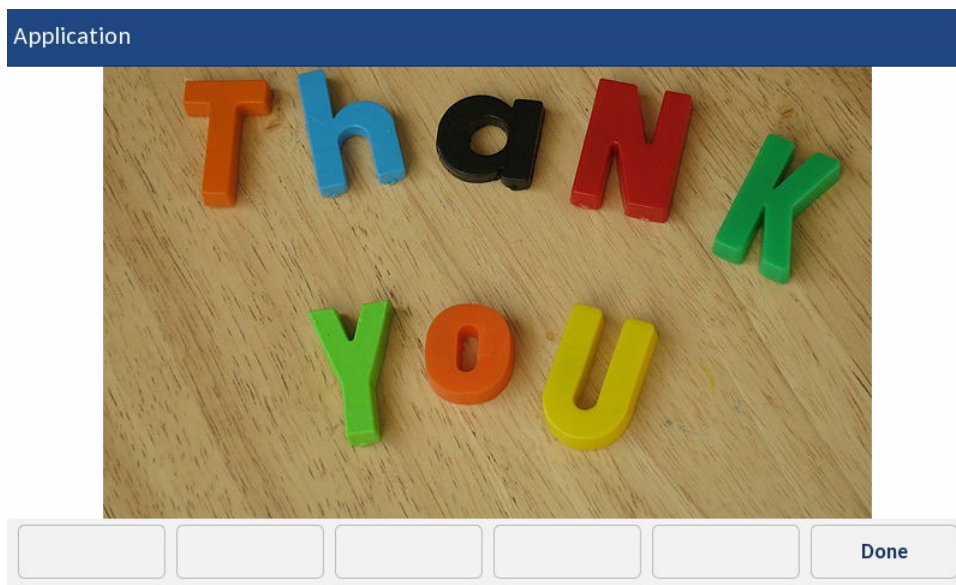


Figure 41: ImageScreen Example 2 (6940)

3.6 INPUTSCREEN OBJECT – SINGLE INPUT FIELD

The InputScreen object allows developers to create a screen capable of gathering user input. The Mitel IP phones support seven input types:

- IP Addresses,
- Numbers (integers plus * and #),
- Strings,
- Dates (US and international format)
- Time (US and international format)

Each parameter has a URL tag that is used to send information back to the HTTP server. The label in the parameter tag is appended to the address in the URL tag and sent via HTTP GET.

3.6.1 IMPLEMENTATION (6920/6930)

The **Up/Down** keys allow up and down scrolling in order to display the previous/next input field.

The **Select** key is mapped to the doneAction tag (Submit by default).

Object native interaction

- **Submit** (SoftKey:Submit) Completes the user input by submitting the programmed URI and value.
“SoftKey:Exit”

Common default Softkeys (6920)

Position	Label	Description	URIs
3	Done	Completes the user input by submitting the programmed URI and value.	SoftKey:Submit
4	Cancel	Redisplays the previous XML object present in the phone browser. Not available if LockIn set to “yes”.	SoftKey:Exit

Common default Softkeys (6930)

Position	Label	Description	URIs
4	Done	Completes the user input by submitting the programmed URI and value.	SoftKey:Submit
5	Cancel	Redisplays the previous XML object present in the phone browser. Not available if LockIn set to “yes”.	SoftKey:Exit

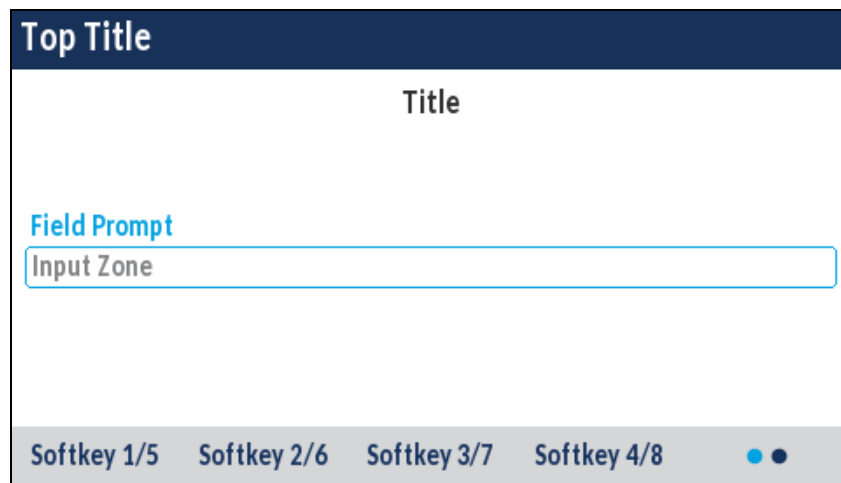


Figure 42: InputScreen implementation on 6930

On the 6920/30, like “Softkey::Submit” the URI custom softkeys are also appended with the parameter values as well as the selection, this allows to exit the InputScreen object without the validity check.

3.6.2 IMPLEMENTATION (6940)

On the 6940, the input for a field is done via a virtual keyboard which is displayed when the user touches the input zone.

The  key on the virtual keyboard triggers the submission of the entered value.

The  key on the virtual keyboard removes the keyboard from the display.

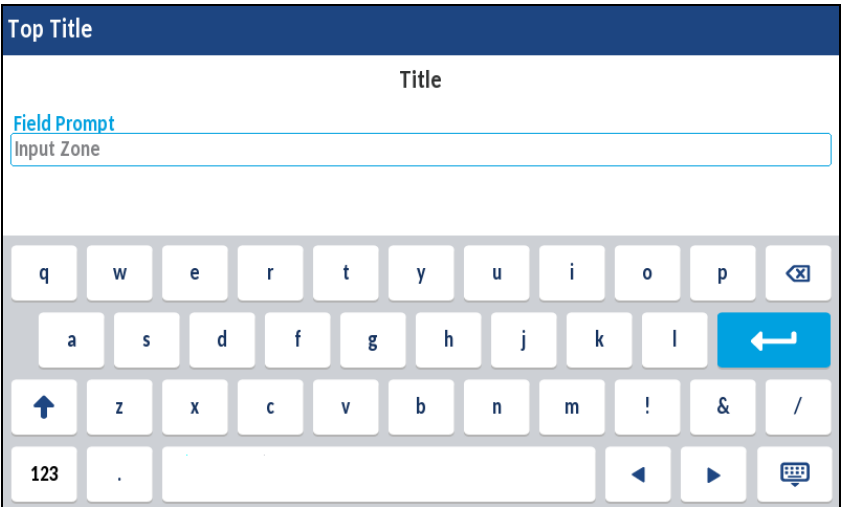


Figure 43: InputScreen implementation on 6940

Object native interaction

- **Submit** (SoftKey:Submit) Completes the user input by submitting the programmed URI and value.
- “SoftKey:Exit”

Object default Softkeys

Ten customizable softkeys are available for this object.

Position	Label	Interaction	URIs
5	Done	Completes the user input by submitting the programmed URI and value.	SoftKey:Submit
6	Cancel	Exit: Redisplays the previous XML object present in the phone browser.	SoftKey:Exit

3.6.3 XML DESCRIPTION

“Red” tags indicates that the tag does not apply to all phones, when not supported the tags are just ignored.

```

<AstralPPhoneInputScreen
  type = "IP/string/stringN/number/timeUS/timeInt/dateUS/dateInt"
  password = "yes/no"
  editable = "yes/no"
  destroyOnExit = "yes/no"
  cancelAction = "some URI"
  Beep = "yes/no"
  Timeout = "some integer"
  bgColor="white/black..."
  defaultFocus = "yes/no"
  LockIn = "yes/no"
  CallProtection = "yes/no/notif"
  GoodbyeLockInURI = "some URI"
  inputLanguage = "English / French / German / Italian / Spanish / Portuguese/ Russian / Nordic"
>
  <Title Color="white/black..."
  >Title string</Title>
  <TopTitle icon="icon index"
  Color="white/black..."
  >Top Title</TopTitle>
  <Prompt Color="white/black..."
  >Guidance for the input</Prompt>
  <URL>Target receiving the input</URL>
  <Parameter Color="white/black..."
  >name of the parameter added to URL</Parameter>
  <Default>Default Value</Default>
<!--Additional Softkey Items may be added -->
<!--Additional Icon Items may be added -->
</AstralPPhoneInputScreen>

```

XML Document Objects

Document Object	Position	Type	Comments
AstralPPhoneInputScreen	Root tag	Mandatory	Root object
type	Root tag	Optional	Specifies the type of input, possible values are "IP", "string", "stringN", "number", "timeUS", "timeInt", "dateUS", "dateInt". Default value is "string"
password	Root tag	Optional	Specifies if the input is masked by "*" characters. Default value is "no"
editable	Root tag	Optional	Specifies if the user is allowed to modify the input. Default value is "yes"
destroyOnExit	Root tag	Optional	"yes/no" indicates if the object is kept or not in the phone browser after exit. If not specified, the object is kept in the browser.
cancelAction	Root tag	Optional	Defines the URI to be called when the user cancels the XML object.
Beep	Root tag	Optional	"yes" or "no" to indicate if a notification beep must be generated by the phone.

Document Object	Position	Type	Comments
Timeout	Root tag	Optional	Overrides the default 45 seconds timeout of the UI XML object. A Timeout set to "0" will disable the timeout feature. See section 4.3.2 for more details
bgColor	Root tag	Optional	Set the background color of the object. the possible values are detailed in chapter 4.2.3. If not specified, the default value is "white".
LockIn	Root tag	Optional	If set to "yes", the phone ignores all events that would cause the screen to exit without using the keys defined by the object. Default value is "no". See section 4.3.3 for more details.
CallProtection	Root tag	Optional	If set to "yes", the phone will not destroy the XML object being displayed on an incoming call. If set to "notif", the behavior is the same as "yes" but a notification is displayed on the screen providing the caller ID details.
GoodbyeLockInURI	Root tag	Optional	Valid only if LockIn="yes", this tag defines a URI to be called when the "Goodbye" key is pressed during a locked XML session. This URI overrides the native behavior of the "Goodbye" key which is to destroy the current XML object displayed.
defaultFocus	Root tag	Optional	If set to "yes", the input field automatically goes to edit mode (keyboard displayed). Default value is "no". <i>6940 only</i>
inputLanguage	Root tag	Optional	Defines the language character set used for the input. English by default. See section 12 for the localized key mapping.
Title	Body	Optional	Text to be used as title for the object
Color	Title tag	Optional	Label color, the possible values are detailed in chapter 4.2.3. If not specified, the default value is "white".
TopTitle	Body	Optional	Text to be used as top title for the object
Icon	TopTitle tag	Optional	Index of the icon to be used for

Document Object	Position	Type	Comments
			the top title.
Color	TopTitle tag	Optional	Label color, the possible values are detailed in chapter 4.2.3. If not specified, the default value is "white".
Prompt	Body	Mandatory	Text to be displayed as guidance for the user input.
Color	Prompt tag	Optional	Label color, the possible values are detailed in chapter 4.2.3. If not specified, the default value is "yellow".
URL	Body	Mandatory	URI called when user completes his input.
Parameter	Body	Mandatory	Name of the parameter to be added to the URL after input is complete.
Color	Parameter tag	Optional	Label color, the possible values are detailed in chapter 4.2.3. If not specified, the default value is "white".
Default	Body	Mandatory	Default value to be displayed in the input field.
SoftKey	Body	Optional	See section 4.1 for more details

3.6.4 INPUT TYPE: IP

When the type is set to IP, the user input is restricted to integers only. The phone will validate the user input; if an invalid IP address is entered, nothing will be sent to the server and the user will receive an error message.

Object default Softkeys (6920/6930/6940)

Position	Label	Description	URIs
1	Backspace	Deletes the character before the cursor in the input field.	SoftKey:BackSpace
2	Dot "."	Inserts a "." in the user input at the cursor position	SoftKey:Dot
One before last	Done	Completes the user input by submitting the programmed URI and value.	SoftKey:Submit
Last	Cancel	Redisplays the previous XML object present in the phone browser. Does not appear if LockIn set to "yes".	SoftKey:Exit

XML Example

```
<AstralIPPhoneInputScreen type = "IP">
  <Title>Proxy Server</Title>
  <Prompt>Server IP:</Prompt>
  <URL>http://10.50.10.53/script.pl</URL>
  <Parameter>proxy</Parameter>
  <Default></Default>
</AstralIPPhoneInputScreen>
```



Note: In this example, when the user press “Done” or “Submit” or “Enter” on the phone after entering “192.168.0.100”, the phone will call the following URL
“http://10.50.10.53/script.pl?proxy=192.168.0.100”.

Resulting Screen (6930)

Figure 44: InputScreen “IP” Example (6930)

Resulting Screen (6940)

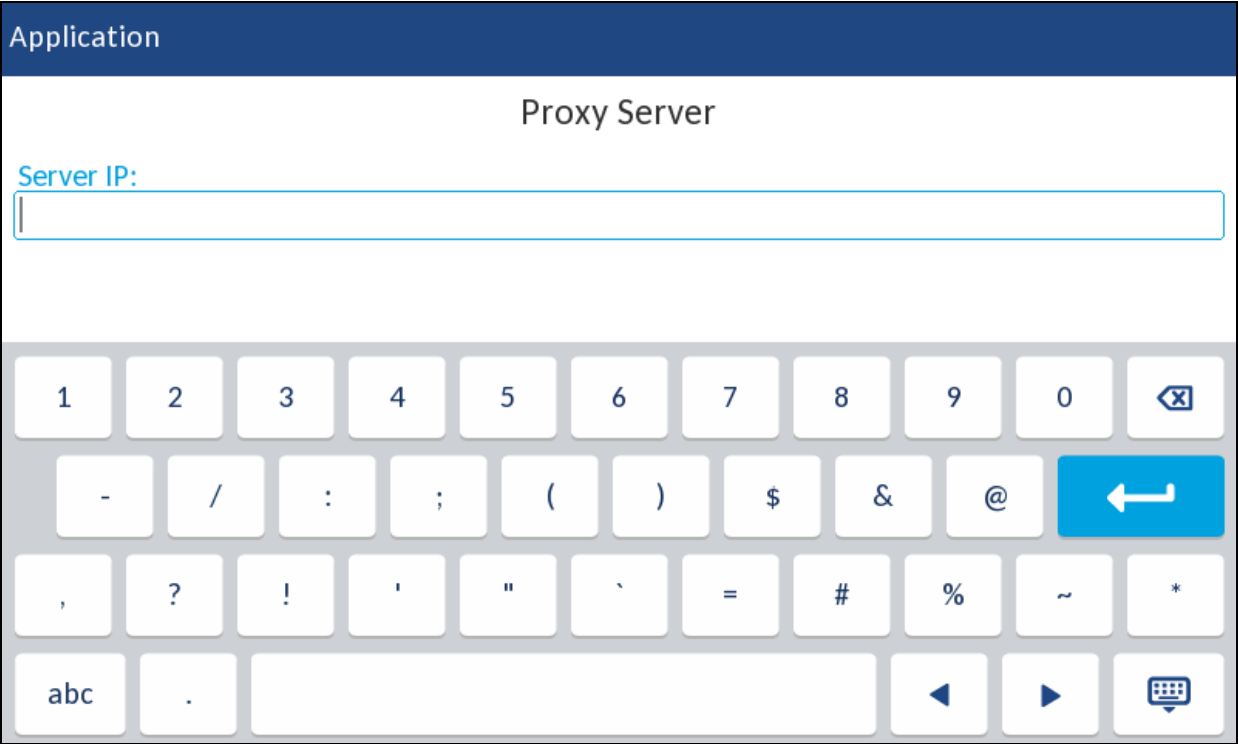


Figure 45: InputScreen “IP” Example (6940)

3.6.5 INPUT TYPE: NUMBER

Like an IP screen, a number input screen restricts the user to numbers only. Field validation is performed on the user input.

Object default Softkeys (6920/6930/6940)

Position	Label	Description	URIs
1	Backspace	Deletes the character before the cursor in the input field.	SoftKey:BackSpace
One before last	Done	Completes the user input by submitting the programmed URI and value.	SoftKey:Submit
Last	Cancel	Redisplays the previous XML object present in the phone browser. Does not appear if LockIn set to “yes”.	SoftKey:Exit

XML Example

```
<AstralPPhoneInputScreen type="number">
  <Title>Proxy Port</Title>
  <Prompt>Port:</Prompt>
  <URL>http://10.50.10.53/script.pl</URL>
  <Parameter>port</Parameter>
  <Default>5060</Default>
</AstralPPhoneInputScreen>
```

Resulting Screen (6930)

Application

Proxy Port

Port:

5060

Backspace

Cancel

Done

Figure 46: InputScreen “Number” Example (6930)

Resulting Screen (6940)

Application

Proxy Port

Port:

5060

1234567890

- / : ; () \$ & @

, ? ! ' " , = # % ~ *

abc .

← →

Figure 47: InputScreen “Number” Example (6940)



Note: In this example, when the user presses “Done” or “Submit” or “Enter” on the phone after entering “5060”, the phone will call the following URL “http://10.50.10.53/script.pl?port=5060”.

3.6.6 INPUT TYPE: STRING OR STRINGN

When the input type is set to string, the user can enter uppercase letters, lowercase letters, symbols, and numbers.

6920/6930/6940 input

The input mode can be switched via the Mode Key (Upper Case, Lower Case and Digits).

The user can scroll through some available input symbols

- . : ; , = _ , - ' & () " \$! by pressing the 1 key repeatedly.
- # / \ @ by pressing the # key repeatedly.
- Space by pressing the * key repeatedly

In Upper Case or Lower Case mode, keys 2-9 scroll through the keypad letters (e.g. pressing repeatedly 2 in Upper case mode will scroll through ABC2)

Object default Softkeys (6920/6930/6940)

Position	Label	Description	URIs
1	Backspace	Deletes the character before the cursor in the input field.	SoftKey:BackSpace
2	Dot “.”	Inserts a “.” in the user input at the cursor position	SoftKey:Dot
3	ABC>	Toggle between input modes, “ABC”, “123”, “abc”. Type is “string”	SoftKey:ChangeMode
3	123>	Toggle between input modes, “123”, “ABC”, “abc”. Type is “stringN”	SoftKey:ChangeMode
4	NextSpace	Inserts a space in the user input at the cursor position	SoftKey:NextSpace
One before last	Done	Completes the user input by submitting the programmed URI and value.	SoftKey:Submit
Last	Cancel	Redisplays the previous XML object present in the phone browser. Does not appear if LockIn set to “yes”.	SoftKey:Exit

XML Example

```
<AstralPPhoneInputScreen
  type = "string"
  password = "yes">
  <Title>Settings</Title>
  <Prompt>Enter something</Prompt>
  <URL>http://10.50.10.53/script.pl</URL>
  <Parameter>passwd</Parameter>
  <Default></Default>
</AstralPPhoneInputScreen>
```

Resulting Screen (6930)

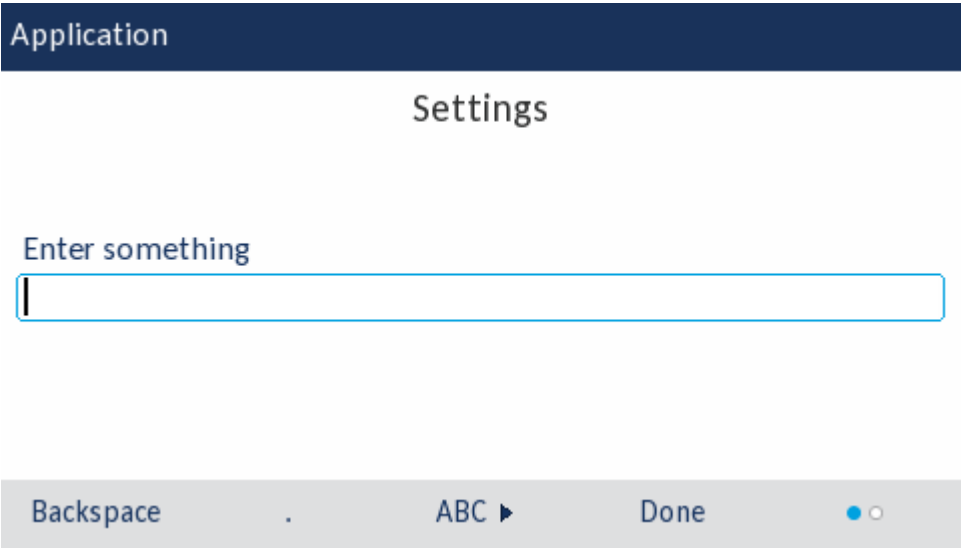


Figure 48: InputScreen “String” Example (6930)

Resulting Screen (6940)

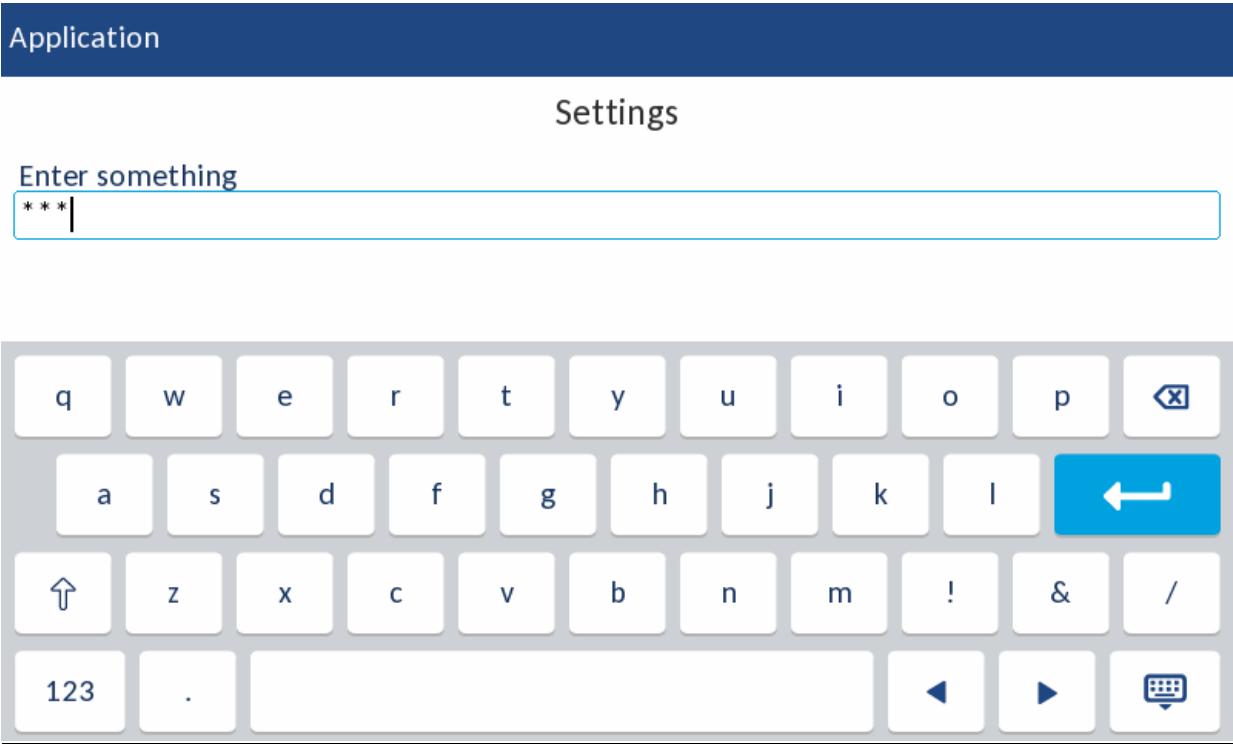


Figure 49: InputScreen “String” Example (6940)



Note: In this example, when the user press “Done” or “Submit” or “Enter” on the phone after entering “12345”, the phone will call the following URL “http://10.50.10.53/script.pl?passwd=12345”.

3.6.7 INPUT TYPE: TIMEUS

When the input type is set to “timeUS”, the user can enter a time using the US format with 12 hours cycle (HH:MM:SS AM/PM with HH from 00 to 12).

The user navigates between the various fields using the left and right navigation arrow keys, the toggle between AM and PM is done using the right navigation arrow key with the cursor positioned right before the AM/PM field.

Notes:



- the “password” attribute has no effect on this input type.
- the format of the “Default” attribute must be HH:MM:SSXX
- where XX is AM or PM, HH between 00 and 12 and MM/SS between 00 and 59.
- If the “Default” tag is empty, the phone displays “12:00:00 AM”.
- On the softkey phones, an extra softkey “AM/PM” is also available.

Object default Softkeys (6920/6930/6940)

Position	Label	Description	URIs
1	Backspace	Deletes the character before the cursor in the input field.	SoftKey:BackSpace
One before last	Done	Completes the user input by submitting the programmed URI and value.	SoftKey:Submit
Last	Cancel	Redisplays the previous XML object present in the phone browser. Does not appear if LockIn set to “yes”.	SoftKey:Exit

XML Example

```
<AstralPPhoneInputScreen type = "timeUS">
  <Title>Time US</Title>
  <Prompt>Enter time:</Prompt>
  <URL>http://10.50.10.53/script.pl</URL>
  <Parameter>time</Parameter>
  <Default></Default>
</AstralPPhoneInputScreen>
```

Resulting Screen (6930)

Application

Time US

Enter time:
12:00:00 AM

Backspace

AM/PM

Cancel

Done

Figure 50: InputScreen “TimeUS” Example (6930)

Resulting Screen (6940)

Application

Time US

Enter time:
12:00:00 AM

Backspace

AM/PM

Cancel

Done

Figure 51: InputScreen “TimeUS” Example (6940)



Note: In this example, when the user press “Done” or “Submit” or “Enter” on the phone after entering “10:20:00 AM”, the phone will call the following URL “http://10.50.10.53/script.pl?time=10:20:00AM” (actually the phone URL encodes the URL so the “.” character is replaced by %3A but transparent to the receiving code)

3.6.8 INPUT TYPE: TIMEINT

When the input type is set to “timeInt”, the user can enter a time using the international format with a 24 hours cycle (HH:MM:SS with HH from 00 to 23).

The user navigates between the various fields using the left and right navigation arrow keys.

Notes:



- the “password” attribute has no effect on this input type.
- the format of the “Default” attribute must be HH:MM:SS with HH from 00 to 23. If the “Default” tag is empty, the phone displays “00:00:00”.

Object default Softkeys (6920/6930/6940)

Position	Label	Description	URIs
1	Backspace	Deletes the character before the cursor in the input field.	SoftKey:BackSpace
One before last	Done	Completes the user input by submitting the programmed URI and value.	SoftKey:Submit
Last	Cancel	Redisplays the previous XML object present in the phone browser. Does not appear if LockIn set to “yes”.	SoftKey:Exit

XML Example

```
<AstralPPhoneInputScreen type = "timeInt">
  <Title>Time International</Title>
  <Prompt>Enter time:</Prompt>
  <URL>http://10.50.10.53/script.pl</URL>
  <Parameter>time</Parameter>
  <Default></Default>
</AstralPPhoneInputScreen>
```

Resulting Screen (6930)

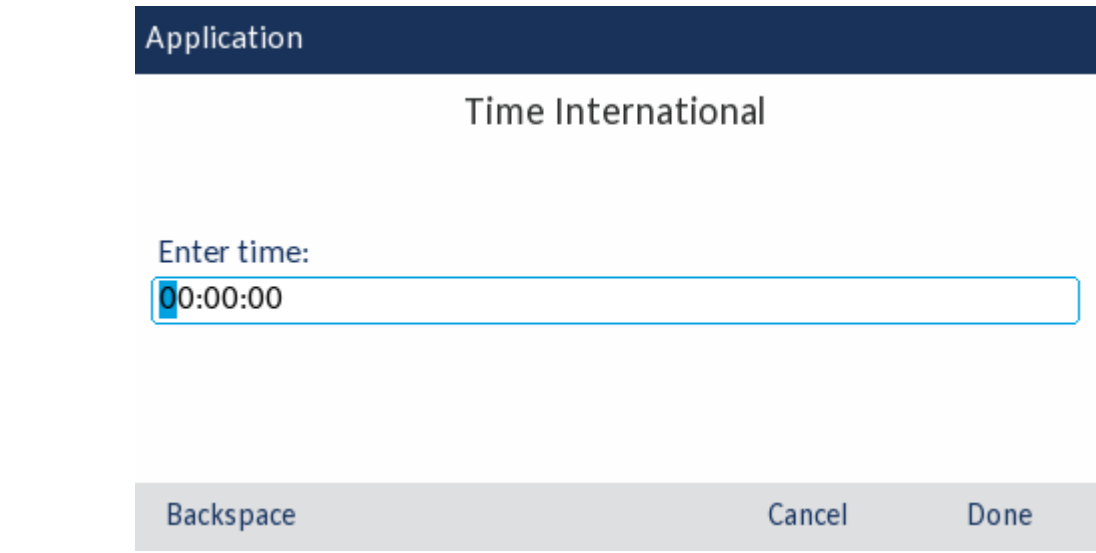


Figure 52: InputScreen “TimeInt” Example (6930)

Application

Time US

Enter time:

12:00:00 AM

Backspace

AM/PM

Cancel

Done

Figure 53: InputScreen “TimeInt” Example (6940)

 **Note:** In this example, when the user press “Done” or “Submit” or “Enter” on the phone after entering “14:20:00”, the phone will call the following URL `http://10.50.10.53/script.pl?time=14:20:00` (actually the phone URL encodes the URL so the “.” character is replaced by %3A but transparent to the receiving code)

3.6.9 INPUT TYPE: DATEUS

When the input type is set to “dateUS”, the user can enter a date using the US format (MM/DD/YYYY).

The user navigates between the various editable fields using the left and right navigation arrow keys.

Notes:

- 
- the “password” attribute has no effect on this input type.
 - the format of the “Default” attribute must be MM/DD/YYYY. If “Default” tag is empty, today’s date is displayed.

Object default Softkeys (6920/6930/6940)

Position	Label	Description	URIs
1	Backspace	Deletes the character before the cursor in the input field.	SoftKey:BackSpace
One before last	Done	Completes the user input by submitting the programmed URI and value.	SoftKey:Submit
Last	Cancel	Redisplays the previous XML object present	SoftKey:Exit

in the phone browser. Does not appear if
LockIn set to "yes".

XML Example

```
<AstralPPhoneInputScreen type = "dateUS">  
  <Title>Date US</Title>  
  <Prompt>Enter date</Prompt>  
  <URL>http://10.50.10.53/script.pl</URL>  
  <Parameter>date</Parameter>  
  <Default></Default>  
</AstralPPhoneInputScreen>
```

Resulting Screen (6930)

Application

Date US

Enter date

04/22/2017

Backspace Cancel Done

Figure 54: InputScreen "DateUS" Example (6930)

Application

Date US

Enter date

04/22/2017

Backspace

Cancel

Done

Figure 55: InputScreen “DateUS” Example (6940)

Notes:



- In this example, when the user press “Done” or “Submit” or “Enter” on the phone after entering “06/14/2009”, the phone will call the following URL “http://10.50.10.53/script.pl?date=06/14/2009”.
- the InputScreen object does not perform any control on the validity of the date entered by the user.

3.6.10 INPUT TYPE: DATEINT

When the input type is set to “dateInt”, the user can enter a date using the international format (DD/MM/YYYY).
The user navigates between the various editable fields using the left and right navigation arrow keys.

Notes:



- the “password” attribute has no effect on this input type.
- the format of the “Default” attribute must be DD/MM/YYYY. If “Default” tag is empty, today’s date is displayed.

Object default Softkeys (6920/6930/6940)

Position	Label	Description	URIs
One before last	Done	Completes the user input by submitting the programmed URI and value.	SoftKey:Submit
Last	Cancel	Redisplays the previous XML object present in the phone browser. Does not appear if	SoftKey:Exit

Position	Label	Description	URIs
		LockIn set to "yes".	

XML Example

```
<AstralPPhoneInputScreen type = "dateInt">
  <Title>Date International</Title>
  <Prompt>Enter date</Prompt>
  <URL>http://10.50.10.53/script.pl</URL>
  <Parameter>date</Parameter>
  <Default></Default>
</AstralPPhoneInputScreen>
```

Resulting Screen (6930)

Application

Date International

Enter date

22/04/2017

Backspace Cancel Done

Figure 56: InputScreen “DateInt” Example (6930)

Application

Date International

Enter date

22/04/2017

Backspace

Cancel

Done

Figure 57: InputScreen “DateInt” Example (6940)

Notes:



- In this example, when the user press “Done” or “Submit” or “Enter” on the phone after entering “14/06/2009”, the phone will call the following URL “http://10.50.10.53/script.pl?date=14/06/2009”.
- **Note:** the InputScreen object does not perform any control on the validity of the date entered by the user.

3.7 INPUTSCREEN OBJECT – MULTIPLE INPUT FIELDS

The InputScreen object can support up to 10 input fields, each of them potentially having different individual attributes.

The supported input types for each field are the same than the ones supported for the single input object plus an ‘empty’ type.

For each input field, the following attributes can be defined overriding the ones declared in the main object:

- Type
- Editable
- Password
- Prompt
- Default
- Parameter

Of course, only one URL can be defined to be called when the user has completed his inputs; the label in the parameter tags is appended to the address in the URL tag and sent via HTTP GET. If the Selection tag is used the value of the tag is also appended to the URL tag.

Two display modes are available for the Multiple Input AastraIPPhoneInputScreen:

- Normal: similar aspect to the single input field with a prompt and the input field on 2 separate lines, three (6920/6930) or seven (6940) input fields will be displayed per screen and the user will be able to scroll through them.
- Condensed: the prompt and the input field are on the same line, the prompts being right aligned on the longest prompt. Up to 7 fields are displayed on the same screen depending on the phone model.

3.7.1 IMPLEMENTATION (6920/6930)

The **Left/Right** keys are used to navigate between characters in the input field.

The **Up/Down** keys allow up and down scrolling in order to display the previous/next input field.

When multiple fields are displayed, the **Select** key is mapped to go to the next field (like the arrow down key) until the last field is reached where it is then mapped to the doneAction tag (Submit by default).

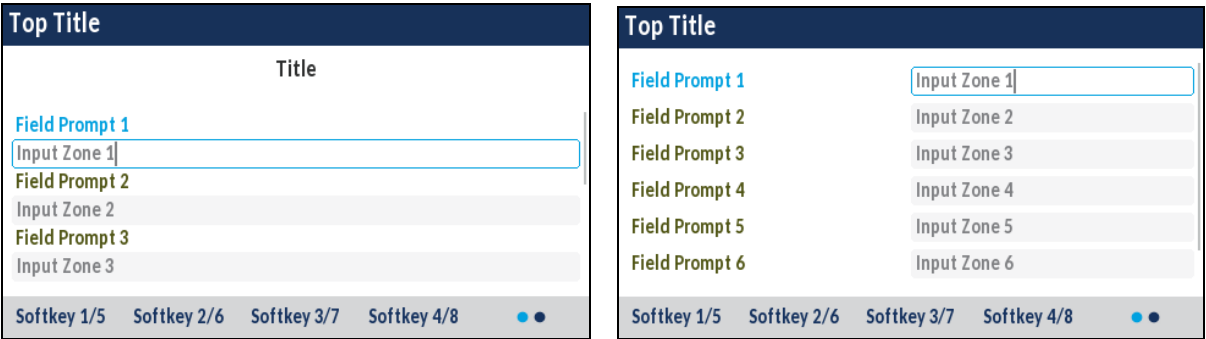


Figure 58: InputScreen implementation on 6930 “normal” mode



Figure 59: InputScreen implementation on 6930 “condensed” mode

The number of fields displayed on a single screen depends on the presence of the optional title.

Object native interaction

- **Done/Submit** Completes the user input by submitting the programmed URI and value.
- **Cancel** Redisplays the previous XML object present in the phone browser

Object default Softkeys (6920)

Six customizable softkeys are available for this object. If more than 4 softkeys are configured or needed the softkeys are displayed on 2 pages.

Position	Label	Interaction	URIs
3	Done	Completes the user input by submitting the programmed URI and value.	SoftKey:Submit
4	Cancel	Redisplays the previous XML object present in the phone browser. Not available if LockIn set to “yes”.	SoftKey:Exit

Object default Softkeys (6930)

Eight customizable softkeys are available for this object. If more than 5 softkeys are configured or needed the softkeys are displayed on 2 pages.

Position	Label	Interaction	URIs
4	Done	Completes the user input by submitting the programmed URI and value.	SoftKey:Submit
5	Cancel	Redisplays the previous XML object present in the phone browser. Not available if LockIn set to “yes”.	SoftKey:Exit

On the 6920/6930, like “Softkey::Submit” the URI custom softkeys are also appended with the parameter values as well as the selection, this allows to exit the InputScreen object without the validity check. This allows for instance to create an input screen with a field which could be a selection amongst a list of values and come back to the screen without losing the user inputs.

3.7.2 IMPLEMENTATION (6940)

The **Left/Right** keys on the virtual keyboard are used to navigate between characters in the input field.

Scrolling between the fields is done swiping the finger on the screen up and down. Scrolling is also available when the virtual keyboard is displayed.

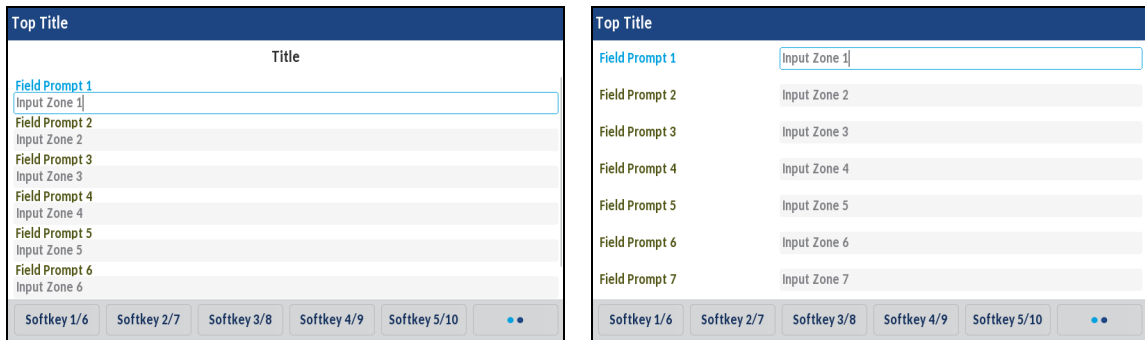


Figure 60: InputScreen implementation on 6940 “normal” mode

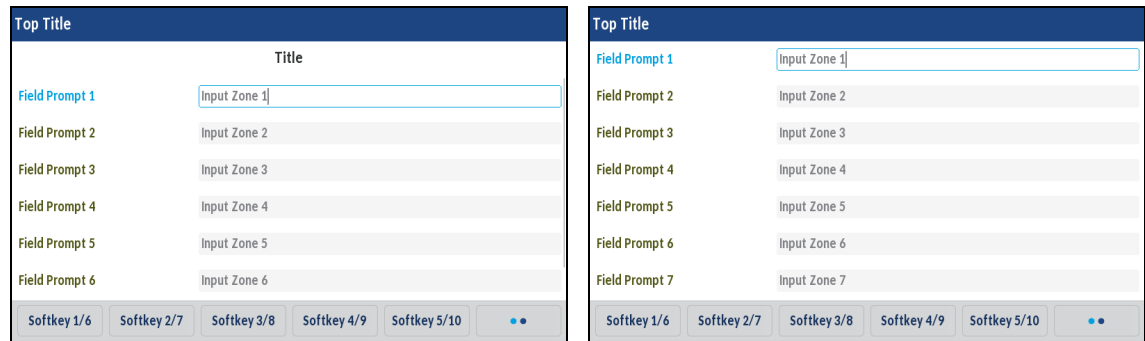


Figure 61: InputScreen implementation on 6940 “condensed” mode

The number of fields displayed on a single screen depends on the presence of the optional title.

Object native interaction

- **Done/Submit** Completes the user input by submitting the programmed URI and value.
- **Cancel** Redisplays the previous XML object present in the phone browser

Object default Softkeys

Eight customizable softkeys are available for this object. If more than 5 softkeys are configured or needed the softkeys are displayed on 2 pages.

Position	Label	Interaction	URIs
5	Done	Completes the user input by submitting the programmed URI and value.	SoftKey:Submit
6	Cancel	Redisplays the previous XML object present in the phone browser. Not available if LockIn set to “yes”.	SoftKey:Exit

On the 6940, like “Softkey::Submit” the URI custom softkeys are also appended with the parameter values as well as the selection, this allows to exit the InputScreen object without the validity check. This allows for instance to create an input screen with a field which could be a selection amongst a list of values and come back to the screen without losing the user inputs.

3.7.3 XML DESCRIPTION

“Red” tags indicate that the tag does not apply to all phones, when not supported the tags are just ignored.

```
<AstralPPhoneInputScreen
  type = “IP/string/stringN/number/timeUS/timeInt/dateUS/dateInt”
```

```

password = "yes/no"
editable = "yes/no"
destroyOnExit = "yes/no"
cancelAction = "some URI"
Beep = "yes/no"
Timeout = "some integer"
bgColor="white/black..."
LockIn = "yes/no"
CallProtection = "yes/no/notif"
GoodbyeLockInURI = "some URI"
defaultFocus = "yes/no"
displayMode = "normal/condensed"
inputLanguage = "English / French /German /Italian / Spanish / Portuguese /
Russian / Nordic"
>
  <Title   Color="white/black..."
  >Title string</Title>
  <TopTitle      icon="icon index"
                  Color="white/black..."
  >Top Title</TopTitle>
  <Prompt Color="white/black..."
  >Guidance for the input</Prompt>
  <URL>Target receiving the input</URL>
  <Parameter      Color="white/black..."
  >name of the parameter added to URL</Parameter>
  <Default>Default Value</Default>
  <InputField
    type = "IP/string/stringN/number
           timeUS/timeInt/dateUS/dateInt"
    password = "yes/no"
    editable = "yes/no"
  >
    <Prompt Color="white/black..."
    >Guidance for the input</Prompt>
    <Parameter      Color="white/black..."
    >parameter name added to URL</Parameter>
    <Default>Default Value</Default>
    <Selection>Selection</Selection>
  <!--Additional Softkey Items may be added -->
  </InputField>
  <!--Additional Input fields Items may be added -->
  <!--Additional Softkey Items may be added -->
  <!--Additional Icon Items may be added -->
</AstralPPhoneInputScreen>

```

XML Document Objects

Document Object	Position	Type	Comments
AstralPPhoneInputScreen	Root tag	Mandatory	Root object
Type	Root tag	Optional	Specifies the type of input, possible values are "IP", "string", "stringN", "number", "timeUS", "timeInt", "dateUS", "dateInt". Default value is "string".
Password	Root tag	Optional	Specifies if the input is masked by "*" characters. Default value is "no"
Editable	Root tag	Optional	Specifies if the user is allowed to modify the input. Default

Document Object	Position	Type	Comments
			value is "yes"
destroyOnExit	Root tag	Optional	"yes/no" indicates if the object is kept or not in the phone browser after exit. If not specified, the object is kept in the browser.
cancelAction	Root tag	Optional	Defines the URI to be called when the user cancels the XML object.
Beep	Root tag	Optional	"yes" or "no" to indicate if a notification Beep must be generated by the phone.
Timeout	Root tag	Optional	Overrides the default 45 seconds timeout of the UI XML object. A Timeout set to "0" will disable the timeout feature. See section 4.3.2 for more details
bgColor	Root tag	Optional	Set the background color of the object. the possible values are detailed in chapter 4.2.3. If not specified, the default value is "white".
LockIn	Root tag	Optional	If set to "yes", the phone ignores all events that would cause the screen to exit without using the keys defined by the object. Default value is "no". See section 4.3.3 for more details.
CallProtection	Root tag	Optional	If set to "yes", the phone will not destroy the XML object being displayed on an incoming call. If set to "notif", the behavior is the same as "yes" but a notification is displayed on the screen providing the caller ID details.
GoodbyeLockInURI	Root tag	Optional	Valid only if LockIn="yes", this tag defines a URI to be called when the "Goodbye" key is pressed during a locked XML session. This URI overrides the native behavior of the "Goodbye" key which is to destroy the current XML object displayed.
defaultFocus	Root tag	Optional	If set to "yes", the first input field defined by defaultIndex automatically goes to edit mode (keyboard displayed). Default value is "no".

Document Object	Position	Type	Comments
			<i>6940 only</i>
displayMode	Root tag	Optional	If set to "normal" the input fields will be displayed with the prompt and the input field on 2 lines. If set to "condensed" both the prompt and the input field are on the same line. Default value is "normal"
defaultIndex	Root tag	Optional	Defines the field where the user will start his input amongst the multiple field inputs. Default 1.
Title	Body	Optional	Text to be used as title for the object
Color	TopTitle tag	Optional	Label color, the possible values are detailed in chapter 4.2.3. If not specified, the default value is "white".
TopTitle	Body	Optional	Text to be used as top title for the object.
Icon	TopTitle tag	Optional	Index of the icon to be used for the top title
Color	TopTitle tag	Optional	Label color, the possible values are detailed in chapter 4.2.3. If not specified, the default value is "white".
Prompt	Body	Optional	Text to be displayed as guidance for the user input. Is used as the default value for each input field.
Color	Prompt tag	Optional	Label color, the possible values are detailed in chapter 4.2.3. If not specified, the default value is "yellow".
URL	Body	Mandatory	URI called when user completes his input.
Parameter	Body	Optional	Name of the parameter to be added to the URL after input is complete. Is used as the default value for each input field.
Color	Parameter tag	Optional	Label color, the possible values are detailed in chapter 4.2.3 If not specified, the default value is "white".
Default	Body	Mandatory	Default value to be displayed in the input field. Is used as the default value for each input

Document Object	Position	Type	Comments
			field.
InputField	Body	Optional	
Type	InputField tag	Optional	Specifies the type of input for the field, possible values are "IP", "string", "number", "timeUS", "timeInt", "dateUS", "dateInt" or "empty". Overrides the value set in the root tag for the field. An "empty" value will create one blank line in condensed mode and 2 blank lines in normal mode.
Password	InputField tag	Optional	Specifies if the input is masked by "*" characters. Default value is "no". Overrides the value set in the root tag for the field.
Editable	InputField tag	Optional	Specifies if the user is allowed to modify the input. Default value is "yes". Overrides the value set in the root tag for the field.
Prompt	InputField Body	Optional	Text to be displayed as guidance for the user input. Overrides the value set in the object for the field.
Color	Prompt tag	Optional	Label color, the possible values are detailed in chapter 4.2.3. If not specified, the default value is "dark blue"
Parameter	InputField Body	Optional	Name of the parameter to be added to the URL after input is complete. Overrides the value set in the object for the field.
Color	Parameter tag	Optional	Label color, the possible values are detailed in chapter 4.2.3. If not specified the default value is "dark blue"
Default	InputField Body	Optional	Default value to be displayed in the input field. Overrides the value set in the object for the field.
Selection	InputField Body	Optional	The content of this tag will be added when the "Submit" key is pressed while editing this field.
SoftKey	Body	Optional	See section 4.1 for details.

Note: when the InputField type is set to 'empty', a non editable blank line replaced the input field on the display when the XML object is in condensed mode, 2 blank lines for the normal mode. Also an empty field is a proper field; the defaultIndex value must consider the empty field as a plain field.

3.7.4 EXAMPLES

XML Example

```
<AstralPPhoneInputScreen
defaultIndex = "3"
>
<Title>Multi Entry Input Screen</Title>
<URL>http://10.102.43.70/XML/example.pl</URL>
<Default/>

<InputField type="string" >
<Prompt Color="red">1/ Username:</Prompt> To
<Parameter>user</Parameter>
<Selection>1</Selection>
<Default>User</Default>

<SoftKey index="1">
<Label>Erase</Label>
<URI>SoftKey:BackSpace</URI>
</SoftKey>

<SoftKey index="2">
<Label>ABC</Label>
<URI>SoftKey:ChangeMode</URI>
</SoftKey>
</InputField>

<InputField type="string" password="yes">
<Prompt Color="yellow">2/ Password:</Prompt>
<Parameter>passwd</Parameter>
<Selection>2</Selection>
</InputField>

<InputField type="IP" password="no">
<Prompt Color="cyan">3/ Proxy IP</Prompt>
<Parameter>Proxy</Parameter>
<Selection>3</Selection>

<SoftKey index="1">
<Label>Erase</Label>
<URI>SoftKey:BackSpace</URI>
</SoftKey>
</InputField>

<InputField type="number" password="no">
<Prompt Color="magenta">4/ Port number</Prompt>
<Parameter>Port</Parameter>
<Selection>4</Selection>
<Default>5060</Default>

<SoftKey index="1">
<Label>Erase</Label>
<URI>SoftKey:BackSpace</URI>
</SoftKey>
</InputField>

</AstralPPhoneInputScreen>
```

Resulting Screen (6930)

Application

Multi Entry Input Screen

1/ Username:

User

2/ Password:

3/ Proxy IP

Erase

Figure 62: Multiple Input Fields Example (6930)

Resulting Screen (6940)

Application

Multi Entry Input Screen

1/ Username:

User

2/ Password:

3/ Proxy IP

4/ Port number

5060

Erase

Figure 63: Multiple Input Fields Example (6940)

3.8 PHONEEXECUTE OBJECT (ALL MODELS)

The PhoneExecute object allows an external application to ask the phone to execute a sequence of local actions using a URI.

3.8.1 IMPLEMENTATION

The actions can be:

- Any supported uri
- [http\(s\)://myserver.com/myscript.pl](http(s)://myserver.com/myscript.pl), you can now use variables in the URI, variables are parsed by the phone.
- Dial:XXXXX
- DialLine:X:YYYYY (X is the line number, YYYYY is the number or URI to dial) see chapter 4.9 for more details on this feature.
- Led: XXXXXX=on/off/slowflash/fastflash:red/yellow/green see chapter 4.10 for more details on the LED control.
- Led: all=on/off/slowflash/fastflash:red/yellow/green to control all the phone XML keys see chapter 4.10 for more details on the LED control.
- Key: XXXXX see chapter 4.11 for more details on the simulated keypress feature.
- Phone Reboot (URI="Command: Reset"), the phone will restart (if the phone is idle) and process the complete boot sequence (configuration, language packs, firmware...)
- Phone Fast Reboot (URI="Command: FastReboot"), the phone will restart (if the phone is idle) but will reduce the boot sequence as it will not check for new firmware.
- Clear local configuration and reboot (URI="Command: ClearLocal")
- Clear all call history records (URI="Command: ClearCallHistoryList")
- Clear the callers list and missed calls indicator (URI="Command: ClearCallersList")
- Clear the local directory (URI="Command: ClearDirectory")
- Clear the redial list (URI="Command: ClearRedialList")
- Retrieve crash and configuration file (URI="Command: UploadSystemInfo") , see chapter 4.12 for more details.
- Launch the directory application (URI="Command: LaunchDirectory")
- Launch the callers list application (URI="Command: LaunchCallersList")
- Launch the redial list application (URI="Command: LaunchRedialList")
- Do nothing (URI="")

In order to prevent a pushed dial uri from putting an active call on hold the PhoneExecute object supports an optional tag "interruptCall". By default, the attribute will allow the current call to be interrupted. To prevent this, set the attribute to "no".

3.8.2 XML DESCRIPTION

```
<AstralPPhoneExecute
  Beep                                     = "yes/no"
  triggerDestroyOnExit="yes/no"
>
  <ExecuteItem URI="" interruptCall="yes/no" />
<!--Additional ExecuteItems may be added -->
</AstralPPhoneExecute>
```

XML Document Objects

Document Object	Position	Type	Comments
AstralPPhoneExecute	Root tag	Mandatory	Root object
Beep	Root tag	Optional	"yes" or "no" to indicate if a notification beep must be generated by the phone.
triggerDestroyOnExit	Root tag	Optional	If the XML object is sent as an answer to a UI object, setting this parameter to "yes" will trigger the exit of the calling object as if it was a UI object. See section 4.3.4 for more details.

Document Object	Position	Type	Comments
ExecuteItem	Body	Optional	Tag for the action to be executed
URI	ExecuteItem tag	Optional	URI describing the action to be executed. Only NULL URIs are supported.
interruptCall	ExecuteItem tag	Optional	When the URI is a dial command, if this tag is set to 'no' an existing call will not be put on hold.

3.8.3 EXAMPLES

```
<AstralPPhoneExecute>
  <ExecuteItem URI=http://myserver.com/myscript.php/>
  <ExecuteItem URI="Dial:12345" interruptCall="no"/>
  <ExecuteItem URI="Command: ClearCallersList"/>
  <ExecuteItem URI="Command: ClearDirectory"/>
  <ExecuteItem URI="Command: ClearRedialList"/>
  <ExecuteItem URI="Command: Reset"/>
</AstralPPhoneExecute>
```

This example will make the phone execute 6 action:

- Do an HTTP GET to myserver.com/myscript.php
- Dial 12345 without putting any current call on hold
- Clear the Callers List
- Clear the local directory
- Clear the Redial List
- Reset the phone

Notes:



- the "do nothing" can be used when an application needs to display nothing as an answer to a HTTP GET.
- you must be careful when you use the "Dial:" URI as the state of phone is unknown at the time of the XML GET.
- the FastReboot command will speed up the boot process of the phone which may be useful in the self-configuration application.

3.9 PHONESOFTKEY OBJECT (6920/6930/6940)

The PhoneSoftkey object allows an external application to modify the configuration of an “URL Line” Programmable softkeys dynamically. This includes changing icons, label and url (value).

3.9.1 IMPLEMENTATION

“URL Line” key layout

The following diagram shows the visual elements of the layout for the “URL Line” softkey



“URL Line” key attributes

Dynamic attributes

The attributes are indexed by the top softkey index the same way it is done in the phone configuration file. So the attributes are defined like regular top softkey attributes “ prgkeyXX attribute” for instance “ prgkey2 attribute”.

Attribute	Description	Default value
label	text or sequences of text to be displayed on the key	Comes from the MiVB configuration
value	URI to be called when the key is pressed	Comes from the MiVB configuration
icon	allows to set the icon, possible values are “Icon:Nolcon” or “An empty value” to display a default icon and downloaded image should delete from the phone “Icon location” to display a dynamic icon located on a server	Icon:Nolcon or empty

Notes:



- Dynamic attribute configurations are lost when phone reboots or when the key has a configuration change via MIVB or minet link reset using IPA and also downloaded image gets deleted from the phone.
- Phone ignores to download the same image again and again.

Default layout

When a key with type “URL Line” is created, this is the default layout

- Default Icon
- Label left justified and default color
- No background color
- For 6940 the active touch zone is the whole key



“URL Line” key uri

The “URL Line” key has a label field and an assigned uri as a value which is called when the button configured for the respective prgkey is pressed.

Notes:



- The label attribute is a dynamic parameter that lists the label messages that can also be Updated via the AstraIPPhoneSoftkey object.

3.9.2 XML DESCRIPTION

```
<AstraIPPhoneSoftkey
  Beep = "yes/no"
  triggerDestroyOnExit = "yes/no"
>
  <SoftkeyItem>
    <Parameter>parameter#</Parameter>
    <Value>value</Value>
  </SoftkeyItem>
<!--Additional SoftkeyItems may be added -->
</AstraIPPhoneSoftkey>
```

The number of parameters to be sent in a single PhoneSoftkeyobject is only limited by the overall size of the XML object (10000 bytes). Practically 30 parameters can be sent in a single object.

XML Document Objects

Document Object	Position	Type	Comments
AstraIPPhoneSoftkey	Root tag	Mandatory	Root object

Beep	Root tag	Optional	“yes” or “no” to indicate if a notification beep must be generated by the phone.
CallProtection	Root tag	Optional	If set to “yes”, the phone will not destroy the XML object being displayed on an incoming call. If set to “notif”, the behavior is the same as “yes” but a notification is displayed on the screen providing the caller ID details.
triggerDestroyOnExit	Root tag	Optional	In case the XML object is sent as an answer to a UI object, setting this parameter to “yes” will trigger the exit of the calling object as if it was a UI object. See section 4.3.5 for more details.
SoftkeyItem	Body	Optional	Tag for the configuration change
Parameter	SoftkeyItem tag	Mandatory	Key attribute to be changed
Value	SoftkeyItem tag	Mandatory	New value of the key attribute.

3.9.3 EXAMPLES

Example 1

XML Script

```

<AastraIPPhoneSoftkey>
  <SoftkeyItem>
    <Parameter>prgkey2 label</Parameter>
    <Value>key2</Value>
  </SoftkeyItem>
  <SoftkeyItem>
    <Parameter>prgkey2 icon</Parameter>
    <Value>tftp://10.112.123.82/image1.png</Value>
  </SoftkeyItem>
  <SoftkeyItem>
    <Parameter>prgkey3 label</Parameter>
    <Value>key3</Value>
  </SoftkeyItem>
  <SoftkeyItem>
    <Parameter>prgkey3 icon</Parameter>
    <Value>tftp://10.112.123.82/image2.png</Value>
  </SoftkeyItem>
  <SoftkeyItem>
    <Parameter>prgkey4 label</Parameter>
    <Value>key4</Value>
  </SoftkeyItem>
  <SoftkeyItem>
    <Parameter>prgkey4 icon</Parameter>
    <Value>tftp://10.112.123.82/image3.png</Value>
  </SoftkeyItem>
</AastraIPPhoneSoftkey>

```

Resulting screen (6940)

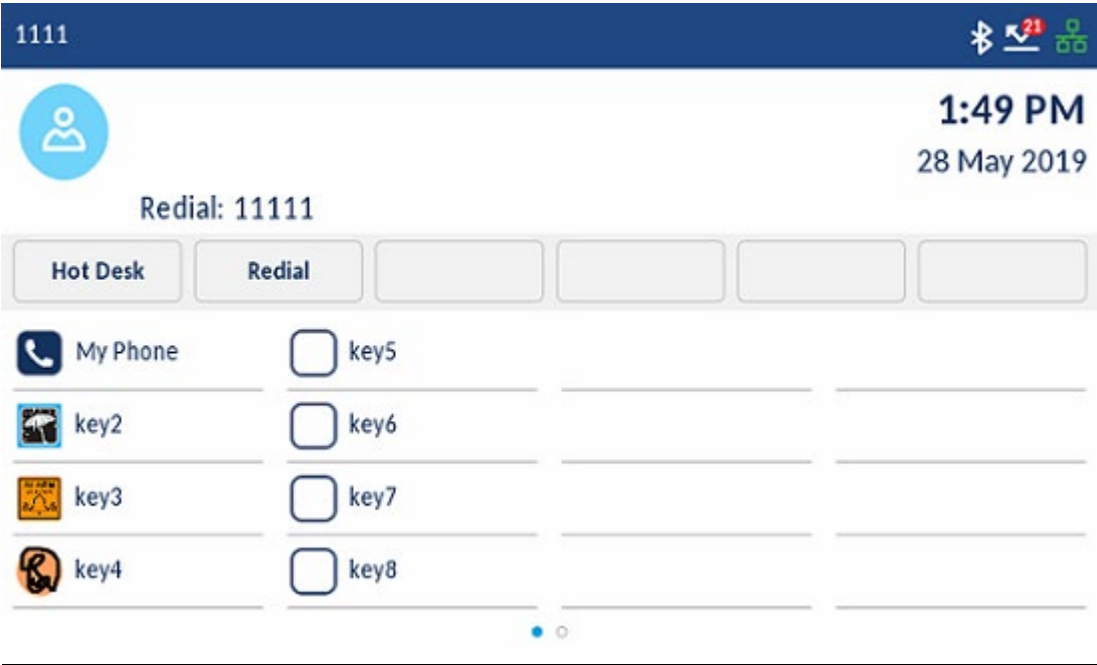


Figure 64: PhoneSoftkey Example (6940)

3.10 IMAGEMENU OBJECT (6920/6930/6940)

The ImageMenu object allows using a bitmap image to serve as a menu. This is desirable when a user wants to display menu choices in some non-ASCII character set or with pictures only. Each menu selection is linked to a keypad key (0-9, *, #).allows developers to create a numerical list of choices.

3.10.1 IMPLEMENTATION (6920)

For the 6920, the image is either a 24/32 bit depth “png” or a “jpeg” file located on a server and which can be downloaded by the phone using TFTP, FTP, HTTP or HTTPS. See chapter 4.2.1.1 for more details.

Two image sizes are supported:

- Up to 320x180 pixels (mode=regular or extended).
- Up to 320x240 pixels (mode=fullscreen). In that case, the softkeys are not displayed.

If the image is bigger than the size supported by the requested mode, the image is clipped based on the requested alignment.

If the image is smaller than the size supported by the requested mode, the image is displayed based on the requested alignment

The **Left** key is mapped to the cancelAction tag, if not specified the XML object is destroyed.

Note:



- The left Arrow key default interaction is disabled if the LockIn tag is set to “yes”.
- If the LockIn tag is set to “yes” and the cancelAction tag is configured, pressing the left Arrow key triggers the configured cancelAction.
- The up, down, left and right arrow key default interactions can be overridden using the scrollUp, scrollDown, scrollLeft and scrollRight tags.

Object default Softkeys

Six (four physical) customizable softkeys are available for this object.

Position	Label	Interaction	URIs
4	Done	Exit: Redisplays the previous XML object present in the phone browser.	SoftKey:Exit

The following figure details how the AastraIPPhoneImageMenu is implemented on the Mitel 6920.



Figure 65: ImageMenu 6920 implementation (mode regular/extended and full screen)

Also a URI can be programmed to be called when the user presses the **Select** key (imageAction root tag). If this tag is empty, phone will use the URI configured by the doneAction tag (SoftKey:Exit by default).

So by default pressing the **Select** key destroys the current XML object unless an imageAction or a doneAction is configured.

Native Interaction

- “SoftKey:Exit”

3.10.2 IMPLEMENTATION (6930)

For the 6930, the image is either a 24/32 bit depth “png” or a “jpeg” file located on a server and which can be downloaded by the phone using TFTP, FTP, HTTP or HTTPS. 4.2.1.1 for more details.

Two image sizes are supported:

- Up to 480x204 pixels (mode=regular or extended).
- Up to 480x272 pixels (mode=fullscreen). In that case, the softkeys are not displayed.

If the image is bigger than the size supported by the requested mode, the image is clipped based on the requested alignment.

If the image is smaller than the size supported by the requested mode, the image is displayed based on the requested alignment

The **Left** key is mapped to the cancelAction tag, if not specified the XML object is destroyed.

Note:



- The left Arrow key default interaction is disabled if the LockIn tag is set to “yes”.
 - If the LockIn tag is set to “yes” and the cancelAction tag is configured, pressing the left Arrow key triggers the configured cancelAction.
 - The up, down, left and right arrow key default interactions can be overridden using the scrollUp, scrollDown, scrollLeft and scrollRight tags.
-

Object default Softkeys

Eight (five physical) customizable softkeys are available for this object.

Position	Label	Interaction	URIs
5	Done	Exit: Redisplays the previous XML object present in the phone browser.	SoftKey:Exit

The following figure details how the `AastraIPPhoneImageMenu` is implemented on the Mitel 6930.

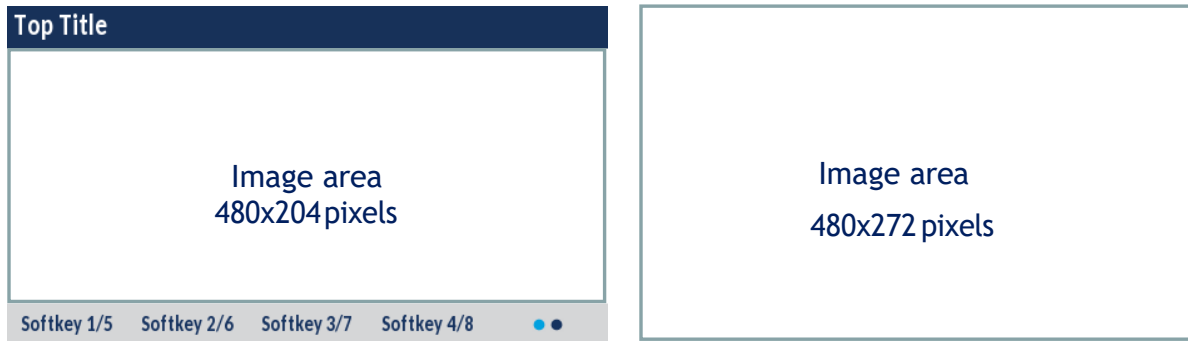


Figure 66: ImageMenu 6930 implementation (mode regular/extended and full screen)

Also a URI can be programmed to be called when the user presses the **Select** key (`imageAction` root tag). If this tag is empty, phone will use the URI configured by the `doneAction` tag (`SoftKey:Exit` by default).

So by default pressing the **Select** key destroys the current XML object unless an `imageAction` or a `doneAction` is configured.

Native Interaction

- “SoftKey:Exit”

3.10.3 IMPLEMENTATION (6940)

For the 6873i, the image is either a 24/32 bit depth “png” or a “jpeg” file located on a server and which can be downloaded by the phone using TFTP, FTP, HTTP or HTTPS. See chapter 4.2.1.1 for more details.

Two image sizes are supported:

- Up to 800x372 pixels (mode=regular or extended).
- Up to 800x480 pixels (mode=fullscreen). In that case, the softkeys are not displayed.

If the image is bigger than the size supported by the requested mode, the image is clipped following the requested alignment.

If the image is smaller than the size supported by the requested mode, the image is displayed following the requested alignment.

Note: as the 6940 does not have navigation keys, the scroll actions are mapped to screen swipes



- `scrollUp`, by swiping from top to bottom
- `scrollDown`, by swiping from bottom to top
- `scrollLeft`, by swiping from left to right
- `scrollRight`, by swiping from right to left

Object default Softkeys

Ten (six physical) customizable softkeys are available for this object.

Position	Label	Interaction	URIs
6	Done	Exit: Redisplays the previous XML object present in the phone browser.	SoftKey:Exit

The following figure details how the `AastraIPPhoneImageMenu` is implemented on the Mitel 6940.

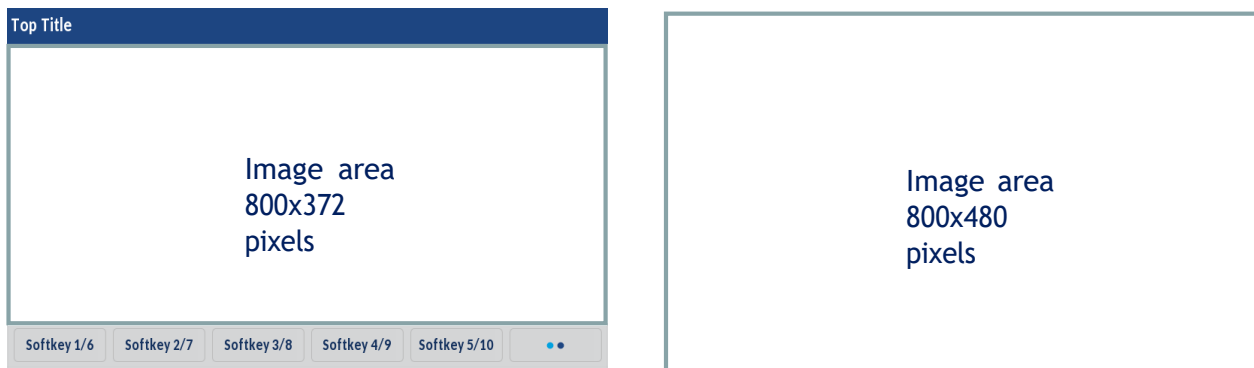


Figure 67: ImageMenu 6940 implementation (mode regular/extended and full screen)

Also a URI can be programmed to be called when the user presses on the displayed image (`imageAction` root tag). If this tag is empty, phone will use the URI configured by the `doneAction` tag (`SoftKey:Exit` by default).

So, by default, pressing on the image destroys the current XML object unless an `imageAction` or a `doneAction` is configured.

Native Interaction

- “SoftKey:Exit”

3.10.4 XML DESCRIPTION

```

<AastraIPPhoneImageMenu
    destroyOnExit = "yes/no"
    cancelAction = "some URI"
    doneAction = "some URI"
    imageAction = "some URI"
    Beep = "yes/no"

    Timeout = "some integer"
    bgColor="white/black..."
    LockIn = "yes/no"

    CallProtection = "yes/no/notif"
    GoodbyeLockInURI = "some URI"
    allowDTMF = "yes/no"

    scrollUp = "some URI"
    scrollDown = "some URI"
    scrollLeft = "some URI"
    scrollRight = "some URI"

    mode = "regular/extended/fullscreen"

>

    <TopTitle    icon="icon index"

                Color="white/black..."

    >Top Title</TopTitle>

    <Image

                verticalAlign = "top,middle,bottom"
                horizontalAlign = "left,middle,right"
                height = "height in pixels"

                width = "width in pixels"

    >Image as hexadecimal characters or URL</Image>

    <!--Base attribute is optional-->

    <URIList base = "http://someserver/">

        <URI key = "0">link1.php</URI>

        <URI key = "#">link3.php</URI>

        <!--Additional URI entries may be added (0-9,* and #)-->

    </URIList>

    <!--Additional Softkey Items may be added -->

</AastraIPPhoneImageMenu>

```

XML Document Objects

Document Object	Position	Type	Comments
AstralPPhoneImageMenu	Root tag	Mandatory	Root object
destroyOnExit	Root tag	Optional	“yes/no” indicates if the object is kept or not in the phone browser after exit. If not specified, the object is kept in the browser.
cancelAction	Root tag	Optional	Defines the URI to be called when the user cancels the XML object.
doneAction	Root tag	Optional	Defines the URI to be called when the user selects the “Done” softkey or ‘SoftKey:Exit’.
imageAction	Root tag	Optional	Defines the URI to be called when the user presses on the image. <i>6920, 6930 and 6940 only.</i>
Beep	Root tag	Optional	“yes” or “no” to indicate if a notification beep must be generated by the phone.
Timeout	Root tag	Optional	Overrides the default 45 seconds timeout of the UI XML object. A Timeout set to “0” will disable the timeout feature. See section 4.3.2 for more details
bgColor	Root tag	Optional	Set the background color of the object. the possible values are detailed in chapter 4.2.3. If not specified, the default value is “white”.
LockIn	Root tag	Optional	If set to “yes”, the phone ignores all events that would cause the screen to exit without using the keys defined by the object. Default value is “no”. See section 4.3.3 for more details.
CallProtection	Root tag	Optional	If set to “yes”, the phone will not destroy the XML object being displayed on an incoming call. If set to “notif”, the behavior is the same as “yes” but a notification is displayed on the screen providing the caller ID details.
GoodbyeLockInURI	Root tag	Optional	Valid only if LockIn=“yes”, this tag defines a URI to be called when the “Goodbye” key is pressed during a locked XML session. This URI overrides the native behavior of the “Goodbye” key which is to destroy the current XML object displayed.

Document Object	Position	Type	Comments
allowDTMF	Root tag	Optional	This tag allows letting keypad strokes as DTMF when the phone is in the connected status. Allow the DTMF keys for Idle state if CallProtection is enabled and allowDTMF is set to “yes”
scrollUp	Root tag	Optional	This tag allows overriding the default behavior of the Up arrow key. <i>Not available on 6940</i>
scrollDown	Root tag	Optional	This tag allows overriding the default behavior of the Down arrow key. <i>Not available on 6940</i>
scrollLeft	Root tag	Optional	This tag allows overriding the default behavior of the Left arrow key. <i>Not available on 6940</i>
scrollRight	Root tag	Optional	This tag allows overriding the default behavior of the Right arrow key. <i>Not available on 6940</i>
mode	Root tag	Optional	Configures the display mode, “regular”, “extended” or “fullscreen”.
TopTitle	Body	Optional	Text to be used as top title for the object.
Icon	TopTitle tag	Optional	Index of the icon to be used for the top title.s
Color	TopTitle tag	Optional	Label color, the possible values are detailed in chapter 4.2.3. If not specified, the default value is “white”.
Image	Body	Mandatory	Image to be displayed as a series of hexadecimal characters or a URL. See section 4.2.1 for more details.
verticalAlign	Image tag	Optional	Vertical position of the image (“top”, “middle” or “bottom”). If the tag is not specified, the object will use “middle” as a default value.
horizontalAlign	Image tag	Optional	Horizontal position of the image (“left”, “middle” or “right”). If the tag is not specified, the object will use “middle” as a default value.
height	Image tag	Mandatory	Height in pixels. Must match the image height.
width	Image tag	Mandatory	Width in pixels. Must match the image width.
URIList	Body	Mandatory	Master tag of the URI list linked to a keypad key (0-9, * and #)
Base	URIList tag	Optional	The value of this attribute is pre-pended to the value in the URI tags.
URI	URIList body	Mandatory	URI to be used if the user presses the key defined in the “key” tag.

SoftKey	Body	Optional	See section 4.1 for details
---------	------	----------	-----------------------------

3.10.5 EXAMPLES

XML Example (6930)

```

<AastraIPPhoneImageMenu destroyOnExit="yes">
  <Image>tftp://server.com/menu.png</Image>
  <URIList> key="1">http://myserver.com?choice=1</URI>
  <URI key="2">http://myserver.com?Choice=2</URI>
</URIList>
  <SoftKey index="1"><Label>Label</Label>
  <URI>http://myserver.com/script.php?action=1</URI>
</SoftKey>
  <SoftKey index="5">
    <Label>Exit</Label>
    <URI>SoftKey:Exit</URI>
  </SoftKey>
</AastraIPPhoneImageMenu>

```

Resulting Screen



Figure 68: ImageMenu Example (6930)

3.11 PHONESTATUS OBJECT

The `AastraIPPhoneStatus` object provides the ability to display a status message or icon on the phone's idle screen when XML information is pushed from the servers.

If the phone receives multiple messages, the first message received displays first and the remaining messages scroll consecutively one at a time.

The `AastraIPPhoneStatus` object supports 2 modes for the information to be displayed on the idle screen:

- Icon mode (6920/6930 and 6940 only): **up to 4 icons** are displayed on the top line of the display and remain there until they are removed/replaced (by the server) or after a phone reboot.
- Toaster mode (6920/6930 and 6940 only): the message is displayed in the “toaster” notification which scrolls under the status bar.

➔ **Note:** You can set the amount of time, in seconds, that a message displays to the phone before scrolling to the next message (See *Scroll delay* below).

3.11.1 IMPLEMENTATION (6920/6930/6940)

The phones display the messages on the second line in the phone idle screen (where “No Service” would be displayed if there was no service. If there is no service on the phone, the “No Service” message overrides the XML object message).

The following graphic shows how the different types of message are displayed.

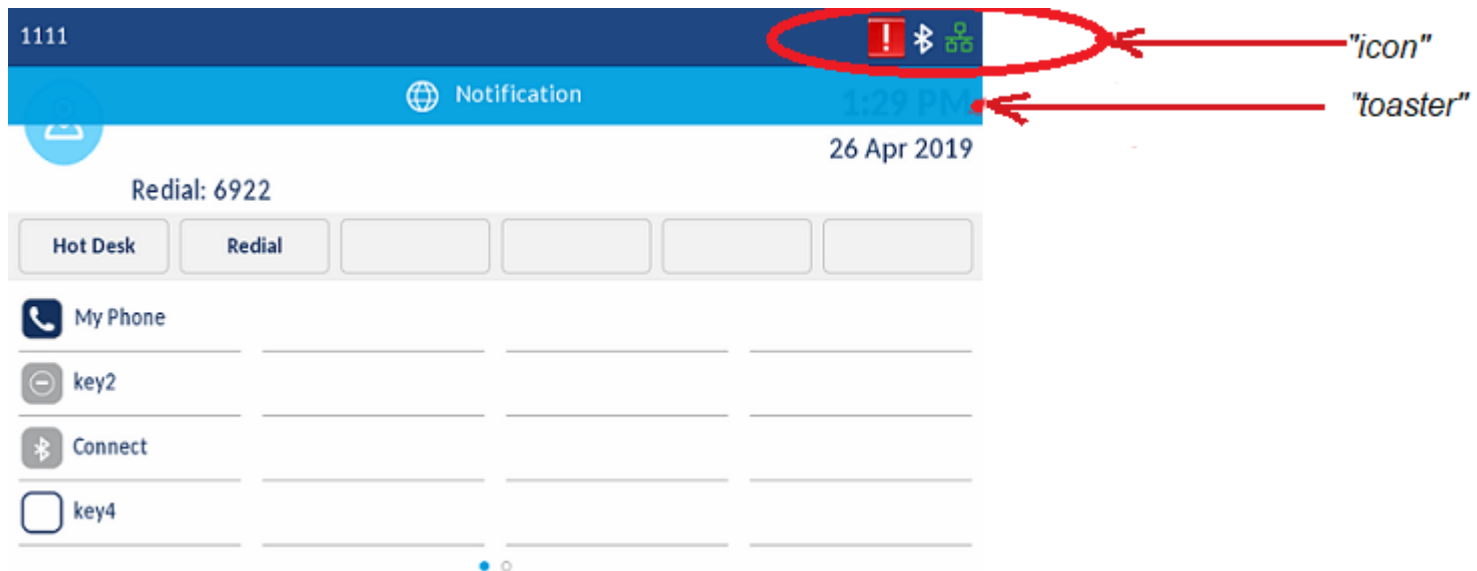


Figure 69: PhoneStatus Object (6940)

On the 6940 the user can also press the “toaster” notification to execute the URI attached to the message if any.

3.11.2 XML DESCRIPTION

```
<AastraIPPhoneStatus

  Beep = "yes/no"

  triggerDestroyOnExit = "yes/no">

    <Session>Session ID</Session>

    <Message

      Index = "index"

      Color="white/black..."

      Type = "normal/alert/icon/toaster"
      Timeout = "timeout"

      URI = "some URI"

      icon = "icon index"

    >Message</Message>

    <!--Additional Message Items may be added -->

    <!--Additional Icon Items may be added -->

  </AastraIPPhoneStatus>
```

Notes:

- The *Session ID* must be unique to the application sending the XML object to the phone. It is up to the application to generate that session ID, which does not have to be limited to just numbers. It could be a combination of letters and numbers. There could only be one Session tag per PhoneStatusMsg object. If the Session tag is not provided, the phone assumes a default value (0) for it; this can be used if you don't have multiple applications displaying messages on the idle screen.
- ➡ • The type="icon" indicates the icon mode. The four icons are referenced by the index value (0 to 3), the session value is ignored. Dynamic icons are supported.
- The type="toaster" indicates the toaster mode. The message is displayed in an animated bar under the phone status bar, Dynamic icons are supported.

The following table details which tags are relevant for each message type.

	<u>Session</u>	<u>Index</u>	<u>Color</u>	<u>Timeout</u>	icon	<u>URI</u>
--	----------------	--------------	--------------	----------------	------	------------

icon		✓			✓	
<u>toaster</u>					✓	✓ (6940 only)

XML Document Objects

Document Object	Position	Type	Comments
AstralPPhoneStatus	Root tag	Mandatory	Root object
Beep	Root tag	Optional	“yes” or “no” to indicate if a notification beep must be generated by the phone.
triggerDestroyOnExit	Root tag	Optional	If the XML object is sent as an answer to a UI object, setting this parameter to “yes” will trigger the exit of the calling object as if it was a UI object. See section 4.3.4 for more details.
Session	Body	Optional	Session ID used to identify the application displaying message. It allows message change and message reset. Ignored if type is “icon”.
Message	Body	Mandatory	Message to be displayed or empty to reset the message. In “icon” mode this is the value of the counter to be displayed on the icon if value is superior to zero.
index	Message tag	Mandatory	Index of the message for the session, the index starts at 0, ignored if type is “toaster”. If type is “icon” the index must be from 0 to 3 (4 icons).
type	Message tag	Optional	Type of message, “icon” to indicate an update of the status bar icons. “toaster” to indicate that the message is displayed in the animated notification under the status bar.

			If not specified, the message stays on the screen until it is reset by an empty message or after a phone reboot.
URI	Message tag	Optional	URI to be called when the user presses the notification message button. <i>6940 only</i>
icon	Message tag	Optional	Index of the icon to be used for this message.
IconList	Body	Optional	List of icons used in the object.

3.11.3 EXAMPLES

XML Example1

This XML example displays in the status bar

- the custom “aa.png” icon (at index 2)
- the custom “bb.png” icon (at index 3)

```
<AastraIPPhoneStatus>
<Session/>
<Message index="2" type="icon" icon="2"/>
<Message index="3" type="icon" icon="3"/>
<IconList>
<Icon index="2">tftp://10.112.123.82/aa.png</Icon>
<Icon index="3">tftp://10.112.123.82/bb.png</Icon>
</IconList>
</AastraIPPhoneStatus>
```



Figure 70: PhoneStatus icons example



Note: The `PhoneStatus` object can also be used to remove icons from the display using the same `index` but setting the icon to “Icon:None”.

To just remove the “bb.png” icon, the following content must be sent

```
<AastraIPPhoneStatus>
<Session/>
<Message index="3" type="icon" icon="3"/>
<IconList>
<Icon index="3">Icon:None</Icon>
</IconList>
</AastraIPPhoneStatus>
```

Or

```
<AastraIPPhoneStatus>
<Session/>
<Message index="3" type="icon"/>
</AastraIPPhoneStatus>
```



Figure 71: PhoneStatus icons example

XML Example 2

This XML example displays a “toaster” notification using the standard icon and another one with a custom icons.

```

<AastraIPPhoneStatus>

  <Session/>

  <Message type="toaster">External notification standard icon</Message>

  <Message type="toaster" icon="1">External notification custom icon</Message>

  <IconList>

    <Icon index="1">Icon:Home</Icon>

  </IconList>

</AastraIPPhoneStatus>

```

Resulting screen



Followed by



Figure 72: PhoneStatus toaster example

4 XML EXTENSIONS

4.1 CUSTOMIZABLE SOFTKEYS ((6920/6930/6940))

The Softkey object can be used to override the default softkeys in each of the XML objects. It allows developers to link arbitrary URIs to keys in the XML screens and invoke softkey behavior native to each XML screen type.

XML Description

6 softkeys are available (8 for the 6930, 10 for the 6940)

```
<SoftKey index = "1-10" >
<Label>Text</Label>
<URI>
http(s)://someserver/somepage
OR SoftKey:XXXXXX
OR Dial:somenummer
</URI>
</SoftKey>
<!--As many as used in the softkey definition -->
```

Notes:



- Custom softkeys are only available for the UI XML objects.
- If you use custom softkeys, the default softkeys of the XML object are no longer displayed. This means they have to be recreated as custom softkeys.

XML Document Objects

Document Object	Position	Type	Comments
SoftKey	Body	Mandatory	Softkey Root object (up to 6 or 10 for 6940)
Index	SoftKey Root tag	Mandatory	Indicates the softkey number
Label	SoftKey Body	Mandatory	Label of the softkey
URI	SoftKey Body	Mandatory	URI called if the softkey is pressed

Available object commands

The following softkey functionality is available to the developer for the purpose of reordering or preserving the default functionality of the XML screens.

- “SoftKey:Select” is available for AastralPPhoneTextMenu only and calls the URI tag of the selected MenuItem.
- “SoftKey:Exit” is available for all UI XML objects and redisplay the previous XML object present in the phone browser.
- “SoftKey:Dial” is available to screens that allow input. The dial string for the “Dial” function is taken from the menu items URI on the Menu Screen, and from the editor field input on the Input Screen.
- “SoftKey:Dial2” is available only for the AastralPPhoneTextMenu, AastralPPhoneTextScreen and AastralPPhoneFormattedTextScreen objects and will dial the number set by the “Dial” tag.
- “SoftKey:Submit” is available only for the AastralPPhoneInputScreen object, it completes the user input by submitting the programmed URI (URL tag) and value (Parameter tag).
- “SoftKey:BackSpace” is available only for the AastralPPhoneInputScreen object, it deletes the character placed before the cursor.
- “SoftKey:NextSpace” is available only for the AastralPPhoneInputScreen object, it inserts a “space” character at the cursor position.
- “SoftKey:Dot” is available only for the AastralPPhoneInputScreen object, it inserts a “.” character at the cursor position.

- “SoftKey:ChangeMode” is available only for the AstralPPhoneInputScreen object, it allows a toggle between lower case, upper case and digit inputs. “SoftKey:SymbolList=“XYZ”” is available only for the AstralPPhoneInputScreen object, it allows an input of a custom list of characters at the cursor position.

The format is:

```
<SoftKey index="1">
  <Label>Symbols</Label>
  <URI>SoftKey:SymbolList="@."</URI>
</SoftKey>
```

The content of the Symbol List must be encapsulated with quotes.

 **Note:** SymbolList supports only standard ASCII characters.

Note that there can be multiple SymbolList softkeys with different list of symbols. There are some special characters that need to be encoded due to XML limitations (see chapter 2.6 for the encoded value).

Available object commands

“SoftKey:Exit” is available for all UI XML objects and redisplayes the previous XML object present in the phone browser.

“SoftKey:Submit” is available for AstralPPhoneInputScreen.

Softkey availability per object

SoftKey	TextScreen	TextMenu	IconMenu	InputScreen
Select		X	X	
Exit	X	X	X	X
Dial		X	X	X
Dial2	X	X	X	
Submit				X
BackSpace				X
NextSpace				X
Dot				X
ChangeMode				X
SymbolList				X

SoftKey	Formatted TextScreen	ImageScreen	ImageMenu
Select			
Exit	X	X	X

SoftKey	Formatted TextScreen	ImageScreen	ImageMenu
Dial			
Dial2	X		
Submit			
BackSpace			
NextSpace			
Dot			
ChangeMode			
SoftKey List			

You can define up to six, eight or ten softkeys depending on the phone model before the closing tag of any object.

4.2 GRAPHICS (6920/6930/6940)

4.2.1 IMAGES

4.2.1.1 Implementation on the 6920

The image tag is used by the ImageScreen and ImageMenu XML objects.

Format

The 6920 support 24/32 bits depth png or jpeg files, the image tag must include the URL to retrieve the png or jpeg file. The supported protocols are:

- TFTP
- HTTP
- HTTPS

The area available to display the picture is 320x240 pixels, the phone uses the provided width and height tags as the display zone, if the image sent is bigger than the display zone, the picture is clipped using the top left corner as the origin of the clipping area.

Same thing if the width or height is larger than the supported resolution, the phone will use the minimum value and will clip if needed.

If the phone cannot retrieve the picture using the provided URL, the “unknown” image is displayed.

4.2.1.2 Implementation on the 6930

The image tag is used by the ImageScreen and ImageMenu XML objects.

Format

The 6930 support 24/32 bits depth png or jpeg files, the image tag must include the URL to retrieve the png or jpeg file. The supported protocols are:

- TFTP
- HTTP
- HTTPS

The area available to display the picture is 480x272 pixels, the phone uses the provided width and height tags as the display zone, if the image sent is bigger than the display zone, the picture is clipped using the top left corner as the origin of the clipping area.

Same thing if the width or height is larger than the supported resolution, the phone will use the minimum value and will clip if needed.

If the phone cannot retrieve the picture using the provided URL, the “unknown” image is displayed.

4.2.1.3 Implementation on the 6940

The image tag is used by the ImageScreen and ImageMenu XML objects.

Format

The 6940 support 24/32 bits depth png or jpeg files, the image tag must include the URL to retrieve the png or jpeg file. The supported protocols are:

- TFTP
- HTTP
- HTTPS

The area available to display the picture is 800x480 pixels, the phone uses the provided width and height tags as the display zone, if the image sent is bigger than the display zone, the picture is clipped using the top left corner as the origin of the clipping area.

Same thing if the width or height is larger than the supported resolution, the phone will use the minimum value and will clip if needed.

If the phone cannot retrieve the picture using the provided URL, the “unknown” image is displayed.

4.2.2 ICONS

The 6920/6930/6940 also support icons in some XML objects:

- AastralPPhoneTextMenu in the menuitem tags
- All UI objects in TopTitle tag

But not in the Custom softkeys for the UI XML objects.

Two types of icons are supported:

- Predefined icons residing in the phone identified by a unique name
- Dynamic icons identified by a URI, the icon files are downloaded by the phone every time they are needed

Predefined icons

The predefined icons shown below are directly available from the phone. To use the predefined icons you must set the Icon tag to Icon:IconName where IconName is the name of the predefined icon.

Each XML icon has 2 variants:

- One for the regular UI XML application designed for a white background
- One for the status bar designed for a dark background

Name	Icon (XML app)	Icon (status bar)	Name	Icon (XML app)	Icon (status bar)
Icon:Add			Icon:Park		

Name	Icon (XML app)	Icon (status bar)	Name	Icon (XML app)	Icon (status bar)
Icon:Alarm			Icon:PhoneConnected		
Icon:AlarmFilled			Icon:PhoneDial		
Icon:ArrowDown			Icon:PhoneDiverted		
Icon:ArrowLeft			Icon:PhoneDivertedA		
Icon:ArrowRight			Icon:PhoneDivertedB		
Icon:ArrowUp			Icon:PhoneOffHook		
Icon:ArrowUpAndDown			Icon:PhoneOnHold		
Icon:BLF_Busy			Icon:PhoneOnHook		
Icon:BLF_Hold			Icon:PhoneRecall		
Icon:BLF_Ringing			Icon:PhoneRinging		
Icon:BLF_Unknown			Icon:PhoneRingingA		
Icon:Book			Icon:PhoneRingingB		
Icon:Calendar			Icon:PresenceAbsent		
Icon:CallFailed			Icon:PresenceAvailable		
Icon:CallFwd			Icon:PresenceBusy		
Icon:CallTransfer			Icon:PresenceMeeting		
Icon:CellPhone			Icon:PresenceNotAvailable		
Icon:CheckBoxCheck			Icon:PresenceSignedOut		

Name	Icon (XML app)	Icon (status bar)	Name	Icon (XML app)	Icon (status bar)
Icon:CheckBoxUnCheck			Icon:PresenceUnknown		
Icon:CircleBlue			Icon:Prohibit		
Icon:CircleGreen			Icon:Reset		
Icon:CircleRed			Icon:RingTone		
Icon:CircleYellow			Icon:RoomInspected		
Icon:Delete			Icon:RoomNotClean		
Icon:DND			Icon:RoomNotInspected		
Icon:Edit			Icon:RoomOccupied		
Icon:Envelope			Icon:RoomVacant		
Icon:EnvelopeOpen			Icon:Save		
Icon:GroupRoomOccupied			Icon:Search		
Icon:GroupRoomVacant			Icon:Settings		
Icon:Home			Icon:Speaker		
Icon:Incoming			Icon:StarBlue		
Icon:IncomingMissed			Icon:StarYellow		
Icon:Information			Icon:TailArrowDown		
Icon:Key			Icon:TailArrowUp		

Name	Icon (XML app)	Icon (status bar)	Name	Icon (XML app)	Icon (status bar)
Icon:Lock			Icon:UnLock		
Icon:MissedCall			Icon:WakeUpExpired		
Icon:Mute			Icon:WakeUpPending		
Icon:Office			Icon:Warning		
Icon:Outgoing			Icon:World		
Icon:OutgoingMissed					

Icons declaration

Icons are identified by an index in the XML objects; this index refers to the IconList tag which must be present each an icon is used. For AastralPPhoneSoftkey the icon name is directly used.

```
<IconList>
  <Icon index = "iconindex">Icon:Iconname or file location</Icon>
  <Icon index = "iconindex">Icon:Iconname or file location</Icon>
  <!--As many as used in the object definition -->
  <!--Up to 22 icons, index can be 1 to 21 -->
</IconList>
```

Dynamic icons

Dynamic icons are identified by an URI which is the location of the graphical file on a server.

Icon files must be:

- png files
- resolution 28x28 pixels (6920/6930i) and 40x40 pixels (6940), if a larger icon is used, the icon is clipped by the phone to the expected size
- 24 bits or 32 bits depth
- Less than 150 kB

Supported protocols are:

- TFTP
- HTTP
- HTTPS

Example

```
<IconList>
  <Icon index="1">http://myserver/icons/icon.png</Icon>
</IconList>
```

4.2.3 COLORS

The phones support 26 colors which can be used in the XML applications for labels and backgrounds.

The following table provides the XML color map.

Color name	HTML Value	RGB value	Display
white	#FFFFFF	255,255,255	
black	#000000	0,0,0	
darkred	#C51D23	197,29,35	
red	#ED1C24	237,28,36	
lightred	#F26065	242,96,101	
darkgreen	#399041	57,144,65	
green	#3FAC49	63,172,73	
lightgreen	#78C57F	120,197,127	
darkblue	#15325F	21,50,95	
blue	#0081B3	0,129,179	
lightblue	#4CBDEF	76,189,239	
darkyellow	#CDA806	205,168,6	
yellow	#F8CA00	248,202,0	
lightyellow	#FADA4C	250,218,76	
darkgray	#404141	64,65,65	
gray	#BFBFC0	191,191,192	
lightgray	#EDEEEF	237,238,239	
darkbrown	#2B1D0E	43,29,14	
brown	#52361B	40,26,13	
lightbrown	#9F6934	159,105,52	
darkmagenta	#8B008B	139,0,139	
magenta	#FF00FF	255,0,255	
lightmagenta	#FF46FF	255,70,255	
darkcyan	#008B8B	0,139,139	

Color name	HTML Value	RGB value	Display
cyan	#00FFFF	0,255,255	
lightcyan	#E0FFFF	224,255,255	

Figure 73: XML colors (6920/6930/6940)

4.3 SPECIAL ATTRIBUTES

4.3.1 BEEP

You can enable or disable a Beep option in all the Mitel XML objects via the Beep attribute in the root tag. When the phone receives an XML object, the Beep notifies the user that an object is being displayed.

This attribute is optional. If the Beep attribute is set to "yes" (i.e. Beep="yes") then it is an indication to the phone to sound a beep when it receives the object. If the Beep attribute is set to "no" (i.e. Beep="no") or not present, then the default behavior is no beep is heard when the object arrives to the phone.

4.3.2 TIMEOUT

The Timeout attribute is an optional root tag attribute for all of the current UI XML objects. When the phone receives an XML object with this attribute set it will override the default 45 seconds timeout specified for custom applications.

Setting Timeout to "0" will disable the timeout feature. It's not recommended to set Timeout = 0 as it may cause burn in effect on phone's display LCD.

This timeout is reset by button presses.

4.3.3 LOCKIN

The LockIn attribute is an optional root tag attribute for all of the current UI XML objects. This attribute allows the XML designer to specify that a screen cannot be cancelled.

When a phone receives an UI XML object with the attribute set to "yes" it ignores all events that would cause the screen to exit without using the keys defined by the object.



Note: An incoming call still cancels the "locked" screen as telephony events have priority over XML applications.

Setting LockIn will disable the default timeout feature (45 seconds) unless the Timeout attribute is also set in the root tag.

4.3.4 TRIGGERDESTROYONEXIT

The triggerDestroyOnExit is an optional root tag attribute for all of the current non-UI XML objects (PhoneStatus, PhoneExecute and PhoneConfiguration). Its default value is "no".

By default, if a UI XML object gets a non-UI XML object as an answer to an exit URI ("Select" on a TextMenu or "Done" on a InputScreen...) the UI XML object is not destroyed and stays on the phone display even if its destroyOnExit tag is set to "yes".

By setting triggerDestroyOnExit to "yes", the previous UI XML object is destroyed if its destroyOnExit tag is set to "yes".

4.3.5 CALLPROTECTION

The CallProtection attribute is an optional root tag for all the current UI XML objects. This attribute allows the XML application designer to control the effect of an incoming call on the XML screen. It complements the LockIn attribute but is completely independent of it.

The possible values are:

- "no", an incoming call will cancel the current screen (even if LockedIn is enabled), this is the default behavior.

- “yes”, an incoming call will not cancel the current screen
- “notif”, an incoming call will not cancel the current screen and a toaster notification is displayed providing the caller ID (name/number). Not supported on 6863i and 6865i.

When set to “yes” or “notif”, the capability for the user to answer the call (off-hook, Line key, hands-free, ...) will depend on the `LockedIn` attribute.

4.4 TEXTMENU OR ICONMENU USER SELECTION (6920, 6930, 6940)

It is also possible to send information related to a user’s choice in a Text Menu.

“Select” and “Dial” system keys use the URI passed in the MenuItem attribute, a custom key might need to know the user selection in the TextMenu, and this is the purpose of the **Selection** attribute.

When a user accesses an arbitrary URI via a custom softkey, the Text Menu will send along a bit of information regarding the user’s currently selected option using an optional XML attribute **Selection**.

XML Example

```
<AstralPPhoneTextMenu destroyOnExit = "yes">
<Title>Parameter Tester</Title>
<MenuItem>
  <Prompt>First Selection</Prompt>
  <URI>http://someserver/somepage.xml</URI>
  <Selection>dataToAppend</Selection>
</MenuItem>
<SoftKey index = "1">
  <Label>Custom Key</Label>
  <URI>http://someotherserver/someotherpage.xml</URI>
</SoftKey>
</AstralPPhoneTextMenu>
```

When the user selects item 1 and presses softkey 1, the URI requested is **http://someotherserver/someotherpage.xml?selection=dataToAppend**

Notes:



- If a “?” already exists in the URI, then a “&” is used to separate the parameters. Note the parameter name “selection” is automatic. If the Selection attribute is omitted, then nothing extra is appended to the URI.
- the Selection attribute can be more than one parameter, for instance, the following Selection attribute is valid `<Selection>200&action=set</Selection>` (don’t forget to escape encode the attribute for the “&”).

4.5 “SELECT” AND “DIAL” IN THE SAME TEXTMENU OR ICONMENU OBJECT

The only way to have a “Select” and “Dial” keys in the same text menu, the only option is to use “Dial” as a system command and mimic “Select” with a custom key.

```

<AstralPPhoneTextMenu destroyOnExit = "yes">
<Title>Dial and Select</Title>
<MenuItem>
<Prompt>First Selection</Prompt>
<URI>200</URI>
<Selection>200</Selection>
</MenuItem>
<MenuItem>
<Prompt>Second Selection</Prompt>
<URI>201</URI>
<Selection>201</Selection>
</MenuItem>
<SoftKey index = "1">
<Label>Dial</Label>
<URI>SoftKey:Dial</URI>
</SoftKey>
<SoftKey index = "2">
<Label>Select</Label>
<URI>http://myserver.com/script.php</URI>
</SoftKey>
</AstralPPhoneTextMenu>

```

When the user selects item 1 and presses

Softkey 1, the phone dials 200

Softkey 2, the requested URI requested is <http://myserver.com/script.php?selection=200>

4.6 "SELECT", "DIAL" AND "DIAL2" BEHAVIOR IN A TEXTMENU OR ICONMENU OBJECT

The following table describes what actions will be performed for each possible combination of items and actions in a <MenuItem>.

MenuItem Content	Select Softkey	Offhook	Dial2 Softkey	Dial Softkey
<URI>http://XYZ.</URI>	GET(http://XYZ)	-	-	Dial(http://XYZ)
<Dial>123</Dial>	-	Dial(123)	Dial(123)	-
<URI>http://XYZ.</URI> <Dial>123</Dial>	GET(http://XYZ)	Dial(123)	Dial(123)	Dial(http://XYZ)
<URI>Dial:456</URI> <Dial>123</Dial>	Dial(456)	Dial(123)	Dial(123)	Dial(Dial:456)



Note: the main purpose of the "Dial" tag in the TextMenu is to allow the Mitel phones to dial a number going off-hook.

4.7 REFRESH OF AN XML PAGE

All XML UI objects have the ability to be refreshed. This is accomplished by adding a Refresh setting to the HTTP header. This setting comprises two parameters:

- a time in seconds
- a URL.

The Refresh setting is set by the XML application and it is up to the application to decide which objects they want to refresh. So, it is an optional setting.

If the setting appears, then both parameters must be set for the Refresh to work.

The format of the HTTP header is:

```
Refresh: timeout; url=page to load
```

Example

```
Refresh: 3; url=http://10.50.10.140/update.xml
```

All of the parameters are mandatory.

With php you will have to use the header command in your script.

```
header("Refresh: 1; url= http://10.50.10.140/update.xml");
```

4.8 HTTP HEADERS FORMAT FOR A PHONE HTTP GET

When the Mitel performs a GET to an HTTP server, it sets multiple variables in the HTTP header:

- 'User-Agent' providing information on the characteristics of the phone (type, MAC address, firmware),
- 'Accept-Language' providing information on the language used on the phone.

Example:

```
User-Agent: MitelMINET_6940 MAC:08-00-0f-19-94-B7 V:1.2.0.26  
Accept-Language: en
```

4.8.1 USER-AGENT

The User-Agent header includes:

- The type of phone
- The MAC address of the phone
- The firmware version of the phone

Usually, the HTTP server is able to retrieve the User Agent parameters. The Mitel 69xx phones send the following string for the User-Agent.

Phone_Model[space]MAC:-XX-XX-XX-XX-XX-XX[space]V:Version

This allows the application to adapt to the type of phone it is dealing with but also allows to design a single application to provide XML (for the phones) and HTML (for a web browser).

The following source code is a php example that can be used to decode the User-Agent header sent by a Mitel 69xx phone.

```

function Decode_HTTP_header()
{
$user_agent=$_SERVER["HTTP_USER_AGENT"];
if(strpos($user_agent,"Aastra"))
{
$value=preg_split("/ MAC:/",$user_agent);
$end=preg_split("/ /",$value[1]);
$value[1]=preg_replace("/-/","", $end[0]);
$value[2]=preg_replace("/V:/","", $end[1]);
}
else
{
$value[0]="Unknown";
$value[1]="NA";
$value[2]="NA";
}
$header['model']=$value[0];
$header['mac']=$value[1];
$header['firmware']=$value[2];

return($header);
}

```

This function returns an array with

Array['model'] is the phone model

- "MitelMINET_6920"
- "MitelMINET_6930"
- "MitelMINET_6940"

Array['mac'] is the MAC address

Array['firmware'] is the firmware version

Example of execution

- Array['model'] => MitelMINET_6940
- Array['mac'] => 08000F031C81
- Array['firmware'] => 1.2.0.66

4.8.2 ACCEPT-LANGUAGE

The phone sends the standard **Accept-Language** in the headers request which describes the language of the phone configured by user. For example:

- "en" for English
- "fr" for French
- "de" for German
- "es-MX" Spanish-LA
- "pt-BR" Portuguese-Brazil
- "nl" for Dutch
- "it" for Italian
- "ro" for Romanian
- "pt" for Portuguese
- "es" for Spanish
- "ru" for Russian

- “sv” for Swedish
- “pl” for Polish

To access it using php the syntax is `$_SERVER["HTTP_ACCEPT_LANGUAGE"]`)

Example

```
echo $_SERVER["HTTP_ACCEPT_LANGUAGE"]
returns “en”
```

4.9 DIAL AND DIALLINE URIS

The phone supports 2 URI commands to launch an outgoing call from XML.

- Dial:XXXXX
- DialLine:Y:XXXXX

Where XXXXX is the number/URI to dial and Y is the Programmable key configured as SINGLE LINE or MULTICALL or KEY SYSTEM.

The difference between Dial and DialLine is that with the DialLine command you can specify which line to use, Dial will use the first available line on the phone.

Dial and DialLine are generic URI and can be used anywhere a URI is requested by an XML object, this can be a custom softkey, an item in a TextMenu.

4.10 LED CONTROL

You can control the status of the LED associated to a phone key **as long as the key is configured as an “URL Line” key.**

```
“Led:XXXXX=on/off/fastflash/slowflash[:green/yellow/red]”
```

Where XXXXX represents the key you want to modify the LED or “all” for all the keys.

The first part of the command controls the state of the physical LED attached to a softkey and, if the second part of the command is omitted, a matching color for the pseudo-LED.

The second part of the command controls the state of the pseudo-LED by allowing a color (red, green, yellow).



Note: the LED state is lost after a phone reboot.

The supported LED states are:

- on, the LED is steady on
- off, the LED is steady off
- slowflash, the LED blinks at a slow pace
- fastflash, the LED blinks at a fast pace

Keys supported on an expansion module (6920/6930/6940)

- expmodX key1 to expmodX key74
- Where X is the expansion module number (1 to 3)

4.10.1 IMPLEMENTATION (6920/6930)

Though the 6920/6930 has physical LEDs for the topsoftkeys, the LED status is also indicated on the screen via the pseudo-LED.

The following table shows the mapping between the LED state providing in the command with the physical LED and the pseudo-LED when the pseudo-LED state is not provided.

LED state	LED (physical led)	Pseudo-LED state	Pseudo-LED (top)
"off"	Off	"default"	 Top Softkey
"on"	On	"red"	 Top Softkey
"slowflash"	Slow flashing	"yellow"	 Top Softkey
"fastflash"	Fast Flashing	"yellow"	 Top Softkey

The following table shows the pseudo-LED based on its state, when specified the pseudo-LED state overrides the state mapping coming from the LED state.

Pseudo-LED state	Pseudo-LED (top)
"default"	 Top Softkey
"red"	 Top Softkey

"green"	 Top Softkey
"yellow"	 Top Softkey

4.10.2 IMPLEMENTATION (6940)

As the 6940 has no physical LED attached to the softkeys, the LED status is indicated on the screen via the pseudo-LED which blinks at the requested rate instead of showing a different color.

The following table shows the mapping between the LED state providing in the command with the physical LED and the pseudo-LED when the pseudo-LED state is not provided.

LED state	Pseudo-LED state	Icon/Border	Pseudo-LED (top)
"off"	"default"	Off	 Top Softkey

LED state	Pseudo-LED state	Icon/Border	Pseudo-LED (top)
"on"	"red"	Steady	 Top Softkey
"slowflash"	"red"	Slow blink	 Top Softkey
"fastflash"	"red"	Fast blink	 Top Softkey

The following table shows the pseudo-LED based on its state, when specified the pseudo-LED state overrides the state mapping coming from the LED state.

Pseudo-LED state	Pseudo-LED (top)
"default"	 Top Softkey
"red"	 Top Softkey
"green"	 Top Softkey
"yellow"	 Top Softkey

4.10.3 XML EXAMPLES

```
<AastraIPPhoneExecute>
  <ExecuteItem URI="Led: prgkey2=on"/>
  <ExecuteItem URI="Led: prgkey2=off"/>
  <ExecuteItem URI="Led: prgkey3=fastflash"/>
  <ExecuteItem URI="Led: expmod1 key1=slowflash"/>
  <ExecuteItem URI="Led: all=off"/>
</AastraIPPhoneExecute>
```

On the 6920/6930/6940 pseudo-LED can also be controlled.

```

<AastraIPPhoneExecute>
  <ExecuteItem URI="Led: prgkey2=on:green"/>
  <ExecuteItem URI="Led: prgkey5=on:yellow"/>
</AastraIPPhoneExecute>

```

The following example shows the difference in pseudo-LED behavior between a 6920/6930 and a 6940

```

<AastraIPPhoneExecute>
  <ExecuteItem URI="Led: prgkey2=on"/>
  <ExecuteItem URI="Led: prgkey3=on:green"/>
</AastraIPPhoneExecute>

```

Resulting screen (6940)

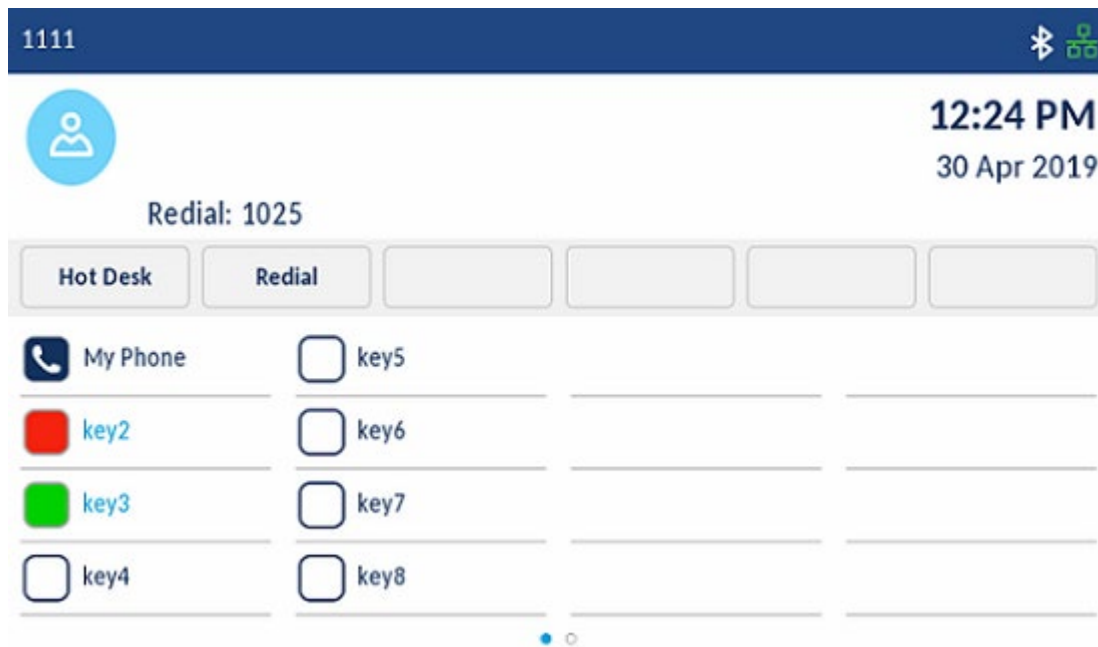


Figure 74: XML LEDs on 6940

Resulting screen (6930)

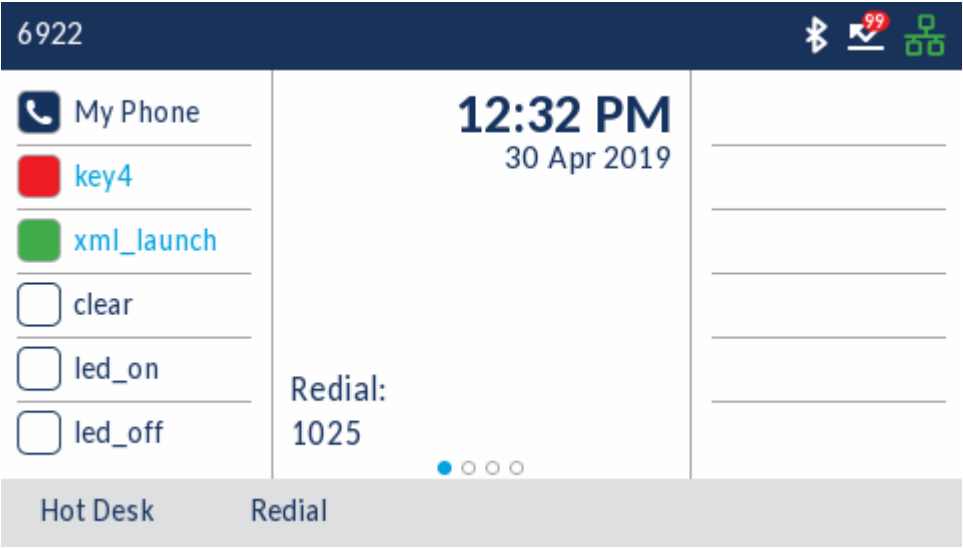


Figure 75: Figure 74: XML LEDs on 6930

4.11 KEYPRESS EMULATION

The Mitel minet Phones support a “Key” URI to allow the XML application to send an event that a key has been pressed on the phone. The system initiates the event as if the button was physically pressed.

The phone maps key events to the phone physical keys and not to their mapped logical keys.

Notes:



- If the key is a valid key, the phone executes the key regardless of the current phone status.
- Phone does not send a response detailing the success or failure of a simulated key.

Keypad

- Key:KeyPad0 to Key:KeyPad9 Numeric Keypad buttons 0-9
- Key:KeyPadStar *Star buttons
- Key:KeyPadPound # Hash button

Programmable keys (depending on model)

- Key:SoftKey1 to Key:SoftKey55 soft keys 1-55
- Key:PrgKey1 to Key:PrgKey11 programmable keys 1-11

Volume and Voice path control

- Key:VolDwn Decrease button
- Key:VolUp Increase button
- Key:Headset Headset
- Key:Speaker Speaker
- Key:Mute Mute

Note: for both keys (headset and speaker) the behaviour is as if the "speaker/headset" button is



pressed, and it does not switch to headset for "Headset" key event or to speaker for "Speaker" key event.

Telephony

- Key:Redial Redial
- Key:Callers Callers List
- Key:Directory Directory
- Key:Options Options



Note: Since the phones do not have physical keys for Pickup and Park, those features will be only available if they are mapped to a programmable or soft key

Navigation (not available on 6940)

- Key:NavUp Navigation up key
- Key:NavDwn Navigation down key
- Key:NavEnter Navigation center key (Enter)
- Key:NavLeft Navigation left key
- Key:NavRight Navigation right key

Example:

This example asks the phone to display the phone directory.

```
<AastraIPPhoneExecute>  
  
<ExecuteItem URI = "Key:Directory"/>  
  
</AastraIPPhoneExecute>
```

4.12 CRASH AND CONFIGURATION FILES RETRIEVAL

The phone has the capability to upload the last crash file as well as the configuration files to a pre-configured location. This operation can be triggered manually via the Web UI or via the phone (if configured) but also via an XML command.

Command

```
<AastraIPPhoneExecute>
<ExecuteItem URI="Command: UploadSystemInfo"/>
</AastraIPPhoneExecute>
```

Sending this command to the phone make the phone uploads the last crash file as well as log files to the configured server (using TFTP in this example). Uploaded files are indexed by the MAC address, date and time.

4.13 SOME DEVELOPMENT GUIDELINES

There are some simple rules that you should follow when you develop XML applications for the Mitel phones.

Don't forget the "Exit" key when you use custom softkeys

Place custom softkeys as they are for the standard objects, also it is better to use the same labels. The Exit key should preferably be placed in position 6.

Setting **destroyOnExit** attribute to "yes" is always preferable when you write complex applications as it is the only way to properly control the pages that are stacked in the phone.

Avoid using the Directory Object; TextMenu object with custom keys is much more flexible.

If you want to access data from the internet, it is preferable to use a RSS feed or a SOAP interface than Web scraping as Web sites frequently change their layout interface.

For XML applications using data from the Internet, if they are not real time you might consider to "cache" the data on the server and update them on a regular basis (a weekly update is necessary for Movie schedules but a 10 minutes update might be necessary for a weather application for severe conditions).

As the XML objects are limited to 10000 bytes, you might sometime hit this limit for a complex TextMenu with a lot of custom keys and custom icons. Don't forget to use the base attribute in the MenuItem object as it is a good way to reduce the size of the object.

5 URL FORMAT AND VARIABLES

5.1 URL FORMAT

To access to an XML application, the phone performs a HTTP (or HTTPS) GET on a URL. The supported syntax is:

`http://host[:port]/dir/file`

or

`https://host[:port]/dir/file`

where:

- host is the hostname or IP address of the Web server supporting the XML application
- port is the port number the phones is using for the HTTP request.



Note: If the port is not specified, the phone uses port 80.

The phone also supports the following URLs:

- Dial: XXXXX to have the phone dial XXXXX

5.2 URL VARIABLES

The phones support system variables in the HTTP URL to pass dynamic data to the XML applications.

The variables are in the form `$$VARIABLENAME$$` and the following variables are supported:

Variable Name	Value
SIPUSERNAME	Line user name (Phone DN)
PROXYURL	Proxy of the prime line (connected ICP)
LOCALIP	Phone local IP address
REMOTENUMBER	Remote number for incoming and outgoing call

At the time of the HTTP call, the variables are replaced with the value of the appropriate variable.

The variables are available in all the HTTP URL for programmable/soft keys as well as for the action URI. They can also be used in the XML objects themselves (TextMenu, InputScreen...) but not in a PhoneExecute command.

REMOTENUMBER variable is not supported for calls that are associated with BT Line.

Example

For example, if the administrator specifies an XML softkey with the value:

`http://10.50.10.140/script.pl?name=$$SIPUSERNAME$$`

This softkey executes a GET on:

`http://10.50.10.140/script.pl?name=42512`

Assuming that the dn of the phone is 42512.

The variables can be used, for instance, in a context where the phone has multiple dns (cases of hot desking) and the http server has to know which DN.

5.3 HTTPS

If you want to secure the communication between the Mitel 69xx Phone and your HTTP server hosting the XML applications, you can use HTTPS instead of HTTP.



Note: HTTPS provides a reasonable level of security to your XML applications for man-in-the-middle attacks but you must still make sure that the HTTP server itself is secure, only the data transport is protected by HTTPS.

To allow HTTPS connections, the web server must have a root certificate; this certificate must be signed by a certificate authority of one form or another, which certifies that the certificate holder is indeed the entity it claims to be.

Mitel 69xx phones come with the signing certificates of

- Verisign
- GeoTrust
- Thwate
- Let's Encrypt
- Comodo

So, only certificates signed by these Certification Authorities can be verified by the phone. Certificates that are signed by other providers will not verify on the phone but to overcome this problem a special phone parameter can be configured to disable certificate validation. Refer 6.3 for details.



Note: For more information regarding HTTPS and associated certificates please refer to the x.509 standard at <http://www.ietf.org/html.charters/pkix-charter.html> .

5.4 XML OBJECTS PUSHED TO THE PHONE

The phone can request an XML object via HTTP GET, or an object can be pushed to the phone via a POST. The phone parses this object immediately upon receipt and displays the information to the screen.

The HTTP POST packet must contain an “xml=” line in the message body. The string to parse is located after the equals sign in the message. HTML forms that post objects to the phone must use a field named “xml” to send their data. Any applications that construct HTTP packets on the fly must also specify this line.



Note: the content of the HTTP POST is not url-encoded.

Sample php source code

Below is a sample php source code that sends an XML object to a Mitel phone. In this example, the phone is located at 192.168.0.150 and the server pushing the XML object at 192.168.0.112.

```

<?php
#
function push2phone($server,$phone,$data)
{
$xml = "xml=".$data;

$post = "POST / HTTP/1.1\r\n";
$post .= "Host: $phone\r\n";
$post .= "Referer: $server\r\n";
$post .= "Connection: Keep-Alive\r\n";
$post .= "Content-Type: text/xml\r\n";
$post .= "Content-Length: ".strlen($xml)."\r\n\r\n";

$fp = @fsockopen ( $phone, 80, $errno, $errstr, 5);
if($fp)
{
    fputs($fp, $post.$xml);
    flush();
    fclose($fp);
}
}

#####
$xml = "<AstralPPhoneTextScreen>\n";
$xml .= "<Title>Push test</Title>\n";
$xml .= "<Text>This is a test for pushing a screen to a phone. It is a way to demonstrate that we can push XML objects to an Mitel Phone.</Text>\n";
$xml .= "</AstralPPhoneTextScreen>\n";
push2phone("192.168.0.112","192.168.0.150",$xml);
?>

```

Sample perl source code

Below is a sample perl source code that sends an XML object to an Mitel phone. In this example, the phone is located at 192.168.0.150.

```

#!c:/perl/bin/Perl.exe
# Create a user agent object
use LWP::UserAgent;
use LWP::ConnCache;
use HTTP::Request::Common;
my $ua = LWP::UserAgent->new(agent => 'Mozilla/4.0 (compatible; MSIE 5.5; Windows 98)');

$ua->conn_cache(LWP::ConnCache->new());
$ua->request(
    POST 'http://192.168.0.150'
    ,Content_Type => 'application/x-www-form-urlencoded'
    ,Content      => 'xml=<AstralPPhoneTextScreen><Title>Push test</Title><Text>This is a test for pushing a screen to a phone. It is a way to demonstrate that we can push XML objects to an Mitel Phone.</Text></AstralPPhoneTextScreen>';

```

Notes:



- To accept a pushed page the “**xml application post list**” parameter must be configured with the list of the HTTP servers allowed pushing a page. Refer 6.2 for details.

When a XML object is pushed to the phone, the phone acts like a Web server listening to TCP port 80, which means that if the phone is behind a NAT and the server on the other side of the NAT, which is a typical configuration for a remote user, the port 80 must be forwarded to the IP phone at the router level.

6 XML CONFIGURATION

After creating an XML application to use on the IP phone, the application can be accessed as a Service or a Key.

6.1 CONFIGURING A PROGRAMMABLE KEY USING THE MIVB ESM

In addition to linking an XML application to a custom service, an application can also be linked to programmable key.

To support key programmed applications, the required URI must be programmed to a Programmable key. This can be done via ESM:

Search By

Number

12501

Search Results (1 match)

12501

6930 IP

Full Service

User Profile

Service Profile

Device Details

Service Details

Access and Authentication

Phone Applications

Keys

Save Changes

Cancel

Copy Keys

Clear All Keys

Clear Key

Button Number	Label	Line Type	URL
> 1		Single Line	
▼ 2	xml	URL Line	http://10.112.95.119/hello/new-menu.xml

Label

xml

Line Type

URL Line

Float

☐

URL

http://10.112.95.11

Button Dir. Number

Ring Type

MiXML Application Feature

Not Assigned

Phone Application Feature

Choose Line type “URL Line” for the desired key.

Enter the URL in the URL field.

Save above settings using Save Changes button.

Note: Reboot is not required.

6.2 ACTION URIS

Configuration parameters have been introduced to configure the phone to make an HTTP GET based on events, these events are:

- End of the boot sequence action uri startup
- Time based action uri poll
- On-hook action uri onhook
- Off-hook action uri offhook
- Incoming call action uri incoming
- Outgoing call action uri outgoing
- Connect action uri connected
- Disconnect action uri disconnected
- Successful registration action uri registered

6.2.1 CONFIGURATION

A URI can be configured for each type of events; the configuration can be made using:

- a special cfg file - AppInfo-<SetType>.cfg file and has to be uploaded on MiVB.

See chapter 6.3 for more details on the MiVB ESM Configuration.

Configuration description

These URIs will be configurable via the configuration files using the following parameters.

- **action uri startup:** <URI to GET on startup>
- **action uri incoming:** <URI to GET on an incoming call>
- **action uri outgoing:** <URI to GET on outgoing call>
- **action uri offhook:** <URI to GET on an off-hook event>
- **action uri onhook:** <URI to get on an on-hook event>
- **action uri connected:** <URI to get on a connect event>
- **action uri disconnected:** <URI to get on a disconnect event>
- **action uri poll:** <URI to get when configured timer expires>
- **action uri registered:** <URI to GET on successful registration>

6.2.2 ACTION URI DETAILED BEHAVIOR

Notes:



The action uri GET requests are non blocking.

If the GET request triggered by an action uri fails, the phone does not display any error message.

6.2.2.1 ACTION URI OFFHOOK

The offhook uri is triggered only when the user performs one of the following actions:

- Release the handset
- Press the Speaker/Headset key

This means that the offhook uri is not triggered when the user presses a speeddial key even if the phone goes off- hook and dials a number. Also if the user directly dials a number and press the “dial” key the action uri is not triggered (but the outgoing uri is).

6.2.2.2 ACTION URI ONHOOK

The *action uri onhook* is triggered at the end of any active call:

- The user presses the 'Goodbye' key.
- The user hangs up the handset after a connected call.



Note: The *onhook* event is a subset of the *disconnect* event, you should avoid to configure both action uri as most of the time they will both be called at the same time.

6.2.2.3 ACTION URI INCOMING

The *action uri incoming* is triggered each time there is an incoming call presented to the phone and processed to make the phone ring.



Note: the action uri incoming is not triggered if the phone is call forwarded or is in DND mode .

6.2.2.4 ACTION URI OUTGOING

The *action uri outgoing* is triggered each time an outgoing call is performed by the phone.



Note: the action uri outgoing is also triggered when the user makes a second call to perform a transfer or a conference.

6.2.2.5 ACTION URI CONNECTED

The *action uri connected* is triggered each time the phone transitions to a connected state. This includes regular phone calls.

6.2.2.6 ACTION URI DISCONNECTED

The *action uri disconnected* is triggered when the phone transitions from any active (non idle) call state to idle. This includes regular calls .

The difference between action uri disconnected and action uri onhook is that the disconnect event is generated even when the call fails or if a call is not answered.

The action uri onhook is triggered every time the phone comes back to the idle state where the action uri disconnect is triggered only when the phone was on an active call.



Note: the action uri disconnected is not triggered if the phone is call forwarded or is in DND mode.

6.2.2.7 ACTION URI STARTUP

The *action uri startup* is triggered at the end of the phone boot sequence.

6.2.2.8 ACTION URI REGISTERED

The *action uri registered* is triggered the first time the phone registers successfully a line to its proxy/registrar and also on minet link reset.

If the phone has multiple registrations, the action uri registered is triggered for each line.

6.2.2.9 ACTION URI POLL

The *action uri poll* is called each time the configured timer (*action uri poll interval* parameter) expires. The first timer starts at the end of the phone boot sequence. Phone supports upto 3 polling XML apps.

6.2.3 APPLICATIONS

Action URIs are very powerful as they allow an external application to take control of the display when an event occurs. Here are some examples of potential applications.

Self-configuration

Using the startup URI, it is possible to show a custom menu at the phone startup.

Mass Notification

Using the action uri poll, it is possible to display a status information on the phone after configured interval.

6.3 CONFIGURING ACTION URIS USING THE MIVB ESM

AppInfo-<Settype>.cfg file is used to configure startup XML App and also used to poll for an XML app after a specific interval. 6920, 6930 and 6940 phone uses AppInfo-6920.cfg, AppInfo-6930.cfg and AppInfo-6940.cfg file respectively are used to get the specific configuration. The format & supported parameters are -

action uri poll : <http://myserver.com/myappli.xml>
action uri poll interval : 15
action uri poll2 : <http://myserver.com/myappli2.xml>
action uri poll interval2 : 15
action uri poll3 : <http://myserver.com/myappli3.xml>
action uri poll interval3 : 15
action uri startup : <http://myserver.com/mystartupappli.xml>
xml application post list : myserver1, myserver2 ..
https validate certificates : 0
action uri registered: <http://myserver.com/myregisteredappli.xml>
action uri incoming: <http://myserver.com/myincomingappli.xml>
action uri outgoing: <http://myserver.com/myoutgoingappli.xml>
action uri offhook: <http://myserver.com/myoffhookappli.xml>
action uri onhook: <http://myserver.com/myonhookappli.xml>
action uri connected: <http://myserver.com/myconnectedappli.xml>
action uri disconnected: <http://myserver.com/mydisconnectedappli.xml>

In this particular example, when phone starts up, it will pull the mystartupappli.xml file and display the corresponding app on the phone screen.

Phone will periodically poll for myappli.xml, myappli2.xml, myappli3.xml after every 15 seconds and show the contents accordingly. However, phone can only display one XML app at a time. There is a randomization algorithm resides on phone where it will avoid clashing of polling XML apps if poll intervals are set to the same value. Design recommends to either use same values for all 3 polling intervals or use multiples of the smallest value e.g. 15, 30, 45.

When phone is in teleworker mode (connected to MBG), then MBG will be used as Web Proxy i.e. all the http and https requests (including XML apps) from phone will be proxied through MBG. In this mode, phone will create persistent HTTP(s) connections¹ to poll on configured poll URI's to optimize the socket management on MBG.

Xml application post list can be provided by adding list of HTTP(s) push servers separated by commas. Set will only accept the HTTP post from the servers in the list. Upto 80 IP address/FQDNs can be given in HTTP Post Authenticated Server list.

To disable certificate validation (cases where deployments are using self signed certificates) https validate certificates parameter must be set to 0. Default value of this parameter is 1 forcing the set to perform certificate validation for HTTPs GET connections for XML Apps.

When the phone successfully registers for the first time, it will pull the myregisteredappli.xml file and display the corresponding app on the phone screen.

When the phone receives an incoming call then it will display the myincomingappli.xml app on the phone screen.

When the phone makes a valid outgoing call, it will pull the myoutgoingappli.xml file and display the corresponding app on the phone screen.

When the phone goes offhook, it will display the myoffhookappli.xml app on the phone screen.

When the phone goes back on hook after an active call, it will display the myonhookappli.xml app on the phone screen.

When the phone goes to the connected state, it will display the myconnectedappli.xml app on the phone screen.

¹ Phone will add Keep-Alive header in HTTP(s) requests.

When the phone goes back to the idle state, it will display the mydisconnectedappli.xml app on the phone screen.

Example:

Steps to configure AppInfo-6920.cfg file for 6920 phone on MiVB system.

- a. Open ESM form Online Service URLs and configure Application Info URL to specify the address from where this file will be downloaded (format is <proto>://server/dir, proto is http, https, tftp) If this URL is blank the file will be downloaded from MiVB via tftp.

Online Service URLs on Local_125

[Change](#)

Online Service URLs	
Avatar	
Branding	
Screen Saver	
Desktop Portal	
Application Info	http://10.112.95.9

- b. Open form Phone Application Update and upload AppInfo-6920.cfg created as per applicable parameters.
- c. Once b is completed, the phone will download the AppInfo-6920.cfg file and parse & starts polling if poll interval and URI is configured.
- d. When phone is rebooted, it will download the AppInfo-6920.cfg again & show startup app (if configured) and starts polling (if configured).
- e. To stop polling, remove AppInfo-6920cfg file from MiVB server or program poll interval as 0.

7 TROUBLESHOOTING XML APPLICATIONS

7.1 INTRODUCTION

The following figure shows the HTTP call flows when the phone is performing an XML operation.

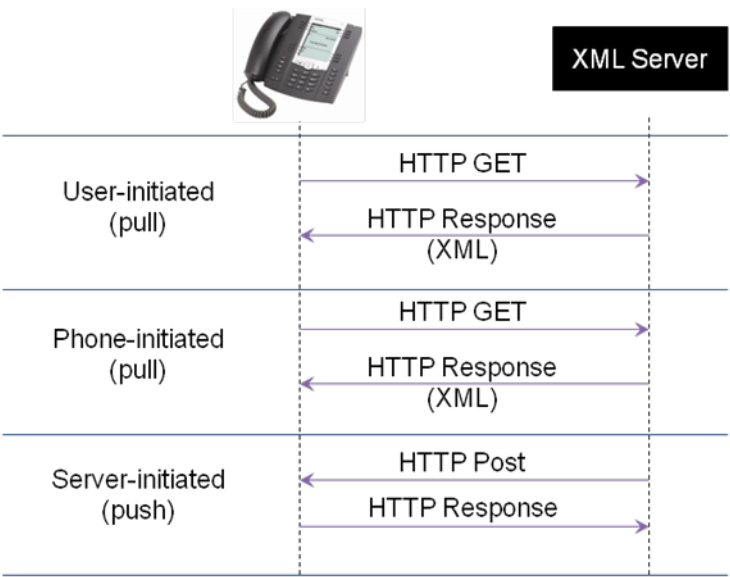


Figure 76:HTTP flow for XML applications

When the phone performs an HTTP GET there are many potential sources of error such as:

Server is not reachable	Answer
Network is down	Cannot display
Server is down	Cannot display
HTTP server application is down	Cannot display
URI cannot be resolved (DNS issue)	Cannot display
Server is reachable	Answer
Wrong file name on the server	File not found
Parsing error of the XML answer	Cannot display
Timeout reached before getting an answer	Connect Timeout
URI cannot be parsed (wrong format)	Invalid URI

Most of the time, an error will translate into a “Cannot display” which is not very helpful to find the origin of the problem as well as very frustrating for the XML developer.

The most common error is a XML parsing error and often the most difficult one to fix.



Note: Using the php classes provided in the toolkit is a good way to limit the number of parsing errors as the object are built following the XSD scheme.

When the server performs a PUSH to the phone, the common errors are:

Nothing happens, most of the time the problem comes from a wrong configuration of the XML push list.

“Cannot display”, if the XML object pushed to the phone creates a parsing error (object not properly formatted).

7.2 TROUBLESHOOTING TOOLS

The following tools will help you troubleshoot problems with the XML services.

Standard web browser (Microsoft Internet Explorer 6.0 or a later version)

- Verify the connectivity
- Verify the validity of the URI called by the phone
- Verify if the XML answer has a correct syntax
- Network packet analyzer such as Wireshark
- Verify what is exchanged between the phone and the server
- HTTP Server log
- Verify if the HTTP GET reached the server
- Verify the parameters of the HTTP GET
- XML validator tools
- Verify the XML syntax and the compliance with the XSD model
- IPA Phone log
- Phone’s message logs
- Verify how the phone processes a XML request
- Verify Wireshark HTTP logs

This last tool is the most powerful as it allows you to read phone internal traces and understand why something went wrong. It is the best way to debug a parsing error.

8 WHY XML APPLICATIONS FOR AN IP PHONE?

XML applications can be split in following categories:

- Media and information
- Vertical applications

8.1 MEDIA AND INFORMATION

These are the applications getting information from the Web.

- Weather Alerts
- Stock Alerts
- Stock Prices
- Worldwide Time/Temperature
- Server Alarms and Notifications
- Server Status
- Account Balances
- Current Gas Prices
- Local Movie Times
- Upcoming Concerts-By Category
- Order Flowers-by Category
- Send Order to Starbucks
- Send SMS Messages
- Track FedEx Package (or Airborne, UPS, etc.)
- Calling Card Minutes Remaining
- Reserve Meeting Rooms
- Contact center Metrics (Calls Waiting, Longest Hold, Performance against Service Level, etc.)
- Pro Sports Scores, Vegas Betting Lines
- Multitude of Banking apps: Balances, Transfers, etc.
- Language Translation
- Daily Horoscope
- Broadcast Joke Of The Day/Inspirational Quote of the Day

8.2 VERTICAL APPLICATIONS

This chapter details potential vertical applications that can be developed as an XML application for the Mitel 69xx phones. The list is far from being exhaustive.

8.2.1 HUMAN RESOURCES

- Available Vacation Days
- Available Personal Days
- 401K balance
- Clock-In/Clock-Out

8.2.2 TRAVEL/HOTEL

- Current Balance
- In-Room Dining Ordering
- Delivery Dining Options
- Extend Stay
- Schedule Airport Shuttle
- Request Housekeeping/Engineering
- Leave Feedback
- Wake-Up Call
- Book Corporate Travel

8.2.3 HEALTH CARE

- Test Results
- Manage Appointment
- Appointment Reminders
- Take-Your-Medicine Reminders
- Order Hospital Meals
- Check Pharmacy Inventory
- Schedule Blood Donation

8.2.4 EDUCATION

- Attendance
- Request Substitute Teacher (used by primary teacher)
- Review Open Requests for Substitute Teacher (used by potential subs)
- Schedule Classes
- Request Dorm Room Change
- School Closing Notification/Status
- Parent Contact Info

8.2.5 LAW ENFORCEMENT

- Amber Alerts
- Traffic Ticket Plead By Phone
- Fugitive Alerts

9 SAMPLE XML APPLICATIONS

Here is the list of XML applications provided by Mitel, these XML applications are available either through an Internet server hosted by Mitel or as source code (part of the Mitel XML API SDK) to be executed from a local PC running XAMPP for example.

- [Area code \(US\)](#)
- [Biorhythms](#)
- [CNN News](#)
- [Currency Converter](#)
- [ESPN News](#)
- [Fox News](#)
- [Horoscope](#)
- [IP Geolocation](#)
- [Local Weather \(US\)](#)
- [Movies](#)
- [Netflix](#)
- [Stock](#)
- [Today...](#)
- [World Clock](#)
- [Yahtzee](#)
- [All the above](#)

9.1 ACCESS FROM THE INTERNET

Mitel has made available, **for demonstration purpose only**, a list of XML applications on the Internet.

Notes:

 Mitel does not guarantee the availability of these applications. The applications can change at any time without notice.

These applications should not be used commercially; any abusive use of these applications will be detected and the phone will be automatically banned from the applications.

Your Mitel phone must have Internet access to use these applications.

The XML applications can be configured individually on a phone key or as a global menu.

9.2 LOCAL SERVER USING XAMPP

XAMPP is a free, cross-platform, easy-to-use web server capable of serving dynamic pages. XAMPP consists mainly of the Apache HTTP Server, MySQL database and interpreters for scripts written in the PHP and Perl programming languages. The program is released under the GNU General Public License.

 **Notes:** your server must have Internet access to use these applications.

XAMPP is an acronym for:

- **X** (any of the four operating systems Windows, Linux, Sun Solaris and Mac OS X)
- **A**pache
- **M**ySQL
- **P**HP
- **P**erl

The XAMPP 1.6.6 Basic Package includes the following main components:

- Apache HTTPD 2.2.8 + OpenSSL 0.9.8g
- MySQL 5.0.51
- PHP 5.2.5 + PHP 4.4.8 rc2 dev + PEAR
- SQLite 2.8.15
- phpMyAdmin 2.11.4

- Mercury Mail Transport System v4.52
- FileZilla FTP Server 0.9.25



Note: the XML scripts provided by Mitel use only Apache and PHP.

9.2.1 XAMPP INSTALLATION

The nice thing with XAMPP is that it does not need to be installed. Simply extracting the archive to the root folder of your disk is enough. No registry keys are written, no files are copied to the Windows directory. You can even put XAMPP on a memory stick and run it from there. This makes it very portable and easy to use.

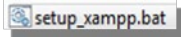
XAMPP can be downloaded from <http://www.apachefriends.org/en/xampp.html>, where you can select which Operating System you want to use XAMPP.

Please refer to the XAMPP web site for detailed information on how to install the software.

Notes:



- Make sure you extract XAMPP to the root directory of your disk or memory stick (e.g. C:\), otherwise you may run into troubles later
 - If you have an existing PHP installation on your computer, please uninstall it or remove the Windows environment variable PHPRC (open the Windows Control Panel, go to System, Advanced, Environment Variables and remove the PHPRC variable). Otherwise XAMPP might use the wrong PHP settings.
-

Once XAMPP is installed, just run  which is located in the xampp directory. This configures the installation of XAMPP on your computer.

To uninstall, just remove the xampp directory.

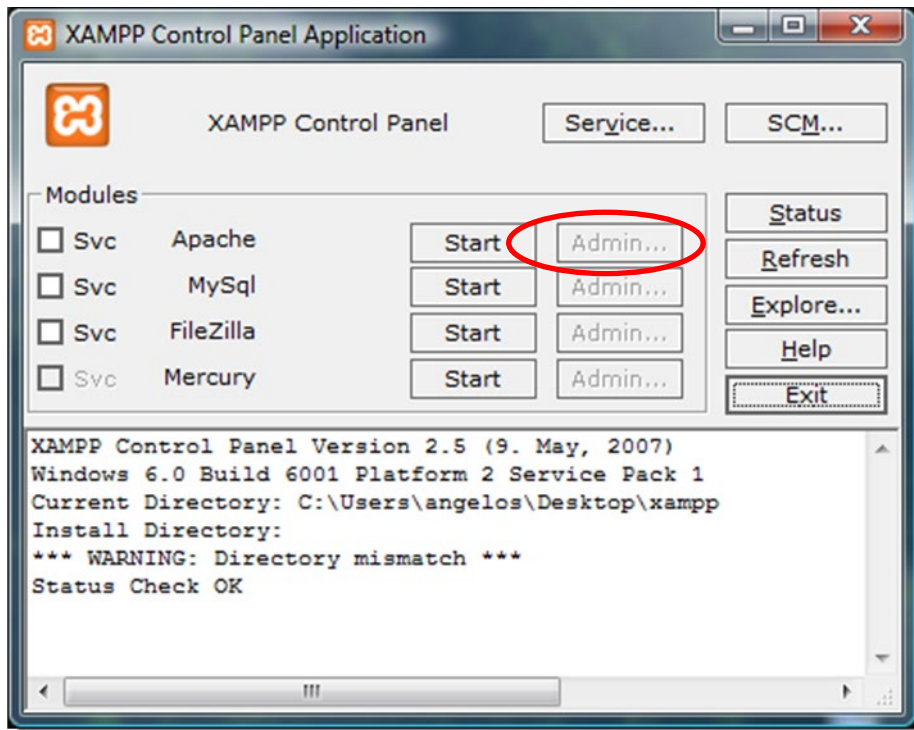
9.2.2 XML SCRIPTS INSTALLATION

Just create a directory named “xml” under “xampp/htdocs” directory and unzip the xml-xampp.zip file provided in the Mitel XML API SDK in this directory.

Also create a “cache” directory under the xampp directory. This directory will be used to cache data on the server.

9.2.3 XAMPP START AND STOP

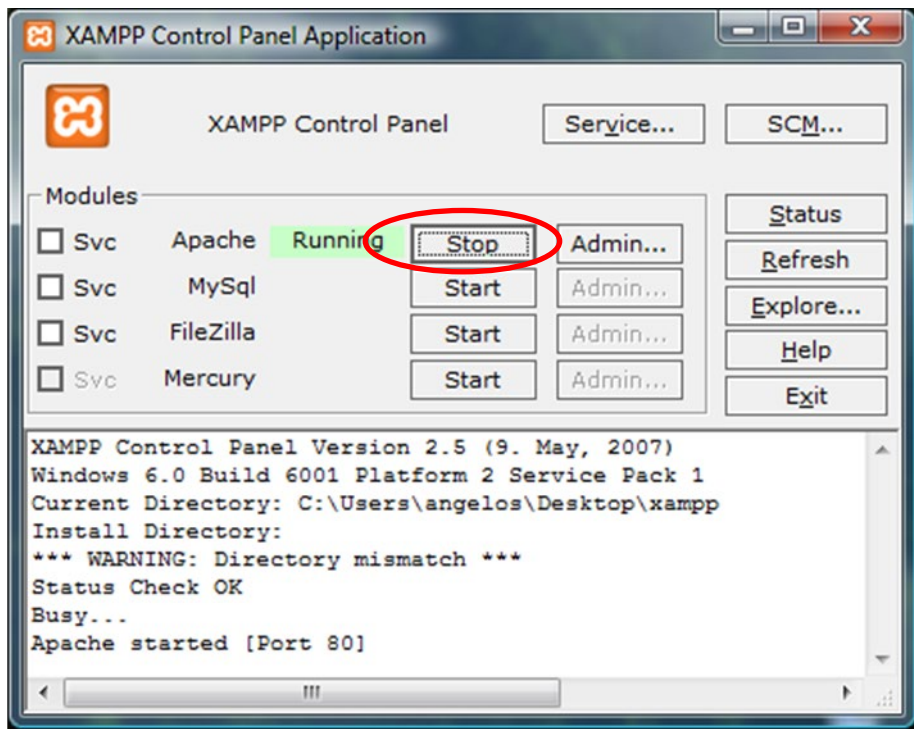
In the xampp directory run . To start the Apache HTTP server, just click on the “Start” button next to the Apache module.



Apache is now running.



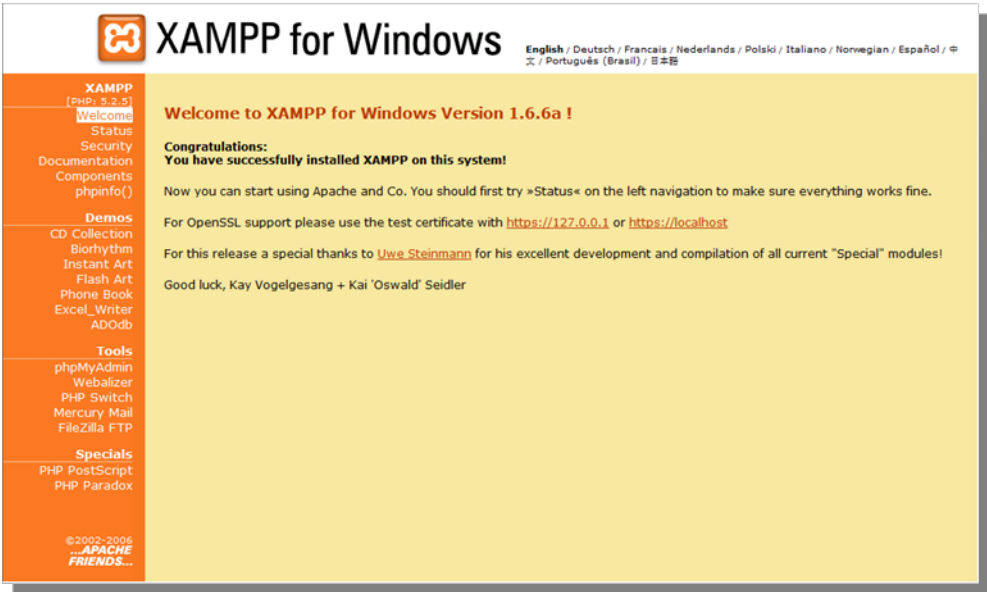
Note: In case you have a firewall running (native for Windows XP or Vista), you will be prompted to authorize Apache HTTP Server.



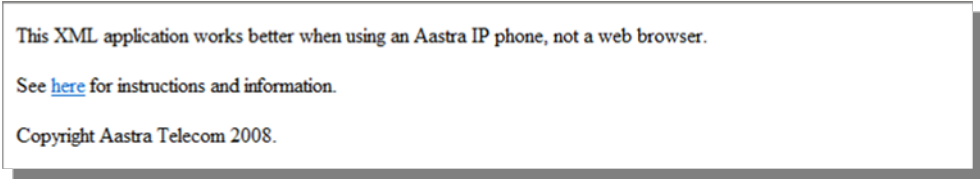
To stop Apache, simply click on the Stop-button. To close the XAMPP Control Panel, click on the Exit button.

9.2.4 TEST YOUR INSTALLATION

To test the HTTP server, direct your Web browser to `http://localhost/xampp/`. You should see the following page:



To test the XML scripts, direct your Web browser to `http://localhost/xml/area/area.php`. You should see the following page:



9.2.5 TROUBLESHOOTING APACHE

Most common problem is that another program is already using TCP server port 80 (HTTP) and port 443 (HTTPS). Run the program `xampp-portcheck.exe` in the xampp directory.

Check which program is using server port 80 and/or 443. Shutdown that program.

9.3 APPLICATIONS

This chapter describes all the sample applications available either from the Internet or from XAMPP running on a server (represented below as myserver.com).

9.3.1 AREA CODE LOOKUP (US)

Description	
This application allows the user to lookup for the State/Cities of any given US area code.	
Phone compatibility	
✓	Mitel 6920/6930/6940
Configuration (Internet)	
uri= <code>http://65.36.55.137/xml/area/area.php</code>	
Configuration (XAMPP)	

uri=http://myserver.com/xml/area/area.php	
Screenshots	
Area code finder Enter area code 972 Backspace Cancel Done	Application Area code 972 USA Texas Addison, Allen, Carrollton, Celina, Dallas, Duncanville, Ennis, Farmers Branch, Frisco, Garland, Grand Prairie, Irving, Italy, Kaufman, Lavon, Lewisville, McKinney, Mesquite, Midlothian, Plano, Renner, Richardson, Seagoville, Terrell, Waxahatchie. Back Exit

9.3.2 BIORHYTHMS

Description	
Check your Biorhythms.	
Phone compatibility	
✓	Mitel 6920/6930/6940
Configuration (Internet)	
uri=http://65.36.55.137/xml/games/biorhythms.php	
Configuration (XAMPP)	
uri=http://myserver.com/xml/games/biorhythms.php	
Screenshots (6920/6930/6940)	
Biorhythms Birth Date (MM/DD/YYYY) 01/01/2015 Cancel Done	Biorhythms Intellect Emotional Physical Back Exit
Comments	
User's birth date is stored on the server. Only phones with graphical display (6920, 6930 and 6940) will display the charts PHP-GD extension is needed for this XML script.	

9.3.3 CNN NEWS

Description

RSS feed from CNN.com including the following topics:

Top Stories	Entertainment
World	Travel
U.S.	Education
Politics	Video
Law	Offbeat
Technology	Most Popular
Science and Space	Most Recent
Health	

Phone compatibility

✓	Mitel 6920/6930/6940
---	----------------------

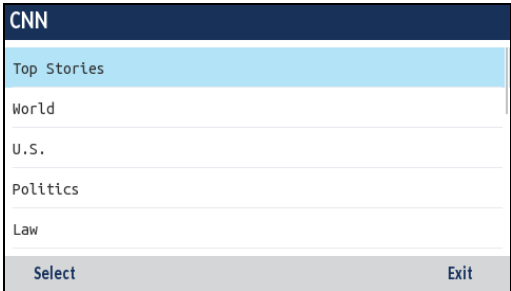
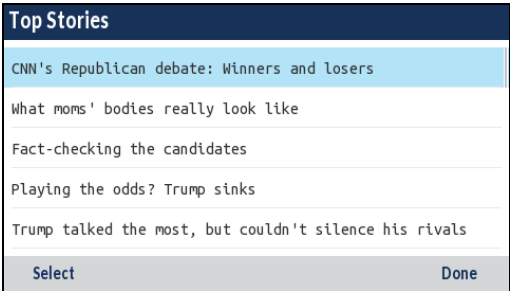
Configuration (Internet)

uri=http://65.36.55.137/xml/rss/rss.php?feed=cnn

Configuration (XAMPP)

uri=http://myserver.com/xml/rss/rss.php?feed=cnn

Screenshots

9.3.4 CURRENCY CONVERTER

Description

This application uses www.yahoo.com to convert any currency in to another currency. It also allows the user to setup a list of favorite conversions.

Phone compatibility

✓	Mitel 6920/6930/6940
---	----------------------

Configuration (Internet)

uri=http://65.36.55.137/xml/stock/currency.php

Configuration (XAMPP)

uri=http://myserver.com/xml/stock/currency.php

Screenshots

Currency converter

Source

USD

Target

EUR

Backspace

Dot "."

ABC

Done

Currency converter

USD to EUR

Last Trade: 0.8848

Ask: 0.8848

Bid: 0.8848

Date: 9/17/2015

Time: 4:46pm

Powered by Yahoo

Done

Comments

User's last request and favorites are stored on the server.

9.3.5 ESPN NEWS

Description

RSS feed from ESPN.com bringing news for the most popular sports in North America.

Top Headlines	Motorsports
NFL	Soccer
NBA	College Basketball
MLB	College Football
NHL	

Phone compatibility

✓

Mitel 6920/6930/6940

Configuration (Internet)

uri=http://65.36.55.137/xml/rss/rss.php?feed=espn

Configuration (XAMPP)

uri=http://myserver.com/xml/rss/rss.php?feed=espn

Screenshots

ESPN News

Top Headlines

NFL

NBA

MLB

NHL

Select

Exit

9.3.6 FOX NEWS

Description	
RSS feed from foxnews.com including the following topics:	
Latest Headlines	Business
National	SciTech
World	Health
Politics	Entertainment
Phone compatibility	
✓	Mitel 6920/6930/6940
Configuration (Internet)	
uri=http://65.36.55.137/xml/rss/rss.php?feed=fox	
Configuration (XAMPP)	
uri=http://myserver.com/xml/rss/rss.php?feed=fox	
Screenshots	
FOX News Latest Headlines National World Politics Business Select Exit Latest Headlines (1/2) Can Florina capitalize on fiery Trump skirmish, impassioned 'shame on us' challenge to defund Planne... TSUNAMI ADVISORY Hawaii, Calif. urge caution after 8.3 Chile quake SCANDAL IN THE AIR? At least 2 marshals caught up in prostitution probe HOUSE OF HORRORS Fla. woman arrested at home full of blades 'TOTAL FAILURE' Military admits only 4 or 5 US-trained troops in Syria Select Done	

9.3.7 HOROSCOPE

Description	
RSS feed from dailyhoroscopes.com.	
Phone compatibility	
✓	Mitel 6920/6930/6940
Configuration (Internet)	
uri=http://65.36.55.137/xml/rss/rss.php?feed=horoscope	
Configuration (XAMPP)	
uri=http://myserver.com/xml/rss/rss.php?feed=horoscope	
Screenshots	

Daily Horoscope Scorpio Sagittarius Capricorn Aquarius Pisces Select Exit	Pisces Pisces - Pisces Horoscope for Sep 17, 2015 Worry thoughts could ruin your happiness. Don't allow as it only obeys destructive power and sows the depressing seed in intelligence. Encourage yourself to always expect best and try to see ludicrous even in mishaps. Unexpected bills will increase financial burden. You can expect problems at home- if you do not spend time with family members. The day is fabulous in terms of your love life. Keep making love. Don't mix business with pleasure. Do not reveal information that is personal and confidential. After marriage, the sin becomes worship, and you might worship a lot today. RSS feed provided by: Done
--	---

9.3.8 IP GEOLOCATION

Description	
Identify the location (Country, City...) of any given public IP address. This application uses an API provided by http://www.geoplugin.net .	
Phone compatibility	
✓	Mitel 6920/6930/6940
Configuration (Internet)	
uri= http://65.36.55.137/xml/area/ip.php	
Configuration (XAMPP)	
uri= http://myserver.com/xml/area/ip.php	

9.3.9 MOVIES

Description	
RSS feed from yahoo.com including the following topics:	
Top 10 Box Office Opening This Week Coming Soon	
Phone compatibility	
✓	Mitel 6920/6930/6940
Configuration (Internet)	
uri= http://65.36.55.137/xml/rss/rss.php?feed=movies	
Configuration (XAMPP)	
uri= http://myserver.com/xml/rss/rss.php?feed=movies	

9.3.10 NETFLIX

Description	
RSS feed from netflix.com including the following topics:	
Top 100	Horror Top 25
New Releases*	Sci-Fi Top 25
Action/adventure Top 25	Television Top 25
Comedy Top 25	Thrillers Top 25
Drama Top 25	
Phone compatibility	
✓	Mitel 6920/6930/6940
Configuration (Internet)	
uri=http://65.36.55.137/xml/rss/rss.php?feed=netflix	
Configuration (XAMPP)	
uri=http://myserver.com/xml/rss/rss.php?feed=netflix	
Comments	
*Not available if you are using XAMPP under Microsoft Windows due to memory limitation for PHP.	

9.3.11 STOCK QUOTES

Description	
This application uses www.yahoo.com to get the value of any given stock. Please refer to yahoo.com for the syntax of the stock ticker.	
Phone compatibility	
✓	Mitel 6920/6930/6940
Configuration (Internet)	
uri=http://65.36.55.137/xml/stock/stock.php	
Configuration (XAMPP)	
uri=http://myserver.com/xml/stock/stock.php	
Screenshots	

Stock quotes	Stock quotes
Enter Ticker MITL	Enter Ticker MITL
Backspace Dot "." ABC ▸ Done ● ●	Backspace Dot "." ABC ▸ Done ● ●

Comments

User's last request and favorites are stored on the server.

Only large screen phones have access to the favorites.

9.3.12 TODAY...

Description	
RSS feed from answers.com including the following topics: Word of the Day Birthdays today This day in History Quote of the Day	
Phone compatibility	
✓	Mitel 6920/6930/6940
Configuration (Internet)	
uri=http://65.205.17.13/xml/rss/rss.php?feed=day	
Configuration (XAMPP)	
uri=http://myserver.com/xml/rss/rss.php?feed=day	

9.3.13 LOCAL WEATHER

Description	
RSS feed from rssweather.com providing weather forecast for any given US ZIP code	
Phone compatibility	
✓	Mitel 6920/6930/6940
Configuration (Internet)	
uri=http://65.36.55.137/xml/weather/weather.php	
Configuration (XAMPP)	

uri=http://myserver.com/xml/weather/weather.php
Screenshots
US Local Weather Enter ZIP code 75034 Backspace Submit Exit Frisco, Texas Weather Currently 81F PARTLY CLOUDY RSS feed provided by rssweather.com and brought to you by Mitel. Currently Today Tonight Friday
Comments
User's last request is stored on the server

9.3.14 YAHTZEE

Description
A Yahtzee game...
Phone compatibility
✓ Mitel 6920/6930/6940
Configuration (Internet)
uri=http://65.36.55.137/xml/games/yahtzee.php
Configuration (XAMPP)
uri=http://myserver.com/xml/games/yahtzee.php

9.3.15 GLOBAL MENU

Description
This application is an XML application aggregator. It can be configured as an XML key or as the XML custom menu behind the "Services" key
Phone compatibility
✓ Mitel 6920/6930/6940
Configuration (Internet)
uri=http://65.36.55.137/xml/menu/mymenu.php
Configuration (XAMPP)
uri=http://myserver.com/xml/menu/mymenu.php
Screenshots

Application XML applications Area Code Biorhythms CNN News Currency converter SelectDone
Comments
Only Mitel phones with softkeys are able to customize the list of applications.

10 APPENDIX A: XSL MODEL

The xsl model is provided as an xsd file in this SDK.

11 APPENDIX B: OBJECT ORIENTED PHP CLASSES

Mitel also provides an object oriented API to develop XML applications which is included in the XML SDK.



Note: The PHP objects are taking care of the XML escape encoding when they are needed.

11.1 AASTRAIPPHONEEXECUTE()

This class allows you to create a XML PhoneExecute object.

Include

- AstralPPhoneExecute.class.php

Methods

- setBeep() to enable a notification beep with the object (optional)
- setEncodingUTF8() to change encoding from default ISO-8859-1 to UTF-8 (optional)
- generate() to return the generated XML for the object
- output(flush) to display the object
 - flush boolean optional, output buffer to be flushed out or not.
- setTriggerDestroyOnExit() to set the triggerDestroyOnExit tag to "yes" (optional)
 - addEntry (url,interruptCall,title) to add an action to be executed.
 - url string
 - interruptCall string, optional, should be always "yes"
 - title string, optional, title to be displayed

Example

```
require_once('AstralPPhoneExecute.class.php');
$execute = new AstralPPhoneExecute();
$execute->addEntry('http://myserver.com/script.php?choice=2');
$execute->addEntry('Command: Reset');
$execute->output();
```

11.2 AASTRAIPPHONEFORMATTEDTEXTSCREEN()

This class allows you to create a XML FormattedTextScreen object.

Include

- AstralPPhoneFormattedTextScreen.class.php

Methods

- setTopTitle(title,color,icon_index) to set the Top Title of the XML screen
 - title string
 - color string, "red", "blue", ... (optional)
 - icon_index integer, icon number
- setCancelAction(uri) to set the cancel parameter with the URI to be called on Cancel (optional)
 - uri string
- setDestroyOnExit() to set DestroyonExit parameter to 'yes', 'no' by default (optional)

- `setLockIn(uri)` to set the Lock-in tag to 'yes' and the GoodbyeLockInURI (optional)
 - `uri` string, GoodbyeLockInURI
- `setCallProtection(notif)` to protect the XML object against incoming calls
 - `notif` enable/disable the display of an incoming call notification (optional)
- `setTimeout(timeout)` to define a specific timeout for the XML object (optional)
 - `timeout` integer (seconds)
- `addSoftkey(index,label,uri)` to add custom softkeys to the object (optional)
 - `index` integer, softkey number
 - `label` string
 - `uri` string
- `setRefresh(timeout,URL)` to add Refresh parameters to the object (optional)
 - `timeout` integer (seconds)
 - `URL` string
- `setEncodingUTF8()` to change encoding from default ISO-8859-1 to UTF-8 (optional)
- `generate()` to return the generated XML for the object
- `output(flush)` to display the object
 - `flush` boolean optional, output buffer to be flushed out or not.
- `addLine(text,size,align,color,wrap, blink)` to add a formatted line
 - `text` string
 - `size` string, optional, "double"
 - `align` string, optional, "left", "right" or "center"
 - `color` string, optional
 - `wrap` string, optional, "yes", "no" (default)
 - `blink` string, optional, "slow", "fast" or "no" (default)
- `setScrollStart()` to define the beginning of the scrolling section
- `setScrollEnd()` to define the end of the scrolling section
- `setAllowDTMF()` to allow DTMF passthrough on the object
- `setDial(number,line)` to set the number to be dialed as well as the line to use when going off-hook or with the custom softkey
Softkey::Dial2
 - `number` string
 - `line` integer (optional)

Example

```
require_once('AstralPPhoneFormattedTextScreen.class.php');
$text = new AstralPPhoneFormattedTextScreen();
$text->setDestroyOnExit();
$text->addLine('Formatted Screen','double','center','red');
$text->setScrollStart();
$text->addLine('Scrolled text1');
$text->addLine('Scrolled text2');
$text->addLine('Scrolled text3');
$text->addLine('Scrolled text4');
$text->addLine('Scrolled text5');
$text->setScrollEnd();
$text->addLine('Footer',NULL,'center','blue');
```

```
$ftext->addSoftkey('10', 'Exit', 'SoftKey:Exit');  
$ftext->output();
```

Output

Application

Formatted Screen

```
Scrolled text1  
Scrolled text2  
Scrolled text3  
Scrolled text4  
Scrolled text5
```

Footer



11.3 AASTRAIPPHONEICONMENU()

This class allows you to create a XML IconMenu object.

Include

- AastralPPhonelconMenu.class.php

Methods

- setDestroyOnExit() to set DestroyonExit parameter to 'yes', 'no' by default (optional)
- setBeep() to enable a notification beep with the object (optional)
- setLockIn(uri) to set the Lock-in tag to 'yes' and the GoodbyeLockInURI (optional)
 - uri string, GoodByeLockInURI
- setCallProtection(notif) to protect the XML object against incoming calls
 - notif enable/disable the display of an incoming call notification (optional)
- setTimeout(timeout) to define a specific timeout for the XML object (optional)
 - timeout integer (seconds)
- setBackgroundColor(color) to change the XML object background color (optional)
 - color string, "red", "blue", ...
- addSoftkey(index,label,uri) to add custom soktkeys to the object (optional)
 - index integer, softkey number

- label string
 - uri string
- setRefresh(timeout,URL) to add Refresh parameters to the object (optional)
 - timeout integer (seconds)
 - URL string
- setEncodingUTF8() to change encoding from default ISO-8859-1 to UTF-8 (optional)
- generate() to return the generated XML for the object
- output(flush) to display the object
 - flush boolean optional, output buffer to be flushed out or not.
- setLayout(layout) to set the desired layout
 - layout 1 or 2
- setMode(mode) to set the screen mode
 - mode 'regular' or 'extended'
- setDefaultIndex(index) to set the default selection in the list (optional)
 - index index (1-24)
- setFontMono(fontMono) to decide the use of the monotype font
 - fontMono boolean ("yes"/"no")
- addEntry(url,selection,icon,fontMono,dial,line) to add an element in the list to be displayed
 - url string
 - selection string (optional)
 - icon integer (optional)
 - fontMono boolean (optional)
 - dial string, phone number to dial (optional)
- addLine(text,align,color) to add a line in the current item
 - text string
 - align string, optional, "left", "right" or "center"
 - color string, optional, "red", "black", ...

Example

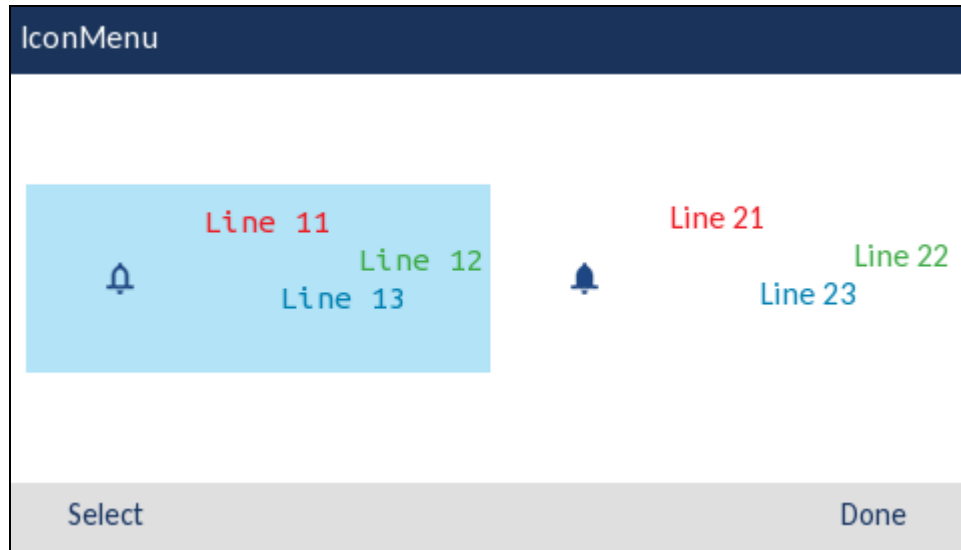
```
require_once('AstralPPhonelconMenu.class.php');
$menu = new AstralPPhonelconMenu();
$menu->setDestroyOnExit();
$menu->setTopTitle('IconMenu');
$menu->setDeFaultIndex('3');
$menu->setLayout('1');
$menu->setMode('regular');
$menu->setFontMono('yes');
$menu->addEntry( 'http://myserver.com/script.php?choice=1',
                'Value=1',
                'Icon:Alarm',
                'yes'
                );
$menu->addLine('Line 11','left','red');
$menu->addLine('Line 12','right','green');
$menu->addLine('Line 13','center','blue');
$menu->addEntry( 'http://myserver.com/script.php?choice=2',
                'Value=2',
```

```

        'Icon:AlarmFilled',
        'no'
    );
$menu->addLine('Line 21','left','red');
$menu->addLine('Line 22','right','green');
$menu->addLine('Line 23','center','blue');
$menu->addSoftkey('1', 'My Select', 'http://myserver.com/script.php?action=1');
$menu->addSoftkey('6', 'Exit', 'SoftKey:Exit');
$menu->output();

```

Output



11.4 AASTRAIPPHONEIMAGESCREEN()

This class allows you to create a XML ImageScreen object.

Include

- AastralPPhoneImageScreen.class.php

Methods

- setTopTitle(title,color,icon_index) to set the Top Title of the XML screen
 - title string
 - color string, "red", "blue", ... (optional)
 - icon_index integer, icon number
- setCancelAction(uri) to set the cancel parameter with the URI to be called on Cancel (optional)
 - uri string
- setDestroyOnExit() to set DestroyonExit parameter to 'yes', 'no' by default (optional)
- setBeep() to enable a notification beep with the object (optional)
- setLockIn(uri) to set the Lock-in tag to 'yes' and the GoodbyeLockInURI (optional)
 - uri string, GoodByeLockInURI
- setCallProtection(notif) to protect the XML object against incoming calls
 - notif enable/disable the display of an incoming call notification (optional)

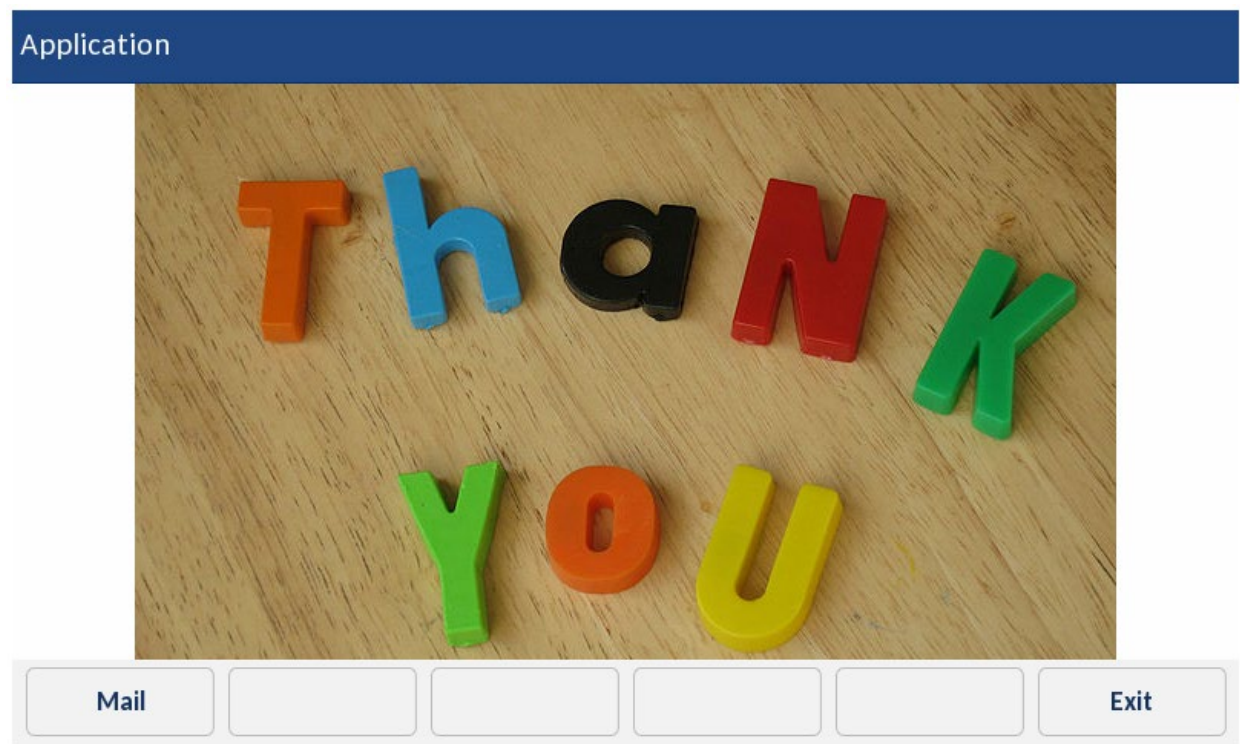
- `setTimeout(timeout)` to define a specific timeout for the XML object (optional)
 - `timeout` integer (seconds)
- `setBackgroundColor(color)` to change the XML object background color (optional)
 - `color` string, "red", "blue", ...
- `addSoftkey(index,label,uri,icon_index)` to add custom softkeys to the object (optional)
 - `index` integer, softkey number
 - `label` string
 - `uri` string
 - `icon_index` integer, icon number
- `setRefresh(timeout,URL)` to add Refresh parameters to the object (optional)
 - `timeout` integer (seconds)
 - `URL` string
- `setEncodingUTF8()` to change encoding from default ISO-8859-1 to UTF-8 (optional)
- `addIcon(index,icon)` to add custom icons to the object (optional)
 - `index` integer, icon index
 - `icon` string, icon name or definition
- `generate()` to return the generated XML for the object
- `output(flush)` to display the object
 - `flush` boolean optional, output buffer to be flushed out or not.
- `setImage(image)` to define the image to be displayed
 - `image` string
- `setGDImage(GDImage)` to use a GDImage for display, the size is forced to 40x144
 - `GDImage` GDImage
- `setAlignment(vertical,horizontal)` to define image alignment
 - `vertical` string, "top", "middle", "bottom"
 - `horizontal` string, "left", "middle", "right"
- `setSize(height,width)` to define image size
 - `height` integer (pixels)
 - `width` integer (pixels)
- `setAllowDTMF()` to allow DTMF passthrough on the object
- `setScrollUp(uri)` to set the URI to be called when the user presses the Up arrow (optional)
 - `uri` string
- `setScrollDown(uri)` to set the URI to be called when the user presses the Down arrow (optional)
 - `uri` string
- `setScrollLeft(uri)` to set the URI to be called when the user presses the Left arrow (optional)
 - `uri` string
- `setScrollRight(uri)` to set the URI to be called when the user presses the Right arrow (optional)
 - `uri` string
- `setMode(mode)` to define the image mode to be displayed (normal, extended, fullscreen) (optional)
 - `mode` string enum normal, extended, fullscreen

- `setDoneAction(uri)` to set the URI to be called when the user selects the default "Done" key (optional)
 - `uri` `string`
- `setImageAction(uri)` to set the `imageAction` parameter with the URI to be called when user presses on the displayed image (optional)
 - `uri` `string`

Example

```
require_once('AstralPPhoneImageScreen.class.php');
$images = new AstralPPhoneImageScreen();
$images->setDestroyOnExit();
$images->setSize(40,40);
$images->setImage('http://10.112.95.119/image.png');
$images->addSoftkey('1', 'Mail', 'http://myserver.com/script.php?action=1','1');
$images->addSoftkey('6', 'Exit', 'SoftKey:Exit');
$images->addIcon('1', 'Icon:Envelope');
$images->output();
```

Output



11.5 AASTRAIPPHONEINPUTSCREEN() – SINGLE INPUT FIELD

This class allows you to create a XML InputScreen object.

Include

- `AstralPPhoneInputScreen.class.php`

Methods

- `setTitle(title)` to setup the title of an object (optional)
 - `title` `string`

- `setTopTitle(title,color,icon_index)` to set the Top Title of the XML screen
 - `title` string
 - `color` string, "red", "blue", ... (optional)
 - `icon_index` integer, icon number
- `setCancelAction(uri)` to set the cancel parameter with the URI to be called on Cancel (optional)
 - `uri` string
- `setDestroyOnExit()` to set DestroyonExit parameter to 'yes', 'no' by default (optional)
- `setLockIn(uri)` to set the Lock-in tag to 'yes' and the GoodbyeLockInURI (optional)
 - `uri` string, GoodByeLockInURI
- `setCallProtection(notif)` to protect the XML object against incoming calls
 - `notif` enable/disable the display of an incoming call notification (optional)
- `setTimeout(timeout)` to define a specific timeout for the XML object (optional)
 - `timeout` integer (seconds)
- `setBackgroundColor(color)` to change the XML object background color (optional)
 - `color` string, "red", "blue", ...
- `addSoftkey(index,label,uri)` to add custom softkeys to the object (optional)
 - `index` integer, softkey number
 - `label` string
 - `uri` string
- `setRefresh(timeout,URL)` to add Refresh parameters to the object (optional)
 - `timeout` integer (seconds)
 - `URL` string
- `setEncodingUTF8()` to change encoding from default ISO-8859-1 to UTF-8 (optional)
- `generate()` to return the generated XML for the object
- `output(flush)` to display the object
 - `flush` boolean optional, output buffer to be flushed out or not.
- `setURL(url)` to set the URL to called after the input
 - `url` string
- `setType(type)` to set type of input, 'string' by default
 - `type` enum ('IP', 'string', 'stringN', 'number', 'dateUS'...)
- `setDefault(default)` to set default value for the input (optional)
 - `default` string
- `setParameter(param)` to set the parameter name to be parsed after the input
 - `param` string
- `setInputLanguage(language)` to set the language of the input (optional)
 - `language` enum ("English", "French"....)
- `setPassword()` to set the Password parameter to 'yes', 'no' by default (optional)
- `setNotEditable()` to set the editable parameter to 'no', 'yes' by default (optional)
- `setEditable()` is now replaced by `setNotEditable` but kept for compatibility reasons (optional)

- `setPrompt(prompt)` to set the prompt to be displayed for the input.
 - `prompt` `string`

Example

```
require_once('AstralPPhoneInputScreen.class.php');
$input = new AstralPPhoneInputScreen();
$input->setTitle('Title');
$input->setPrompt('Enter your password');
$input->setParameter('param');
$input->setType('string');
$input->setURL('http://myserver.com/script.php');
$input->setPassword();
$input->setDestroyOnExit();
$input->setDefault('Default');
$input->output();
```

Output

Application

Title

Enter your password

*****|

Backspace

.

ABC ►

NextSpace

Cancel

Done

11.6 AASTRAIPPHONEINPUTSCREEN() – MULTIPLE INPUT FIELDS

This class allows you to create a XML InputScreen object with multiple input fields.

Include

- `AstralPPhoneInputScreen.class.php`

Methods

- `setTitle(title)` to setup the title of an object (optional)
 - `title` `string`
- `setTopTitle(title,color,icon_index)` to set the Top Title of the XML screen

- title string
 - color string, "red", "blue", ... (optional)
 - icon_index integer, icon number
- setCancelAction(uri) to set the cancel parameter with the URI to be called on Cancel (optional)
 - uri string
- setDestroyOnExit() to set DestroyonExit parameter to 'yes', 'no' by default (optional)
- setLockIn(uri) to set the Lock-in tag to 'yes' and the GoodbyeLockInURI (optional)
 - uri string, GoodByeLockInURI
- setCallProtection(notif) to protect the XML object against incoming calls
 - notif enable/disable the display of an incoming call notification (optional)
- setAllowAnswer() to set the allowAnswer tag to 'yes' (optional)
- setAllowDrop() to set the allowDrop tag to 'yes' (optional)
- setAllowXfer() to set the allowXfer tag to 'yes' (optional)
- setAllowConf() to set the allowConf tag to 'yes' (optional)
- setTimeout(timeout) to define a specific timeout for the XML object (optional)
 - timeout integer (seconds)
- setBackgroundColor(color) to change the XML object background color (optional)
 - color string, "red", "blue", ...
- addSoftkey(index,label,uri) to add custom softkeys to the object (optional)
 - index integer, softkey number
 - label string
 - uri string
- setRefresh(timeout,URL) to add Refresh parameters to the object (optional)
 - timeout integer (seconds)
 - URL string
- setEncodingUTF8() to change encoding from default ISO-8859-1 to UTF-8 (optional)
- generate() to return the generated XML for the object
- output(flush) to display the object
 - flush boolean optional, output buffer to be flushed out or not.
- setURL(url) to set the URL to called after the input
 - url string
- setType(type) to set the default type of input 'string' by default
 - type enum ('IP', 'string', 'number', 'dateUS'...)
- setDefault(default) to set default default value for the input (optional)
 - default string
- setParameter(param) to set the default parameter name to be parsed after the input
 - param string
- setPassword() to set the default Password parameter to 'yes', 'no' by default (optional)
- setNotEditable() to set the default editable parameter to 'no', 'yes' by default (optional)

- `setEditable()` is now replaced by `setNotEditable` but kept for compatibility reasons (optional)
- `setPrompt(prompt)` to set the default prompt to be displayed for the input.
 - `prompt` `string`
- `setDefaultIndex(index)` to define the field index the object will use to start (optional)
 - `index` `integer`, optional, default is 1
- `setDisplayMode(display)` to define the aspect of the display, normal/condensed (optional)
 - `display` `enum` ("normal", "condensed"), default is "normal".
- `setInputLanguage(language)` to set the language of the input (optional)
 - `language` `enum` ("English", "French"....)
- `addField(type)` to add an input field and setting its type
 - `type` (IP, string, stringN, number, dateUS, timeUS, dateInt, timeInt or empty) if the type is an empty string then the type is inherited from the main object.
- `setFieldPassword(password)` to set the password mode for the input field, overrides the value set by `setPassword` for the field
 - `password` `enum` ("yes", "no")
- `setFieldEditable(editable)` to set the input field editable mode ('yes', 'no'), overrides the value set by `setEditable` or `setNotEditable` for the field
 - `editable` `enum` ("yes", "no")
- `setFieldParameter(parameter)` to set the parameter name to be parsed after the global input, overrides the value set by `setParameter` for the field
 - `parameter` `string`
- `setFieldPrompt(prompt)` to set the prompt to be displayed for the input field, overrides the value set by `setPrompt` for the field
 - `prompt` `string`
- `setFieldSelection(selection)` to set the Selection tag for the field
 - `selectionstring`
- `setFieldDefault(default)` to set default value for the input field, overrides the value set by `setDefault` for the field
 - `default` `string`

Example

```
require_once('AastralPPhoneInputScreen.class.php');
$input = new AastralPPhoneInputScreen();
$input->setTitle('Restricted application');
$input->setDisplayMode('condensed');
$input->setURL($XML_SERVER);
$input->setDestroyOnExit();
$input->addField('empty');
$input->addField('string');
$input->setFieldSelection('1');
$input->setFieldPrompt('Username:');
$input->setFieldParameter('user');
$input->setFieldSelection('1');
$input->addField('number');
$input->setFieldPassword('yes');
$input->setFieldPrompt('Password:');
$input->setFieldParameter('password');
$input->setFieldSelection('2');
$input->addSoftkey('10', 'Exit', 'SoftKey:Exit');
$input->output();
```

Output

Application

Restricted application

Username:

Password:

Backspace

.

ABC ►

NextSpace

Cancel

Done

11.7 AASTRAIPPHONETEXTMENU()

This class allows you to create a XML TextMenu object.

Include

- AstralPPhoneTextMenu.class.php

Methods

- setTitle(title) to setup the title of an object (optional)
 - title string
- setCancelAction(uri) to set the cancel parameter with the URI to be called on Cancel (optional)
 - uri string
- setDestroyOnExit() to set DestroyonExit parameter to 'yes', 'no' by default (optional)
- setBeep() to enable a notification beep with the object (optional)
- setLockIn(uri) to set the Lock-in tag to 'yes' and the GoodbyeLockInURI (optional)
 - uri string, GoodByeLockInURI
- setTimeout(timeout) to define a specific timeout for the XML object (optional)
 - timeout integer (seconds)
- setBackgroundColor(color) to change the XML object background color (optional)
 - color string, "red", "blue", ...
- addSoftkey(index,label,uri) to add custom soktkeys to the object (optional)
 - index integer, softkey number
 - label string

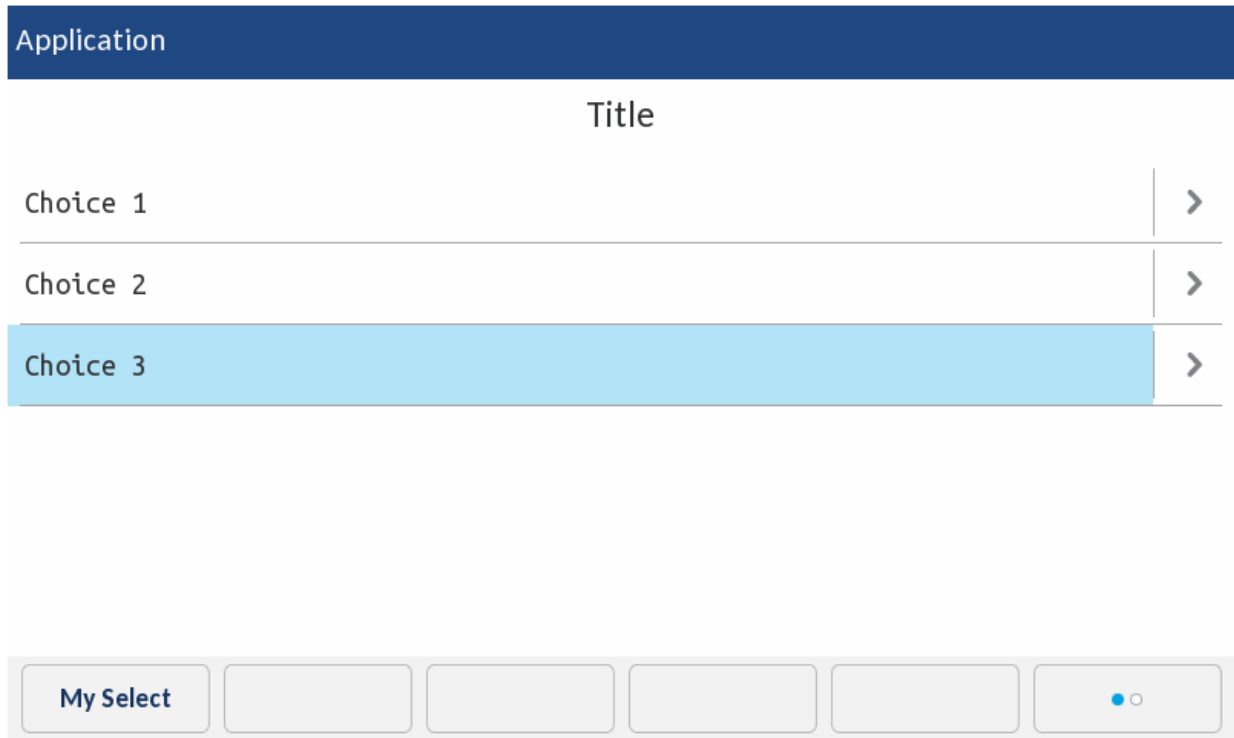
- uri string
- setRefresh(timeout,URL) to add Refresh parameters to the object (optional)
 - timeout integer (seconds)
 - URL string
- setEncodingUTF8() to change encoding from default ISO-8859-1 to UTF-8 (optional)
- generate() to return the generated XML for the object
- output(flush) to display the object
 - flush boolean optional, output buffer to be flushed out or not.
- setDefaultIndex(index) to set the default selection in the list (optional)
 - index index (1-30)
- setBase(base) to configure the menuitem base URI
 - base string
- resetBase() to reset the menuitem base URI
- setScrollConstrain() to avoid the list to wrap
- setNumberLaunch() to allow keypad selection
- setTouchLaunch() to allow quick item selection
- addEntry(name,url,selection,icon,dial,line,color,split,iconr1,iconr2,iconr3,iconr4) to add an element in the list to be displayed
 - name string or array(0=>Line1,1=>Line2,2=>Offset,3=>Char,4=>Mode)
 - url string
 - selection string
 - icon integer
 - dial string, phone number to dial
 - color string, "red", "black", ... (optional)
 - split integer, position of the split between line 1 and line 2 (optional)
 - iconr1 integer (optional)
 - iconr2 integer (optional)
 - iconr3 integer (optional)
 - iconr4 integer (optional)
- setScrollUp(uri) to set the URI to be called when the user presses the Up arrow (optional)
 - uri string
- setScrollDown(uri) to set the URI to be called when the user presses the Down arrow (optional)
 - uri string
- natsortbyname() to order the list, must not be used in conjunction with setBase or resetBase

Example

```
require_once('AstralPPhoneTextMenu.class.php');
$menu = new AstralPPhoneTextMenu();
$menu->setTitle('Title');
$menu->setDestroyOnExit();
$menu->setDeFaultIndex('3');
$menu->addEntry('Choice 2', 'http://myserver.com/script.php?choice=2', 'Value=2');
```

```
$menu->addEntry('Choice 1', 'http://myserver.com/script.php?choice=1', 'Value=1');
$menu->addEntry('Choice 3', 'http://myserver.com/script.php?choice=3', 'Value=3');
$menu->natsortByName();
$menu->addSoftkey('1', 'My Select', 'http://myserver.com/script.php?action=1');
$menu->addSoftkey('10', 'Exit', 'SoftKey:Exit');
$menu->output();
```

Output



11.8 AASTRAIPPHONETEXTSCREEN()

This class allows you to create a XML TextScreen object.

Include

- AastralPPhoneTextScreen.class.php

Methods

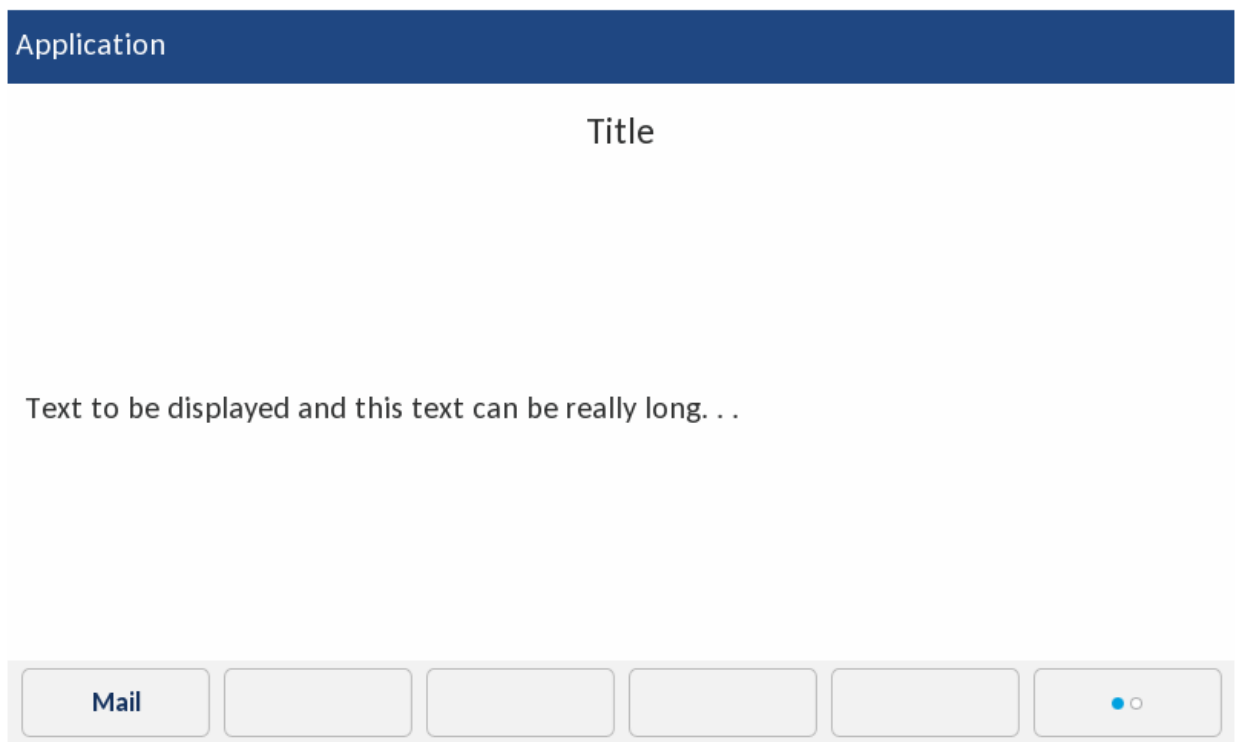
- setTitle(title) to setup the title of an object (optional)
 - title string
- setTopTitle(title,color,icon_index) to set the Top Title of the XML screen
 - title string
 - color string, "red", "blue", ... (optional)
 - icon_index integer, icon number
- setCancelAction(uri) to set the cancel parameter with the URI to be called on Cancel (optional)
 - uri string
- setDestroyOnExit() to set DestroyonExit parameter to 'yes', 'no' by default (optional)
- setBeep() to enable a notification beep with the object (optional)
- setLockIn(uri) to set the Lock-in tag to 'yes' and the GoodbyeLockInURI (optional)

- uri string, GoodByeLockInURI
- setAllowAnswer() to set the allowAnswer tag to 'yes' (optional)
- setAllowDrop() to set the allowDrop tag to 'yes' (optional)
- setAllowXfer() to set the allowXfer tag to 'yes' (optional)
- setAllowConf() to set the allowConf tag to 'yes' (optional)
- setTimeout(timeout) to define a specific timeout for the XML object (optional)
 - timeout integer (seconds)
- setBackgroundColor(color) to change the XML object background color (optional)
 - color string, "red", "blue", ...
- addSoftkey(index,label,uri) to add custom softkeys to the object (optional)
 - index integer, softkey number
 - label string
 - uri string
- setRefresh(timeout,URL) to add Refresh parameters to the object (optional)
 - timeout integer (seconds)
 - URL string
- setEncodingUTF8() to change encoding from default ISO-8859-1 to UTF-8 (optional)
- addIcon(index,icon) to add custom icons to the object (optional)
- generate() to return the generated XML for the object
- output(flush) to display the object
 - flush boolean optional, output buffer to be flushed out or not.
- setText(text) to set the text to be displayed.
 - text string
- setAllowDTMF() to allow DTMF passthrough on the object
- setDial(number,line) to set the number to be dialed as well as the line to use when going off-hook or with the custom softkey Softkey::Dial2
 - number string
 - line integer (optional)

Example

```
require_once('AastralPPhoneTextScreen.class.php');
$text = new AastralPPhoneTextScreen();
$text->setTitle('Title');
$text->setText('Text to be displayed and this text can be really long...');
$text->setDestroyOnExit();
$text->addSoftkey('1', 'Mail', 'http://myserver.com/script.php?action=1');
$text->addSoftkey('10', 'Exit', 'SoftKey:Exit');
$text->output();
```

Output



11.9 AASTRAIPPHONEIMAGEMENU()

This class allows you to create a XML ImageMenu object.

Include

- AstralPPhoneImageMenu.class.php

Methods

- setTitle(title,color,icon_index) to set the Top Title of the XML screen
 - title string
 - color string, "red", "blue", ... (optional)
 - icon_index integer, icon number
- setCancelAction(uri) to set the cancel parameter with the URI to be called on Cancel (optional)
 - uri string
- setDestroyOnExit() to set DestroyonExit parameter to 'yes', 'no' by default (optional)
- setBeep() to enable a notification beep with the object (optional)
- setLockIn(uri) to set the Lock-in tag to 'yes' and the GoodbyeLockInURI (optional)
 - uri string, GoodByeLockInURI

- `setCallProtection(notif)` to protect the XML object against incoming calls
 - `notif` enable/disable the display of an incoming call notification (optional)
- `setAllowAnswer()` to set the `allowAnswer` tag to 'yes' (optional)
- `setAllowDrop()` to set the `allowDrop` tag to 'yes' (optional)
- `setAllowXfer()` to set the `allowXfer` tag to 'yes' (optional)
- `setAllowConf()` to set the `allowConf` tag to 'yes' (optional)
- `setTimeout(timeout)` to define a specific timeout for the XML object (optional)
 - `timeout` integer (seconds)
- `setBackgroundColor(color)` to change the XML object background color (optional)
 - `color` string, "red", "blue", ...
- `addSoftkey(index,label,uri,icon_index)` to add custom softkeys to the object (optional)
 - `index` integer, softkey number
 - `label` string
 - `uri` string
 - `icon_index` integer, icon number
- `setRefresh(timeout,URL)` to add Refresh parameters to the object (optional)
 - `timeout` integer (seconds)
 - `URL` string
- `setEncodingUTF8()` to change encoding from default ISO-8859-1 to UTF-8 (optional)
- `addIcon(index,icon)` to add custom icons to the object (optional)
 - `index` integer, icon index
 - `icon` string, icon name or definition
- `generate()` to return the generated XML for the object
- `output(flush)` to display the object
 - `flush` boolean optional, output buffer to be flushed out or not.
- `setImage(image)` to define the image to be displayed
 - `image` string

- `setGDImage(GDImage)` to use a `GDImage` for display, the size is forced to 40x144
 - `GDImage` `GDImage`
- `setAlignment(vertical, horizontal)` to define image alignment
 - `vertical` `string, "top", "middle", "bottom"`
 - `horizontal` `string, "left", "middle", "right"`
- `setSize(height, width)` to define image size
 - `height` `integer (pixels)`
 - `width` `integer (pixels)`
- `setURIBase(uriBase)` to define the base URI for the selections
 - `uriBase` `string`
- `addURI(key, uri)` to add a selection key with its URI
 - `key` `string (1-9, * and #)`
 - `uri` `string`
- `setMode(mode)` to define the image mode to be displayed (normal, extended, fullscreen) (optional)
 - `mode` `string enum normal, extended, fullscreen`
- `setDoneAction(uri)` to set the URI to be called when the user selects the default "Done" key (optional)
 - `uri` `string`
- `setImageAction(uri)` to set the `imageAction` parameter with the URI to be called when user presses on the displayed image (optional)
 - `uri` `string`

Example

[illegible]

Output



11.10 AAstraIPPhoneSoftkey()

This class allows you to create a XML PhoneSoftkey object.

Include

- AastraIPPhoneSoftkey.class.php

Methods

- setEncodingUTF8() to change encoding from default ISO-8859-1 to UTF-8 (optional)
- generate() to return the generated XML for the object
- output(flush) to display the object
 - flush boolean optional, output buffer to be flushed out or not.
- addEntry(parameter,value) to add a configuration change for a dynamic attribute
 - parameter string, parameter name
 - value string, parameter value
- setTriggerDestroyOnExit() to set the triggerDestroyOnExit tag to "yes" (optional)

Example

```
require_once('AastraIPPhoneSoftkey.class.php');

$configuration = new AastraIPPhoneSoftkey();

$configuration->addEntry('prgkey2 icon','Icon:Home');

$configuration->addEntry('prgkey2 label','Demo Text');

$configuration->output();
```

In this example, the icon and the label of the prgkey2 are updated.

11.11 AASTRAIPPHONESTATUS()

This class allows you to create a XML PhoneStatus object.

Include

- AastralPPhoneStatus.class.php

Methods

- setEncodingUTF8() to change encoding from default ISO-8859-1 to UTF-8 (optional)
- generate() to return the generated XML for the object
- output(flush) to display the object
 - flush boolean optional, output buffer to be flushed out or not.
- setSession(session) to setup the session ID
 - session string
- setTriggerDestroyOnExit() to set the triggerDestroyOnExit tag to "yes" (optional)
- addEntry(index,message,type,timeout,uri,icon) to add a message to be displayed on the idle screen.
 - index integer
 - message string
 - type enum ("alert") optional
 - timeout integer (seconds) optional
 - uri string (unused)
 - icon integer (graphical phones only, optional)

11.12 EXAMPLES PROVIDED WITH THE PHP API

With the PHP API, Mitel provides some source code on how to use it.

sample.php for the regular XML objects

The whole directory "php_classes" must be installed on a HTTP server anywhere behind the root directory sample.php

On any type of phone, create a URL Line programmable key to call the sample.php script using the following uri:

<http://myserver.com/mydirectory/sample.php>

It includes examples for

- PhoneExecute
- PhoneInputScreen (3)
- PhoneTextMenu (2)
- PhoneTextScreen

- PhoneFormattedTextScreen
- PhoneStatus(2)

12 APPENDIX D: LOCALIZED INPUT CHARACTER SET

12.1 ENGLISH

Key	Upper case	Lower case
1	1.;=_,'&()	1.;=_,'&()
2	ABC2	abc2
3	DEF3	def3
4	GHI4	ghi4
5	JKL5	jkl5
6	MNO6	mno6
7	PQRS7	pqrs7
8	TUV8	tuv8
9	WXYZ9	wxyz9
0	0+	0+
*	* <SPACE>	* <SPACE>
#	#\#@	#\#@

12.2 FRENCH/FRANÇAIS

Key	Upper case	Lower case
1	1.;=_,'&()	1.;=_,'&()
2	ABC2ÂÂÇÁÂÆ	abc2ââçáâæ
3	DEF3ÉÊËË	def3éèêë
4	GHI4Ï	ghi4ï
5	JKL5	jkl5
6	MNO6ÑÓÔÕÖ	mno6ñóòôö
7	PQRS7	pqrs7
8	TUV8ÚÛÜÜ	tuv8úûüü
9	WXYZ9	wxyz9
0	0+	0+
*	* <SPACE>	* <SPACE>
#	#\#@	#\#@

12.3 SPANISH/ESPAÑOL

Key	Upper case	Lower case
1	1.;=_,'&()	1.;=_,'&()
2	ABC2ÁÂÇ	abc2áâç
3	DEF3ÉÊ	def3éè
4	GHI4Ï	Ghi4ïí
5	JKL5	Jkl5
6	MNO6ÑÓÔ	mno6ñóò

Key	Upper case	Lower case
7	PQRS7	pqrs7
8	TUV8ÚÜ	tuv8úü
9	WXYZ9	wxyz9
0	0+	0+
*	* <SPACE>	* <SPACE>
#	#\@	#\@

12.4 GERMAN/DEUTSCH

Key	Upper case	Lower case
1	1.;;=_,'&()	1.;;=_,'&()
2	ABC2ÄÀ	abc2äà
3	DEF3É	def3é
4	GHI4	ghi4
5	JKL5	jkl5
6	MNO6Ö	mno6ö
7	PQRS7ß	pqrs7ß
8	TUV8Ü	tuv8ü
9	WXYZ9	wxyz9
0	0+	0+
*	* <SPACE>	* <SPACE>
#	#\@	#\@

12.5 ITALIAN/ITALIANO

Key	Upper case	Lower case
1	1.;;=_,'&()	1.;;=_,'&()
2	ABC2ÀÇ	abc2àç
3	DEF3ÈÊ	def3èè
4	GHI4	ghi4
5	JKL5	jkl5
6	MNO6ÓÒ	mno6óò
7	PQRS7	pqrs7
8	TUV8Ù	tuv8ù
9	WXYZ9	wxyz9
0	0+	0+
*	* <SPACE>	* <SPACE>
#	#\@	#\@

12.6 PORTUGUESE/PORTUGÊS

Key	Upper case	Lower case
1	1.;=_,-'&()	1.;=_,-'&()
2	ABC2ÁÂÃÄÇ	abc2áâãäç
3	DEF3ÉÊ	def3éê
4	GHI4Í	ghi4í
5	JKL5	jkl5
6	MNO6ÓÔÕ	mno6óôõ
7	PQRS7	pqrs7
8	TUV8ÚÛ	tuv8úü
9	WXYZ9	wxyz9
0	0+	0+
*	* <SPACE>	* <SPACE>
#	#\#@	#\#@

13 APPENDIX D: CSV BASED DIRECTORY

13.1 INTRODUCTION

The application described in this chapter is a generic directory application using a CSV file as data source as well as a server-side speed dial application which can be linked from the directory application.

The directory application allows the user to perform a lookup based on

- The first letters of the first name
- The first letters of the last name
- The first letters of the Company name
- (at least 3) Letters anywhere in the complete name or company name

Also, it allows the user to configure:

- The results display format: 'first name last name' or 'last name first name'
- Sorting index: 'last name' or 'first name'

The directory application can also be password protected (configured in config/directory.conf), the password is requested only for the first use, once authenticated by its MAC address, the phone no longer requires the password check.

The source code is provided in the XML SDK as a zip file called csv_directory_1.5.1.zip



Note: the source code is not supported by Mitel, it is provided only as an example.

13.2 PHONE COMPATIBILITY

The directory and speed dial applications are available for

- Mitel 6920/6930/6940

But some limitations apply in the directory application when the phone is a non softkey phone a user cannot:

- set a speed dial from the directory application
- change sorting index and display format
- access to record details
- perform a search using first name or last name

13.3 INSTALLATION

The provided scripts must be extracted under the ROOT directory of the HTTP server in a 'xml' directory, a cache directory (default

/var/cache/aastra) must also be created with read/write access for the HTTP server user.

The cache directory can be configured in config/server.conf, see chapter 13.6.1 for more details.

13.4 XML KEY CONFIGURATION DIRECTORY

The uri to use is

<http://myserver.com/xml/directory/directory.php?user=USER&source=S>

[SOURCE](#) Where

- myserver.com is the name or IP address of your HTTP server
- USER is the userID, if not provided the script uses the phone MAC address as the userID.
- SOURCE is the name of the csv file located in the 'directory' directory, if not provided, the script uses the file configured as default in config/directory.conf

Speed dial

The uri to use is

<http://myserver.com/xml/directory/speed.php?user=>

[USER](#) Where

- myserver.com is the name or IP address of your HTTP server
- USER is the userID, if not provided the script uses the phone MAC address as the userID.



Note: USER must be the same for both applications in order to have the speed dial integrated in the directory application.

13.5 CSV FILE FORMAT

The CSV file must respect the following format:

First name, Last name, Company, Title, Work, Home, Mobile

The application is provided with a sample database 'directory/directory.txt' where all the fields have been randomly generated.

13.6 CONFIGURATION FILES

13.6.1 SERVER.CONF

This file includes configuration parameters for the XML server.

```
[General]

# Public IP address (optional), if not provided $_SERVER['HTTP_HOST'] is
used public=

# Path for the Mitel cache directory (optional), if not provided
/var/cache/aastra is used cache=

# Path for the TFTP root directory (optional) if not provided /tftpboot is
used tftp=

# Path for the XML directory behind the HTTP Server (optional),
default is xml xmldirectory=
```

When activated, the traces are stored in a log file located in the 'cache' directory (by default /var/cache/aastra) indexed by the date, a file is generated per day.

13.6.2 DIRECTORY.CONF

This file includes configuration parameters for the Directory application.

```
#####

#
directory.conf
#

# Configuration file for the directory application
#####

#####

# Global
configuration #

# [general]

# password To protect the application with a password

# speeddial Configure is speeddial is enabled (0 or
1)(optional) # defaultName of the default csv file located in
the directory

#####

[General]
password=
speeddial=
default=directo
ry

#####
```

```

#      Dialing
profile #

# [Dialplan]

# country          country code

# long distance    long distance
prefix            #      international
international prefix #      outgoing
outgoing prefix

# local            list of local phone numbers prefixes separated by a
comma
#####
#####

[Dialplan
]
country=1

long distance=1
international=0
11 outgoing=9

local=

```

13.7 APPLICATION FILES

/	Root Directory
language.ini	Language configuration file
license.txt	License file
/directory/	Directory for the applications
directory.php	Directory application
directory.txt	Sample CSV file
speed.php	Speed dial application
/config/	Directory for configuration files
server.conf	Server configuration

directory.conf	Directory configuration
/include/	Directory for common includes
AastraCommon.php	Common Mitel functions
AastraIPPhone.php	
AastraIPPhone.class.php	Mitel XML objects php classes
AastraIPPhoneConfiguration.class.php	
AastraIPPhoneConfigurationEntry.class.php	
AastraIPPhoneExecute.class.php	
AastraIPPhoneExecuteEntry.class.php	
AastraIPPhoneInputScreen.class.php	
AastraIPPhoneInputScreenEntry.class.php	
AastraIPPhoneSoftkey.class.php	
AastraIPPhoneSoftkeyParam.class.php	
AastraIPPhoneSoftkeyEntry.class.php	
AastraIPPhoneStatus.class.php	
AastraIPPhoneStatusEntry.class.php	
AastraIPPhoneTextMenu.class.php	
AastraIPPhoneTextMenuEntry.class.php	
AastraIPPhoneTextScreen.class.php	

14 APPENDIX E: LDAP DIRECTORY

14.1 INTRODUCTION

This appendix describes how to provide access to the Corporate Directory (Active Directory) from the Mitel phones. As LDAP protocol is used to access Active Directory, this document is also valid for any other LDAP enabled directory server. This document is solely about user initiated directory lookup, it does not cover Corporate Directory based caller ID lookup for incoming or outgoing calls.

There is no native LDAP client in the Mitel phone, but the phone's inbuilt XML browser can be used to access a XML Proxy Server which then communicates with the LDAP server. This architecture offers more flexibility than a native LDAP client in the phone since the logic of the LDAP directory application lies in the XML LDAP application (e.g. a PHP script) running on the XML Proxy Server rather than in the LDAP client embedded in the phone's firmware. The following features can be easily controlled by changing the XML LDAP application on the XML Proxy Server:

- Customized search fields on the phone (e.g. first name / last name / any name / department / ...)
- Customized LDAP-attributes to be included in the search result (such as title, department, address)
- Number translation (digits to be removed from / added to the numbers found in the LDAP before dialling – e.g. external prefix)

The message flow is given by the diagram below:

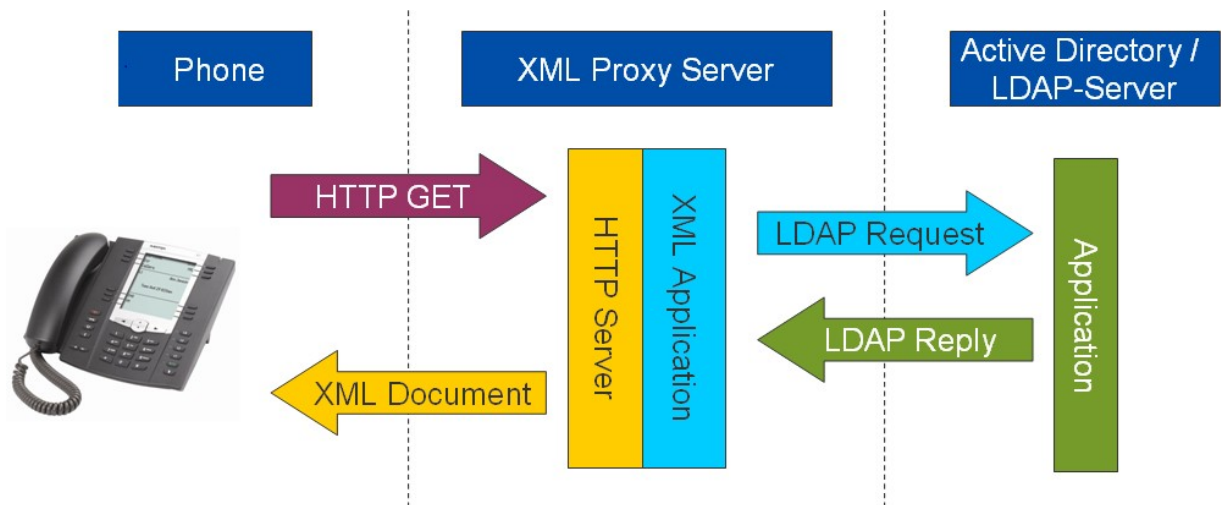


Figure 77: LDAP directory message flow

The directory application allows the user to perform a lookup based on

- The first letters of the first name AND last name
- (at least 3) Letters anywhere in the complete name

The source code is provided in the XML SDK as a zip file called ldap_directory_1.5.1.zip



Note: the source code is not supported by Mitel, it is provided only as an example.

14.2 INSTALLATION

The provided scripts must be extracted under the ROOT directory of the HTTP server in a 'xml' directory, a cache directory (default

/var/cache/aastra) must also be created with read/write access for the HTTP server user.

The cache directory can be configured in config/server.conf, see chapter 14.5.1 for more details.

The Web server must support PHP 5.x or higher and the PHP LDAP extension must be available in PHP. Typically, Apache Web server is used. Apache runs on most operating systems.

Windows platforms

To enable the PHP LDAP extension, this line must be inserted into php.ini:

extension=php_ldap.

Linux platforms

Just install the php-ldap package (via yum or via rpm)

To check whether PHP 5.x (or higher) and the PHP LDAP extensions are properly installed, create a PHP script with the following content and save it under the Web server's root directory:

```
<?php phpinfo(); ?>
```

Then point your Web browser to this script. Check the PHP version and make sure there is a section "ldap" with a table containing the key "LDAP Support" and corresponding value "enabled"

14.3 NETWORK REQUIREMENTS

14.3.1 PHONE -> XML PROXY SERVER

The phone must be able to reach the XML Proxy Server. Typically, the XML Proxy Server will run on port 80/tcp, but any other port can be used as well. The phone can be behind NAT as the TCP requests will always be phone initiated.

In case there is a firewall between the phone and the XML Proxy Server, the firewall must allow phone initiated TCP sessions towards the Web server port on the XML Proxy Server.

14.3.2 XML PROXY SERVER -> LDAP-SERVER

The XML Proxy Server must be able to reach the LDAP-Server on port 389/tcp (default LDAP port) or 3268/tcp (default LDAP port on Active Directory Server) or any other port that is used for LDAP access.

Requests will always be XML Proxy Server initiated.

14.4 ACTIVE DIRECTORY / LDAP-SERVER REQUIREMENTS

The LDAP / Active Directory Server must allow LDAP search requests. Depending on the server's configuration, anonymous access (anonymous bind) is allowed. If not, a username (User DN) and password must be provided. Active Directory typically does not allow anonymous access.

Also the Search Base / Root DN needs to be known. This information should be requested from the LDAP / Active Server administrator.

To test the LDAP server access, use a tool like "Softerra LDAP Browser" (free). If anonymous access is supported, setup a new profile with "Anonymous Bind" selected. Otherwise provide User DN and password. Configure the server's IP address and port plus the "Base DN" and you should be able to browse the LDAP tree.

14.5 CONFIGURATION FILES

14.5.1 SERVER.CONF

This file includes configuration parameters for the XML server.

```
[General]

# Public IP address (optional), if not provided $_SERVER['HTTP_HOST'] is
used public=

# Path for the Mitel cache directory (optional), if not provided /var/cache/aastra
is used cache=

# Path for the TFTP root directory (optional) if not provided /tftpboot is
used tftp=

# Path for the XML directory behind the HTTP Server (optional), default is
xml xmldirectory=
```

When activated, the traces are stored in a log file located in the 'cache' directory (by default /var/cache/aastra) indexed by the date, a file is generated per day.

14.5.2 LDAP_DIRECTORY.CONF

This file includes configuration parameters for the LDAP Directory application.

```

#####
#####

#
ldap_directory.conf
#

# Configuration file for the LDAP directory application
#####
#####

#####

# Global
configuration #

# [LDAP_Server]

# hostname Hostname or IP address of LDAP server

# port TCP server port of LDAP server. Typically this is 389
or 3268 # baseDN Base / Root Search DN. Example: dc=company, dc=com

# userdomain User domain to connect to the LDAP server (optional)

# username User DN to connect to the LDAP server. Leave blank if anonymous
bind is used.

# password Password for the User DN. Leave blank if anonymous bind is used.

#####
#####

[LDAP_Serve
r]
hostname=
port=3268
basedn=
userdomain=
username=
password=

#####
#####

# Dialing
profile #

# [Dialplan]

# countrycode Country code

```

```

#           If phone number starts with "+<countrycode>", this will be
removed #

# longdistance Long distance prefix

#           This prefix will be added to the number in case +<countrycode>
has been stripped.

#           Leave blank if not
needed. #

# international International prefix

#           The "+" sign will be replaced with this
prefix #

# outgoing Outgoing prefix

#           Prefix that needs to be appended for outgoing calls

#           (all call but local calls), e.g. 0
or 9 #

# local List of local PBX number prefixes separated by a comma (in
national format).

#           Prefix will be removed, no outgoing prefix will be added.

#           Examples: local=905760,978262

#           --> 9057602222 will be replaced by 2222 and 9782623333 will be
replaced by 3333.

#

# localextlen Numbers with this number of digits or less will be treated as local
PBX extensions

#           No prefix added (useful in case numbers are stored in local
format in LDAP)

```

```
#####  
###  
[Dialplan]  
countrycode=1  
longdistance=1  
international=011  
outgoing=  
1 1
```

58016588-00

This page is intentionally left blank