MODEL

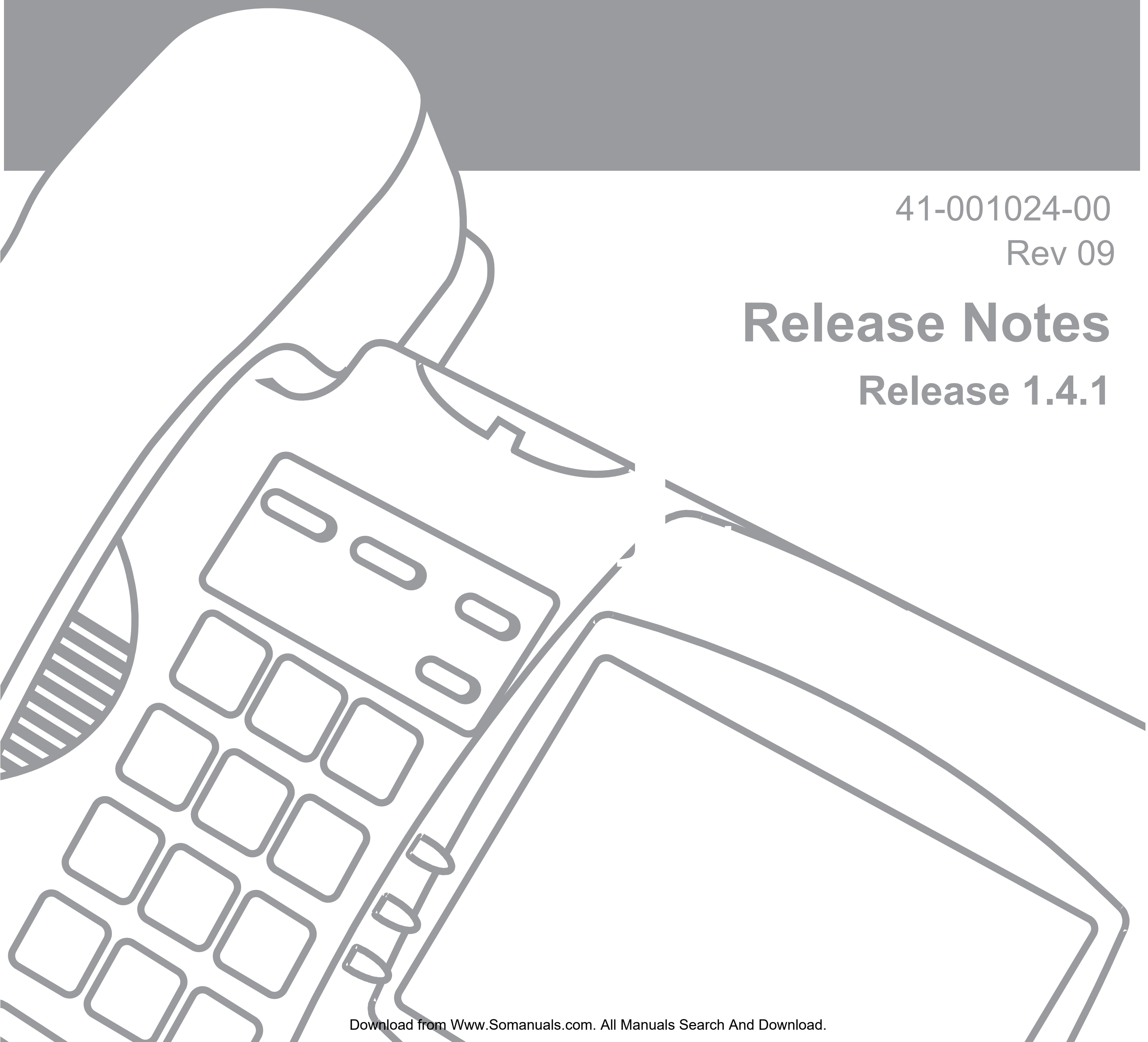
480i, 480i CT, 9112i, 9133i

SIP IP PHONE

41-001024-00
Rev 09

Release Notes

Release 1.4.1



Aastra Telecom will not accept liability for any damages and/or long distance charges, which result from unauthorized and/or unlawful use. While every effort has been made to ensure accuracy, Aastra Telecom will not be liable for technical or editorial errors or omissions contained within this documentation. The information contained in this documentation is subject to change without notice.

Contents

SIP IP Phone Models 480i, 480i CT, 9112i, 9133i Release Notes 1.4.1

About this Document

This document provides information specific to the SIP IP Phone release 1.4.1. It includes the following information:

- General Information (release content, hardware supported, bootloader requirements, and upgrade notes)
- New Features in 1.4.1
- Issues Resolved in Release 1.4.1
- Known Anomalies in 1.4.1
- Contacting Aastra Telecom Support

General Information

Release Content Information

This document provides release content information on the Aastra 480i, 480i CT, 9112i, and 9133i SIP IP phone firmware.

Model	Release Name	Release Version	Release Filename	Release Date
480i	Generic SIP	1.4.1	FC-000032-01-09	November 2006
480i CT	Generic SIP	1.4.1	FC-000040-00-09	November 2006
9112i	Generic SIP	1.4.1	FC-000058-01-09	November 2006
9133i	Generic SIP	1.4.1	FC-000046-01-09	November 2006

Hardware Supported

This release of firmware is compatible with the following Aastra IP portfolio products:

- 480i
- 480i CT
- 9112i
- 9133i

Bootloader Requirements

This release of firmware is compatible with the following Aastra IP portfolio product bootloader versions:

- 480i - Bootloader 1.1.0.4 or above
- 480i CT - Bootloader 1.1.0.4 or above
- 9112i - Bootloader 1.1.0.10 or above
- 9133i - Bootloader 1.1.0.10 or above

Upgrade Notes

This section provides notes that customers should be aware of before upgrading to IP phone firmware release 1.4.1. Aastra Telecom recommends that customers read this section completely and thoroughly prior to the upgrade to avoid any known issues with the upgrade process.

- The recommended base IP Phone firmware to upgrade to this new firmware release is 1.2.2 or above.
- Users who previously adjusted the audio transmit and receive gains for the Aastra IP Phones should reset the values to "0" and re-tune their phones again with release 1.4.1 if necessary. This is required to avoid high transmit levels heard by the far-end after upgrading to 1.4.1. To set audio transmit and receive gains in the 1.4.1 configuration files, see the SIP IP Phone Administrator Guide Release 1.4 or the SIP IP Phone Release Notes Version 1.3.1.
- Users upgrading from release 1.2.2 to 1.4.1 may experience post-upgrade configuration issues IF they previously configured their phones solely using the telephone user interface or Aastra Web user interface. Users may need to reconfigure their settings again after the upgrade due to new file system enhancements. This issue does not affect customers who configure their phones using a TFTP/FTP/HTTP server.
- Users who wish to downgrade from release 1.4.1 to 1.2 must downgrade to release 1.2.2 prior to downgrading to release 1.2, in order to restore file system structure. Failure to downgrade to version 1.2.2 first may result in configuration parameters being ignored.

New Features in 1.4.1

Description

This section describes the features new to the IP phones in release 1.4.1. The features apply to all IP Phone models (480i, 480i CT, 9112i, and 9133i), unless specifically stated that only a particular IP phone model supports the feature.

The following table lists the new features in release 1.4.1. The paragraphs following this table describe each feature in detail.

Feature	Description	Page
All Models		
SIP Timers	**The following are SIP timers that were added or changed on the IP phones:**	
	SIP Registration Renewal Timer - A new parameter called **sip registration renewal timer** has been added that allows an administrator to control when the phone renews SIP registrations. Configurable via configuration files and Aastra Web UI.	page 7
	SIP Registration Timeout Retry Timer - An administrator can now set the time, in seconds, that the phone waits until it attempts to register the phone after a REGISTER message times outs. This new parameter is called **sip registration timeout retry timer**. Configurable via configuration files and Aastra Web UI.	page 9
	"SIP Registration Retry Timer" Web UI Parameter Name Change - In the Aastra Web UI only, the "registration retry timer" parameter name has changed to **registration failed retry timer.**	page 10
	Explicit Message Waiting Indicator (MWI) Subscription Period - An administration can now set the length of time, in seconds, of an explicit MWI subscription. This value allows the phone to re-subscribe the explicit MWI subscription before the timeout value is reached.	page 11
Prefix Dialing	An administrator can now enter a digit(s) at the end of the Local Dial Plan parameter string, which allows the IP phone to automatically add the digit as a prefix to a dialed number.	page 13

Feature	Description	Page
Last Number Redial	A user can now press the REDIAL key to dial the last number dialed. They can also press the REDIAL button once, scroll the list of numbers, then press the REDIAL button again to dial the number that displays on the screen.	page 15
XML Objects and Enhancements	**The following are new XML objects on the IP phone:**	
	• XML: AastraIPPhoneStatus object - The IP phones now display a status message on a single designated line on the phone's idle screen when XML information is pushed from the servers. The 480i/480i CT phones display messages on the second line (where "No Service" would display). The 9112i/9133i phones display messages on the first line (overriding the DisplayName). Long messages that are wider then the phone screen get truncated.	page 16
	• XML: AastraIPPhoneExecute Object - This object provides the ability to execute commands on the phone using XML. Specific commands you can use with this object are **Reset** and **NoOp**.	page 23
	The following are XML enhancements on the IP phone:	
	• XML: Action URI - This feature provides administrators the ability to specify a URI for the phone to GET when certain events occur.	page 25
	• XML: Softkey URI - This feature allows the user to specify variables in the XML softkey URIs that are bound when the key is pressed.	page 30
	• XML: HTTP Refresh Header - All current XML screen objects now have the ability to be refreshed by adding a **Refresh** and **URL** setting to the HTTP headers.	page 33
Backup Proxy/Registrar Support	The IP phones now support the use of a backup SIP proxy/ registrar. You can configure this feature on a global or per-line basis via the configuration files or the Aastra Web UI.	page 34
Auto-discovery Using mDNS	Release 1.4.1 introduces a process that allows the phones to auto-discover a TFTP server using mDNS, and subsequently be automatically configured by a TFTP server.	page 39

Feature	Description	Page
IP Phone Features for Sylantro Servers	**The following are new features added to the IP phones for Sylantro servers:** Last Call Return (lcr) Support - A new feature has been added to the IP phones that allow a user or administrator to configure a "last call return" function on a softkey or programmable key. This feature is for Sylantro servers only. You can configure the “lcr” softkey feature via the configuration files and the Aastra Web UI. Support for additional “Alert Info” keywords for distinctive ringing - The IP phones now allow you to configure new distinctive ringing priority alert parameters for Sylantro servers. These “info” parameters allow you to configure specific priority alert tones for each parameter that may appear as key words in the “Alert-Info” header of a Sylantro server. The new keywords are **alert-acd**, **alert-community-1**, **alert-community-2**, **alert-community-3**, and **alert-community-4**. You can configure these new parameters via the configuration files or the Aastra Web UI.	 page 40 page 41
Startup Enhancement	Upon phone startup, the maximum time the phone spends attempting to contact the configuration server (TFTP, FTP or HTTP) has been reduced.	page 45
480i CT Only		
Single Call Restriction (480i CT only)	A new feature has been implemented on the 480i CT that allows an administrator to enable or disable a single call restriction between the 480i CT and a call server.	page 46
9112i/9133i Only		
Addition of DisplayName1 & DisplayName2 (now also applies to 9112i/9133i)	The 9112i and 9133i IP phones now support the parameters, displayName1 and displayName2. Configurable via the configuration files and the Aastra Web UI.	page 48

SIP Timers

The following paragraphs describe SIP timers that have been added or changed in Release 1.4.1.

SIP Registration Renewal Timer

A new parameter has been added to the IP phones that enables an administrator to control when registration renewals occur. The new parameter **sip registration renewal timer** specifies the length of time, in seconds, before the expiration of an existing registration, that the registration is renewed. For example, if the value is set to 20, then 20 seconds before the registration is due to expire, a new REGISTER message is sent to the registrar to renew the registration.

The parameter may be set via the configuration files and the Aastra Web UI.

Configuring via the Configuration Files

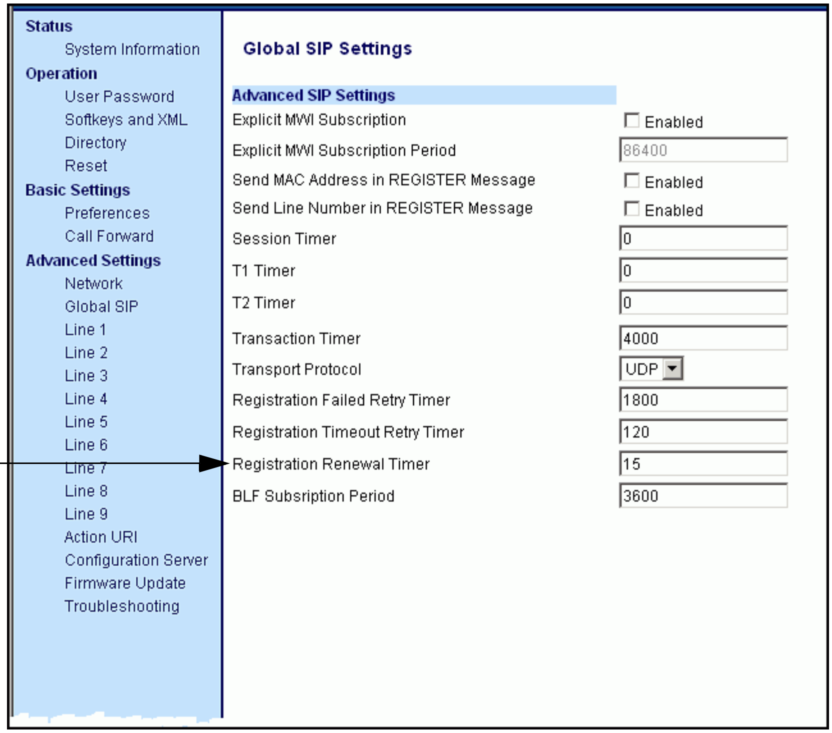
You use the following parameter in the configuration files to control when the phone renews SIP registration:

Parameter – *sip registration renewal timer* *Registration Renewal Timer* (in Web UI)	**Aastra Web UI** Advanced Settings->Global SIP-> Advanced SIP Settings **Configuration Files** aastra.cfg, <mac>.cfg
Description	The length of time, in seconds, that the phone renews registrations.
Format	Integer
Default Value	15
Range	0 to 214748364 The value set for this parameter should be between 0 and the value set for the registration period.
Example	sip registration renewal timer: 10

Configuring via the Aastra Web UI

You can set the Registration Renewal Timer in the Aastra Web UI at **Advanced Settings->Global SIP->Advanced SIP Settings**.

Registration Renewal Timer



SIP Registration Timeout Retry Timer

An administrator can now set the length of time, in seconds, that the phone waits until it re-attempts to register after a REGISTER message times out. This parameter is called, **sip registration timeout retry timer**, and is configurable via the configuration files and the Aastra Web UI.

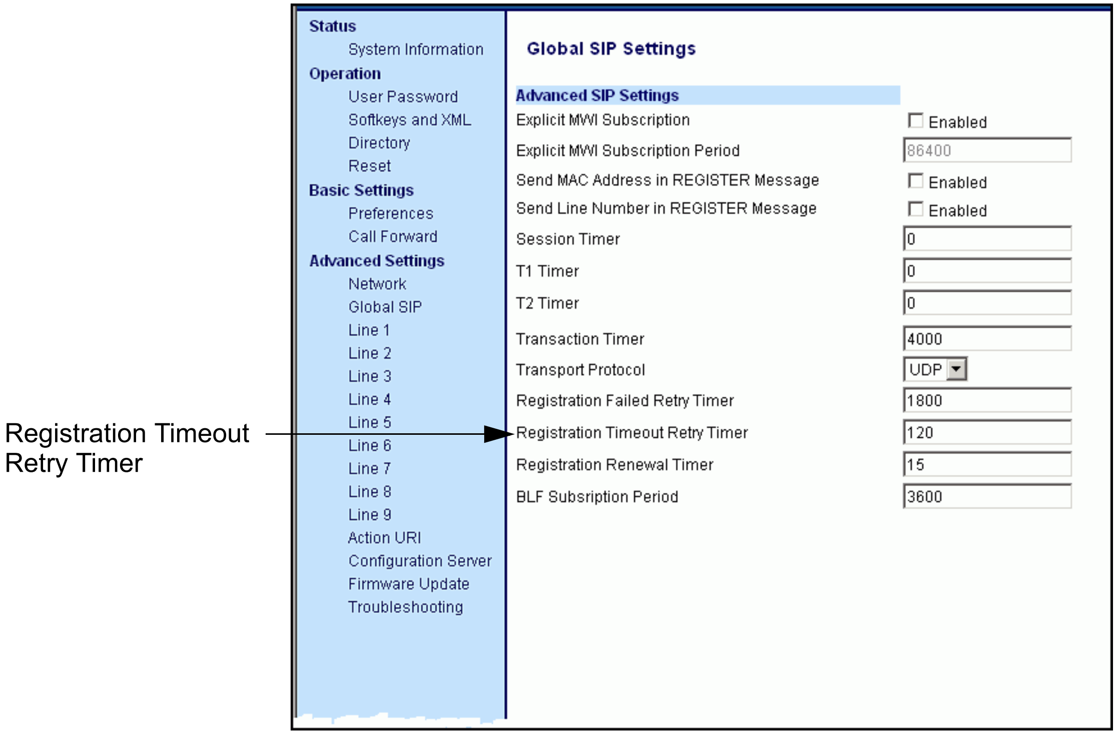
Configuring via the Configuration Files

You use the following parameter in the configuration files to configure the registration timeout retry timer.

Parameter – *sip registration timeout retry timer* *Registration Timeout Retry Timer* (in Web UI)	**Aastra Web UI** **Configuration Files**	Advanced Settings->Global SIP-> Advanced SIP Settings aastra.cfg, <mac>.cfg
Description	Specifies the length of time, in seconds, that the phone waits until it re-attempts to register after a REGISTER message times out. **Note:** If this parameter is set lower than 30 seconds, the phone uses a minimum timer of 30 seconds.	
Format	Integer	
Default Value	120	
Range	30 to 214748364	
Example	sip registration timeout retry timer: 150	

Configuring via the Aastra Web UI

You can set “Registration Timeout Retry Timer” in the Aastra Web UI at **Advanced Settings->Global SIP->Advanced SIP Settings.**



“SIP Registration Retry Timer” Web UI Parameter Name Change

In the Aastra Web UI only, the **registration retry timer** parameter name has been changed to “**registration failed retry timer**”. The functionality of this parameter has not changed. The default value is still 1800 seconds.

Explicit Message Waiting Indicator (MWI) Subscription Period

The IP phones support a new feature for MWI that allows the administrator to set a requested duration, in seconds, before an explicit MWI subscription times out. The phone re-subscribes to MWI before the subscription period ends.

The new parameter you can configure is "**sip explicit mwi subscription period**". You can configure this parameter using the configuration files or the Aastra Web UI.

Configuring via the Configuration Files

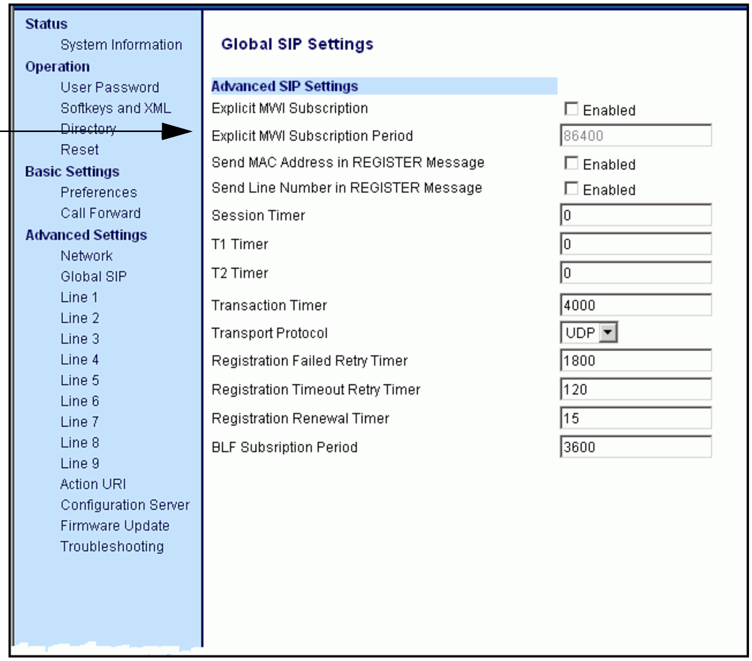
You configure the explicit MWI timeout using the following parameter in the configuration files.

Parameter – *sip explicit mwi subscription period* *Explicit MWI Timeout* (in Web UI)	**Aastra Web UI** Advanced Settings->Global SIP-> Advanced SIP Settings **Configuration Files** aastra.cfg, <mac>.cfg
Description	The requested duration, in seconds, before the MWI subscription times out. The phone re-subscribes to MWI before the subscription period ends.
Format	Integer
Default Value	86400
Range	30 - 214748364
Example	sip explicit mwi timeout: 30

Configuring via the Aastra Web UI

You configure "**Explicit MWI Subscription Period**" in the Aastra Web UI at **Advanced Settings->Global SIP->Advanced SIP Settings**.

Explicit MWI Subscription Period



Prefix Dialing

The IP phones now support a prefix dialing feature for outgoing calls.

How it works

You can manually dial a number or dial a number from a list. The phone automatically maps the pre-configured prepended digit in the configuration, to the outgoing number. When a match is found, the prepended digits are added to the beginning of the dial string and the call is dialed.

Note: The prepend digits are also added if the dialing times-out on a partial match.

You can enable this feature by adding a prepend digit(s) to the end of the **Local Dial Plan** parameter string in the configuration files or the Aastra Web UI at *Basic Settings->Preferences->General*.

For example, if you add a prepend map of "[2-9]XXXXXXXXX,91", the IP phone adds the digits "91" to any 10-digit number beginning with any digit from 2 to 9 that is dialed out. Other examples of prepend mappings are:

- **1X+#,9** (Prepends 9 to any digit string beginning with "1" and terminated with "#".)
- **6XXX,579** (Prepends "579" to any 4-digit string starting with "6".)
- **[4-6]XXXXXX,78** (Prepends "78" to any 7-digit string starting with "4", "5", or "6".)

Note: You can configure a local dial plan via the configuration files or the Aastra Web UI.

Example

If you enter the following dial string for a local dial plan:

```
sip dial plan: 1+#,9
```

where “9” is the prepended digit, and you dial the following number:

```
15551212
```

the IP phone automatically adds the “**9**” digit to the beginning of the dialed number before the number is forwarded as **915551212**.

Configuring via the Configuration Files

You use the following parameter to configure local dial plan with a prepended digit(s).

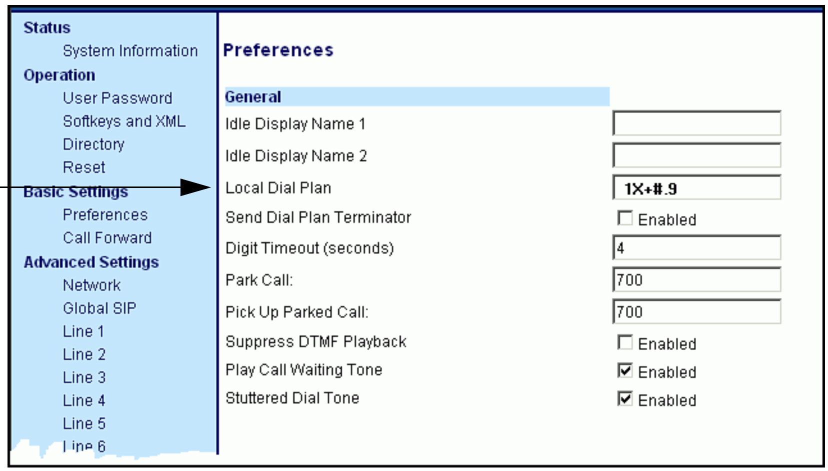
Parameter – *sip dial plan* *Local Dial Plan* (in Web UI)	**Aastra Web UI** Basic Settings->Preferences **Configuration Files** aastra.cfg, <mac>.cfg
Description	A dial plan describes the number and pattern of digits that a user dials to reach a particular telephone number. The SIP local dial plan is as follows: **Symbol** — **Description** 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 — Digit symbol X — Match any digit symbol (wildcard) *, #, . — Other keypad symbol | — Expression inclusive OR + — 0 or more of the preceding digit symbol or [] expression [] — Symbol inclusive OR - — Used only with [], represent a range of acceptable symbols; For example, [2-8] “,” (open/close quotes) — In the configuration files, enter the sip dial plan value using quotes. **Note:** You can configure prefix dialing by adding a prepend digit to the dial string. For example, if you add a prepend map of “[2-9]XXXXXXXXX,91”, the IP phone adds the digits “91” to any 10-digit number beginning with any digit from 2 to 9 that is dialed out. Other examples of prepend mappings are: • **1X+#,9** (• **6XXX,579** (Prepends “579” to any 4-digit string starting with “6”.) • **[4-6]XXXXXX,78** (Prepends “78” to any 7-digit string starting with “4”, “5”, or “6”.
Format	Alphanumeric characters
Default Value	X+#|XX+*

Range	Up to 127 alphanumeric characters
Example	sip dial plan: "1X+#,9”

Configuring via the Aastra Web UI

You configure a local dial plan string in the Aastra Web UI at **Basic Settings->Preferences**. The following illustration shows a Local Dial Plan dial string in the Aastra Web UI using a prepended digit.

Local Dial Plan



Last Number Redial

The IP phones now have an enhanced redial user interface that allows a user to quickly redial the last number that was dialed out from the phone. You can:

- Press the REDIAL button twice to redial the last number dialed.
- Press the REDIAL button once, scroll the list of numbers, then press the REDIAL button again to dial the number that displays on the screen.

This feature is static and is not configurable.

XML Objects and Enhancements

The following paragraphs describe the XML enhancements in release 1.4.1.

XML: AastraIPPhoneStatus object

A new XML **AastraIPPhoneStatus** object has been implemented on the IP phones. This object provides the ability to display a status message on a single designated line on the phone's idle screen when XML information is pushed from the servers.

The 480i/480i CT phones display messages on the second line in the phone window. (where "No Service" would display if there was no service. If there is no service on the phone, the "No Service" message overrides the XML object message). The 9112i/9133i phones display messages on the first line (overriding the DisplayName). Long messages that are wider then the phone screen get truncated.

If the phone receives multiple messages, the first message received displays first and the remaining messages scroll consecutively one at a time. Messages remain displayed until they are removed (by the server) or the phone reboots. The AastraIPPhoneStatus object feature is always enabled.

Note: You can set the amount of time, in seconds, that a message displays to the phone before scrolling to the next message. For more information about this feature, see "Scroll Delay Option" on page 21.

AastraIPPhoneStatus Structure

The **AastraIPPhoneStatus** object describes the structure of the XML document that is used to send status messages to the phone. The basic structure of the AastraIPPhoneStatus object is:

```
<AastraIPPhoneStatus>
    <Session>My session ID</Session>
    <Message index="Msg index">Message</Message>
<! -- Additional status messages may be added under new Message tags-->
<AastraIPPhoneStatus/>
```

The ”**My Session ID**” attribute must be unique to the application sending the XML object to the phone. The application generates the session ID, which could be a combination of letters and numbers. There is a maximum of one **<Session**> tag per PhoneStatus object, so the **<Session**> tag is optional.

Examples

Example 1: The following is an example of using the AastraIPPhoneStatus object:

```
<AastraIPPhoneStatus>
   <Session>abc12345</Session>
<Message index="3">Server side call forwarding disabled</Message>
<AastraIPPhoneStatus/>
```

In this example, the AastraIPPhoneStatus object sends the default behavior with the status message (i.e., the status message is added to the scroll list).

Example 2: You can also use the AastraIPPhoneStatus object to remove status messages from the display, by setting an empty tag for the <Message index> tag.

The following example removes the status message that was posted to the phone in Example 1.

```
<AastraIPPhoneStatus>
   <Session>abc12345</Session>
<Message index="3"/>
<AastraIPPhoneStatus/>
```

Beep Option

You can enable or disable a BEEP option in the AastraIPPhoneStatus object. When the phone receives a status message, the BEEP notifies the user that the message is being displayed. The following attribute enables/disables the BEEP from being heard:

`< AastraIPPhoneStatus  Beep="yes|no">` (case sensitive)

This attribute is optional. If notification is required, the attribute must be in the ROOT. 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.

Beep Option via Configuration Files and Aastra Web UI

The BEEP option can also be enabled or disabled via the configuration files and the Aastra Web UI using the following parameters:

- **xml beep notification** (via configuration files)
- **XML Beep Support** (via the Aastra Web UI)

The value set in the configuration files and Aastra Web UI override the attribute you specify in the AastraIPPhoneStatus object.

For example, if the AastraIPPhoneStatus object has the attribute of **Beep="yes"**, and you uncheck (disable) the "**XML Beep Support**" in the Aastra Web UI, the phone does not beep when it receives an AastraIPPhoneStatus object.

Setting the BEEP option in the configuration files and the Aastra Web UI is dynamic and applies to the phone immediately.

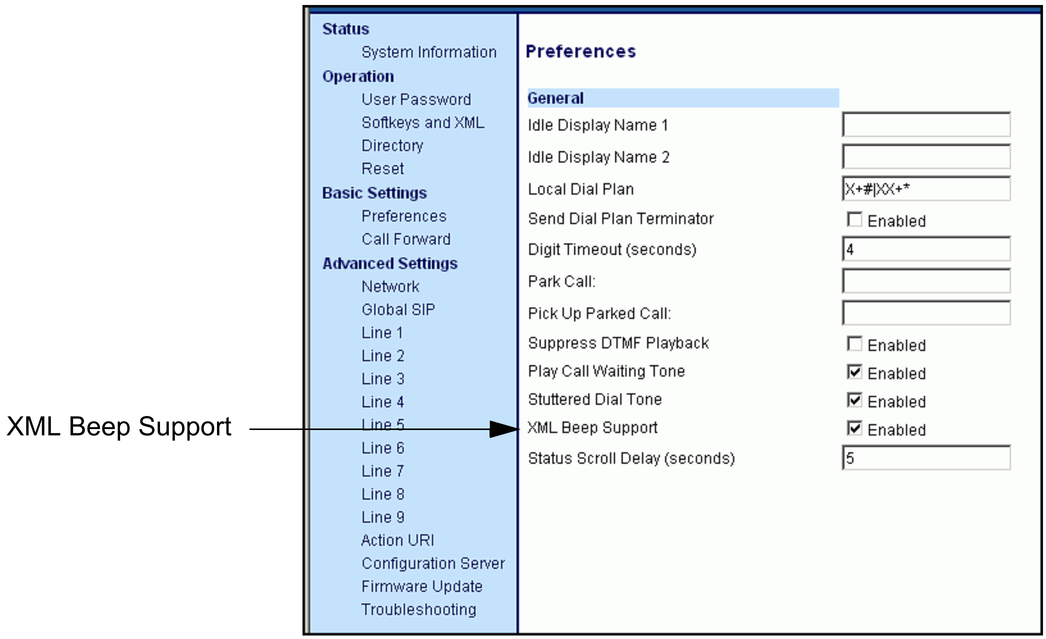
Configuring via the Configuration Files

You enable/disable the BEEP option using the following parameter in the configuration files:

Parameter – *xml beep notification* *XML Beep Support* (in Web UI)	**Aastra Web UI** Basic Settings->Preference**s** **Configuration Files** aastra.cfg, <mac>.cfg
Description	Enables or disables a BEEP notification on the phone when an AastraIPPhoneStatus object containing a “beep” attribute arrives to the phone.
Format	Boolean
Default Value	1 (ON)
Range	0 (OFF)No beep is audible even if the beep attribute is present in the XML object. 1 (ON)The phone beeps when an XML object with the “beep” attribute arrives to the phone.
Example	xml beep notification: 0

Configuring via the Aastra Web UI

You enable/disable the BEEP option in the Aastra Web UI at **Basic Settings->Preferences.**



Scroll Delay Option

The IP phones support a scroll delay option that allows you to set the time delay, in seconds, between the scrolling of each status message on the phone. The default time is 5 seconds for each message to display before scrolling to the next message. You can configure this option via the configuration files or the Aastra Web UI. Changes are dynamic and apply to the phone immediately.

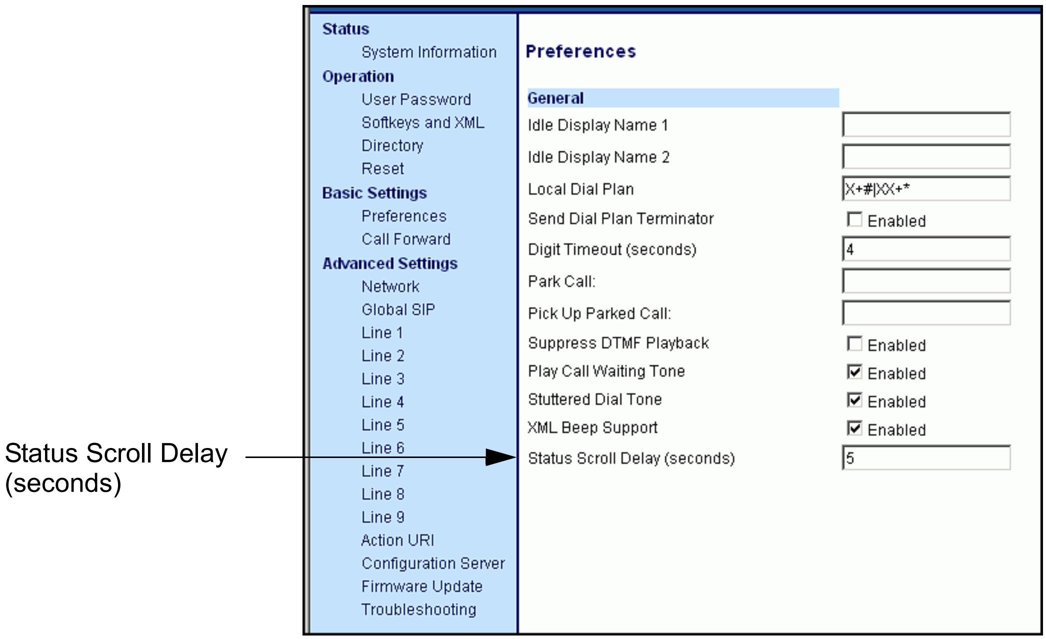
Configuring via the Configuration Files

You set a value for the scroll delay option using the following parameter in the configuration files.

Parameter – *xml status scroll delay* *Status Scroll Delay (seconds)* (in Web UI)	**Aastra Web UI** **Configuration Files**	Basic Settings->Preferences aastra.cfg, <mac>.cfg
Description	Specifies the length of time, in seconds, that each XML status message displays on the phone.	
Format	Integer	
Default Value	5	
Range	1 to 25	
Example	xml status scroll delay: 3	

Configuring via the Aastra Web UI

You set the scroll delay option in the Aastra Web UI at **Basic Settings->Preferences.**



XML: AastraIPPhoneExecute Object

A new XML AastraIPPhoneExecute object has been implemented on the IP phones. This object provides the ability to execute commands on the phone using XML. Release 1.4.1 supports the following Execute object commands:

- **Reset** - This command waits until the phone is idle and then executes a reset.
- **NoOp** - This command has no affect on the IP phone. It is made up of a blank URI. You can use this feature when you need to press a key on the phone to access a feature, and it is not necessary to display anything.

Since the server forces phone firmware changes, the AastraIPPhoneExecute object was implemented to send the reset command to the phone.

AastraIPPhoneExecute Object Structure

The AastraIPPhoneExecute object describes the structure of the XML document that is used to send a command to the phone. It delivers multiple execution requests to the phone. The basic structure of the AastraIPPhoneExecute object is:

```
<AastraIPPhoneExecute>
   <ExecuteItem URI ="the URL or URI to be executed"/>
<! -- Additional execution items may be added under new ExecuteItem tag-->
</AastraIPPhoneExecute>
```

Using the Reset Command

The <ExecuteItem URI =""/> tag can be entered with the command the phone should execute. Upon receiving an AastraIPPhoneExecute object, the phone begins executing the URL or URI specified.

The following example shows an AastraIPPhoneExecute object using the **Reset** command:

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

Note: If you specify a command as a URI attribute (instead of a URL), the keyword "**Command**" must be prepended in the value of the URI attribute so that the phone recognizes it as a URI attribute value. If you enter a URI and leave out the “**Command**” keyword, the phone interprets the value in the URI attribute as a URL containing network resources.

The following example shows the AastraIPPhoneExecute object using a URL:

```
<AastraIPPhoneExecute>
       <ExecuteItem URI="http://aastraserver/message.xml"/>
</AastraIPPhoneExecute>
```

When the phone receives this object, it displays the specified XML URI page.

Using the NoOp Command

You can use the AastraIPPhoneExecute object as an object to create a blank display (it has no affect on the IP phone). It is made up of a blank URI. You can use this feature when you need to press a key on the phone to access a feature, and it is not necessary to display anything.

The following example shows an AastraIPPhoneExecute object using a blank URI:

```
<AastraIPPhoneExecute>
       <ExecuteItem URI=””/>
<AastraIPPhoneExecute>
```

XML: Action URI

This feature provides the administrators the ability to specify a uniform resource identifier (URI) that triggers a GET when certain events occur. The IP phone events that support this feature are:

- Startup
- Successful registration
- Incoming call
- Outgoing call
- Offhook
- Onhook

The following table identifies the configurable action URI parameters in the configuration files and the Aastra Web UI. This table also identifies the variables that apply to specific parameters.

Configuration File Parameters	Aastra Web UI Parameters at Advanced Settings->Action URI	Applicable Variables
action uri startup	Startup	-
action uri registered	Successful Registration	$$SIPUSERNAME$$ $$SIPAUTHNAME$$ $$PROXYURL$$
action uri incoming	Incoming Call	$$REMOTENUMBER$$ $$DISPLAYNAME$$ $$SIPUSERNAME$$ $$INCOMINGNAME$$
action uri outgoing	Outgoing Call	$$REMOTENUMBER$$ $$SIPUSERNAME$$
action uri offhook	Offhook	-
action uri onhook	Onhook	-

How it works

When a startup, successful registration, incoming call, outgoing call, offhook, or onhook call event occurs on the phone, the phone checks to see if the event has an action URI configured. If the phones finds a URI configured, any variables configured (in the form $$VARIABLENAME$$) are replaced with the value of the appropriate variable. After all of the variables are bound, the phone executes a GET on the URI. The Action URI binds all variables and is not dependant on the state of the phone.

For example, if you enter the following string for the **action uri outgoing** parameter:

```
action uri outgoing: http://10.50.10.140/
outgoing.pl?number=$$REMOTENUMBER$$
```

and you dial out the number 5551212, the phone executes a GET on:

```
http://10.50.10.140/outgoing.pl?number=5551212
```

Note: If the phone can't find the Action URI you specify, it returns a "NULL" response. For example,
`http://10.50.10.140/outgoing.pl?number=`

You can configure this feature via the configuration files or the Aastra Web UI.

Configuring via the Configuration Files

You use the following parameters in the configuration files to configure the XML: action URI.

Parameter – *action uri startup* *Startup* (in Web UI)	**Aastra Web UI** **Configuration Files**	Advanced Settings->Action URI aastra.cfg, <mac>.cfg
Description	Specifies the URI for which the phone executes a GET on when a startup event occurs.	
Format	Fully qualified URI	
Default Value	Not Applicable	
Range	Up to 128 ASCII characters	
Example	action uri startup: http://10.50.10.140/startup	

Parameter – *action uri registered* *Successful Registration* (in Web UI)	**Aastra Web UI** **Configuration Files**	Advanced Settings->Action URI aastra.cfg, <mac>.cfg
Description	Specifies the URI for which the phone executes a GET on when a successful registration event occurs. This parameter can use the following variables: $$SIPUSERNAME$$ $$SIPAUTHNAME$$ $$PROXYURL$$	
Format	Fully qualified URI	
Default Value	Not Applicable	
Range	Up to 128 ASCII characters	
Example	action uri registered: http://10.50.10.14/registered.php?auth name=$$SIPAUTHNAME$$	

Parameter – *action uri incoming* *Incoming Call* (in Web UI)	**Aastra Web UI** **Configuration Files**	Advanced Settings->Action URI aastra.cfg, <mac>.cfg
Description	Specifies the URI for which the phone executes a GET on when an incoming call event occurs. This parameter can use the following variables: $$REMOTENUMBER$$ $$DISPLAYNAME$$ $$SIPUSERNAME$$ $$INCOMINGNAME$$	
Format	Fully qualified URI	
Default Value	Not Applicable	
Range	Up to 128 ASCII characters	
Example	action uri incoming: http://10.50.10.140/ incoming.php?number=$$REMOTENUMBER$$	

Parameter – *action uri outgoing* *Outgoing Call* (in Web UI)	**Aastra Web UI** **Configuration Files**	Advanced Settings->Action URI aastra.cfg, <mac>.cfg
Description	Specifies the URI for which the phone executes a GET on when an outgoing call event occurs. This parameter can use the following variables: $$REMOTENUMBER$$ $$SIPUSERNAME$$	
Format	Fully qualified URI	
Default Value	Not Applicable	
Range	Up to 128 ASCII characters	
Example	action uri outgoing: http://10.50.10.140/ outgoing.php?number=$$REMOTENUMBER$$	

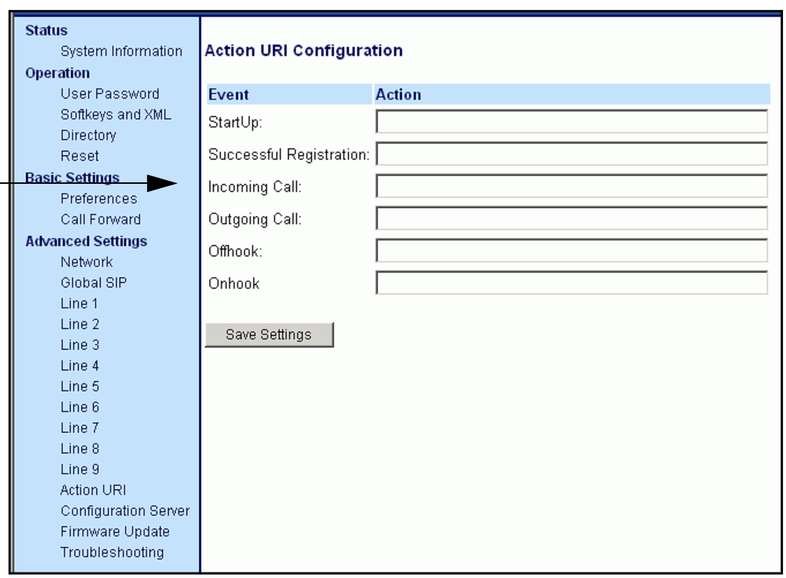
Parameter – *action uri offhook* *Offhook* (in Web UI)	**Aastra Web UI** **Configuration Files**	Advanced Settings->Action URI aastra.cfg, <mac>.cfg
Description	Specifies the URI for which the phone executes a GET on when an offhook event occurs. **Note:** The only supported use of the offhook action URI is to perform a GET that returns a NoOp Execute XML object. Although the phone correctly displays all other returned XML objects, the interaction with other phone features, such as speeddial, redial, etc., is undefined.	
Format	Fully qualified URI	
Default Value	Not Applicable	
Range	Up to 128 ASCII characters	
Example	action uri offhook: http://10.50.10.140/offhook	

Parameter – *action uri onhook* *Onhook* (in Web UI)	**Aastra Web UI** Advanced Settings->Action URI **Configuration Files** aastra.cfg, <mac>.cfg
Description	Specifies the URI for which the phone executes a GET on when an onhook event occurs.
Format	Fully qualified URI
Default Value	Not Applicable
Range	Up to 128 ASCII characters
Example	action uri onhook: http://10.50.10.140/onhook

Configuring via the Aastra Web UI

You can set the XML: action parameters in the Aastra Web UI at **Advanced Settings->Action URI.**

Action URI parameters



XML: Softkey URI

In addition to specifying variables for the Action URIs, you can also specify variables in the XML softkey URIs that are bound when the key is pressed. These variables are the same as those used in the Action URIs.

When an administrator enters an XML softkey URI either via the Aastra Web UI or the configuration files, they can specify the following variables:

- $$SIPUSERNAME$$
- $$SIPAUTHNAME$$
- $$PROXYURL$$
- $$REMOTENUMBER$$
- $$DISPLAYNAME$$
- $$INCOMINGNAME$$

When the softkey is pressed, if the phone finds a URI configured with variables (in the form $$VARIABLENAME$$), they are replaced with the value of the appropriate variable. After all of the variables are bound, the softkey executes a GET on the URI.

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 sip username of the specific line is 42512.

You can configure the XML softkey URI variables via the configuration files or the Aastra Web UI.

Configuring via Configuration Files

You use the following parameters to configure XML softkeys using variable binding:

- softkeyN type
- softkeyN label
- softkeyN value
- prgkeyN type
- prgkeyNvalue

For example, on a 480i/480i CT:

softkey1 type: xml
softkey1 label: JohnSmith
softkey1 value: **`http://10.50.10.140/script.pl?name=$$SIPUSERNAME$$`**

On a 9112i/9133i:

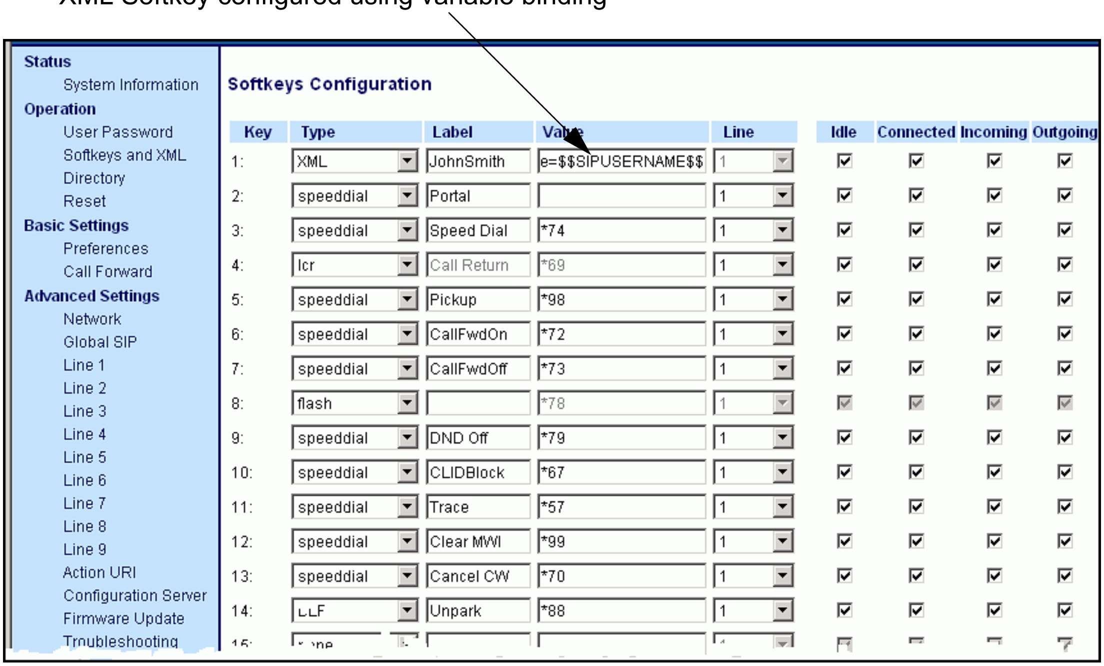
prgkey1 type: xml
prgkey1 value: **`http://10.50.10.140/script.pl?name=$$SIPUSERNAME$$`**

Configuring via the Aastra Web UI

For 480i/480i CT, you configure XML softkey variable binding at **Operation->Softkeys and XML**.

For 9112i/9133i, you configure XML softkey variable binding at **Operation->Programmable Keys.**

XML Softkey configured using variable binding



XML: HTTP Refresh Header

A new HTTP refresh header feature has been implemented on the IP phones. This feature includes the following:

- All current XML screen objects have the ability to be refreshed by adding a **Refresh** and **URL** setting to the HTTP headers. (see Refresh setting format below)
- The Refresh setting is set by the XML application and it is up to the application to decide which objects it wants to refresh.

Note: This HTTP refresh header feature only applies to objects that display to the screen.

The Refresh setting must be included in the HTTP header. The XML application decides which objects it wants to use with this setting. The phone recognizes this setting when parsing the HTTP header. If the setting is present, then it passes along the refresh timeout and the URL to the ParserData object, which all XML screen objects inherit from. The ParserData class also has a timer, which must be set to expire at the next refresh time. When the timer expires (time to refresh the screen), the phone requests the URL again and displays the refreshed screen.

Refresh Setting Format

The following is the Refresh setting format for the HTTP header:

```
Refresh: <timeout>; URL=<page to load>
```

The following example is a Refresh setting for use in an HTTP header:

```
Refresh: 3; URL=http://10.50.10.140/cgi-bin/update.xml
```

Note: You must use the **Refresh** and **URL** parameters in order for this feature to work in the HTTP header.

Backup Proxy/Registrar Support

The IP phones now support a backup SIP proxy and backup SIP registrar feature. If the primary server is unavailable, the phone automatically switches to the backup server allowing the user's phone to remain in service.

How it Works

All SIP registration messages are sent to the primary registrar first. If the server is unavailable, then a new registration request is sent to the backup registrar. This also applies to registration renewal messages, which try the primary server before the backup.

Similarly, any outgoing calls attempt to use the primary proxy first, then the backup if necessary. In addition, subscriptions for BLF, BLA, and explicit MWI can also use the backup proxy when the primary fails. Outgoing calls and the previously mentioned subscriptions behave the same as registrations, where the primary proxy is tried before the backup.

You can configure the backup SIP proxy on a global or per-line basis via the configuration files or the Aastra Web UI.

Configuring via the Configuration Files

You can configure the following parameters for a backup SIP proxy and backup SIP registrar.

Global parameters

Parameter – *sip backup proxy ip* *Backup Proxy Server* (in Web UI)	**Aastra Web UI** **Configuration Files**	Advanced Settings->Global SIP-> Basic SIP Network Settings aastra.cfg, <mac>.cfg
Description	The IP address of the backup SIP proxy server for which the IP phone uses when the primary SIP proxy is unavailable.	
Format	IP address or fully qualified Domain Name	
Default Value	0.0.0.0	
Range	Not Applicable	
Example	sip backup proxy ip: 192.168.0.102	

Parameter – *sip backup proxy port* *Backup Proxy Port* (in Web UI)	**Aastra Web UI** Advanced Settings->Global SIP-> Basic SIP Network Settings **Configuration Files** aastra.cfg, <mac>.cfg
Description	The backup proxy's port number.
Format	Integer
Default Value	0
Range	Not Applicable
Example	sip backup proxy port: 5060

Parameter – *sip backup registrar ip* *Backup Registrar Server* (in Web UI)	**Aastra Web UI** Advanced Settings->Global SIP-> Basic SIP Network Settings **Configuration Files** aastra.cfg, <mac>.cfg
Description	The address of the backup registrar (typically, the backup SIP proxy) for which the IP phone uses to send *REGISTER* requests if the primary registrar is unavailable. A global value of 0.0.0.0 disables backup registration. However, the phone is still active and you can dial using username@ip address of the phone. If the backup registrar IP address is set to 0.0.0.0 for a per-line basis (i.e, line 1, line 2, etc.), then the backup register request is not sent, the "No Service" message does not display, and the message waiting indicator (MWI) does not come on.
Format	IP address or fully qualified Domain Name
Default Value	0.0.0.0
Range	Not Applicable
Example	sip backup registrar ip: 192.168.0.102

Parameter – *sip backup registrar port* *Backup Registrar Port* (in Web UI)	**Aastra Web UI** **Configuration Files**	Advanced Settings->Global SIP-> Basic SIP Network Settings aastra.cfg, <mac>.cfg
Description	The backup registrar's (typically the backup SIP proxy) port number.	
Format	Integer	
Default Value	0	
Range	Not Applicable	
Example	sip backup registrar port: 5060	

Per-line parameters

Parameter – *sip linex backup proxy ip* *Backup Proxy Server* (in Web UI)	**Aastra Web UI** **Configuration Files**	Advanced Settings->Line*N*-> Basic SIP Network Settings aastra.cfg, <mac>.cfg
Description	The IP address of the backup SIP proxy server for which the IP phone uses when the primary SIP proxy is unavailable.	
Format	IP address or fully qualified Domain Name	
Default Value	0.0.0.0	
Range	Not Applicable	
Example	sip line1 backup proxy ip: 192.168.0.102	

Parameter – *sip linex backup proxy port* *Backup Proxy Port* (in Web UI)	**Aastra Web UI** **Configuration Files**	Advanced Settings->Line*N*-> Basic SIP Network Settings aastra.cfg, <mac>.cfg
Description	The backup proxy's port number.	
Format	Integer	
Default Value	0	
Range	Not Applicable	
Example	sip line1 backup proxy port: 5060	

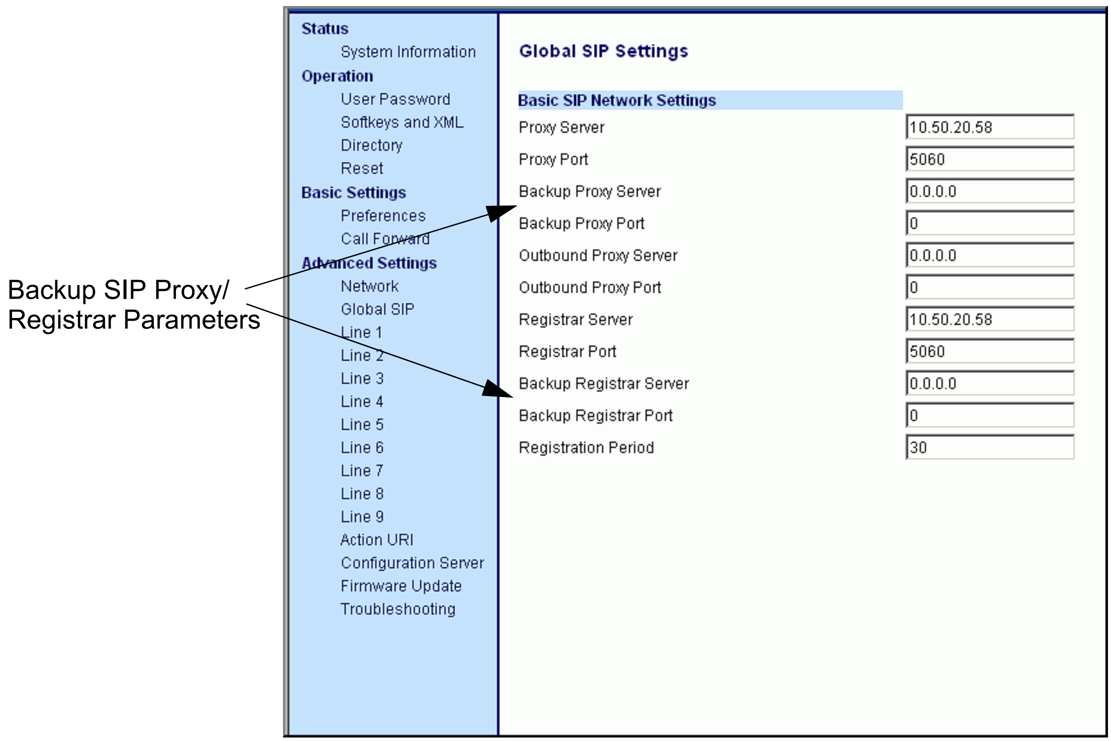
Parameter – *sip linex backup registrar ip* *Backup Registrar Server* (in Web UI)	**Aastra Web UI** **Configuration Files**	Advanced Settings->Line*N*-> Basic SIP Network Settings aastra.cfg, <mac>.cfg
Description	The address of the backup registrar (typically, the backup SIP proxy) for which the IP phone uses to send *REGISTER* requests if the primary registrar is unavailable. A global value of 0.0.0.0 disables backup registration. However, the phone is still active and you can dial using username@ip address of the phone. If the backup registrar IP address is set to 0.0.0.0 for a per-line basis (i.e, line 1, line 2, etc.), then the backup register request is not sent, the "No Service" message does not display, and the message waiting indicator (MWI) does not come on.	
Format	IP address or fully qualified Domain Name	
Default Value	0.0.0.0	
Range	Not Applicable	
Example	sip line1 backup registrar ip: 192.168.0.102	

Parameter – *sip linex backup registrar port* *Backup Registrar Port* (in Web UI)	**Aastra Web UI** **Configuration Files**	Advanced Settings->Line*N*-> Basic SIP Network Settings aastra.cfg, <mac>.cfg
Description	The backup registrar's (typically the backup SIP proxy) port number.	
Format	Integer	
Default Value	0	
Range	Not Applicable	
Example	sip line1 backup registrar port: 5060	

Configuring via the Aastra Web UI

For global configuration, you can set the backup SIP proxy/registrar parameters at **Advanced Settings->Global SIP->Basic SIP Network Settings.**

For per-line configuration, you can set the backup SIP proxy/registrar parameters at **Advanced Settings->Line*N*->Basic SIP Network Settings.**



Auto-discovery Using mDNS

Release 1.4.1 introduces a process that allows the phones to auto-discover all servers on a network using mDNS. When the IP phone discovers a TFTP server, it is automatically configured by that TFTP server.

An unconfigured phone (phone right out of the box) added to a network, attempts to auto-discover a configuration server on the network without any end-user intervention. When it receives DHCP option 66 (TFTP server), it automatically gets configured by the TFTP server.

An already configured phone (either previously configured by auto-discovery or manually configured) added to a network, uses its predefined configuration to boot up.

Notes:

1. Configuration parameters received via DHCP do not constitute configuration information, with the exception of a TFTP server. Therefore, you can plug a phone into a DHCP environment, still use the auto-discovery process, and still allow the use of the TFTP server parameter to set the configuration server.

2. DHCP option 66 (TFTP server details) overrides the mDNS phase of the auto-discovery. Therefore, the DHCP option takes priority and the remaining process of auto-discovery continues.

3. As the phone performs auto-discovery, all servers in the network (including the TFTP server), display in the phone window. However, only the server configured for TFTP automatically configures the phone.

IP Phone Features for Sylantro Servers

Last Call Return (lcr) Support

A new feature has been added to the IP phones that allow a user or administrator to configure a "last call return" function on a softkey or programmable key. This feature is for Sylantro servers only.

You can configure the “lcr” softkey feature via the configuration files and the Aastra Web UI.

How it works

If you configure "lcr" on a softkey or programmable key, and a call comes into your phone, after you are finished with the call and hang up, you can press the key configured for “**lcr**” and the phone dials the last call you received. When you configure an “lcr” softkey, the label “LCR” displays next to that softkey on the IP phone. When the Sylantro server detects an “lcr” request, it translates this request and routes the call to the last caller.

For the 480i/480i CT, applicable parameters to configure for “lcr” are:

- **softkeyN type** (“Type” in Web UI)
- **softkeyN line** (“Line” in Web UI)
- **softkeyN states** (“Idle, Connected, Incoming, Outgoing” in Web UI)

For the 9112i/9133i, applicable parameters to configure for “lcr” are:

- **prgkeyN type** (“Type” in Web UI)
- **prgkeyN line** (“Line” in Web UI)

Note: For more information about “lcr” see the *SIP IP Phone Administrator Guide*, *Release 1.4.1.*

Support for additional “Alert Info” keywords for distinctive ringing

New configurable, distinctive ringing support has been added to release 1.4.1 for Sylantro servers. The following “info” parameters allow you to configure specific priority alert tones for each parameter that may appear as keywords in the “Alert-Info” header of a Sylantro server:

- **alert-acd (auto call distribution)**
- **alert-community-1**
- **alert-community-2**
- **alert-community-3**
- **alert-community-4**

You can configure these priority alert parameters via the configuration files or the Aastra Web UI.

Configuring via the Configuration Files

In addition to the priority alert parameters that already exist (from previous releases), you can also now configure the following parameters for priority alerting.

Parameter – *alert auto call distribution* auto call distribution (in Web UI)	**Aastra Web UI** **Configuration Files**	Basic Settings->Preferences-> Priority Alerting Settings aastra.cfg, <mac>.cfg
Description	When an "alert-acd" keyword appears in the header of the INVITE request, the configured Bellcore ring tone is applied to the IP phone.	
Format	Integer	
Default Value	0 Normal ringing	
Range	0 Normal ringing (default) 1 Bellcore-dr2 2 Bellcore-dr3 3 Bellcore-dr4 4 Bellcore-dr5 5 Silent	
Example	alert auto call distribution: 2	

Parameter – *alert community 1* community-1 (in Web UI)	**Aastra Web UI** **Configuration Files**	Basic Settings->Preferences-> Priority Alerting Settings aastra.cfg, <mac>.cfg
Description	When an "alert community-1" keyword appears in the header of the INVITE request, the configured Bellcore ring tone is applied to the IP phone.	
Format	Integer	
Default Value	0 Normal ringing	
Range	0 Normal ringing (default) 1 Bellcore-dr2 2 Bellcore-dr3 3 Bellcore-dr4 4 Bellcore-dr5 5 Silent	
Example	alert community 1: 3	

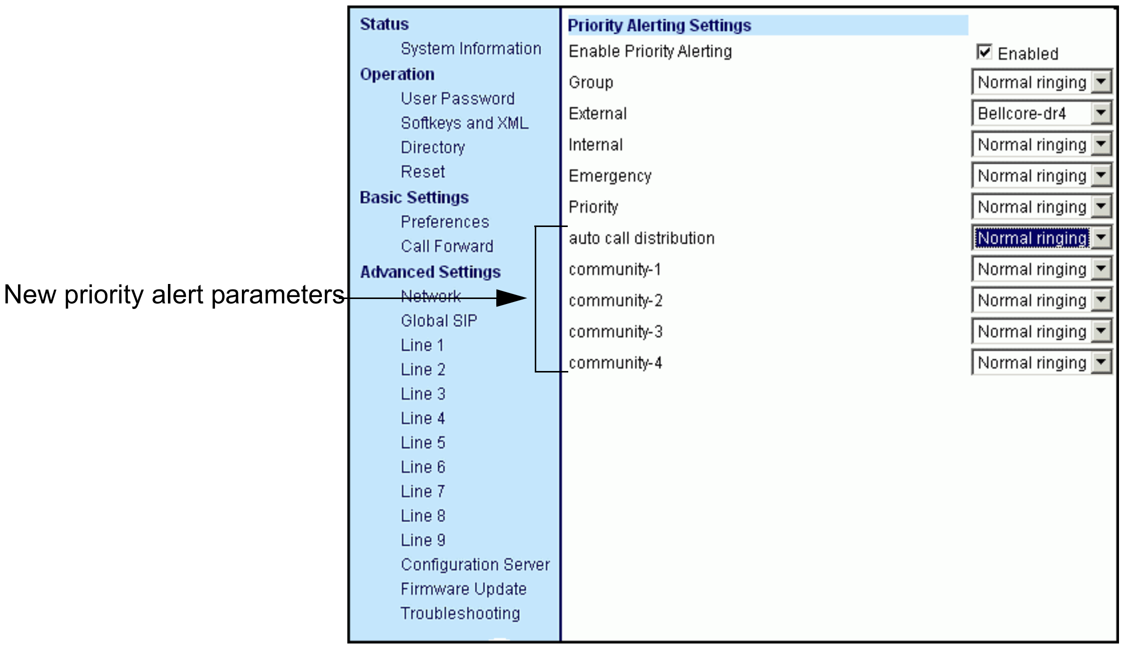
Parameter – *alert community 2* community-2 (in Web UI)	**Aastra Web UI** **Configuration Files**	Basic Settings->Preferences-> Priority Alerting Settings aastra.cfg, <mac>.cfg
Description	When an "alert community-2" keyword appears in the header of the INVITE request, the configured Bellcore ring tone is applied to the IP phone.	
Format	Integer	
Default Value	0 Normal ringing	
Range	0 Normal ringing (default) 1 Bellcore-dr2 2 Bellcore-dr3 3 Bellcore-dr4 4 Bellcore-dr5 5 Silent	
Example	alert community 2: 4	

Parameter – *alert community 3* community-3 (in Web UI)	**Aastra Web UI:** Basic Settings->Preferences->Priority Alerting Settings **Configuration Files**: aastra.cfg, <mac>.cfg
Description	When an "alert community-3" keyword appears in the header of the INVITE request, the configured Bellcore ring tone is applied to the IP phone.
Format	Integer
Default Value	0 Normal ringing
Range	0 Normal ringing (default) 1 Bellcore-dr2 2 Bellcore-dr3 3 Bellcore-dr4 4 Bellcore-dr5 5 Silent
Example	alert community 3: 1

Parameter – *alert community 4* community-4 (in Web UI)	**Aastra Web UI:** Basic Settings->Preferences->Priority Alerting Settings **Configuration Files**: aastra.cfg, <mac>.cfg
Description	When an "alert community-4" keyword appears in the header of the INVITE request, the configured Bellcore ring tone is applied to the IP phone.
Format	Integer
Default Value	0 Normal ringing
Range	0 Normal ringing (default) 1 Bellcore-dr2 2 Bellcore-dr3 3 Bellcore-dr4 4 Bellcore-dr5 5 Silent
Example	alert community 4: 2

Configuring via the Aastra Web UI

You can configure the new parameters for priority alerting in the Aastra Web UI at **Advanced Settings->Action URI.**



Startup Enhancement

During startup, the maximum time the phone spends attempting to contact the configuration server (TFTP, FTP or HTTP) has been reduced from 10 minutes to approximately 30 seconds.

When the phone boots up, it attempts to contact the configuration server (if the configuration server parameters are configured on the phone), for 4 seconds. The phone then displays a "**Skip**" button. After an additional 30 seconds has expired, or if the user presses the "**Skip**" button, the phone continues the boot process using the stored configuration.

At any point during this period, if the phone successfully contacts the configuration server, it automatically continues and attempts to download its configuration and any new firmware from that server.

480i CT Only

Single Call Restriction (480i CT only)

A new feature has been implemented on the 480i CT that allows an administrator to enable or disable a single call restriction between the 480i CT base unit and a call server.

When this feature is enabled (set to 1), you can make separate active calls from the 480i CT base unit and from the cordless handset. If this feature is disabled (set to 0), only one call can be active at a time either from the base unit or from the handset. When this feature is disabled, and you make an active call on either the base unit or the handset, any other attempt to make an active call is put on hold. Also, when this feature is disabled, more than one call can negotiate complex audio codecs since only a single call is decoding audio at a time.

You can configure this feature via the configuration files or the Aastra Web UI.

Configuring via the Configuration Files

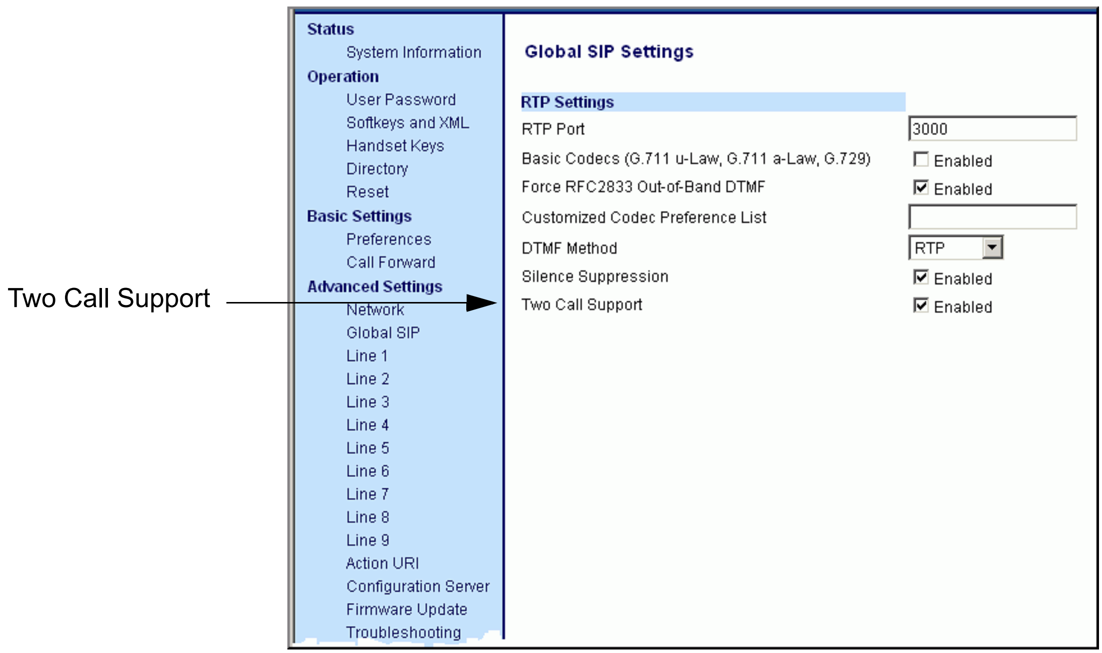
You can configure the following parameter for the 480i CT single media path restriction.

Parameter – *two call support* *Two Call Support* (in Web UI)	**Aastra Web UI** **Configuration Files**	Advanced Settings->Global SIP->RTP Settings aastra.cfg, <mac>.cfg
Description	Enables or disables the single media path restriction between the 480i CT base unit and the handset. When this feature is enabled (set to 1), you can make separate active calls from the 480i CT base unit and from the cordless handset. If this feature is disabled (set to 0), only one call can be active at a time either from the base unit or from the handset. When this feature is disabled, and you make an active call on either the base unit or the handset, any other attempt to make an active call is put on hold. Also, when this feature is disabled, more than one call can negotiate complex audio codecs since only a single call is decoding audio at a time.	
Format	Boolean	

Default Value	1
Range	0 - Disable 1 - Enable
Example	two call support: 0

Configuring via the Aastra Web UI

You can enable or disable the 480i CT single media path restriction at **Advanced Settings->Global SIP->RTP Settings.**



9112i/9133i Only

Addition of DisplayName1 & DisplayName2 (now also applies to 9112i/9133i)

The 9112i/9133i IP phones now support the use of **displayName1** and **displayName2**. The value in these fields display on the idle screen of the IP phone. Previous IP phone releases supported these parameters on the 480i/480i CT only.

You can configure these parameters via the configuration files or the Aastra Web UI.

Configuring via the Configuration Files

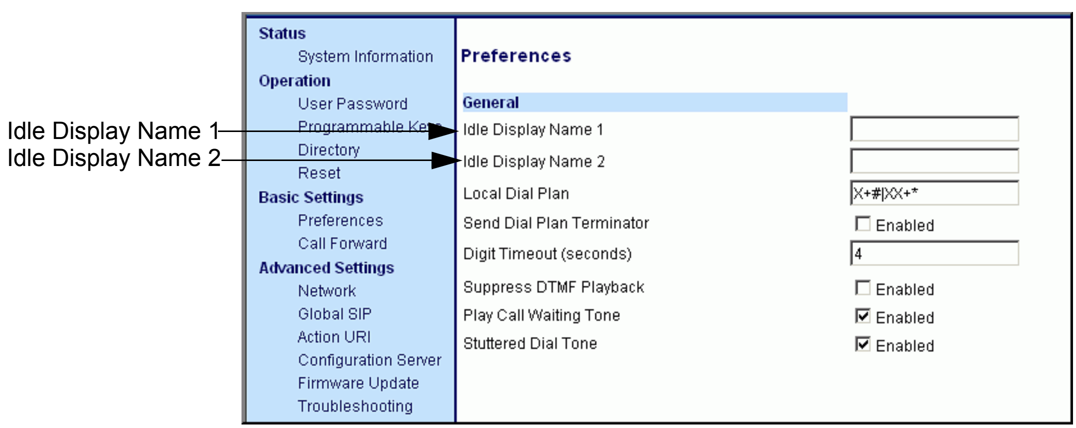
You use the following parameters in the configuration files to configure displayName1 & displayName2.)

Parameter – *displayName1* Idle Display Name 1 (in Web UI)	**Aastra Web UI** Basic Settings->Preferences **Configuration Files** aastra.cfg, <mac>.cfg
Description	The name displayed on the idle screen rather than the screen name and phone number
Format	Alphanumeric characters
Default Value	Not Applicable
Range	**For 480i/480i CT**: Up to 21 characters (width of LCD) **For 9112i/9133i**: Up to 16 characters (width of LCD)
Example	displayName1: SIPphone1

Parameter – *displayName2* Idle Display Name 2 (in Web UI)	**Aastra Web UI** Configuration Files	Basic Settings->Preferences aastra.cfg, <mac>.cfg
Description	The name displayed on the idle screen rather than the screen name and phone number	
Format	Alphanumeric characters	
Default Value	Not Applicable	
Range	**For 480i/480i CT**: Up to 21 characters (width of LCD) **For 9112i/9133i**: Up to 16 characters (width of LCD)	
Example	displayName2: SIPphone2	

Configuring via the Aastra Web UI

On the 9112i/9133i, you can configure Idle Display Name 1& Idle Display Name 2 in the Aastra Web UI at **Basic Settings->Preferences->General**.



Issues Resolved in Release 1.4.1

Description

This section describes the issues resolved in release 1.4.1. The following table provides the issue number and a brief description of each fix.

Note: Unless specifically indicated, these resolved issues apply to all phone models.

Issue Number	Description of Fix
CLN04606	With the 480i, the phone no longer hangs on startup when global call park and call pickup keys are configured.
CLN04933	The IP phones no longer crash during transfers using BroadWorks release 1.2.
DEF04090	Phone no longer adds extra "/" in XML requests.
DEF04096	The cordless handsets now support sending DTMF events as INFO packets.
DEF04174	The message "No Service" no longer appears momentarily on the 480i during registration. A parameter "sip registration renewal timer" has been added that allows you to set the time, in seconds, prior to the expiration of when the registration is renewed.
DEF04175	Ringing is no longer interrupted by the BLF ringing alert tone.
DEF04178	Added support for URI dialing to the 9133i programmable keys.
DEF04188	The Calls between the cordless handset and the base now get a voice path when the call is not made using the intercom function.
DEF04210	When the media port is equal to 0 in the INVITE, the phone no longer sends "486 Busy Here" after sending "180 Ringing when using a MetaSwtich.
DEF04234	The IP phones no longer reject calls when receiving an attribute in SDP that it didn't recognize. It now correctly ignores those attributes. This fixes an issue with the Eyebeam soft client.
DEF04302	You can now toggle to handsfree while on hold.
DEF04450	When using the Aastra Web UI, the dial plan no longer truncates at 85 characters.
DEF04545	The hangup key now returns dialtone when the handset stays off hook. With the 480i CT, the Goodbye key now acts consistently across all phones.

Issue Number	Description of Fix
DEF04757	The call transfer (Xfer) function was not working with Broadsoft SCA using multiple headers. The call transfer now works on an SCA line with Broadsoft R13.
DEF05117	If the CODEC negotiation fails during the processing of a re-INVITE, the phone no longer drops the call, but instead responds with a “488 Not Acceptable Here” message.

Enhancements/Changes

This section describes the enhancements and changes made to the 1.4.1 release..

Note: Unless specifically indicated, these enhancements apply to all phone models.

Issue Number	Description of Fix
DEF04638	XML URIs now support adding a port number to the URI.
ENH04358	Added ability to change debug level from Aastra Web UI. You can now configure a “log level” in the Troubleshooting section of the Web interface.
ENH04371	Web recovery now resets the administrator password and removes local configuration files. You force the phone into recovery mode by holding ‘#’ and ‘1’ on startup.

Known Anomalies in 1.4.1

Description

This section describes the known anomalies in release 1.4.1.

Note: Unless specifically indicated, these known anomalies apply to all phone models.

Issue Number	Description
DEF04448	SIP contact URI is not used in subsequent REFER-TO address.
DEF04535	The TO field displays your own name during an originated call.
DEF04899	In XML APIs, password is not hidden when it is marked as not editable.
DEF04912	Sending a lot of messages (hundreds) within a second, causes the phone to lock up.
DEF05019	The previous IP Phone release 1.4 shows the softkeys for the XML AastraIPPhoneInputScreen object as: • 1 = Backspace • 2 = Dot • 3 = ChangeCase • 4 = Numeric/Alpha • 5 = Cancel • 6 = Done This is incorrect. For softkeys 5 and 6, the key assignment is: • 5 = Done • 6 = Cancel
DEF05056	An incoming intercom call changes the current audio mode (speaker/headset) of existing calls.
ENH05218	When dialing a number on the phone and another call comes in, the phone rings and the digits you dialed are lost.

Contacting Aastra Telecom Support

If you've read this release note, and consulted the Troubleshooting section of your phone model's manual and still have problems, please send inquiries via email to support@aastra.com.

Generic SIP IP Phone
Models 4801, 480i CT, 9112i, 9133i

1.4.1 Release Notes

41-001024-00 Rev 09
SIP IP Phones Release 1.4.1
November 2006