Digium Phone Module for Asterisk Users Guide v1.0.0

Overview

There are myriad VoIP phones on the market that people use with Asterisk. None of them are built specifically for Asterisk, so they don't take any particular advantage of Asterisk's applications. And, each one of them is too difficult to use.

Digium has designed phones to provide a superior user experience for everyone; from the person making the calls, to the person setting up the phone. Phones aren't hard, like everyone thinks that are, and they're not dumb terminals; instead, they're simple and smart.

A Digium phone can communicate with Asterisk, or with any other SIP-based system. In this respect, a Digium phone is somewhat like other SIP phones. In order to provide more than just the capabilities of a regular SIP phone, Digium makes available the Digium Phone Module for Asterisk (DPMA). The DPMA is a binary Asterisk module that provides a secure communications channel for Digium phones and Asterisk. This secure channel provided by the DPMA is used by Asterisk to provision and manage Digium phones, and by the phones to directly access Asterisk's internals - leading to a richer set of phone applications, and happier users.

DPMA: What is it?

The DPMA is a binary Asterisk module that provides a secure communications channel between Digium phones and Asterisk. This secure channel is used to provision and manage the phones and to provide direct access to Asterisk's internal applications. The DPMA is not required to use Digium phones, but offers a number of significant advantages. A single, free license is required for each Asterisk system running the DPMA. For more information on installation and configuration, please see the support pages for Digium phones.

DPMA Compatibility

The DPMA is compatible with the open source Certified Asterisk releases of Asterisk 1.8, beginning with the asterisk-1.8.11-cert1 release. A

DPMA module for the open source "-digiumphones" branch of Asterisk version 10 will be made available at a later date. The DPMA is not compatibile with any other open source version of Asterisk.

Certified Asterisk and DigiumPhones Branches

The DPMA is not compatible with mainline releases of Asterisk because of the release policies of open source Asterisk. Per release policy, once a branch of Asterisk, e.g. 1.4, 1.6.2, 1.8, 10, etc. is created, bugs are fixed, security vulnerabilities are closed, but new features are not added. Over time, this has proven to be an effective policy at limiting the introduction of regressions and making upgrades between branch versions an easy process.

Because the Digium Phone module for Asterisk requires new features - APIs, SIP messaging infrastructure, voicemail changes, etc. - that are not currently available in a mainline version of Asterisk, and because Digium phones will require new changes in the future as additional phone applications are provided, a new branch was required. For Asterisk 1.8 users, all of the code necessary to support the DPMA, as well as changes to Asterisk applications, such as voicemail, parking, user presence, etc. is available in the Certified Asterisk releases, beginning with the asterisk-1.8.11-cert1 release. For more information about Certified Asterisk, please see the overview on the Digium website. Support for Asterisk 10 users will be provided in a 10-digiumphones branch of Asterisk that will track the mainline branch of Asterisk with respect to features and bugfixes, but also support for the DPMA. As with any other branch of Asterisk, both Certified Asterisk and -digiumphones branches are licensed under the GPLv2 and are made available for download via subversion, as tarballs, and as packages.

Provisioning

Digium Phones may be assigned SIP account configuration a number of different ways:

1. Option 66 DHCP directive
2. The phone’s boot menu
    a. Digium Configuration Server (DPMA, manual address)
    b. Fetch Configuration file from URL
        i. HTTP, HTTPs
    c. Manual SIP account entry
3. The phone’s web-based configuration tool
4. DPMA, Bonjour / mDNS

When the phone is configured via Option 66 DHCP directive or it's told to manually fetch its configuration from a URL, the phone will first request a configuration file matching the name <MAC>.cfg, e.g. 0019159bd025.cfg; and, if that file is not found, will then go on to request a configuration file named 000000000000.cfg.

When the phone is configured to talk to a Digium Configuration server, either manually by plugging in the address of the server or automatically through mDNS discovery, the phone will communicate with and retrieve its configuration directly from Asterisk via the DPMA.

DPMA Concepts

DPMA uses two internal concepts:

1. Phones
2. Lines

A **Phone** is a set of configuration parameters that represent the device on your desk. A Phone is abstracted because in most cases, it represents an individual user’s configuration, e.g. Bob’s Phone. Bob’s Phone has line keys that are specific to Bob, it has a contacts list that’s specific to Bob, it has Rapid Dial keys that are specific to Bob.

A **Line** is a key on phone that is mapped to an Asterisk SIP identity ("friend," technically, since that's a better way to go if you want to assign more than one SIP identity to the same physical device) and labeled in a particular way. Because, in Asterisk, there’s a one-to-one relationship between physical device and SIP identity, a Line also maps one-to-one to a SIP identity, since it will only appear on one physical device at a time.

DPMA Configuration

Phones and Lines are configured in the res_digium_phone.conf configuration file – normally located at /etc/asterisk/res_digium_phone.conf. The file contains one reserved section:

- [general]

The **[general]** section contains settings that are specific to the operation of the DPMA itself.

Two other section types are available for user configuration, each contains a **type** definition. The type definition determines the function of the section. The three **types** are:

- phone
- line

- external_line

phone sections contain all settings for a phone profile.
line sections contain all settings for a **line** to be applied to a **phone**
external_line sections contain all settings for registration to an external SIP server, e.g. another Asterisk machine

The phone and lines section of res_digium_phone.conf, are defined by a section identifier encapsulated in square brackets, and are user-definable.

Example

This is an example, using invalid options and functions, of a res_digium_phone.conf configuration file, displaying the organizational layout. The various options and functions are described later in this page.

Example res_digium_phone.conf layout.

```
[general]
; comments are preceded by a comma
;
; the general section contains many options
generaloption=value
generaloption2=othervalue
;
[MyPhone] ; A phone
type=phone
; Phone options go here
phoneoption=value
phoneoption2=othervalue
;
[MyAccount] ; A SIP peer from sip.conf
type=line
; Line options go here
lineoption=value
lineoption2=othervalue
;
[MyExternalAccount] ; An external SIP registration
type=external_line
; External Line options go here
externallineoption=value
externallineoption2=othervalue
;
```

Authentication Options

The General Section provides the following options related to authentication:

Option	Values	Description
globalpin	integer; e.g. 10101019	If userlist_auth is set to globalpin, sets the PIN that must be entered on a phone, at boot, in order to retrieve the list of available phone configurations. If config_auth is set to globalpin, sets the PIN that must be entered on a phone, at the userlist screen, in order to request a particular phone configuration - note that if the globalpin has already been entered to authenticate to retrieve the list of available phone configurations, that it will not required a second time in order to request a particular phone configuration.
userlist_auth	disabled, globalpin	Specifies the kind of authentication required to retrieve the list of available phone profiles from the provisioning server. Options are "disabled," where no authentication is required to retrieve the list of phone profiles; and "globalpin," where entry of the Global PIN is required in order to retrieve the list of phone profiles.
config_auth	mac, pin, globalpin, disabled	Specifies the kind of authentication required to retrieve a phone's configuration from the provisioning server. Options are "mac," where the MAC address of the device requesting the configuration must match the phone profile, and if it does, the phone will be automatically provisioned with the matching phone profile; "pin" where the entered PIN must match the phone profile; "globalpin," where the entered PIN must match the Global PIN; and disabled, where profiles are served without authentication - with this setting, any phone can pull any phone profile defined in this configuration file with no authentication challenges.

Registration Server Options

The General Section provides the following options related to the registration server:

Option	Values	Description
registration_address	IP address or Hostname	Allows for explicit definition of the address to which phones should register. If the option is not set, then the address given to the phones to register is the same address that the phone uses in the To: header when requesting a configuration
registration_port	Port, as an Integer; e.g 5060	The port on which the registration server is running – the same port on which SIP is running on the Asterisk instance.

mDNS Discovery Options

The General Section provides the following options related to mDNS / Avahi Service Discovery:

Option	Values	Description
service_name	string, e.g. Digium Phones Config Server	The name of the service to be used in Avahi service discovery. Defaults to “Digium Phones Config Server” when service_discovery_enabled, registration_address and registration_port are set.
service_discovery_enabled	yes / no	If set to yes, will advertise the config server using Avahi. Defaults to yes. The Discovery Service will only be enabled when registration_address and registration_port are explicitly configured.

Firmware Update Options

The General Section provides the following options related to Firmware Update:

Option	Values	Description
firmware_url_prefix	URL, e.g. http://10.10.10.10/firmware_package_directory	Specifies the URL prefix the phone module should use to tell the phones where to retrieve firmware files.
firmware_package_directory	file path, e.g. /var/www/firmware_package_directory	Specifies the actual disk location of the firmware package provided by the webserver

Files Options

The General Section provides the following options related to Files:

Option	Values	Description
file_directory	path, e.g. /var/lib/asterisk/digium_phones	Specifies a folder from which phones can retrieve txt and binary files. This directory is primarily used for storing contacts.xml and user-generated phone configuration. Defaults to /var/lib/asterisk/digium_phones

Other General Options

The General Section provides the following other options:

message_context	string, e.g. dpma_message_context	The context matching the sip.conf [general] section definition for outofcall_message_context, defaults to dpma_message_context.

Example

In this example:

- The general section has been configured with a Global PIN of 344486 (DIGIUM)
- Userlist authentication has been disabled
- Config Authorization has been set to MAC with Global PIN
- The Registration Server is set to 10.10.10.10 on port 5060
- The Avahi service name has been set to “Go 4 Phones” with discovery enabled
- Firmware will be retrieved from http://10.10.10.10/digium_phones_firmware **which is actually located at /var/www/digium_phones_firmware
- Files are stored in /var/lib/asterisk/digium_phones
- and the dialplan context used for message handing is dpma_message_context

Example General Section configuration.

```
[general]
globalpin=344486
userlist_auth=disabled
config_auth=mac_globalpin
registration_address=10.10.10.10
registration_port=5060
service_name=Go 4 Phones
service_discovery_enabled=yes
firmware_url_prefix=http://10.10.10.10/digium_phones_firmware
firmware_package_directory=/var/www/digium_phones_firmware
file_directory=/var/lib/asterisk/digium_phones
message_context=dpma_message_context
```

Phone Configuration Options

Phones profiles are configured by defining a context with type option equal to “phone.” A Phone profile can have any number of lines associated with it. Each line defined in the configuration is reflected as a separate line key on the phone; and, when provisioned, is ordered on the phone itself as it is in the profile configuration.

The first line key associated with a Phone profile is set internally as the phone’s primary line. The primary line is the only one listed in the User list on boot up. The primary line is also used to automatically match the phone to voicemail boxes.
The following options are provided for Phone configuration:

Option	Values	Description
mac	A MAC address, e.g. 0123456789ab	Optional. When set, and when the general config_auth option requires MAC, locks a phone configuration to a device matching this MAC address.
pin	integer; e.g. 10101019	Optional. When set, and when the general config_auth option requires PIN, one must enter this PIN on the phone before being able to pull the phone's configuration
line	entity defined as line in res_digium_phone.conf	Maps directly to a sip.conf peer/friend entry. More than one line may be defined for a phone configuration. The first line entry defined is adopted as the phone's primary line.
external_line	entity defined as external_line in res_digium_phone.conf	Maps directly to an external_line defined in this configuration file. External line are lines not defined by SIP peers in sip.conf and generally do not register to this instance of Asterisk. If an external line is defined as the primary extension for a phone, many of the advanced phone application features will be disabled
config_file	file, e.g. mycustomconfig.xml	The phone module automatically generates a configuration file for each phone based on the lines assigned to it; but, it is possible to supply a custom configuration file instead. Using this option will direct the DPMA to serve up the specified file, as found in the file_directory defined directory, to the phone. For information about custom configuration files, see [\#xmlconfig|]. Note that using a custom configuration file, as opposed to the provisioning generated by the DPMA, precludes the phone's use of DPMA-specific applications, e.g. voicemail, parking, user status, etc. This option allows users to make use of the DPMA's mDNS provisioning capabilities, providing a simpler alternative to HTTP and Option 66 provisioning, but sacrifices the DPMA-specific features.
full_name	string, e.g. Bob Johnson	The full name of the person who will be using this phone, and what will appear in the user list that the phone pulls
contact	file, e.g. contacts.xml	An xml file in the file_directory containing a list of contacts to serve to the phone. Multiple contact options can exist in a single phone configuration.
blf_contact_group	string, a group from contacts	Every contacts xml file will have at least one group defined in it. This setting controls which loaded group the phone should subscribe to for its Rapid Dial (BLF) keys.
timezone	Timezone String, e.g. America/Chicago	Sets the timezone used for the clock on this phone.
ntp_server	hostname, IP address, e.g. ntp.mycompany.com	Defines the NTP server to which phones will synchronize themselves
ntp_resync	seconds as integer, e.g. 86400	Defines the interval between NTP synchronization

parking_exten	extension, e.g. 700	Sets the extension used for parking calls. When this option is set, and the phone has an in-progress call, it will display a "Park" softkey, allowing for one touch parking.
parking_transfer_type	blind / attended	The type of transfer to perform when parking a call using the "Park" softkey.
ringtone	Alarm, Chimes, Digium, GuitarStrum, Jingle, Office2, Office, RotaryPhone, SteelDrum, Techno, Theme, Tweedle, Twinkle, Vibe	Sets the default ringtone for the phone, defaults to Digium.
web_ui_enabled	no / yes	By default, when using the Digium Phone Module for Asterisk, the phone's built-in Web UI is disabled. To override this and enable the Web UI anyway, which may result in unpredictable behavior if user settings conflict with the settings provided by the DPMA, enable this option.
blf_unused_linekeys	no / yes	Digium phones, by default, place BLF keys on the sidecar, not on unused line keys. To disable this behavior and allow BLF keys to start mapping from the next available unused line key, enable this option.
d40_logo_file	string	The idle screen image for a D40 model phone in PNG format, 150x45 pixels, 8-bit depth, a color type without alpha transparency and less than 10k in size. Loaded from the directory specified by the file_directory\ option. The D40 and D50 screen size is the same; therefore it is permissible to re-use the same logo file for both.
d50_logo_file	string	The idle screen image for a D50 model phone in PNG format, 150x45 pixels, 8-bit depth, a color type without alpha transparency and less than 10k in size. Loaded from the directory specified by the file_directory\ option. The D40 and D50 screen size is the same; therefore it is permissible to re-use the same logo file for both.
d70_logo_file	string	The idle screen image for a D70 model phone in PNG format, 205x85 pixels, 8-bit depth, a color type without alpha transparency and less than 10k in size. Loaded from the directory specified by the file_directory\ option.

Example

In this example:

- the phone configuration is set for a phone whose MAC address matches 01:23:45:67:89:ab
- the phone configuration has a PIN of 10101019
- the phone's primary line is a SIP peer named bob101
- the phone's secondary line is a SIP peer named bob102
- the phone has an external line called bobexternal
- the phone does not load an external configuration file
- the full name of the phone is Bob's Phone
- the phone loads a contacts XML file named bobscontacts.xml
- the phone uses a contact group, from bobscontacts.xml, named "RapidDial" for its BLF keys
- the phone is set for the "America/Los_Angeles" timezone
- the phone's NTP server is set to 0.digium.pool.ntp.org
- the phone's NTP resynchronization time is 86400
- the phone will blind transfer parked calls to extension 700
- the phone's ringtone is a Guitar Strum
- the phone's Web UI is disabled
- the phone's Rapid Dial keys will begin from the side car
- if the phone claiming the profile is a D40, it will use the logo file d40_logo.png
- if the phone claiming the profile is a D50, it will use the logo file d50_logo.png
- if the phone claiming the profile is a D70, it will use the logo file d70_logo.png

Example Phone Configuration.

```
[bobsphone]
type=phone
mac=0123456789ab
pin=10101019
line=bob101
line=bob102
external_line=bobexternal
;config_file
full_name=Bob's Phone
contact=bobstonctacts.xml
blf_contact_group=RapidDial
timezone=America/Los_Angeles
ntp_server=0.digium.pool.ntp.org
ntp_resync==86400
parking_exten=700
parking_transfer_type=blind
ringtone=GuitarStrum
web_ui_enabled=no
blf_unused_linekeys=no
d40_logo_file=d40_logo.png
d50_logo_file=d50_logo.png
d70_logo_file=d70_logo.png
```

Line Configuration Options

Internal lines are SIP peers (friends), but there is Digium Phone-specific data associated with lines that does not otherwise exist in the sip.conf entry. Because of this, advanced line features must be defined separately from sip.conf, here, in res_digium_phone.conf. Advanced line configuration is not a requirement to get a line to work, it only acts as a method of setting advanced phone features to an already-defined sip.conf entry.

Option	Values	Description
exten	string, e.g. 1000	When the sip peer name is different than the actual extension used to contact this line, this option should be set to the line's dialable extension. By default it is assumed that the sip peer name is actual extension, which is true for most Asterisk distributions such as FreePBX and AsteriskNOW, but is not considered a best practice for use of generic Asterisk.
digit_map	Digit mapping, see [\#dialplans|]	The digit mapping to use for this line.
line_label	string, e.g. MD 123	The line label to display on the phone for this line's line key.
language	string, e.g. en_US	The preferred language to use for this line. Currently, only US English is supported.
mailbox	mailbox from voicemail.conf	The voicemail box associated with the line. When not set, this will default first to the mailbox defined for the sip peer associated with the line, and second to a mailbox entry in voicemail.conf matching the name of the line. IF neither of these are found and this option is not set, the line does not have a mailbox and visual voicemail will not be enabled.
voicemail_uri	string, sip:user@host	If the phone's Msgs button should dial a SIP URI rather than opening the visual voicemail application, this option specifies what URI the Msgs button should dial. Setting this option on a phone's primary lie will disable visual voicemail.
outboundproxy_address	Host / IP address	The Outbound SIP proxy address this line should use
outboundproxy_port	Port as integer	The port of the Outbound SIP proxy; defaults to 5060 when outboundproxy_address is set
transport	udp, tcp	SIP transport method this line should use. Defaults to udp
reregistration_timeout	integer, e.g. 300	The number of seconds before re-registering

registration_retry_interval	integer, e.g. 25	The number of seconds to wait before retrying to register after registration fails.
registration_max_retries	integer, e.g. 5	The number of times the phone will attempt to retry registering after registration fails.

Example

In this example:

- the Digit Mapping for the phone is set to [0-8]xxx
- The Label for the line, as it appears on the phone is BobbyJ
- The Mailbox for the line is bob101
- The Voicemail URI (number to be dialed) is 8000 at the 10.10.10.10 PBX.
- The Outbound Proxy is 10.10.10.1 and its port is 5060.
- The Transport type for the signaling is TCP
- The Re-registration timeout is 300 seconds
- The Registration Retry Interval is 25 seconds
- The Maximum Registration Retries is 5 times

Example Line Configuration.

```
[bob101]
type=line
digit_map=[0-8]xx
line_label=BobbyJ
mailbox=bob101
voicemail_uri=sip:8000@10.10.10.10
outboundproxy_address=10.10.10.1
outboundproxy_port=5060
transport=tcp
reregistration_timeout=300
registration_retry_interval=25
registration_max_retries=5
```

If SIP peers are configured using Asterisk Realtime, then the **secret** parameter, as noted in the External Line section, must be populated for standard lines, with the secret used by the actual peer.

External Line Configuration Options

External lines are external to this Asterisk instance; they are lines that are not entries in sip.conf. The external line concept exists to work around the forcing of lines as sip.conf peers. Since external lines are not SIP peers, they require more information than normal line configurations. Here are the external line-specific configuration options.

Note that all internal line options are also available for external lines; but, any internal line options specific to applications on the phone, such as mailbox to enable visual voicemail, will not work unless the server-side component resides on **this** Asterisk instance.

Configuring an external line as the primary line for a phone will result in the advanced PBX features being disabled for the phone

Option	Values	Description
server_address	string, e.g. otherpbx.othercompany.com	The address this line should contact for registration and outbound calls
server_port	integer, e.g. 5060	The port this line should contact for registration and outbound calls. Defaults to 5060
secondary_server_address	string, e.g. backuppbx.othercompany.com	The address this line should contact for registration and outbound calls if the primary server is not available
secondary_server_port	integer, e.g. 5060	The port this line should contact for registration and outbound calls if the primary server is not available

secondary_server_transport	tcp or udp	The transport type used for registration and calling to/from the secondary server
userid	string, e.g. bob1234	This line's SIP username. Defaults to the line's name
authname	string, e.g. bob1234	SIP authorization name if different than userid. If blank, defaults to userid.
secret	string, e.g. mymagicpassword	The SIP secret this line should used
register	yes / no	Indicates whether or not this line should register
callerid	string	Caller ID field to use for this line

*Example: *

In this example:

- The address of the external registration server is otherpbx.mycompany.com
- The contact port of the external registration server is 5061
- The transport method of the external registration server is TCP
- The address of the secondary external registration server is otherpbx2.company.com
- The contact port of the secondary registration server is 5061
- The transport method of the secondary external registration server is UDP
- The SIP username is bob1234
- The SIP authname is bob4321
- The SIP password (secret) is mymagicalpassword
- Registration is enabled
- Caller ID is set to "Bob Jones <555-1234>"

Example External Line Configuration.

```
[bob1234]
type=external_line
server_address=otherpbx.mycompany.com
server_port=5061
transport=tcp
secondary_server_Address=otherpb2.mycompany.com
secondary_server_port=5061
secondary_server_transport=udp
userid=bob1234
authname=bob4321
secret=mymagicalpassword
registration=yes
callerid=Bob Jones <555-1234>
```

Contacts

Digium Phones provide a Contacts application that integrates speed-dial, device busy-lamp field and user presence. The Contacts application is related to the Status application in that both are concerned with presence. The Status application is concerned only with setting the local user's presence. The Contacts application is concerned with the presence status of other users - those to which it's been programmed to subscribe.

Contacts without presence subscriptions may be loaded locally onto the phone by the phone's user. Most deployments of Digium phones will see the administrator specifying XML lists of contacts to load onto phones. Here, we will discuss the methods to direct the phone to load contacts as well as the contents of the contacts XML files.

Loading a Contact List

Contacts files that the phone should load are defined in the phone's configuration XMl file and are served up by the DPMA. Contact file are specified for a phone configuration using the phone configuration parameter:

- contact

as noted above. Multiple "contact" lines may be used for each phone configuration.

Rapid Dial Keys

Rapid Dial, or BLF, keys are set using the phone configuration parameter:

- blf_contact_group

The order of the contacts in this group is important. Those contacts fill in the BLF-keys in the same order as given in the xml, and those contacts are subscribed to, in order, up to a maximum of 40.

Contacts XML skeleton

A basic Contacts XML structure is defined here:

Contactx XML Structure example

```
<phonebooks>
    <contacts>
        <contact>
            <numbers>
                <number />
            </numbers>
            <emails>
                <email />
            </emails>
        </contact>
    </contacts>
</phonebooks>
```

And, a more fleshed-out example looks like:

XML Contact Example

```
<phonebooks>
    <contacts group_name="PBX Directory" editable="0">
        <contact
            prefix="Mr"
            first_name="Robert"
            second_name="Davis"
            last_name="Jones"
            suffix="III"
            contact_type="sip"
            organization="Digium"
            job_title="Direction Manager"
            location="East Texas"
            notes=""
            account_id="104"
            subscribe_to="sip:104@1.2.3.4"
        >
            <numbers>
                <number dial="104" dial_prefix="" label="Extension" primary="1" />
                    <number dial="8005551234" dial_prefix="9" label="Mobile" />
                </numbers>
                <emails>
                    <email address="rdj@mycompany.com" label="Work" primary="1" />
                </emails>
        </contact>
     </contacts>
    <contacts group_name="Default" editable="1" id="1">
    ....
    </contacts>
    <contacts group_name="Family Members" editable="1" id="2">
    ....
    </contacts>
</phonebooks>
```

Dial Plans

The dial plan includes settings that specify the behavior of the phone as a user enters a number in off-hook dialing mode.

The digit map is the setting that describes different patterns of numbers. When a number matches a pattern, the number is sent to Asterisk to place the call. The pattern may include a timer at the end. If no numbers are entered before the time expires, the number matching the pattern will be sent. If additional numbers are entered before the time elapses, the pattern no longer matches.

The syntax of a digit map is:

Digit Map Syntax

```
digitmaplist := digitmap ( '|' digitmap )*

digitmap := digitstring | digitstring timer

digitstring := atom+

atom := literal | class | wildcard

literal := digit | '*' | '#'

digit := '0' | '1' | '2' | '3' | '4' | '5' | '6' | '7' | '8' | '9'

class := '[' range+ ']'

wildcard := '.'

range := digit | digit '-' digit

timer := T digit*
```

A digit map with a timer, but no specified time value, defaults to 4 seconds.

Example:

The following example assumes the following dials will be completed:

- Dial four digit extensions from 0000-8999 after the default timeout.
- Dial 911 immediately
- Dial 9911 immediately
- Dial 411 immediately
- Dial 611 immediately
- Dial numbers beginning with 011 followed by at least 4 more digits after a delay of 3 seconds.
- Dial numbers beginning with 1 followed by 10 digits immediately
- Dial numbers beginning with 2-9 followed by 6 digits immediately
- Dial any three digit number after a delay of 3 seconds.

Digit Map Example.

```
[0-9]xxx|911|9911|411|611|011xxxx.T3|1xxxxxxxxxx|[2-9]xxxxxx|*xx.T3
```

Note that the phone will attempt to immediately dial any pattern that does not have a matching rule

Ringing Types and Intercom

Digium phones support several different types of ringing:

- Normal Ringing - normal
- One Ring followed by automated Answer - ring-answer
- Immediate Answer, No ringing - intercom
- Visual only, No ringing sound - visual

When an incoming call is received, the phone will look for an ALERT_INFO SIP header, and compare the received header against its configuration, as parsed from its Config XML.

Affecting the ring tone from the Asterisk dialplan requires use of the alert_info SIP header. This header should be added into the dialplan before executing a dial to the phone, as such:

Selecting a Ringtone from the Dialplan

```
exten => 100,1,SIPAddHeader("Alert-Info: normal")
exten => 100,n,Dial(SIP/mypeer)
```

Alert-Info elements encapsulated in angle brackets are currently not parsed correctly by the Digium phones. As such, users who require use of angle brackets for the first Alert-Info element, and who are using their own generated configurations, not provided by the DPMA, must use the escaped characters directly in the alert definition.

Firmware

Firmware configuration Management

Firmware configuration is handled globally for the DPMA. When a firmware package directory is set in res_digium_phone.conf, the firmware associated with that package becomes the firmware every phone configured with the DPMA will run from then on. Phones already configured with the DPMA before a firmware package change will have to be sent a reconfigure message or manually restarted to receive the firmware changes.

In res_digium_phone.conf the directory of a firmware package is pointed to. In that firmware package directory there will be a digium_phones_firmware.conf that maps all the firmware (.eff) files in the directory to a phone model and a firmware version.

Example: Example digium_phones_firmware.conf file in a Digium maintained firmware package.

```
[D40]
version=1_0_3_45441
file=1_0_3_45441_D40_firmware.eff

[D50]
version=1_0_3_45441
file=1_0_3_45441_D50_firmware.eff

[D70]
version=1_0_3_45441
file=1_0_3_45441_D70_firmware.eff
```

Each context in that config file represents a specific Digium Phone model along with what firmware to upload to that module and the version associated with it. When a phone requests its config, the DPMA advertises the available firmware version and where to retrieve that firmware.

XML Configuration

This section describes the formatting and options available when creating XML-based configuration files for provisioning Digium phones. Users choosing this method of configuration forgo use of the DPMA, and instead are provisioning phones for use with Asterisk versions that do not support the DPMA.

The configuration elements provided in this section are subject to change between Digium Phone firmware releases.

A Complete XML Configuration Example

A Complete XML Configuration Example

```
<?xml version="1.0" ?>
<config>
    <setting id="login_password" value="789" />
    <setting id="time_zone" value="America/Los_Angeles" />
    <setting id="time_source" value="ntp" />
    <setting id="ntp_server" value="0.digium.pool.ntp.org" />
    <setting id="ntp_resync" value="86400" />
```

```
    <setting id="accept_local_calls" value="any" />
    <setting id="transport_udp_enabled" value="1" />
    <setting id="transport_udp_port" value="5060" />
    <setting id="transport_tcp_enabled" value="1" />
    <setting id="transport_tcp_port" value="5060" />
    <setting id="logo_file" value="user" path="/user_image.png"
url="https://server.example.com/mylogo.png" md5="126cd744583eeea1ab7e44ed8af3d39c" />
    <setting id="display_mc_notifications" value="1" />
    <setting id="brightness" value="10" />
    <setting id="contrast" value="10" />
    <setting id="dim_backlight" value="1" />
    <setting id="backlight_timeout" value="30" />
    <setting id="backlight_dim_level" value="2" />
    <setting id="ringer_volume" value="5" />
    <setting id="speaker_volume" value="5" />
    <setting id="handset_volume" value="5" />
    <setting id="headset_volume" value="5" />
    <setting id="reset_call_volume" value="0" />
    <setting id="default_ringtone" value="Digium" />
    <setting id="active_ringtone" value="Digium" />
    <setting id="headset_answer" value="0" />
    <setting id="desi_strip_enable" value="1" />
    <setting id="enable_blf_on_unused_line_keys" value="0" />
    <setting id="name_format" value="first_last" />
    <setting id="contacts_max_subscriptions" value="40" />
    <setting id="blf_contact_group" value="Default" />
    <setting id="network_enable_dhcp" value="1" />
    <setting id="network_static_ip_address" value="" />
    <setting id="network_subnet_mask" value="" />
    <setting id="network_default_gateway" value="" />
    <setting id="network_domain_name" value="" />
    <setting id="network_primary_dns_server" value="" />
    <setting id="network_secondary_dns_server" value="" />
    <setting id="network_vlan_discovery_mode" value="NONE" network="10.10.0.0/16" />
    <setting id="network_vlan_qos" value="0" />
    <setting id="network_vlan_id" value="44" />
    <setting id="log_level" value="debug" />
    <setting id="enable_logging" value="0" />
    <setting id="log_server" value="10.1.2.3" />
    <setting id="log_port" value="514" />
    <setting id="web_ui_enabled" value="1" />
    <setting id="sip_dscp" value="24" />
    <setting id="rtp_dscp" value="46" />
    <contacts url="https://server.example.com/myfile.xml" id="internal" md5="abcd123" />
    <accounts>
        <account index="0" status="1" register="1" account_id="100" username="100" authname="100"
password="100" passcode="100" line_label="100 Alligator" caller_id="100 Alligator"
dial_plan="[0-8]xxxxx|911|9411|9611|9011xxx.T3|91xxxxxxxxxx|9[2-9]xxxxxx|*xx.T3|[0-8]xx.T3"
visual_voicemail="0" voicemail="sip:800@10.1.2.3" outbound_proxy="" outbound_port=""
conflict="replace">
            <host_primary server="10.10.2.108" port="5060" transport="udp" reregister="300"
retry="25" num_retries="5" />
            <permission id="record_own_calls" value="0" />
        </account>
    </accounts>
    <codecs>
        <codec id="PCMU" priority="255" packetization="20" jitter_min="0" jitter_max="0"
jitter_target="0" enabled="1" />
        <codec id="PCMA" priority="13" packetization="20" jitter_min="0" jitter_max="0"
jitter_target="0" enabled="1" />
        <codec id="G722" priority="11" packetization="20" jitter_min="0" jitter_max="0"
jitter_target="0" enabled="1" />
        <codec id="G726-32" priority="7" packetization="20" jitter_min="0" jitter_max="0"
jitter_target="0" enabled="1" />
        <codec id="G729" priority="4" packetization="20" jitter_min="0" jitter_max="0"
jitter_target="0" enabled="1" />
        <codec id="L16" priority="2" packetization="20" jitter_min="0" jitter_max="0"
jitter_target="0" enabled="1" />
        <codec id="L16-256" priority="1" packetization="20" jitter_min="0" jitter_max="0"
jitter_target="0" enabled="1"/>
```

```
    </codecs>
    <ringtones>
        <tones>
            <tone id="Alarm" display="Alarm" type="phone"/>
            <tone id="Chimes" display="Chimes" type="phone"/>
            <tone id="Digium" display="Digium" type="phone"/>
            <tone id="GuitarStrum" display="Guitar Strum" type="phone"/>
            <tone id="Jingle" display="Jingle" type="phone"/>
            <tone id="Office" display="Office" type="phone"/>
            <tone id="Office2" display="Office 2" type="phone"/>
            <tone id="RotaryPhone" display="Rotary Phone" type="phone"/>
            <tone id="SteelDrum" display="Steel Drum" type="phone"/>
            <tone id="Techno" display="Techno" type="phone"/>
            <tone id="Theme" display="Theme" type="phone"/>
            <tone id="Tweedle" display="Tweedle" type="phone"/>
            <tone id="Twinkle" display="Twinkle" type="phone"/>
            <tone id="Vibe" display="Vibe" type="phone"/>
        </tones>
        <alerts>
            <alert alert_info="normal" ringtone_id="Digium" ring_type="normal" />
            <alert alert_info="ring-answer" ringtone_id="Digium" ring_type="ring-answer" />
            <alert alert_info="intercom"  ringtone_id=""  ring_type="answer" />
            <alert alert_info="visual"  ringtone_id=""  ring_type="visual" />
        </alerts>
    </ringtones>
    <appconfig id="appscreen">
        <application id="contacts" />
    </appconfig>
    <appconfig id="contacts">
        <settings use_local_storage="1" can_transfer_vm="0" />
    </appconfig>
    <firmwares>
        <firmware model="D50" version="1_0_3_45441"
url="http://10.10.4.11/firmware/1_0_3_45441_D50_firmware.eff" />
        <firmware model="D70" version="1_0_3_45441"
url="http://10.10.4.11/firmware/1_0_3_45441_D70_firmware.eff" />
        <firmware model="D40" version="1_0_3_45441"
url="http://10.10.4.11/firmware/1_0_3_45441_D40_firmware.eff" />
    </firmwares>
    <public_firmwares>
        <public_firmware model="D50" version="1_0_3_45441"
url="http://firmware.example.com/1_0_3_45441_D50_firmware.eff" />
        <public_firmware model="D70" version="1_0_3_45441"
url="http://firmware.example.com/1_0_3_45441_D70_firmware.eff" />
        <public_firmware model="D40" version="1_0_3_45441"
```

```
url="http://firmware.example.com/1_0_3_45441_D40_firmware.eff" />
    </public_firmwares>
</config>
```

Setting Elements

XML Setting Elements

XML Setting Elements

```
<?xml version="1.0" ?>
<config>
    <setting id="login_password" value="789" />
    <setting id="time_zone" value="America/Los_Angeles" />
    <setting id="time_source" value="ntp" />
    <setting id="ntp_server" value="pool.ntp.org" />
    <setting id="ntp_resync" value="86400" />
    <setting id="accept_local_calls" value="any" />
    <setting id="transport_udp_enabled" value="1" />
    <setting id="transport_udp_port" value="5060" />
    <setting id="transport_tcp_enabled" value="1" />
    <setting id="transport_tcp_port" value="5060" />
    <setting id="logo_file" value="user" path="/user_image.png"
url="https://server.example.com/mylogo.png" md5="126cd744583eeea1ab7e44ed8af3d39c" />
    <setting id="display_mc_notifications" value="1" />
    <setting id="brightness" value="10" />
    <setting id="contrast" value="10" />
    <setting id="dim_backlight" value="1" />
    <setting id="backlight_timeout" value="30" />
    <setting id="backlight_dim_level" value="2" />
    <setting id="ringer_volume" value="5" />
    <setting id="speaker_volume" value="5" />
    <setting id="handset_volume" value="5" />
    <setting id="headset_volume" value="5" />
    <setting id="reset_call_volume" value="0" />
    <setting id="default_ringtone" value="Digium" />
    <setting id="active_ringtone" value="Digium" />
    <setting id="headset_answer" value="0" />
    <setting id="desi_strip_enable" value="1" />
    <setting id="enable_blf_on_unused_line_keys" value="0" />
    <setting id="name_format" value="first_last" />
    <setting id="contacts_max_subscriptions" value="40" />
    <setting id="blf_contact_group" value="Default" />
    <setting id="network_enable_dhcp" value="1" />
    <setting id="network_static_ip_address" value="" />
    <setting id="network_subnet_mask" value="" />
    <setting id="network_default_gateway" value="" />
    <setting id="network_domain_name" value="" />
    <setting id="network_primary_dns_server" value="" />
    <setting id="network_secondary_dns_server" value="" />
    <setting id="network_vlan_discovery_mode" value="NONE" network="10.10.0.0/16" />
    <setting id="network_vlan_qos" value="0" />
    <setting id="network_vlan_id" value="44" />
    <setting id="log_level" value="debug" />
    <setting id="enable_logging" value="0" />
    <setting id="log_server" value="10.1.2.3" />
    <setting id="log_port" value="514" />
    <setting id="web_ui_enabled" value="1" />
    <setting id="parking_lot_extension" value="700" />
    <setting id="parking_lot_enable_blind_transfer" value="0" />
    <setting id="sip_dscp" value="24" />
    <setting id="rtp_dscp" value="46" />
</config>
```

Each <setting> element represents at least an **id** and **value** pair of attributes. Some <setting> tags may have additional attributes.

General (Login)

Option	Values	Description
login_password	String, e.g. 789	Sets the Admin Password for logging into Web UI or Admin Settings Section on Phone Menu, defaults to 789

General (Time)

time_zone	Timezone String, e.g. America/Chicago	Sets the time zone for the phone
time_source	ntp	Sets the time source for the phone
ntp_server	Hostname or IP address, e.g. 0.digium.pool.ntp.org	Sets the NTP server to which the phone will synchronize itself, defaults to 0.digium.pool.ntp.org
ntp_resync	Seconds as integer, e.g. 86400	Sets the interval between NTP synchronization

General (SIP)

accept_local_calls	any / host	Sets whether to accept calls from any source or only from hosts to which the phone is registered
transport_udp_enabled	boolean	Sets whether to enable UDP transport, defaults to 1
transport_udp_port	Valid integer for ports (1-65535)	Sets the local UDP SIP port, defaults to 5060
transport_tcp_enabled	boolean	Sets whether to enable TCP transport, defaults to 1
transport_tcp_port	Valid integer for ports (1-65535)	Sets the local TCP SIP port, defaults to 5060

Preferences (Idle Screen)

logo_file	**value** as factory / user; **path** as location on disk of file - /factory_asterisk.png for default and /user_image.png for custom ; **url** as optional location to fetch a logo; **md5** as optional when url is used to determine if logo has changed to avoid re-fetching	Sets the idle screen logo, defaults to factory-asterisk.png
display_mc_notifications	boolean	Disables / Enables display of missed calls on the phone, defaults to 1

Preferences (Display)

brightness	integer (0-10)	Sets the LCD screen brightness, defaults to 5
contrast	integer (0-10)	Sets the LCD screen contrast, defaults to 5
dim_backlight	boolean	enable backlight dimming where 1 dims the screen after backlight timeout has been reached and phone is otherwise idle, defaults to 1
backlight_timeout	integer (0-3200)	Time, in seconds, before backlight is set to backlight_dim_level while phone is idle; setting to 0 disables backlight timeout, defaults to 0
backlight_dim_level	integer (0-10)	Brightness level dims to when when dim_backlight is 1, defaults to 2

Preferences (Sounds)

ringer_volume	integer (0-10)	Sets the ringer volume, defaults to 5
speaker_volume	integer (0-10)	Sets the speaker volume, defaults to 5

handset_volume	integer (0-10)	Sets the handset volume, defaults to 5
headset_volume	integer (0-10)	Sets the headset_volume, defaults to 5
reset_call_volume	boolean	If 1, volume changes made during a call do not persist to the next call, defaults to 0
default_ringtone	Tone ID from <tones> provided to phone	Sets the default phone ringtone, defaults to Digium
active_ringtone	Tone ID from <tones> provided to phone	Sets the current user-selected ringtone, defaults to Digium

Preferences (Answering Calls)

headset_answer	boolean	Sets whether to use the headset, rather than the speaker, for answering all calls, defaults to 0

Contacts

desi_strip_enable	boolean	Applies to D50 and D70 models, enables / disables sidecar, defaults to 1
enable_blf_on_unused_line_keys	boolean	If 1, assigns BLFs beginning with first empty line key. If 0, assigns BLFs beginning with first sidecar key. Defaults to false.
name_format	first_last, last_first	Formats the display of contact names, defaults to first_last
contacts_max_subscriptions	integer (0-40)	Sets the maximum number of subscriptions for presence indications - 0 disables presence subscriptions. Defaults to 40
blf_contact_group	Any Group ID from the loaded contacts	The ID of the contact list group to use for the rapid dial list

Network (IP Settings)

network_enable_dhcp	boolean	Disable or Enable DHCP network configuration, defaults to 1
network_static_ip_address	IPv4 address	Defines the network address for the phone
network_subnet_mask	IPv4 netmask	Defines the netmask for the phone
network_default_gateway	IPv4 address	Defines the network gateway for the phone
network_primary_dns_server	IPv4 address	Defines the primary DNS server for the phone
network_secondary_dns_server	IPv4 address	Defines the secondary DNS server for the phone

Network (Virtual LAN)

network_vlan_discovery_mode	**value** of NONE, MANUAL, LLDP; **network** as IP mask	Sets use of none, manual, or LLDP discovered VLAN and, if MANUAL, defines the network; defaults to LLDP
network_vlan_qos	integer (0-7)	Sets the VLAN QOS level
network_vlan_id	integer (0-4095)	Sets the VLAN ID

Logging

log_level	error, warning, debug, info	Sets the logging level, defaults to error
log_server	IPv4 address of syslog server	Specifies remote syslog server
log_port	port as integer	Specifies port of remote syslog server
enable_logging	boolean	Disables or Enables remote syslog, defaults to 0

Miscellaneous

web_ui_enabled	boolean	Disables, Enables the phone's web user interface, defaults to 1 (Enabled)
sip_dscp	integer (0-63)	Specifies the DSCP field of the DiffServ byte for SIP Signaling QoS, defaults to 24
rtp_dscp	integer (0-63)	Specifies the DSCP field of the DiffServ byte for RTP Media QoS, defaults to 46

Contacts Element

Contacts Element Example

Contacts Element Example

```
<?xml version="1.0" ?>
<config>
    <contacts url="https://server.example.com/myfile.xml" id="internal" md5="abcd123" />
</config>
```

Any number of contacts elements may be present directly under the config element.
These <contacts> elements have attributes, most importantly url which allow contacts xml sheets to be downloaded via the DPMA.

Option	Values	Description
contacts	**url** as file link, **id** as unique identifier, **md5** as the md5sum of the xml file	Specifies the contacts XML file to be retrieved by the phone and identifies that file; more than one contacts parameter may be used

Accounts Element

Accounts Element Example

Accounts Element Example

```
<?xml version="1.0" ?>
<config>
    <accounts>
        <account index="0" status="1" register="1" account_id="100" username="100" authname="100"
password="100" passcode="100" line_label="100 Alligator" caller_id="100 Alligator"
dial_plan="[0-8]xxxxx|911|9411|9611|9011xxx.T3|91xxxxxxxxxx|9[2-9]xxxxxx|*xx.T3|[0-8]xx.T3"
visual_voicemail="0" voicemail="sip:800@10.1.2.3" outbound_proxy="" outbound_port=""
conflict="replace">
            <host_primary server="10.10.2.108" port="5060" transport="udp" reregister="300"
retry="25" num_retries="5" />
            <permission id="record_own_calls" value="0" />
        </account>
    </accounts>
</config>
```

Option	Values	Description
index	integer (0-5)	Defines the line key to which the account will be mapped
status	boolean	Enables the line key; if false, will not display the line on the phone or register with the primary host
register	boolean	If 1, then this account will attempt to register with the primary host
account_id	string	SIP username
username	string	SIP username
authname	string	SIP authname
password	string	SIP password
passcode	string	SIP password
line_label	string	The text that shows up next to the line key for this account
caller_id	Name <Number>	Outgoing caller id displayed for this account

dial_plan	Digit mapping, see [\#dialplans|]	The dial plan / digit mapping for this account
visual_voicemail	boolean	Only valid on account with index of 0. Only valid for phones provisioned by Switchvox or the DPMA. If this is set to 1 then the Msgs button action will open the voicemail app. Otherwise it will dial the voicemail extension.
voicemail	digits or SIP URI	A SIP URI or extension to be dialed for voicemail pertaining to this account.
outbound_proxy	IP address / Hostname	Outbound proxy for this account
outobund_port	port	Port for the outbound proxy
conflict	replace	

Host Primary: Child Element of <account>

Option	Values	Description
server	Hostname or IPv4 Address	Sets the server to which calls for this account are directed
port	integer (1-65535)	Sets the server's SIP port
transport	udp, tcp	Sets the transport type, UDP or TCP
reregister	integer in seconds	Sets the re-registration interval
retry	integer	Specifies the number of time to attempt re-registration if registration fails
num_retries	integer	Specifies the number of retries to attempt if registration fails

Permission: Child Element of <account>

Defines line/account based permissions for various phone functions with an **id** and **value** pair.

Option	Values	Description
record_own_calls	boolean	If 1, allows the user to record their own calls using a soft-key. Note that this feature can only enabled when using the DPMA. Users manually provisioning Digium phones should set this to 0 in order to ensure that a non-functional (because the DPMA is not being used) call recording softkey does not appear.

Codecs Element

Codecs Element Example

Codecs Element Example

```
<?xml version="1.0" ?>
<config>
    <codecs>
        <codec id="PCMU" priority="255" packetization="20" jitter_min="0" jitter_max="0"
jitter_target="0" enabled="1" />
        <codec id="PCMA" priority="13" packetization="20" jitter_min="0" jitter_max="0"
jitter_target="0" enabled="1" />
        <codec id="G722" priority="11" packetization="20" jitter_min="0" jitter_max="0"
jitter_target="0" enabled="1" />
        <codec id="G726-32" priority="7" packetization="20" jitter_min="0" jitter_max="0"
jitter_target="0" enabled="1" />
        <codec id="G729" priority="4" packetization="20" jitter_min="0" jitter_max="0"
jitter_target="0" enabled="1" />
        <codec id="L16" priority="2" packetization="20" jitter_min="0" jitter_max="0"
jitter_target="0" enabled="1" />
        <codec id="L16-256" priority="1" packetization="20" jitter_min="0" jitter_max="0"
jitter_target="0" enabled="1"/>
    </codecs>
</config>
```

The codecs element contains all available codecs. each described by an individual <codec> element and its attributes.

Each <codec> element is described by the following attributes:

Option	Values	Description
id	PCMU, PCMA, G722, G726-32, G729, L16, L16-256	A codec supported by the phone
priority	integer (1-255)	Priority of the codec where higher numbers mean the codec is more favored
packetization	integer in 10ms increments per RFC codec guidelines	Packetization (ptime) rate for the specified codec, defaults to 20
jitter_min	integer in ms	Sets the minimum size of the codec jitter buffer
jitter_max	integer in ms	Sets the maximum size of the codec jitter buffer
jitter_target	integer in ms	Sets the target size of the codec jitter buffer
enabled	boolean	Disables / Enables a codec

Ringtones Element

This section has two primary child elements:

1. tones, which are the actual sounds heard when a call is made
2. alerts, which map to a tone and represent a certain call condition

Tones: Child element of <ringtones>

Tones Element Example

Tones Element Example

```
<?xml version="1.0" ?>
<config>
    <ringtones>
        <tones>
            <tone id="Alarm" display="Alarm" type="phone"/>
            <tone id="Chimes" display="Chimes" type="phone"/>
            <tone id="Digium" display="Digium" type="phone"/>
            <tone id="GuitarStrum" display="Guitar Strum" type="phone"/>
            <tone id="Jingle" display="Jingle" type="phone"/>
            <tone id="Office" display="Office" type="phone"/>
            <tone id="Office2" display="Office 2" type="phone"/>
            <tone id="RotaryPhone" display="Rotary Phone" type="phone"/>
            <tone id="SteelDrum" display="Steel Drum" type="phone"/>
            <tone id="Techno" display="Techno" type="phone"/>
            <tone id="Theme" display="Theme" type="phone"/>
            <tone id="Tweedle" display="Tweedle" type="phone"/>
            <tone id="Twinkle" display="Twinkle" type="phone"/>
            <tone id="Vibe" display="Vibe" type="phone"/>
            <tone id="208" display"Fancy" url="http://10.1.2.3/mytone.wav" md5="abc123"
type="user"/>
        </tones>
    </ringtones>
</config>
```

Element lists the <tone> elements, each described by the following attributes:

Option	Values	Description
id	string	Internal Tone identifier
display	string	External Tone Description
url	URL string	Location from which to retrieve a 16kHz, mono WAV sound file, less than 1MB in size
md5	md5sum	MD5 sum of the file to be retrieved
type	phone, user	Indicates the tone's origin; tones that are type phone are embedded into the phone's firmware, tones that are type user are retrieved by URL

Alerts: Child element of <ringtones>

Alerts Element Example

Alerts Element Example

```
<?xml version="1.0" ?>
<config>
    <ringtones>
        <alerts>
            <alert alert_info="normal" ringtone_id="Digium" ring_type="normal" />
            <alert alert_info="ring-answer" ringtone_id="Digium" ring_type="ring-answer" />
            <alert alert_info="intercom"  ringtone_id=""  ring_type="answer" />
            <alert alert_info="visual"  ringtone_id=""  ring_type="visual" />
        </alerts>
    </ringtones>
</config>
```

- Element lists the <alert> elements, each described by the following attributes.
- Alert tones are played when the event designated by 'alert_info' occurs.

Option	Values	Description
alert_info	string	The alert_info header that, as received, applies to this alert

ringtone_id	string	The id of the ring tone for this alert
ring_type	normal, answer, ring-answer, visual	The type of call-answer to affect for this alert.

Firmwares Element

Firmwares Element Example

Firmwares Element Example

```
<?xml version="1.0" ?>
<config>
    <firmwares>
        <firmware model="D50" version="1_0_3_45441"
url="http://10.10.4.11/firmware/1_0_3_45441_D50_firmware.eff" />
        <firmware model="D70" version="1_0_3_45441"
url="http://10.10.4.11/firmware/1_0_3_45441_D70_firmware.eff" />
        <firmware model="D40" version="1_0_3_45441"
url="http://10.10.4.11/firmware/1_0_3_45441_D40_firmware.eff" />
    </firmwares>
</config>
```

- Element lists the <firmwares> elements, each described by the following attributes.
- Network, if specified, allows the phone to load different firmware URLs depending on its own network address mask

Option	Values	Description
model	D40, D50, D70	Model number of the Digium phone
version	string	Version string for the firmware. On boot, the phone will check the version string against an internal copy of the string, as previously loaded. If the strings differ, the phone will load the new firmware
url	http URL string	URL location of the phone firmware

Public Firmwares Element

Public Firmwares Element Example

Firmwares Element Example

```
<?xml version="1.0" ?>
<config>
    <public_firmwares>
        <public_firmware model="D50" version="1_0_3_45441"
url="http://firmware.example.com/1_0_3_45441_D50_firmware.eff" />
        <public_firmware model="D70" version="1_0_3_45441"
url="http://firmware.example.com/1_0_3_45441_D70_firmware.eff" />
        <public_firmware model="D40" version="1_0_3_45441"
url="http://firmware.example.com/1_0_3_45441_D40_firmware.eff" />
    </public_firmwares>
</config>
```

- Element lists the <public_firmwares> elements, each described by the following attributes.
- Specifies a fallback firmare location - more than one public_firmware element may be specified for each model and the public_firmware servers will be tried in the order they are listed, in the event that an internal firmware server cannot be reached

Option	Values	Description
model	D40, D50, D70	Model number of the Digium phone

version	string	Version string for the firmware. On boot, the phone will check the version string against an internal copy of the string, as previously loaded. If the strings differ, the phone will load the new firmware
url	http URL string	URL location of the phone firmware

Appconfig Element

Appconfig Element Example

Appconfig Element Example

```
<?xml version="1.0" ?>
<config>
    <appconfig id="appscreen">
        <application id="contacts" />
    </appconfig>
    <appconfig id="contacts">
        <settings use_local_storage="1" can_transfer_vm="0" />
    </appconfig>
</config>
```

- Element lists the <appconfig> elements

Digium phones when used with the DPMA

SIP Configuration

Configuration of a phone via the Digium Phone module for Asterisk alone is not enough to enable calling between the phone and Asterisk. As with any SIP device that connects to Asterisk, each Digium phone needs a corresponding entry in Asterisk's SIP configuration, e.g. sip.conf. Asterisk provides two types of entities within SIP: peers and friends. Use of either type is permissible, when configuring a Digium phone; however, use of the **peer** type means that Asterisk will not correctly match incoming calls where more than one SIP identity is assigned to the same phone (IP address). General practice then means that the **friend** type is the most flexible - as it matches on the From: username, whereas **peer** matches on IP and port (unless insecure=port has been set).

A minimum SIP.conf entry for a Digium phone then would look like:

SIP Configuration Entry Example

```
[mydigiumphone]
type = friend ; Use of "friend" is good practice, generally
nat = force_rport ; Good security practice dictates enabling nat support by default in both the
general and individual phone sections
host = dynamic ; Dynamic in this case since the device is registering with us
secret = UseGoodPasswords ; Always use good passwords
disallow = all ; Good practice dictates disallowing codecs first, and then allowing only the ones we
want
allow = g722 ; 16kHz at 64kbps
allow = ulaw ; 8kHz at 64kbps, North America
allow = alaw ; 8kHz at 64kbps, Worldwide
allow = g726 ; 8kHz at 32kbps
allow = g729 ; 8kHz at 8kbps - NOTE: This codec should not generally be enabled without installing
Digium's G.729 transcoding module for Asterisk
allow = slin ; 8kHz at 128kbps - NOTE: This codec should generally not be used outside of a LAN
allow = slin16 ; 16kHz at 256kbps - NOTE: This codec should generally not be used outside of a LAN
context = myfancycontext ; The context that incoming calls from this device will arrive into
mailbox = mydigiumphone@default ; The voicemail box associated with the Digium phone
```

Additionally, as Digium phones make use of the out-of-call messaging capabilities within Asterisk, certain modifications to the [general] section of Asterisk's SIP configuration file must be made as well:

1. Out of call messages must be accepted
2. The context for out of call messages should be "dpma_message_context"
3. Message Request authentication must be disabled

Thus, the settings are:

SIP General Section Requirements

```
accept_outofcall_messages = yes
outofcall_message_context = dpma_message_context
auth_message_requests = no
```

Additionally, use of the callcounter SIP configuration option is required for BLF state to properly operate. Thus:

Additional SIP General Section Requirements

```
callcounter=yes
```

callcounter may also be specified per-peer, instead of generally.

> 1.0.0-beta1 versions of res_digium_phone.so require that outofcall_message_context be set to **phone_context**. Versions after 1.0.0-beta1 default to dpma_message_context and have a corresponding configuration option in res_digium_phone.conf. If phone applications and provisioning are not working properly, please check the setting of outofcall_message_context.

Voicemail Configuration

All Digium Phones are provided with a **Msgs** hard button which calls the Voicemail application (visible in the list of Applications on the phone as well). When a Digium phone is not connected to the Digium Phone module for Asterisk, the Voicemail application simply dials a SIP URI, as configured, like any other SIP phone. However, if the Digium phone is connected to the DPMA and Asterisk is correctly configured, then the Msgs button will instead load a Visual Voicemail application that provides an enhanced user experience.

To configure Asterisk correctly, simply ensure that a mailbox entry in Asterisk's Voicemail configuration (voicemail.conf) matches the mailbox configuration parameter for a Digium Phone in res_digium_phone.conf, e.g.:

voicemail.conf

Voicemail Configuration Example for Voicemail Application

```
[fancycontext] ; The voicemail context name
100 => 12345,Bob Bobby,bobbobby@example.com ; The mailbox, its password, the person's name, and their
e-mail address
```

digium_phones.conf

Digium Phones Configuration Example for Voicemail Application

```
[MyPhone] ; The identifier
type=phone ; A phone
mailbox=100@fancycontext ; Sets mailbox to mailbox 100 in the fancycontext of voicemail.conf
; Other Phone options, not the following fake ones, go here
phoneoption=value
phoneoption2=othervalue
```

Parking

When used in combination with the DPMA, Digium phones provide both a parking application as well as one-touch parking. The parking application allows the phone to retrieve a list of all parking lots present on the Asterisk server, along with the calls that are currently parked in each lot. From this list then, a phone may retrieve any parked call. The one-touch parking feature is a softkey on the phone's display that appears when a call is connected. The softkey transfers the connected call (attended or blind) to whatever parking lot extension the phone is configured to use.

In attended transfer mode, once the parking operation is completed, Asterisk will play a prompt back to the parker, indicating the lot number to which the call was parked, and the phone will hang up. That lot number may then be dialed in order to retrieve the call, or one may use the phone's Parking application to browse and directly retrieve a parked call.

In blind transfer mode, the default, once a park is completed, the phone will display a text message on its screen, indicating the lot number in which the call was parked, and the phone will hang up. One may then use the phone's Parking application to browse and directly retrieve a parked call.

features.conf

Asterisk enables call parking by default with the features.conf parameters:

Features Configuration Example for Parking Application

```
parkext => 700 ; The extension to dial to park calls
parkpos => 701-720 ; The extensions onto which calls are parked
context => parkedcalls ; The default parking lot context
```

res_digium_phone.conf

The corresponding res_digium_phone.conf configuration parameters are:

Digium Phones Configuration Example for Parking Application

```
parking_exten=700 ; The extension to program the phone to dial when a call is parked using the park
softkey
parking_transfer_type=blind ; The type of parking to perform, blind or attended.
```

BLF Subscription to a Parking slot

BLF keys on phones are commonly tied to slots in Parking Lots, such that when a caller is waiting in a particular slot, e.g. 701, the lamp for a BLF tied to that parking slot is lit and the user may press the BLF button to retrieve the parked call from the lot.

Tying the status of a BLF lamp to the activity of a parking slot does not require setting the parking_exten, but it does require enabling Asterisk's parking feature, as well as a proper dialplan hint and a proper subscription URI on the Digium phone.

> If parking_exten is not configured for a Digium phone, then the DPMA will not enable the one-touch parking feature of the Digium phone.

Dialplan hint

An example dialplan hint for watching status of a parking slot is:

> Remember to include the context to which calls are parked into the same dialplan context as the Digium phone, as well as the Hint

Dialplan Example for Parking slot Hint

```
include => parkedcalls
exten => 701,hint,park:701@parkedcalls
```

Contacts subscribe_to URI

An example contacts XML file subscribe_to URI for watching parking slot 701:

Contacts Subscription URI for parking slot 701

```
subscribe_to="sip:701@10.24.13.224"
```

User Presence

When used with the DPMA, Digium Phones are capable of seeing both device status and user presence. Device status is simply the device state one can subscribe to over SIP SUBSCRIBE, that maps directly to a hint in the diaplan. User presence is an entirely new concept to Asterisk, and expands upon the usage of dialplan hints, allowing them to represent both device state and user presence at the same time. Digium Phones not connected to the DPMA are capable of only Available and DND (Phone returns 486 to Asterisk) status. Digium Phones using the DPMA are capable of much more, with a Status application that allows users to change their presence on the server, opening up new methods for call routing based on user-presence, and not merely device presence.

Defining User Presence in Asterisk

The fundamentals of how user presence is represented in Asterisk mirrors the concepts currently used with device state. Device state changes are triggered by device state providers.

Example Device State provider mapped to extension.

```
A hint for extension 1111 is mapped to the sip peer 1111 device state provider.  In this example,
endpoints subscribing to hint 1111 will receive a device state update anytime the device state changes
for sip peer 1111.

exten => 1111,hint,SIP/1111

exten => 1111,1,Dial(SIP/1111)
```

Using the same pattern, user presence is changed by a CustomPresence user presence provider. A CustomPresence provider works in the same way a Custom device state provider does. CustomPresence providers are both defined and updated using a dialplan function, PRESENCE_STATE().

Example Device State and Presence State providers mapped to a single extension.

```
A hint for extension 1111 is mapped to both the sip peer 1111 device state provider and the
CustomerPresence:1111 user presence provider.  Endpoints subscribing to hint 1111 will receive both
device state and user presence notifications for extension 1111.

exten => 1111,hint,SIP/1111,CustomPresence:1111

exten => 1111,1,Dial(SIP/1111)
```

Manipulating User Presence through Dialplan and AMI

PRESENCE_STATE() Dialplan Function

User presence information is modified through the use of the PRESENCE_STATE() dialplan function. This function allows a custom user presence provider's information to be both read and written via the dialplan and AMI.

Write Syntax

```
PRESENCE_STATE(<presence state provider>)=value[,subtype[,message[,options]]]
```

Valid State Values

```
"unavailable"

"available"

"away"

"xa"

"chat"

"dnd"
```

Valid Options

```
e: Both subtype and message fields are base64 encoded.  This is necessary for complex strings
containing commas and newline characters.  When this option is used, the PRESENCE_STATE function knows
it must first base64 decode the subtype and message fields before setting them on the CustomPresence
provider.
```

Read Syntax

```
STATE_VALUE = ${PRESENCE_STATE(<presence state provider>,field[,options])}
```

Valid read fields arguments

```
value
subtype
message
```

Valid read options

```
e: Base64 encode the return value when the field argument is subtype or message.
```

Dialplan Examples

Dialplan Write Examples

Example1: Set Batman's state to "Away" with the subtype "In the batcave" with the message, "Making a new batch of batarangs".

```
Set((PRESENCE_STATE(CustomPresence:Batman)=away,In the batcave, Making a new batch of batarangs)
```

Example2: Building on example1, now set Batman's state to "extended away" with no subtype while maintaining the message "Making a new batch of batarangs"

```
Set(PRESENCE_STATE(CustomPresence:Batman)=extended away,,Making a new batch of batarangs)
```

Example3: Setting the state as available without providing a subtype or message string. This will clear any previous message strings.

```
Set(PRESENCE_STATE(CustomPresence:Batman)=avaliable)
```

Example4: Set complex subtype and message strings using base64 encoding.

```
Set(PRESENCE_STATE(CustomPresence:Blah)=away,${BASE64_ENCODE(business)},${BASE64_ENCODE(I will
visiting clients in the San Diego area.\nI will be returning on Oct 11th.\nCall Josh for
emergencies)})
```

Dialplan Read Examples

Read Examples

```
SUBTYPE = ${PRESENCE_STATE(<presence state provider>,subtype)}
MESSAGE = ${PRESENCE_STATE(<presence state provider>,message)}
BASE64_SUBTYPE = ${PRESENCE_STATE(<presence state provider>,subtype,e)}
BASE64_MESSAGE = ${PRESENCE_STATE(<presence state provider>,message,e)}
```

AMI Examples

Example1: Setting both the user state and message using SetVar action in conjunction with PRESENCE_STATE() dialplan function.

```
Action: Setvar
ActionID:1234
Variable: PRESENCE_STATE(CustomPresence:Batman)
Value:away,In the batcave, Making a new batch of batarangs
```

Example2: Setting state information that must be base64 encoded because it contains newlines and/or commas.

```
Action: Setvar
ActionID:1234
Variable: PRESENCE_STATE(CustomPresence:Batman)
Value:away,ISDW982KLJ90==,20DJKL23JK==,e
```

Example3: Reading user state and message using GetVar action.

```
Action: Getvar
ActionID:1234
Variable: PRESENCE_STATE(CustomPresence:Batman,value)

Action: Getvar
ActionID:1234
Variable: PRESENCE_STATE(CustomPresence:Batman,subtype)

Action: Getvar
ActionID:1234
Variable: PRESENCE_STATE(CustomPresence:Batman,message)
```

Example4: Reading subtype and message fields as base64 values.

```
Action: Getvar
ActionID:1234
Variable: PRESENCE_STATE(CustomPresence:Batman,subtype,e)

Action: Getvar
ActionID:1234
Variable: PRESENCE_STATE(CustomPresence:Batman,message,e)
```

User Presence in the DPMA

The DPMA does all of the user presence manipulation of the CustomPresence providers behind the scenes. Phones subscribe to a set of user extensions to receive both device state and user presence updates. The DPMA is in change of defining the hints the phones subscribe to, and mapping those hints to the correct device state and presence state providers. When a phone user updates their user presence, the DPMA internally updates that user's CustomPresence provider to reflect the change using the PRESENCE_STATE() dialplan function. This results in any watcher of the hint mapped to that CustomPresence provider receiving an update indicating the new user presence.

...

v20120424.1