KIRK Wireless Server 300 & 6000

Provisioning

Version 2.2

Table of content

1. Introduction

The purpose of provisioning is to simplify mass configuration and deployment. With provisioning it is possible to have a centralized repository for handling firmware, configuration files and user lists for all devices.

1.1 Document History

Date	Version	Description
2009-04-24	1.0	First Revision
2009-09-22	2.0	Restructure of document and added configuration parameters for KWS300 PCS04__ & KWS6000 PCS04__
2010-03-31	2.2	Added note about TFTP and path. Changed KWS300 from static to DHCP assigned IP address.

2. Provisioning Architecture

The KWS300 and KWS6000 use a common method for provisioning. If a provisioning server is specified (either statically or through a DHCP-server), the KWS contacts the provisioning server to check/update its firmware, configuration and user list.

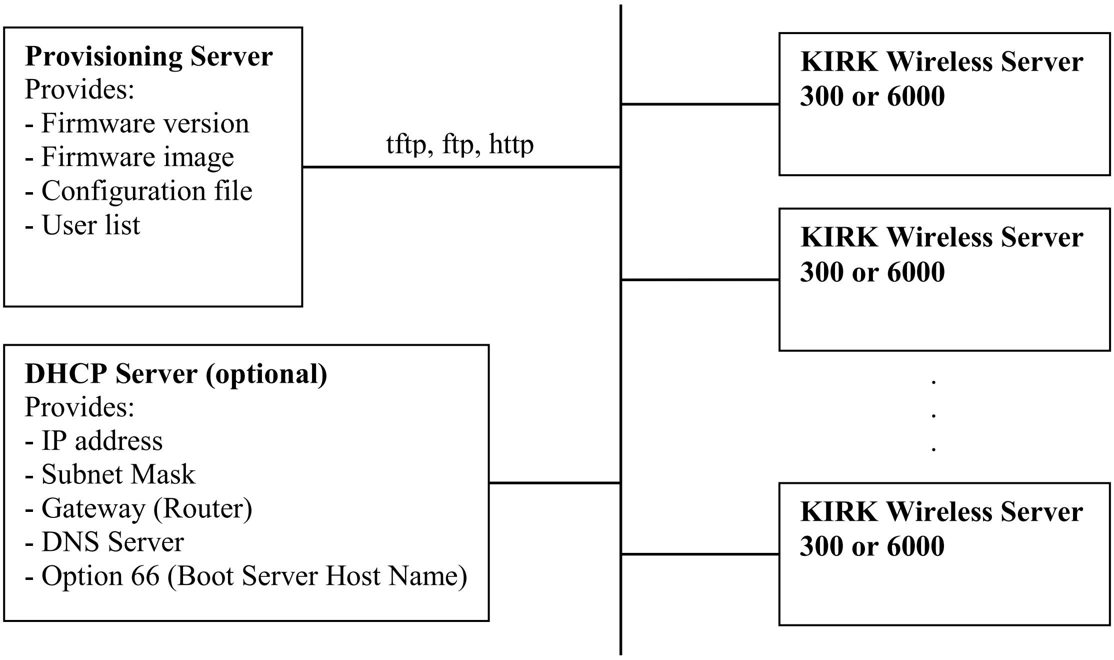


Figure 1: KWS300 and KWS6000 provisioning architecture.

2.1 DHCP Server

When using DHCP, option 66 (TFTP server name) can be used to provide the provisioning server URL. This is a string type option configured on the DHCP server of the network.

2.2 Provisioning Server

A central provisioning server keeps firmware and configuration files for the devices. The firmware and configuration is pulled from the provisioning server by the devices using FTP, TFTP or HTTP. The provisioning server provides the following files to the KWS:

File	Description
Firmwarefilename.bin e.g. kws300firmware.bin or kws6000firmware.bin	A binary file containing the firmware image. The filename is defined in the XML configuration file or entered in the Provisioning -> Firmware -> KWS field in the web GUI

Firmwarefilename.bin.ver e.g. kws300firmware.bin.ver or kws6000firmware.bin.ver	A text file with text describing current firmware version. For example "PCS03 18860". This file must have the name of the binary firmware file with .ver appended. The .ver file is included in the firmware package
<KWS MAC address>-config.xml e.g. 0013d1800032-config.xml	A XML formatted file containing the configuration – see APPENDIX B – Configuration file example
<KWS MAC address>- users.xml e.g. 0013d1800032-users.xml	A XML formatted file containing User data – see APPENDIX D – Users XML Example

2.3 XML File Formatting

XML is a standard format and a few things must be considered when creating and editing XML files.

The character encoding of the XML file must match the character encoding specified in the encoding attribute of the header in the file. Below is a snippet of XML where the header specifies the encoding as UTF-8:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<config>
...
</config>
```

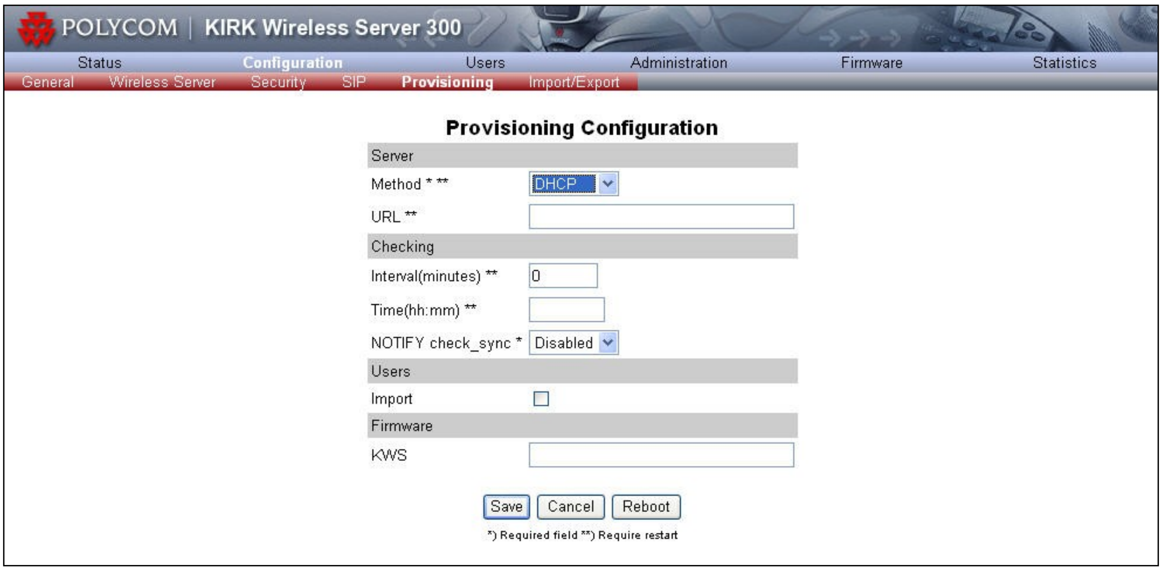
If no encoding is specified in the header the default format is UTF-8. If the encoding does not match international characters (æ, ö and more) may be obscured or the file cannot be parsed.

In order to avoid problems with importing the XML files it is advisable to validate the XML with an external tool before using the files.

Example: If a users XML file is exported from the KWS it is encoded in UTF-8. If the exported file is edited with an editor that is not encoding aware (E.g. Windows Notepad) and international letters are added or changed these may be saved in Windows-1252 encoding. The KWS will not be able to import the file because the encoding of the characters does not match the encoding of the file.

3. Setting up provisioning on KWS

3.1 Setting up Access to Provisioning Server



Provisioning Configuration

Server	
Method * **	DHCP
URL **	
Checking	
Interval(minutes) **	0
Time(hh:mm) **	
NOTIFY check_sync *	Disabled
Users	
Import	□
Firmware	
KWS	

*) Required field **) Require restart

Figure 2: Screen dump from the KWS300 Configuration -> Provisioning Page.

The KWS needs to know the protocol and address of the provisioning server containing firmware, configuration and user-data. This information is handled as an URL in the format:

[<protocol>://[<username>:<password>@]]<host>[/<path>]

Note: The **[/<path>]** part of the URL is not supported for TFTP.

Examples:

- **10.0.0.10** ;tftp used as default protocol
- **tftp://provisioning.test.com**
- **ftp://192.168.0.1**
- **ftp://user:password@provisioning.example.com**
- **http://server.example.com/boot**.

The URL can be obtained through the configuration file or through DHCP.

The KWS can use the following methods to obtain the provisioning server URL:

- Disabled (The KWS will not use provisioning)
- Static (The administrator must manually specify the URL of the provisioning server)
- DHCP Option 66 (default)

If no provisioning server is configured or obtained, the KWS will not use auto provisioning.

Three protocols are available for downloading firmware and configuration: TFTP, FTP and HTTP. All of the protocols are available at the target and no additional software is required. Which protocol to use is specified through the provisioning server's URL.

3.2 Automatic check for new Firmware, Configuration and User data

When new firmware, configuration or user data files are made available at the provisioning server, the KWS must download it. The KWS needs to know when new data is available. Two methods are supplied for this: Periodic polling and SIP notifications.

3.2.1 Polling

When polling is selected the server will initiate a check for updates by itself. The check will be performed at a specified interval or at a specific time of day.

3.2.2 SIP NOTIFY check-sync

The preferred way to handle updates is by notifying the KWS that updates are available. This is with the SIP NOTIFY method and the event "check-sync". A check-sync event is sent to one of extensions/usernames handled by the KWS and when it is received the KWS initiates a check updates. This is illustrated in

Figure 3.

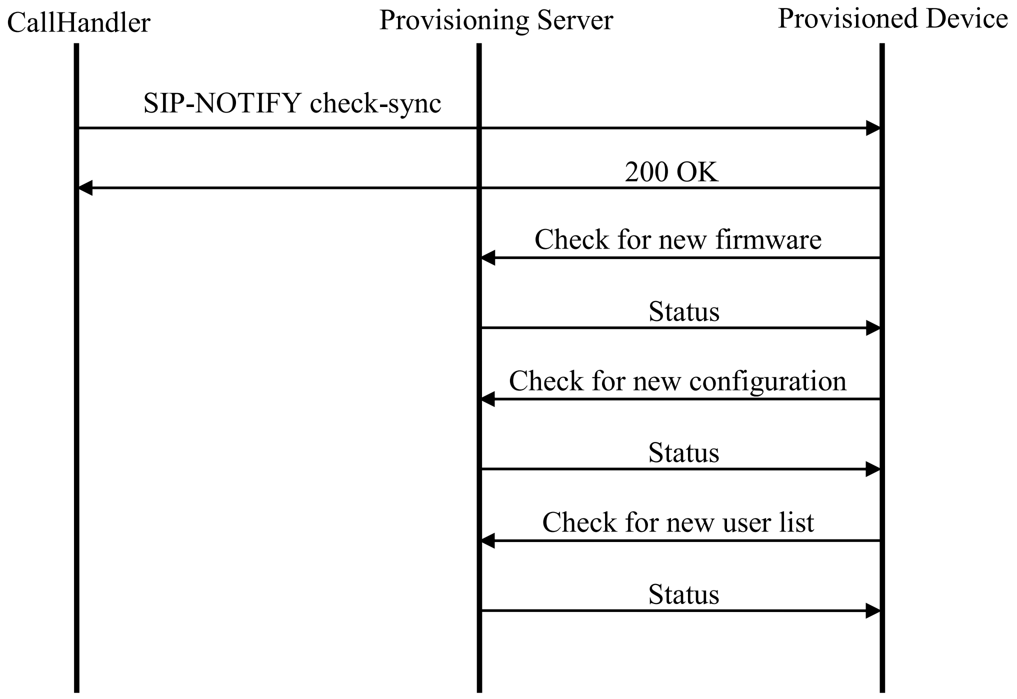


Figure 3: Receiving SIP NOTIFY check-sync.

3.3 Updating the Firmware, Configuration Data or User Data

The KWS will be able to automatically download firmware, configuration and users from the provisioning server. This section specifies how this is done.

3.3.1 Firmware update

The firmware will be stored as a file at the provisioning server. Together with the firmware file a firmware version file will be stored. This file is downloaded to determine the version of the firmware without actually downloading the firmware file, to keep the network load at a minimum.

For flexibility, the name of the firmware file is stored in the XML configuration.

Below is an illustration of the firmware update process:

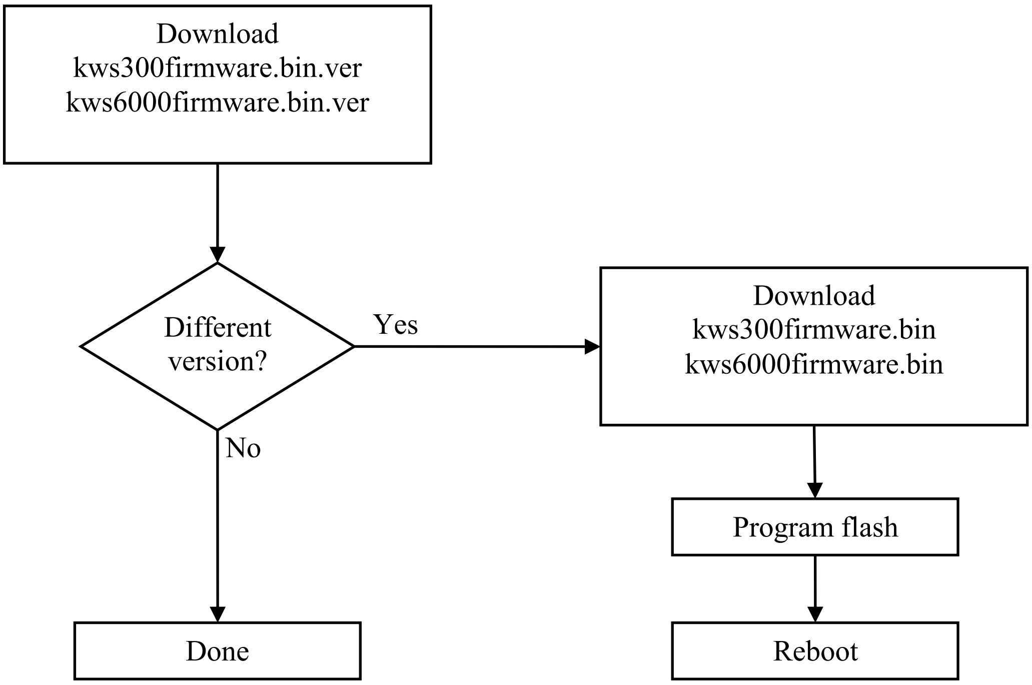


Figure 4: Firmware update process

The firmware version specified in the ".ver" file is compared with the current active firmware version (the version that is currently running). Immediately after the firmware is updated in the flash the KWS is rebooted to make the (updated) firmware residing in the flash become the active firmware.

3.3.2 Configuration update

Figure 5 illustrates the configuration update process.

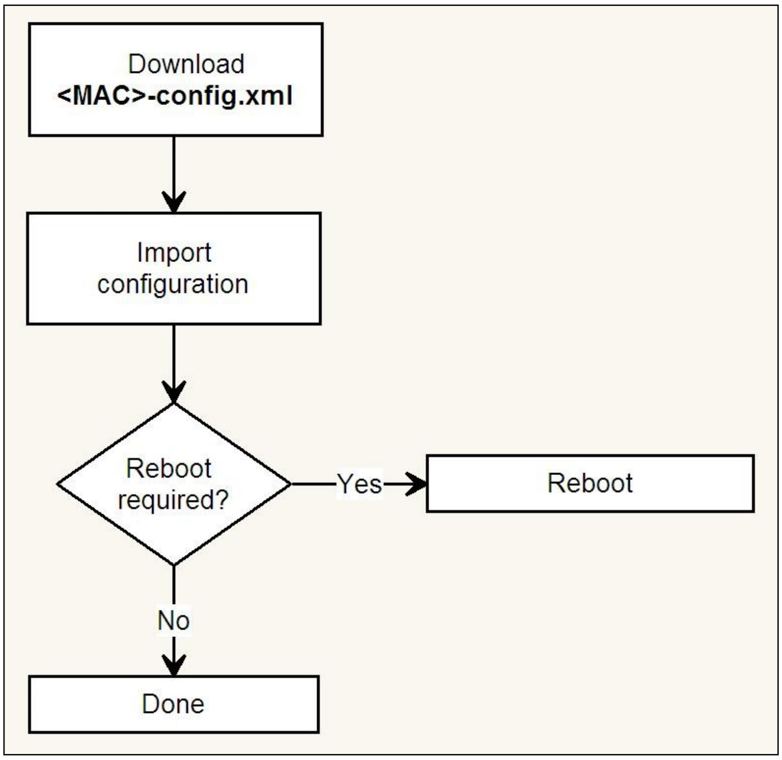


Figure 5: Configuration update process

The XML configuration file is downloaded and imported into the KWS configuration by replacing the existing data. This secures that data located on the provisioning server and at the KWS are identical.

The <MAC>-config.xml replaces the internal configuration. This way the configuration is 100% controlled by the provisioning server.

Be aware: The network configuration of the device is a part of the config.xml. If the network configuration is invalid/missing the device will not be able to boot and a reset to defaults is required. Here is an example of a sufficient network configuration for DHCP:

```
<network>
  <bootproto>dhcp</bootproto>
</network>
```

This way it is not necessary to configure the network configuration in the provisioning.

3.3.3 User list update

The users are stored in a separate “<MAC>-users.xml” file. In an existing KWS installation, the user list file can be retrieved by clicking **Users** -> **Import/Export** -> **Save XML format**.
Each record must as a minimum have a username field.

Changes in the “<MAC>-users.xml” file do not require a reboot of the system.

4. APPENDIX A – Configuration file parameter description

Parameter	Description	Values
DECT		
config.dect.auto_create_users	Adds a new user when a DECT handset tries to subscribe to the system.	true – auto create users false – disabled **Default: false**
config.dect.send_date_time	Specifies if the date and time will be sent to the handsets.	true – send date & time false – do not send date & time **Default: true**
config.dect.subscription_allowed	Specifies if handset subscription is allowed.	true – subscription allowed false – subscription disallowed **Default: true**
config.dect.auth_call	Specifies if DECT authentication should be used when establishing calls.	true – DECT authentication is required when establishing calls. false – DECT authentication of calls is disabled. **Default: true**
config.dect.encrypt_voice_data	Specifies if DECT encryption should be used for voice calls.	Disabled – DECT encryption is disabled. Enabled – DECT encryption is enabled. Enforced – DECT encryption is enforced and calls are terminated if the handset do not support encryption.
Network		
config.network.bootproto	Specifies if the IP configuration is provided by DHCP or static	dhcp – get IP config using DHCP static – the IP config is statically defined **Default: static** **KWS300:** Starting with devices produced with firmware release PCS05B_, the default is **DHCP**. If DHCP is not available, it will fall back to the configured IP address.
config.network.dns1	Specifies the Primary DNS	**Default: Empty**
config.network.dns2	Specifies the secondary DNS	**Default: Empty**

config.network.domain	Specifies the name of the domain the system belongs to.	**Default: Empty**
config.network.gateway	Specifies the IP address of the default gateway.	**Default: Empty**
config.network.ipaddr	Specifies the IP address of the system.	**Default: 192.168.0.1**
config.network.mtu	Specifies the Maximum Transmission Unit.	**Default: Empty**
config.network.netmask	Specifies the network mask.	**Default: 255.255.255.0**
config.network.ntp	Specifies the address of the NTP server.	**Default: Empty**
config.network.timezone	Specifies the time zone.	**Default: CET-1CEST-2,M3.5.0/02:00:00,M10.5.0/03:00:00**
Provisioning		
config.provisioning.check.check_sync	Specifies how the KWS will react to SIP NOTIFY check-sync events.	**disabled** – do not react. **reboot** – reboot and check for updates. **update** – check for updates and reboot if necessary. **Default: disabled**
config.provisioning.check.interval	Specifies an interval for checking for updates.	**0** – do not check for updates periodically. **>= 1** – interval in minutes. **Default: 0**
config.provisioning.check.time	Specifies a specific time for checking each day. The format is HH:MM	00:00 – 23:59 **Default: Empty**
config.provisioning.server.method	Specifies how the KWS will obtain the provisioning server address.	Specifies how the KWS will obtain the provisioning server address. **dhcp** – obtain from DHCP option 66. **static** – use static configured. **disabled** – do not check for updates. **Default: dhcp**

config.provisioning.server.url	Specifies the static provisioning server URL from where the KWS will retrieve configuration information. The format is [<protocol>://[<user>:<password>@]]<host>[/<path>]. Protocol can be either tftp, ftp or http. It is optional to specify a protocol. If the protocol is not specified the KWS will default to tftp. **Note:** The [/<path>] part of the URL is not supported for TFTP.	Example: ftp://kws:ip6000@boot.example.com/phones or 192.168.0.1 **Default: Empty**
config.provisioning.firmware.kws	Specifies the name of the firmware image to use for the KWS. The KWS will check for a version file and a binary file. They must be located as <URL>/<firmware>.ver and <URL>/<firmware>	Example: kws300-flash.bin **Default: Empty**
config.provisioning.users.check	Specifies if the users list should be loaded.	true – load users list false – do not load the user list **Default: Empty**
Security		
config.security.force_https	Specifies if the system should enforce remote access security using HTTPS (TLS).	true – force HTTPS (TLS) false – use HTTP **Default: false**
config.security.username	Username for the user who log on to the web GUI.	**Default: admin**
config.security.password	Encrypted password for the user who log on to the web GUI.	**Default KWS300: kws300 (encrypted)** **Default KWS6000: ip6000 (encrypted)**
SIP		
config.sip.auth.password	Specifies the default password for the handset authentication (if no handset specific authentication password is specified).	**Default: Empty**
config.sip.auth.username	Specifies the default username for the handset authentication (if no handset specific authentication username is specified).	**Default: Empty**
config.sip.defaultdomain	Specifies the default domain for the handset (if no handset specific domain is specified).	**Default: Empty**

config.sip.dtmf.duration	Specifies the length of the DTMF tones in milliseconds.	**Default: 270**
config.sip.dtmf.info	Specifies if the keypad signaling should be sent as SIP INFO.	true – send as SIP INFO false – do not send as SIP INFO **Default: false**
config.sip.dtmf.rtp	Specifies if the keypad signaling should be sent as RTP packets with DTMF code.	true – send as RTP false – no not send as RTP **Default: true**
config.sip.dtmf.rtp_payload_type	Specifies the payload type for RFC2833 in SDP offers.	**Default: 96**
config.sip.localport	Specifies the SIP port.	**Default: 5060**
config.sip.maxforwards	Specifies the maximum number of times the SIP messages can be forwarded through proxies.	**Default: 70**
config.sip.media.codecs	Specifies the codec priority.	**Default:** **1,2 (for KWS300)** **64,1,2,0,0,0 (for KWS6000)**
config.sip.media.port	Specifies the start port for media.	**Default: 58000**
config.sip.media.ptime	Specifies the packet duration for media (milliseconds).	**Default: 20**
config.sip.media.symmetric	Specifies if the KWS should require symmetric RTP for the media.	true – require symmetric RTP false – do not require symmetric RTP **Default: true**
config.sip.media.tos	Specifies the media's TOS/Diffserv.	**Default: 184**
config.sip.mwi.enable	Enables the MWI (Message Waiting Indicator).	true – MWI enabled false – MWI disabled **Default: true**
config.sip.mwi.expire	Specifies the MWI subscription's expiration time (seconds).	**Default: 3600**
config.sip.mwi.subscribe	Enables MWI subscription.	true – MWI subscription enabled false – MWI subscription disabled **Default: false**
config.sip.onholdtone	Specifies if the handset should play the on-hold tone when put on-hold.	true – on-hold tone enabled false – on-hold-tone disabled **Default: true**

config.sip.pound_dials_overlap	Specifies if '#' should dial in overlap dialing.	true – '#' dials in overlap dialing false – '#' does not dial in overlap dialing **Default: false**
config.sip.proxy.domain config.sip.proxy.domain[2-4]	Specifies addresses for SIP proxies.	**Default: Empty**
config.sip.proxy.port config.sip.proxy.port[2-4]	Specifies the port for SIP proxies.	**Default: Empty**
config.sip.proxy.priority config.sip.proxy.priority[2-4]	Specifies the priority for using a SIP proxy. Proxies with lowest priority will be preferred and higher priorities will be used for failover.	Values: 1-4 **Default: 1,2,3,4**
config.sip.proxy.weight config.sip.proxy.weight[2-4]	Specifies the weight for using a proxy. If more proxies have the same priority the KWS will do load balancing using the weight to determine how much each proxy will be loaded.	Values: 0-100 **Default: 100**
config.sip.proxy.transport	Specifies the transport mechanism used for SIP messages.	UDPonly – use UDP and simple DNS for resolving IP addresses DNSsrv – use UDP and DNSSrv for resolving IP addresses **Default: UDPonly**
config.sip.registration_expire	Specifies the number of seconds before a SIP registration will be renewed.	**Default: 3600**
config.sip.send_to_current_registrar	Specifies if the system should send all messages to the current registrar.	true – sends all messages to current registrar false – does not send all messages to current registrar **Default: false**
config.sip.separate_endpoint_ports	Specifies if the endpoints should register on separate ports.	true – register endpoints on separate ports false – don't register endpoints on separate ports **Default: false**
config.sip.showstatustext	Show the information for the call status in the handset display (ring, hold etc)	true: Show text false: Text is not shown **Default: true**
config.sip.tos	Specifies the SIP TOS/Diffserv	**Default: 96**

Phonebook		
config.phonebook.encoding	Specifies the character encoding of the imported CSV file.	utf-8 iso8859-1 windows-1252 **Default - utf-8**
config.phonebook.ldap_attributes	The LDAP attributes to retrieve and use.	Relevant attributes provided by the LDAP server.
config.phonebook.ldap_base	The base path where the users are located in the LDAP structure.	Base path from LDAP server.
config.phonebook.ldap_bind_password	Password used to login to the LDAP server.	Valid LDAP password.
config.phonebook.ldap_bind_user	Username used to login to the LDAP server.	Valid LDAP user name.
config.phonebook.ldap_filter	The filter used for the LDAP query. The (objectClass= person) filter can be used successfully in most cases.	A valid LDAP filter.
config.phonebook.ldap_names	The attribute names assigned to the Attributes specified, separated by a comma.	Text strings.
config.phonebook.ldap_prefixes	The phone number prefixes to strip, separated by a comma. For example, if the phone number is +45678912345, and that user has the 12345 extension, then "+456789" is specified in the Strip prefixes field.	Phone number to strip.
config.phonebook.ldap_refresh_interval	The interval in seconds for querying the LDAP server for updates.	A number of seconds.
config.phonebook.ldap_uri	The URI of the LDAP server.	A valid LDAP URI.
config.phonebook.source	The source of the phone book data.	disabled – do not enable the phonebook. csv – import phone book from CSV file. ldap – query LDAP server for phone book data.
Application		

config.application.enable_msf	Specifies if the MSF application interface is enabled.	true – The MSF interface is enabled and applications can connect. false – The MSF interface is disabled. **Default: true**
config.application.username	Specifies the username required for applications to log in.	**Default: "GW-DECT/admin"**
config.application.password	Specifies the encrypted password required for applications to log in.	**Default:** "f621c2268a8df24955ef4052bfbb80cf" (password "ip6000" encrypted)
config.application.enable_rpc	Specifies if the XML-RPC application interface is enabled.	true – The XML-RPC interface is enabled and applications can connect. false – The XML-RPC interface is disabled. **Default: false**
UPnP		
config.upnp.enable	Specifies if UPnP support is enabled. If enabled the device will respond to UPnP broadcasts.	Values: true/false **Default: true**
config.upnp.broadcast	Specifies if UPnP announcements are broadcasted. If enabled the device will periodically broadcast announcements.	Values: true/false **Default: false**

5. APPENDIX B – Configuration file example

```
<?xml version="1.0" standalone="yes" ?>
<config>
  <dect>
    <auto_create_users>true</auto_create_users>
    <send_date_time>true</send_date_time>
    <subscription_allowed>true</subscription_allowed>
  </dect>
  <media_resource>
    <enabled>true</enabled>
  </media_resource>
  <network>
    <bootproto>static</bootproto>
    <dns1>192.168.0.5</dns1>
    <domain>example.com</domain>
    <gateway>192.168.0.1</gateway>
    <ipaddr>192.168.0.100</ipaddr>
    <mtu>0</mtu>
    <netmask>255.255.240.0</netmask>
    <ntp>192.168.0.5</ntp>
    <timezone>GMT-1</timezone>
  </network>
  <phonebook>
    <encoding>utf-8</encoding>
    <ldap_attributes>displayName, telephoneNumber</ldap_attributes>
    <ldap_base>OU=Users,OU=Site,DC=example,DC=company,DC=com</ldap_base>
    <ldap_bind_password>XXXX_XXXX</ldap_bind_password>
    <ldap_bind_user>someone</ldap_bind_user>
    <ldap_filter>(objectClass=person)</ldap_filter>
    <ldap_names>Name, Phone</ldap_names>
    <ldap_prefixes>+4576281,76281,+45</ldap_prefixes>
    <ldap_refresh_interval>3600</ldap_refresh_interval>
    <ldap_uri>ldap://ldap.example.com</ldap_uri>
    <source>ldap</source>
  </phonebook>
  <security>
    <force_https>false</force_https>
    <password>XXXXXXXXXXXXXXXXXXXXXXXX</password>
    <username>admin</username>
  </security>
  <sip>
    <auth>
      <password>1234</password>
      <username>someone</username>
    </auth>
    <defaultdomain>example.com</defaultdomain>
    <dtmf>
      <duration>270</duration>
      <info>false</info>
      <rtp>true</rtp>
      <rtp_payload_type>96</rtp_payload_type>
      <rtp_payloadtype>96</rtp_payloadtype>
    </dtmf>
    <localport>5060</localport>
    <maxforwards>70</maxforwards>
    <media>
      <codecs>1,2,0,0,0,0</codecs>
      <port>58000</port>
      <ptime>20</ptime>
      <symmetric>true</symmetric>
      <tos>0</tos>
    </media>
    <mwi>
      <enable>true</enable>
      <expire>3600</expire>
      <subscribe>false</subscribe>
    </mwi>
    <onholdtone>true</onholdtone>
    <pound_dials_overlap>true</pound_dials_overlap>
    <proxy>
      <domain>sip.example.com</domain>
      <port>5060</port>
      <transport>UDPonly</transport>
    </proxy>
    <registration_expire>3600</registration_expire>
    <send_to_current_registrar>false</send_to_current_registrar>
    <separate_endpoint_ports>false</separate_endpoint_ports>
    <showstatustext>true</showstatustext>
    <tos>0</tos>
  </sip>
</config>
```

6. APPENDIX C – Users XML Parameters

Parameter	Description	Values
users.user.ipei	The DECT IPEI of the users handset	A valid IPEI in the format XXXXX XXXXXXX or empty.
users.user.accesscode	Access code required for subscribing the handset to the system.	A number with 0-8 digits.
users.user.standbytext	The text displayed in the handset when idle.	A text string.
users.user.username	The user name / extension used when communicating with the SIP server	A valid SIP user name. This field is required.
users.user.domain	SIP domain for the user. Used if the user will have a different domain than the system default.	A valid domain name.
users.user.displayname	The display name send with SIP requests.	A valid SIP display name.
users.user.authuser	User name for authenticating the user.	A valid SIP authentication user name.
users.user.authpassword	Password for authenticating the user.	A valid SIP password.
users.user.disabled	Indicates if the user is disabled and unable to make calls.	true – user is disable. false – user is enabled.

7. APPENDIX D – Users XML Example

```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<users>
  <user>
    <ipei>00077 0000001</ipei>
    <accesscode></accesscode>
    <standbytext>2639</standbytext>
    <username>2639</username>
    <domain></domain>
    <displayname>Jenny Smith</displayname>
    <authuser>2639</authuser>
    <authpassword>1234</authpassword>
    <disabled>true</disabled>
  </user>
  <user>
    <ipei>00077 0000002</ipei>
    <accesscode></accesscode>
    <standbytext>2638</standbytext>
    <username>2638</username>
    <domain></domain>
    <displayname>John Smith</displayname>
    <authuser>2638</authuser>
    <authpassword>1234</authpassword>
    <disabled>true</disabled>
  </user>
</users>
```