Technical Manual

3CX Phone System for Windows

This technical manual is intended for those who wish to troubleshoot issues encountered with implementing 3CX Phone System. It is not intended to replace the manual, rather it explains in some detail how technical problems can be analyzed, and trouble-shooting procedures applied.

http://www.3cx.com

E-mail: info@3cx.com

Version 1.0 – Last updated 14 June 2007

Table of Contents

Introduction to SIP

Introduction

This chapter gives a brief outline of the SIP protocol, on which 3CX Phone System is based.

Background information on SIP & RTP

All SIP Phone calls consist of 2 channels:

- Messaging – SIP
- Media - RTP

Messaging, carried using UDP or TCP (normally UDP) SIP packets – used for device registration, call setup/teardown, on-hold notification, call transfers, DTMF signals. Typically the PBX Server is expecting to receive SIP messages on port 5060, so the PBX Server must be able to receive these packets, requiring accurate firewall configuration to allow this. Typically a SIP Phone will first register with the PBX Server with a SIP message to port 5060 on the PBX Server, indicating inside this registration message the port number which it is expecting to receive calls on. When the PBX Server needs to forward a call to the SIP Phone, it will start the conversation with a SIP message to the port specified by the SIP Phone in the registration message.

Media, typically audio, carried using UDP RTP packets – used to send the actual media for the call, including audio, video, DTMF signals (as a special media type, or embedded in-audio).

A SIP INVITE message contains typically between 4 and 6 header entries with contact information inside them. Different devices or providers use these headers in different ways and therefore, an understanding of the “normal” scope of these headers, together with a close analysis of log messages, will help to understand the cause of certain issues, and also to configure devices or providers in the 3CX Management Console so as to be identified correctly.

SIP Messages Listing

There are essentially 2 message types:

1. Request messages, submitted using one of the SIP Methods listed below.

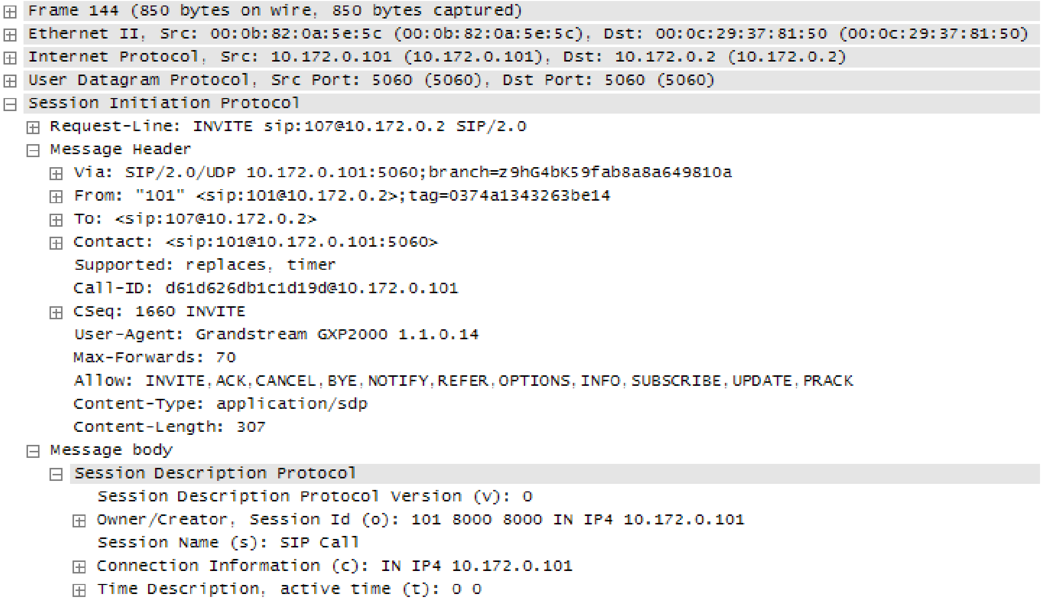
ACK	Used to reply to a SIP Status message in the range 200-699 while in a SIP INVITE dialog.
BYE	Used to end an already-established session, such as a SIP INVITE-based call.
CANCEL	Used to cancel a request (such as an INVITE or call request that is still in progress but unanswered)
INFO	Used to provide additional signaling, not necessarily related to a call in progress. SIP INFO Messages are typically used to provide notification of DTMF strokes (even though RFC 2833 type notifications are preferred), and is also one method used to establish MWI (message waiting indicator) functionality to SIP phones.

INVITE	Used to initiate a session dialog – typically to set up a phone call
NOTIFY	Used by the PBX Server to communicate statistics, typically to a SIP phone. Transferring of one leg of a call from one endpoint to another may be signaled using this SIP Request, within the context of a SIP REFER dialog. MWI information can also be delivered in this way. Normally a SIP Phone would have previously established a SIP dialog using a SIP SUBSCRIBE request, and the SIP NOTIFY messages would be delivered within the context of this dialog. Note however that some SIP Phones do not establish a dialog in this way, and expect to receive SIP NOTIFY messages "out-of-dialog". Note also that some SIP Phones may attempt to establish a SIP dialog with SIP SUBSCRIBE using incorrect methodology (GrandStream GXP-2000 being a typical example), and working (if not entirely correct) functionality can be achieved with such devices if they can be configured to NOT attempt a SIP SUBSCRIBE.
OPTIONS	Used by a SIP client to query another SIP client or SIP proxy (such as the 3CX PBX Server) about its capabilities to discover information about the supported methods, content types, extensions, codecs, and so on, prior to, for example, establishing a call using the SIP INVITE method.
PRACK	Not used in 3CX Phone System
REFER	Used by a SIP client to subscribe for notification of changes in call flow, for instance. Typically seen when a phone requests to transfer a call to another party.
REGISTER	Used to register or unregister a SIP user-agent with a SIP registrar. An unREGISTERed phone CANNOT receive a SIP INVITE Request to receive a call.
SUBSCRIBE	Used by a SIP Phone to establish a SIP dialog for receiving statistics. A typical example is MWI information for the phone's configured Extension Number.
UPDATE	Not used in 3CX Phone System

2. Status messages, submitted with a numeric status indicator.

100-199	Used to indicate a temporary status, such as "100 Trying" or "180 Ringing"
200-299	Used to indicate a final success status, such as "200 OK"
300-399	Not used in 3CX Phone System. Used to indicate a final failure status, but with information about the user's new location, or about alternative services that might be able to answer the call.
400-499	Used to indicate a final error status, but local to the SIP Proxy generating this message – indicating that it may be possible that some other SIP Proxy may be in a position to successfully handle the request.
500-599	Used to indicate that the PBX Server has encountered an internal error. Examples are "500 Server Internal Error" and "501 Not Implemented"
600-699	Used to indicate a final error status, and that this status is global in nature – indicating that the request will in no case be able to be handled by some other SIP Proxy.

A Brief description of the main SIP INVITE Header Fields



```
⊞ Frame 144 (850 bytes on wire, 850 bytes captured)
⊞ Ethernet II, Src: 00:0b:82:0a:5e:5c (00:0b:82:0a:5e:5c), Dst: 00:0c:29:37:81:50 (00:0c:29:37:81:50)
⊞ Internet Protocol, Src: 10.172.0.101 (10.172.0.101), Dst: 10.172.0.2 (10.172.0.2)
⊞ User Datagram Protocol, Src Port: 5060 (5060), Dst Port: 5060 (5060)
⊟ Session Initiation Protocol
  ⊞ Request-Line: INVITE sip:107@10.172.0.2 SIP/2.0
  ⊟ Message Header
    ⊞ Via: SIP/2.0/UDP 10.172.0.101:5060;branch=z9hG4bK59fab8a8a649810a
    ⊞ From: "101" <sip:101@10.172.0.2>;tag=0374a1343263be14
    ⊞ To: <sip:107@10.172.0.2>
    ⊞ Contact: <sip:101@10.172.0.101:5060>
      Supported: replaces, timer
      Call-ID: d61d626db1c1d19d@10.172.0.101
    ⊞ CSeq: 1660 INVITE
      User-Agent: Grandstream GXP2000 1.1.0.14
      Max-Forwards: 70
      Allow: INVITE,ACK,CANCEL,BYE,NOTIFY,REFER,OPTIONS,INFO,SUBSCRIBE,UPDATE,PRACK
      Content-Type: application/sdp
      Content-Length: 307
  ⊟ Message body
    ⊟ Session Description Protocol
        Session Description Protocol Version (v): 0
      ⊞ Owner/Creator, Session Id (o): 101 8000 8000 IN IP4 10.172.0.101
        Session Name (s): SIP Call
      ⊞ Connection Information (c): IN IP4 10.172.0.101
      ⊞ Time Description, active time (t): 0 0
```

Definitions

SIP URI - A SIP URI is a user’s SIP phone number. The SIP URI resembles an e-mail address and is written in the following format: SIP URI = sip:x@y:Port

Request-Line-URI:

The Request-Line-URI includes the destination of the call. It contains the same information as the To field, omitting the display name.

Via:

Every proxy in the request path adds to top of the “Via” the address and port on which it received the message, than forwards it onwards.

When processing responses, each proxy in the return path processes the contents of the “Via” field in reverse order, removing its address from the top.

From:

The “From” header field indicates the identity of the initiator of the request from the point of view of the PBX Server – similar in construction to email addresses (user@domain – where “user” is, for example, the extension number, and “domain” is the server domain or IP address). Like the “To” header field, it contains a URI and optionally a display name. It is used to determine which processing rules to apply to a request.

From the SIP RFC 3261: The “From” header field allows for a display name. A UAC (i.e a phone) SHOULD use the display name "Anonymous", along with a syntactically correct, but otherwise meaningless URI (e.g. From: “Anonymous” <sip:10000@10.172.0.2>), if the identity of the client is to remain hidden.

Typically, the host IP Address will be the internal IP address of the PBX Server.

To:

The “To” header field first and foremost specifies the desired "logical" recipient of the request, or the address-of-record of the user or resource that is the target of this request. This may or may not be the ultimate recipient of the request. The “To” header field MAY contain a SIP URI, but it may also make use of other URI schemes (the tel URL (RFC 2806 [9]), for example) when appropriate. All SIP implementations MUST support the SIP URI scheme.

The To header field allows for a display name (e.g. To: “Target User” <sip:101@10.172.0.2>).

Typically, the “To” field contains a SIP URI that points to the first (next) hop (proxy) that will process the request, but not necessarily the SIP URI of the eventual recipient.

Contact:

The “Contact” header field provides a single SIP URI that can be used to contact the sender of the INVITE for subsequent requests. The Contact header field MUST be present and contain exactly one SIP URI in any request that can result in the establishment of a dialog – in this case, specifically a SIP INVITE. For these requests, the scope of the Contact is global. That is, the Contact header field value contains the URI at which the sender is expecting to receive requests, and this URI MUST be valid even if used in subsequent requests outside of any dialogs (in our case, meaning established calls).

Allow:

This field lists, in comma-separated format, the SIP Methods that the caller can support and use. SIP defines the following methods: ACK, BYE, CANCEL, INFO, INVITE, NOTIFY, OPTIONS, PRACK, REFER, REGISTER, SUBSCRIBE, UPDATE

Analyzing a successful call setup between extensions within 3CX

Introduction

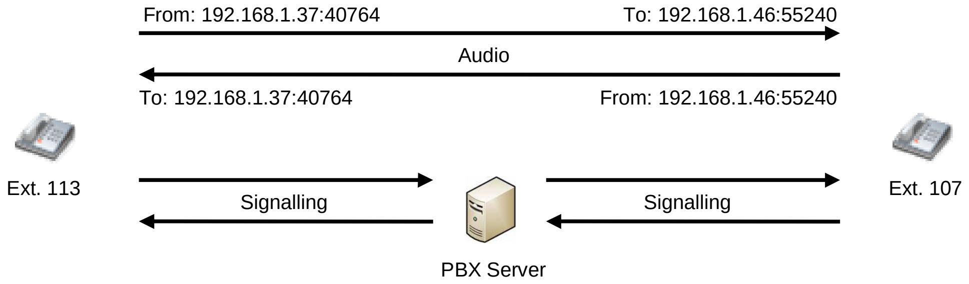
This chapter gives you an overview of how a call is established within 3CX and how you can trace through these steps by reading the server status log. The Server status log can be accessed by going to the 3CX management console and clicking on the Phone System > Server status log.

This server status log lists all important events that 3CX Phone System is processing, and is the most important tool when troubleshooting an issue.

In this chapter, we step through the events being logged for a simple extension to extension call.

Call setup between 2 internal extensions

Log 7	StratLink::onHangUp	[CM104001] Call(C:309): Ext.107 hung up call; cause: BYE; reason: SIP;description=”User Hung Up”
Log 6	CallLegImpl:: onConnected	[CM103001] Call(C:309): Created audio channel for Ext.107 (192.168.1.46:55240) with third party (192.168.1.37:40764)
Log 5	StratInOut::onConnected	[CM104005] Call(C:309): Setup completed for call from Ext.113 to Ext.107
Log 4	CallLegImpl::onConnected	[CM103001] Call(C:309): Created audio channel for Ext.113 (192.168.1.37:40764) with third party (192.168.1.46:55240)
Log 3	CallConf::onProvisional	[CM103003] Call(C:309): Ext. 107 is ringing
Log 2	CallConf::onProvisional	[CM103003] Call(C:309): Ext. 107 is ringing
Log 1	CallConf::onIncoming	[CM103002] Call(C:309): Incoming call from 113 (Ext.113) to sip:107@192.168.1.3



Step 1 – Incoming call detected

Log message 1 [CM103002]

A call, with Call ID 309, has been received with destination “107@192.168.1.3”. This implies that:

1. 113 is the extension number making the call.

2. 107 is the intended destination extension number.
3. 192.168.1.3 is the internal IP Address of the PBX server that is handling the call.

Step 2 – Destination responds

Log messages 2 and 3 [CM103003]

Call ID 309 logs that Extension 107 is responding, but that the call is not yet connected. Typically this means that:

1. The PBX Server has determined that Extension 107 is available, and sends a SIP Status message “100 Trying” to Extension 113.
2. The PBX Server sends a SIP Request message “INVITE” to Extension 107.
3. Extension 107 responds with a SIP Status message “100 Trying”.
4. Extension 107 is ringing, and notifies the PBX server with a SIP Status message “180 Ringing”.
5. The PBX Server will, for each SIP Status message “180 Ringing” received from Extension 107, send a SIP Status message “180 Ringing” to Extension 113 to keep it informed of progress.

Step 3 – Connecting first endpoint to the call

Log message 4 [CM103001]

Call ID 309 logs that it is setting up an audio channel to connect one of the endpoints (Extension 113). This implies that:

1. Extension 107 has sent a SIP Status message “200 OK”. The PBX will generate a SIP Request message “ACK” to confirm.
2. The PBX Server sends a SIP Status message “200 OK” to Extension 113. Extension 113 will generate a SIP Request message “ACK” to confirm.
3. The PBX Server has sufficient information with which to connect the call, and is connecting the first endpoint to this call.
4. Extension 113 will be the first endpoint connected to this call.
5. Extension 113 will be connected directly to Extension 107, because neither of the 2 extensions (113, 107) participating in this call have been configured to connect to Media Server.
6. Extension 113 is an internal extension, and IP Address 192.168.1.37 and Port 40764 will be the source of the audio stream generated by Extension 113. Extension 113 will listen for an incoming audio stream from the other endpoint on the same Port number.
7. Extension 107 is an internal extension, and IP Address 192.168.1.46 and Port 55240 will be the source of the audio stream generated by Extension 107. Extension 107 will listen for an incoming audio stream from Extension 113 on the same Port number.

Step 4 – Connecting the second endpoint to the call

Log message 5 [CM104005]

Log message 6 [CM103001]

Call ID 309 generates these 2 log messages simultaneously upon setting up the audio channel to connect the second endpoint (Extension 107) to the call. This implies that:

1. Extension 107 will be the second endpoint connected to this call.
2. Extension 107 will be connected directly to Extension 113, because neither of the 2 extensions (113, 107) participating in this call have been configured to connect to Media Server.
3. Extension 107 is an internal extension, and IP Address 192.168.1.46 and Port 55240 will be the source of the audio stream generated by Extension 107. Extension 107 will listen for an incoming audio stream from Extension 113 on the same Port number.
4. Extension 113 is an internal extension, and IP Address 192.168.1.37 and Port 40764 will be the source of the audio stream generated by Extension 113. Extension 113 will listen for an incoming audio stream from the other endpoint on the same Port number.
5. The PBX Server has completed signaling for this call between Extensions 113 and 107. The two extensions will send audio to each other directly, instead of having Media Server forwarding audio from one entity to the other.

Step 5 – Call Termination

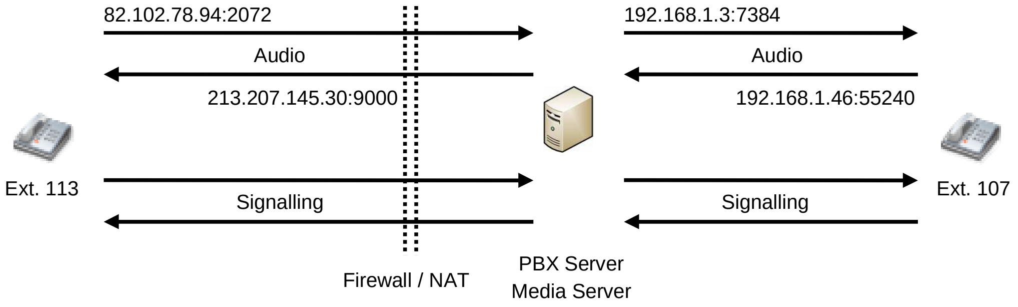
Log message 7 [CM104001]

Extension 107 has hung up the call. This implies that:

1. Extension 107 sends a SIP Request “BYE” message to the PBX Server. The PBX Server responds with a SIP Status “200 OK” message.
2. The PBX Server sends a SIP Request “BYE” message to Extension 113. Extension 113 responds with a SIP Status “200 OK” message.
3. All endpoints to this call are terminated.

Call setup between an external extension and an internal extension.

Log 7	StratLink::onHangUp	[CM104001] Call(C:302): Ext.107 hung up call; cause: BYE; reason: SIP;description=”User Hung Up”
Log 6	CallLegImpl::onConnected	[CM103001] Call(C:302): Created audio channel for Ext.107 (192.168.1.46:55240) with Media Server (192.168.1.3:7384)
Log 5	StratInOut::onConnected	[CM104005] Call(C:302): Setup completed for call from Ext.113 to Ext.107
Log 4	CallLegImpl::onConnected	[CM103001] Call(C:302): Created audio channel for Ext.113 (82.102.78.94:2072) with Media Server (213.207.145.30:9000)
Log 3	CallConf::onProvisional	[CM103003] Call(C:302): Ext. 107 is ringing
Log 2	CallConf::onProvisional	[CM103003] Call(C:302): Ext. 107 is ringing
Log 1	CallConf::onIncoming	[CM103002] Call(C:302): Incoming call from 113 (Ext.113) to sip:107@192.168.1.3



Step 1 – Incoming call detected

Log message 1

A call, with Call ID 302, has been received with destination "107@192.168.1.3". This implies that:

1. 113 is the extension number making the call.
2. 107 is the intended destination extension number.
3. 192.168.1.3 is the internal IP Address of the PBX server that is handling the call.

Step 2 – Destination responds

Log messages 2 and 3

Call ID 302 logs that Extension 107 is responding, but that the call is not yet connected. Typically this means that:

1. The PBX Server has determined that Extension 107 is available, and sends a SIP Status message "100 Trying" to Extension 113.
2. The PBX Server sends a SIP Request message "INVITE" to Extension 107.
3. Extension 107 responds with a SIP Status message "100 Trying".
4. Extension 107 is ringing, and notifies the PBX server with a SIP Status message "180 ringing".
5. The PBX Server will, for each SIP Status message "180 Ringing" received from Extension 107, send a SIP Status message "180 Ringing" to Extension 113 to keep it informed of progress.

Step 3 – Connecting first endpoint to the call

Log message 4

Call ID 302 logs that it is setting up an audio channel to connect one of the endpoints (Extension 113). This implies that:

1. Extension 107 has sent a SIP Status message "200 OK". The PBX will generate a SIP Request message "ACK" to confirm.
2. The PBX Server sends a SIP Status message "200 OK" to Extension 113. Extension 113 will generate a SIP Request message "ACK" to confirm.

3. The PBX Server has sufficient information with which to connect the call, and is connecting the first endpoint to this call.
4. Extension 113 will be the first endpoint connected to this call.
5. Extension 113 will be connected to Media Server, because at least one of the extensions involved in this call (107, 113) is configured to connect to Media Server. Note the difference from the previous example with 2 internal extensions – in this case the Media Server will sit between the two parties, marshalling the audio from one party to the next.
6. Extension 113 is an external extension, and IP Address 82.102.78.94 and Port 2072 will be the source of the audio stream generated by this extension. Extension 113 will listen for an incoming audio stream from the Media Server on the same IP Address and Port number. Note the difference from the previous example, where this time Extension 113 is identifiable as an external extension from the IP Address reported in the log message.
7. Media Server will connect directly with Extension 113, and IP Address 213.207.145.30 and Port 9000 will be the source of the audio stream generated by Media Server. Media Server will listen for an incoming audio stream from Extension 113 on the same IP Address and Port number. Note again the difference from the previous example, where this time Media Server reports that it is communicating with an external extension, identified by Media Server reporting its Public IP Address, and by reporting a Port number from the range specified for use with external devices in the Management Console General Settings.

Step 4 – Connecting the second endpoint to the call

Log messages 5 and 6

Call ID 302 generates these 2 log messages simultaneously upon setting up the audio channel to connect the second endpoint (Extension 107) to the call. This implies that:

1. Extension 107 will be the second endpoint connected to this call.
2. Extension 107 will be connected directly to Media Server, because at least one of the extensions involved in this call (107, 113) is configured to connect to Media Server.
3. Extension 107 is an internal extension, and IP Address 192.168.1.46 and Port 55240 will be the source of the audio stream generated by this extension. Extension 107 will listen for an incoming audio stream from the Media Server on the same Port number.
4. Media Server will connect with Extension 107, and IP Address 192.168.1.3, and Port 7384 will be the source of the audio stream generated by Media Server. Media Server will listen for an incoming audio stream from Extension 107 on the same IP Address and Port number. Note in this case that Media Server will communicate with Extension 107 using its Internal IP Address, and that it reports that it is using a Port number from the range specified for use with internal devices in the Management Console General Settings.

5. The PBX Server has completed signaling for this call between Extensions 113 and 107. Media Server will forward audio from one entity to the other, instead of the two extensions sending audio to each other directly.

Step 5 – Call Termination

Log message 7

Extension 107 has hung up the call. This implies that:

1. Extension 107 sends a SIP Request “BYE” message to the PBX Server. The PBX Server responds with a SIP Status “200 OK” message.
2. The PBX Server sends a SIP Request “BYE” message to Extension 113. Extension 113 responds with a SIP Status “200 OK” message.
3. The PBX Server notifies Media Server that the call has been terminated. All endpoints to this call are terminated

Analyzing registration

Introduction

This chapter describes how an extension or a gateway line registers itself with the PBX Server to make itself available for making and receiving calls, and how the PBX Server will register itself with a VoIP Provider to make itself available for making and receiving calls.

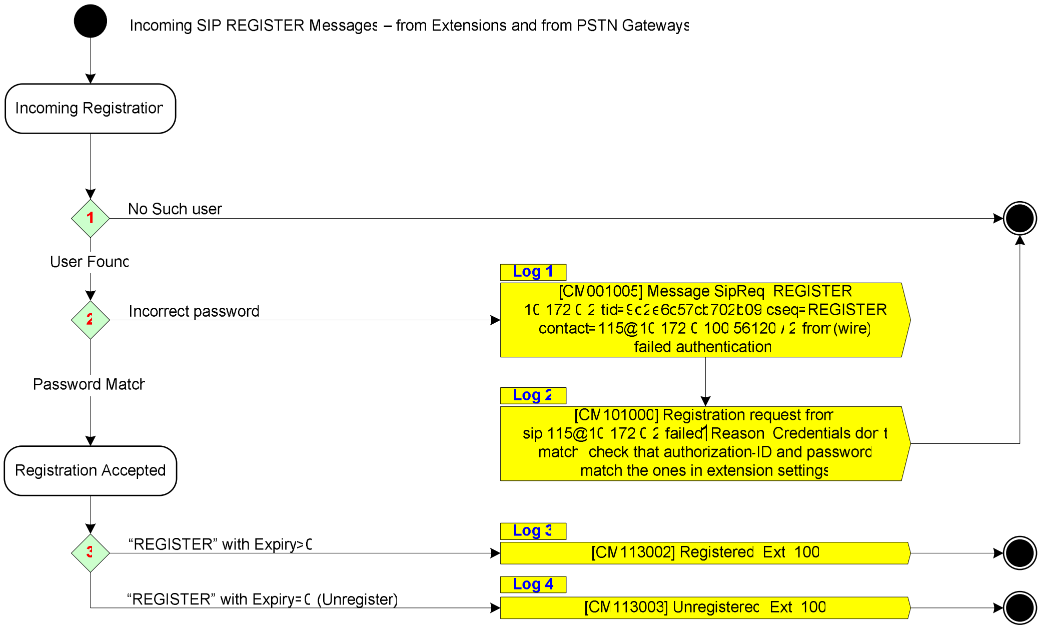
We trace through these steps by reading the server status log. The Server status log can be accessed by going to the 3CX management console and clicking on the Phone System > Server status log.

Registration Flow

Each extension or gateway line must typically register with the 3CX Phone System before the device will be able to forward calls to/from the PBX. Effectively the extension/gateway is a registration client, and the PBX is a registration server, or registrar. This is shown in the flowchart below, together with a flowchart to briefly show the mechanics behind registration expiry.

Registration with providers changes the registration perspective – the PBX is the registration client, and the provider is the registration server, or registrar. This is also shown in the flowchart below. Note that since the PBX is a client in this circumstance, registration expiry will be handled by the registrar and the 3CX Phone System needs to do no work in this regard.

Registration for Gateways and Phones



```
[CM001005] Message SipReq: REGISTER
10.172.0.2 tid=9c2e6c57cb702b09 cseq=REGISTER
contact=115@10.172.0.100:56120 / 2 from(wire)
failed authentication
```

```
[CM101000] Registration request from
sip:115@10.172.0.2 failed. Reason: Credentials don't
match - check that authorization-ID and password
match the ones in extension settings
```

```
[CM113002] Registered: Ext. 100
```

```
[CM113003] Unregistered: Ext. 100
```

Figure 1 – Incoming SIP REGISTER Messages

Steps 1-3 and Log messages 1-4 relate to INCOMING SIP REGISTER requests from extensions and gateway lines – where the 3CX Phone System is the SIP server (registrar).

Step 1 – Determine if user attempting to register is valid

In this step the SIP server determines if this SIP Registration is being made by a user (extension or gateway line) defined in the 3CX Management Console.

If the user (extension or gateway line) is FOUND, the process moves to Step 2.

If the user (extension or gateway line) is NOT FOUND, NO LOG MESSAGES ARE POSTED, and the REGISTER request is denied.

To avoid flooding the server status log with repeated invalid requests, currently this error is not logged.

Step 2 – Determine if user attempting to register is using the correct password

In this step the SIP server determines if this SIP Registration being made contains the correct password for the stated user (extension or gateway line) defined in the 3CX Management Console.

If the password is CORRECT, the process moves to Step 3.

If the password is INCORRECT, log messages 1 and 2 are posted, and the REGISTER request is denied.

AuthMgr::requestCredential	[CM001005] Message SipReq: REGISTER 10.172.0.2 tid=9d2e6c57cb702b09 cseq=REGISTER

	contact=115@10.172.0.100:56120 / 2 from(wire) failed authentication!

Log: 1

AuthMgr::onAuthFailure	[CM101000] Registration request from sip:115@10.172.0.2 failed. Reason: Credentials don't match, check that authorization-ID and password match the ones in extension settings

Log: 2

The SIP Server has received a SIP REGISTER request, and has found the stated user in the database (extension or gateway line), but the password provided does not match. The username attempting to register is “115”, and the IP Address which originated the request is “10.172.0.100”.

To adjust credentials set on the phone or gateway, it will be necessary to review the manual supplied with the SIP Phone or Gateway device in question. The user manual for the 3CX Phone System contains examples for typical phones and gateways.

Step 3 – Determine if the request is REGISTER or unREGISTER

In this step the SIP server determines if this SIP REGISTER request is a request to register or unregister. The distinguishing difference is simply inside the “Expiry” variable inside the SIP REGISTER message.

If the request is to REGISTER, the message specifies the “Expiry” variable to be a value more than “0”, log message 3 will be posted, and the extension or gateway line will be registered to the 3CX Phone System.

If the request is to UNREGISTER, the message specifies the “Expiry” variable to be “0”, log message 4 will be posted, and the extension or gateway line will be unregistered from the 3CX Phone System.

ServRegs::onAdd	[CM113002] Registered: Ln:10400@GW_SPA-3000

Log: 3

The PBX Server has successfully matched username and password for this request to REGISTER.

ServRegs::onRemove	[CM113003] Unregistered: Ext.115

Log: 4

The PBX Server has successfully matched username and password for this request to UNREGISTER.

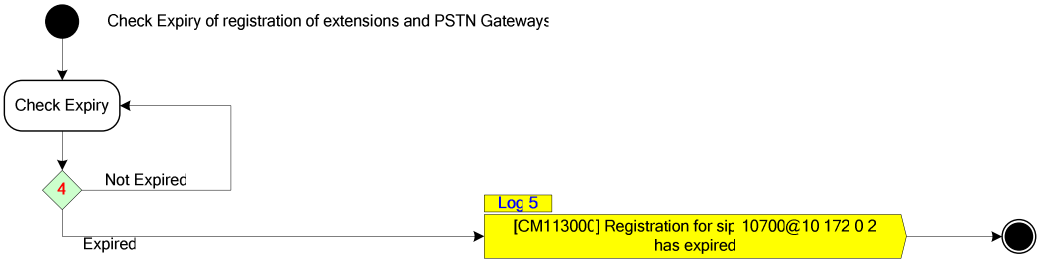

Check Expiry of registration of extensions and PSTN Gateways
Check Expiry
4
Not Expired
Expired
Log 5
[CM11300C] Registration for sip 10700@10 172 0 2 has expired

Figure 2 – Registration Expiry for Extensions and Gateway Lines

Step 4 and Log message 5 relate to SIP registration expiry events generated by the 3CX Phone System timer assigned to each successful registration.

Step 4 – Timer event to trigger deregistration after expiry

In this step the SIP server checks the expiry status of all registered extensions and gateway lines.

If the timer indicates that the registration is NOT EXPIRED, no events are triggered, no log messages are posted, and no action is taken.

If the timer indicates that the registration is EXPIRED, then log message 5 is posted, and the extension or gateway line is deregistered from the PBX server.

ServRegs::checkExpiration	[CM113000] Registration for sip:10700@10.172.0.2 has expired

Log: 5

The PBX Server has determined that the registration for gateway line with internal number 10602 has expired.

Registration for VoIP Providers

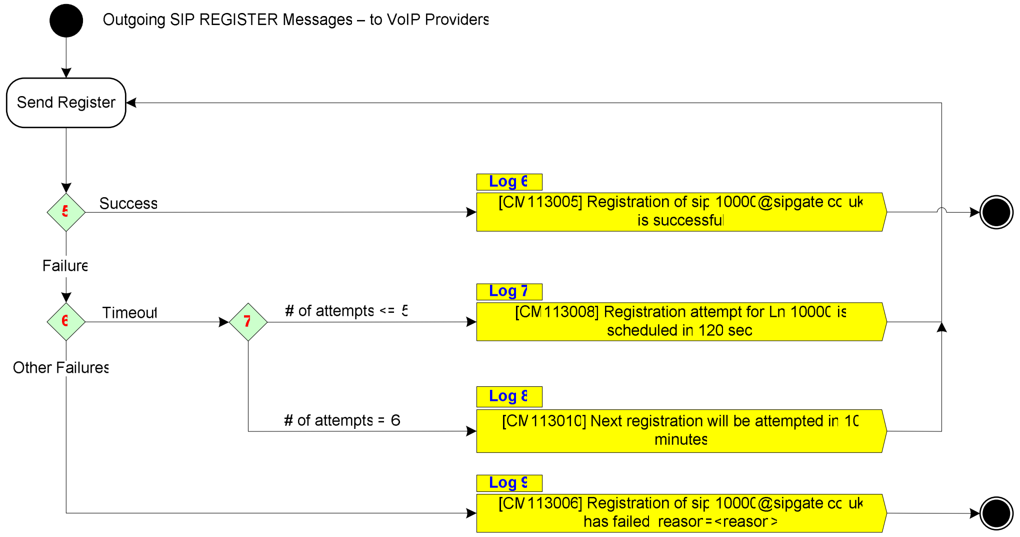

Outgoing SIP REGISTER Messages – to VoIP Providers
Send Register
5
Success
Failure
6
Timeout
Other Failures
7
# of attempts <= 5
# of attempts = 6
Log 6
[CM113005] Registration of sip:10000@sipgate.co.uk is successful
Log 7
[CM113008] Registration attempt for Ln 10000 is scheduled in 120 sec
Log 8
[CM113010] Next registration will be attempted in 10 minutes
Log 9
[CM113006] Registration of sip:10000@sipgate.co.uk has failed reason=<reason>

Figure 3 – Outgoing SIP REGISTER Messages

Steps 5-7 and Log messages 6-9 relate to OUTGOING SIP REGISTER requests to VoIP Provider accounts (lines) – where the 3CX Phone System is the SIP client.

Step 5 – Server checks registration with VoIP Provider

In this step the SIP server attempts to register the line with the VoIP Provider.

ClientRegs::onSuccess	[CM113005] Registration of sip:10000@sipgate.co.uk is successful

Log: 6

If the registration is SUCCESSFUL, log message 6 is posted, and the line is registered and made available for outgoing and incoming calls.

If the registration FAILS, the process moves to Step 6.

Step 6 – The PBX Server checks failure reason

The PBX Server checks why the registration attempt has failed.

ClientRegs::onFailure	[CM113006] Registration of sip:10000@sipgate.co.uk has failed; reason=<reason>

Log: 9

If the registration request was REJECTED, then log message 9 is posted. For example, the VoIP Provider may have responded with a SIP Status message “403 Forbidden”. Typically the reason is reported with this log message.

No further registration attempts will be made for this line.

Please note that in the case of replies “408 Request Timeout” or “480 Temporarily Unavailable”, the PBX Server will silently resubmit the request after a while, since this type of request is a standard response which implies that the registrant (the PBX Server in this case) should retry after some time.

If the registration request receives NO RESPONSE, otherwise knows as a TIMEOUT, then the process moves to Step 7.

Step 7 – The PBX Server has received no response

The PBX Server will decide how to handle the situation when no response has been received from the VoIP Provider.

ClientRegs::onFailure	[CM113008] Registration attempt for Ln.10000 is scheduled in 120 sec.

Log: 7

If the number of attempts is LESS THEN 5, then log message 7 is posted, and the request is resubmitted. The interval between requests is increased incrementally, as follows:

Start	First Request
120 secs	Second Request
240 secs	Third Request
360 secs	Fourth Request
480 secs	Fifth Request

If the number of attempts is EQUAL TO 5, then log message 8 is posted, and the registration process is restarted.

ClientRegs::onFailure	[CM113010] Next registration will be attempted in 10 minutes

Log: 8

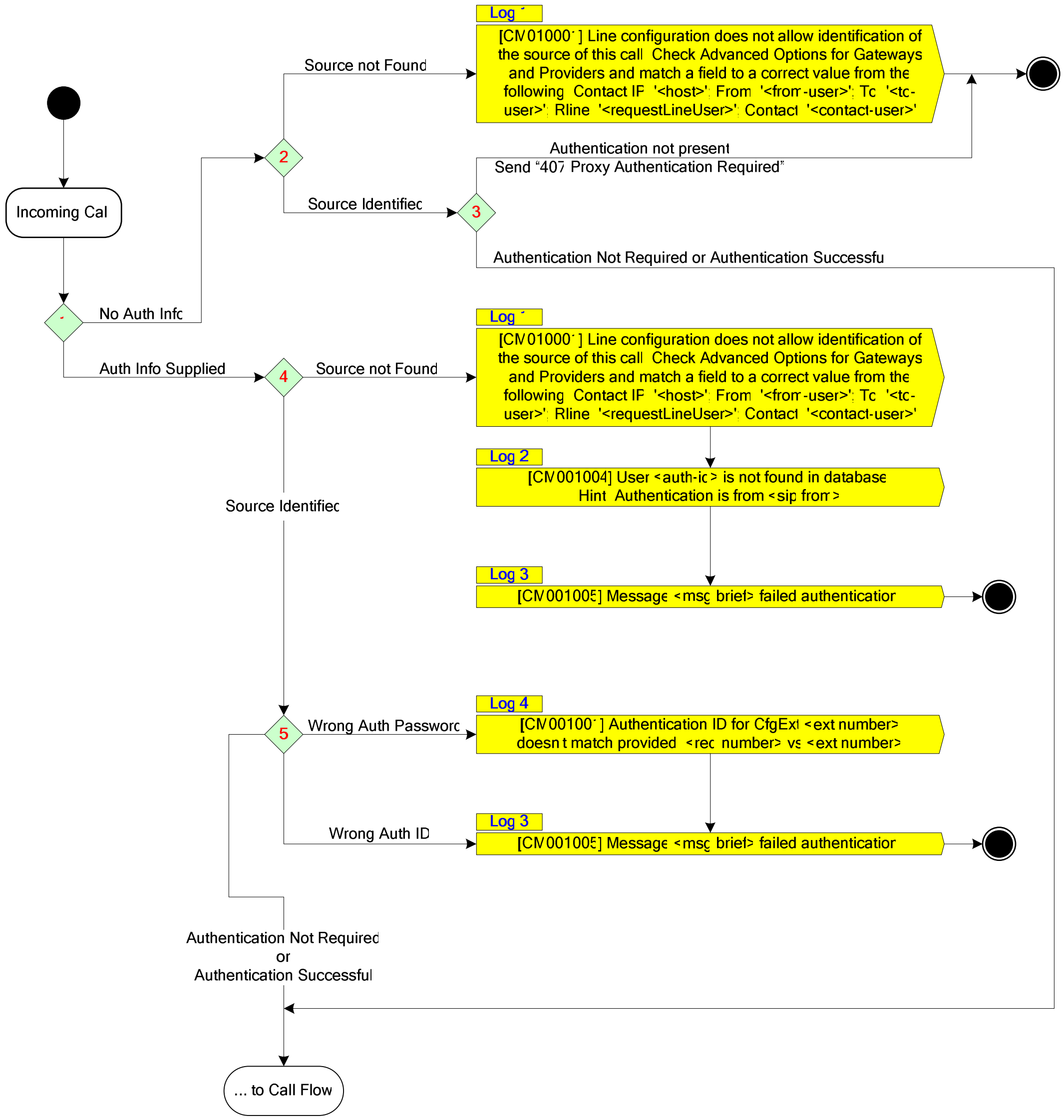
The PBX Server has received no response from the VoIP Provider after 5 attempts. Since this is not an explicit failure, the PBX Server will restart the registration procedure for this line. Typical reasons for this situation are either non-delivery of SIP Messages to the PBX Server because, for example, firewall rules block packets from the VoIP Provider on the SIP port (typically 5060), or because the host being contacted does not have SIP functionality implemented – normally because the wrong host address was specified for this VoIP Provider.

Understanding the Authentication Process and Call Routing

Introduction

This chapter describes and illustrates in more detail how a call is established between two endpoints (i.e. extensions). The authentication and call routing process is analyzed in more detail. Each of the labeled log messages, reported in the Server Status page, is explained with suggested reasons and possible solutions.

Authentication Flow



[CM01000´] Line configuration does not allow identification of the source of this call Check Advanced Options for Gateways and Providers and match a field to a correct value from the following Contact IP '<host>' From '<from-user>' To '<to-user>' Rline '<requestLineUser>' Contact '<contact-user>'

[CM01000´] Line configuration does not allow identification of the source of this call Check Advanced Options for Gateways and Providers and match a field to a correct value from the following Contact IP '<host>' From '<from-user>' To '<to-user>' Rline '<requestLineUser>' Contact '<contact-user>'

[CM001004] User <auth-ic> is not found in database
Hint Authentication is from <sip from>

[CM001005] Message <msg brief> failed authenticatior

[CM00100´] Authentication ID for CfgExt <ext number> doesn t match provided <rec number> vs <ext number>

[CM001005] Message <msg brief> failed authenticatior

Figure 4 - Authentication flow

Each call setup involves each extension/endpoint authenticating itself against the IP PBX. This process is displayed in the flow chart Authentication flow.

Step 1 – Determine if authorization info is present

In this first step the SIP server determines if for this SIP Request, authorization information is included. When a VoIP gateway device makes the first request for a call, it may optionally choose to send this request without authentication information embedded in the request.

If authentication information IS NOT supplied, the process moves to Step 2.

If authentication information IS supplied, the process moves to Step 4.

Step 2 – Identify source of request (no authentication info)

The SIP server will attempt to identify the source of the request. It needs to be able to map an incoming call to a particular VOIP provider account or gateway line in order to authenticate a call. First of all you don’t accept all calls (call spam) and secondly we need to apply inbound routing rules on an incoming call. In order to do any of these we need to identify the source of the call.

We identify the source of an incoming call by a number of means. These are variable/configurable because it is dependent on what device the caller is using and because devices/providers use different methods to communicate “who they are”. We can identify source based on:

- The IP address of the caller (e.g. IP of gateway or provider)
- The internal number assigned to the line that is being called (e.g. 10000 assigned to the first line of an FXO gateway)
- The external number assigned to the line that is being called (e.g. 22444032 that has been assigned to a particular line)

We look for the above parameters in Contact, From, To, Request Line URI and RemoteParty fields of the SIP INVITE message, depending on the way you have configured the line in 3CX Phone System.

If the source of the request CAN be identified, the process moves to Step 3.

If the source of the request CANNOT be identified, the Server Status page logs a message, as described and explained below. The call will subsequently be terminated.

Endpoint::findSource	[CM010001] Line configuration does not allow identification of the source of this call. Check Advanced Options for Gateways and Providers and match a field to a correct value from the following: Contact IP: '<host>'; From: '<from-user>'; To: '<to-user>'; Rline: '<requestLineUser>'; Contact: '<contact-user>'

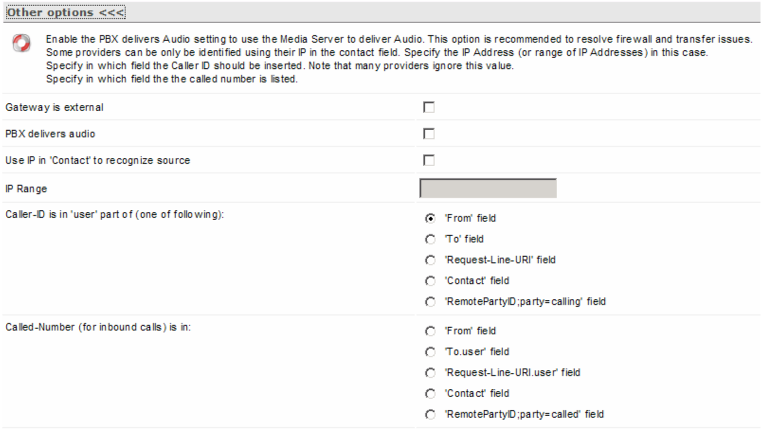
Log: 1

The PBX Server has analyzed this call and cannot identify its source. The message lists the host part of the “Contact” field (e.g. 192.168.23.5 of “10000@192.168.23.5”) – typically the IP Address of the source, the user/first part of the “From” field (e.g. 10000 of

“10000@192.168.23.5”),, the user/first part of the “To” field, the user part of the “Request Line” field, and the user part of the “Contact” field.

This information should assist us to correctly configure the line/device that is receiving this incoming call so that 3CX Phone System will be able to identify the source. Typically, 3CX Phone System will attempt to analyze all the relevant fields and values and dynamically match the incoming call request to the appropriate line. You may need to use the Provider’s IP Address to allow the PBX to determine the source of the call.

1. Go to Lines -> Manage and click “Manage Gateways and Providers” button.
2. Click on the Gateway/Provider name to edit the Gateway/Provider settings, and click the “Other Options>>>” section to expand it and view its contents.



Other options <<<

Enable the PBX delivers Audio setting to use the Media Server to deliver Audio. This option is recommended to resolve firewall and transfer issues.
Some providers can be only be identified using their IP in the contact field. Specify the IP Address (or range of IP Addresses) in this case.
Specify in which field the Caller ID should be inserted. Note that many providers ignore this value.
Specify in which field the the called number is listed.

Gateway is external	☐
PBX delivers audio	☐
Use IP in 'Contact' to recognize source	☐
IP Range	
Caller-ID is in 'user' part of (one of following):	● 'From' field ○ 'To' field ○ 'Request-Line-URI' field ○ 'Contact' field ○ 'RemotePartyID;party=calling' field
Called-Number (for inbound calls) is in:	○ 'From' field ○ 'To.user' field ○ 'Request-Line-URI.user' field ○ 'Contact' field ○ 'RemotePartyID;party=called' field

3. Enable the “Use IP in ‘Contact’ to recognize source” option.
4. Specify the range of IP Addresses used by the VoIP Provider in the “IP Range” field. It may be sufficient to insert the IP Address reported in the log message as the “Contact IP”, but the VoIP Provider may deliver messages using a number of IP Addresses – in this case it will be necessary to set the full range of IP Addresses for predictable behavior.

Step 3 – Determine if authorization is required

This step is reached if a request is received with NO authentication information.

If the source of this request is configured to REQUIRE authentication, the SIP server will respond with a “407 Authentication required” message to the source of the call, which basically asks the initiator to resend the request WITH authentication information.

This is standard behavior, and therefore not technically an error. No log messages are reported since it is normal behavior for the sending device to acknowledge this status and resend the request with authentication information – effectively restarting the authentication process from Step 1 with a new request with authentication information included.

If the source of this request is configured to NOT REQUIRE authentication, the request has successfully completed the source identification and authentication procedure and will proceed directly to Step 6, in the Call Flow section.

Configuration of authentication required or not is configured here:

1. In the Management Console, click on Lines -> Manage to go to the Line Management page. Now click the “Manage Gateways & Providers” icon. Identify the relevant gateway or provider you would like to configure and click on its name. Now click on the “Registration settings >>>” section to expand it and view its contents. Identify the options called “Require (optionally: authorized) registration for” and select the desired behavior.

… OR …

2. In the Management Console, click on Lines -> Manage to go to the Line Management page. Identify the line for which you wish to configure authentication parameters, and click on the Identification field for this line. In the main section, set the password to blank i.e. no password. This configuration option is less flexible, but is a quick and simple way to achieve authorization/registration bypass.

Step 4 – Identify source of request (with authentication info)

The SIP server will attempt to identify the source of the request. It needs to be able to map an incoming call to a particular VOIP provider account or gateway line in order to authenticate a call. First of all you don’t accept all calls (call spam) and secondly we need to apply inbound routing rules on an incoming call. In order to do any of these the 3CX Phone System needs to identify the source of the call.

The source of an incoming call can be identified by a number of means - mainly because it is dependent on what device or provider the caller is using and because devices/providers use different methods to communicate “who they are”. The source can be identified using:

- The IP address of the caller (e.g. IP of gateway or provider)
- The virtual extension number assigned to the line that is being called (e.g. 10000 assigned to the first line of an FXO gateway)
- The identification assigned to the line that is being called (e.g. 22444032 that has been assigned to a particular line)
- The authentication ID for the line. (e.g. myaccount)

3CX Phone System looks for the above parameters in Contact, From, To, Request Line URI and RemoteParty fields of the SIP INVITE message.

If the source of the request CAN be identified, the process moves to Step 3.

If the SIP server CANNOT identify the source of this request, log entries 1, 2, and 3 will be posted for information purposes, and the call will subsequently be terminated.

Endpoint::findSource	[CM010001] Line configuration does not allow identification of the source of this call. Check Advanced Options for Gateways and Providers and match a field to a correct value from the following: Contact IP: '<host>'; From: '<from-user>'; To: '<to-user>'; Rline: '<requestLineUser>'; Contact: '<contact-user>'

Log: 1

The PBX Server has analysed this call and cannot identify its source. The message lists the host part of the “Contact” field (e.g. 192.168.23.5 of “10000@192.168.23.5”), the user/first part of the “From” field (e.g. 10000 of “10000@192.168.23.5”),, the user/first part of the “To” field, the user part of the “Request Line” field, and the user part of the “Contact” field.

This information should assist us to correctly configure the line/device that is receiving this incoming call so that 3CX Phone System will be able to identify the source. Follow the procedures outlined above in Step 2.

AuthMgr::requestCredential	[CM001004] User <auth-id> is not found in database! Hint: Authentication is from <sip:from>

Log: 2

The incoming call was received with an Authentication ID and password that is incorrect. This message also provides the user part of the “From” SIP field to allow the user to identify which gateway or provider has forwarded the call. This message is typically generated when the gateway itself has been configured, but the 3CX Management Console has not yet been configured with the data for this gateway (or the gateway configuration was changed without adjusting the settings in the 3CX Management Console).

AuthMgr::requestCredential	[CM001005] Message <msg.brief> failed authentication!

Log: 3

This log entry gives a summary of the authentication request is provided to help with identifying the problem.

Step 5 – SIP server attempts to authenticate the request

The SIP server has now identified the source of the call and is ready to authenticate the credentials.

If the request CAN authenticate, or if the Authentication ID is correct and the Authentication Password has been set to blank in the Management Console for the source of this request, the request has successfully completed the source identification and authentication procedure and will proceed to Step 6, in the Call Flow section.

If the request CANNOT authenticate because the Authentication Password is incorrect, log entries 4 and 3 will be posted for information purposes, and the call will subsequently be terminated.

If the request CANNOT authenticate because the Authentication ID is incorrect, log entry 3 will be posted for information purposes, and the call will subsequently be terminated.

AuthMgr::requestCredential	[CM001001] Authentication ID for CfgExt:<ext.number> doesn't match provided: <req.number> vs <ext.number>

Log: 4

The incoming call was received with an Authentication ID that is not listed in the database. This message also provides the user part of the “From” SIP field to allow the user to identify which gateway or provider has forwarded the call. This message is typically generated when the gateway itself has been configured, but the 3CX Management Console has not yet been configured with the data for this gateway (or the gateway configuration was changed without adjusting the settings in the 3CX Management Console).

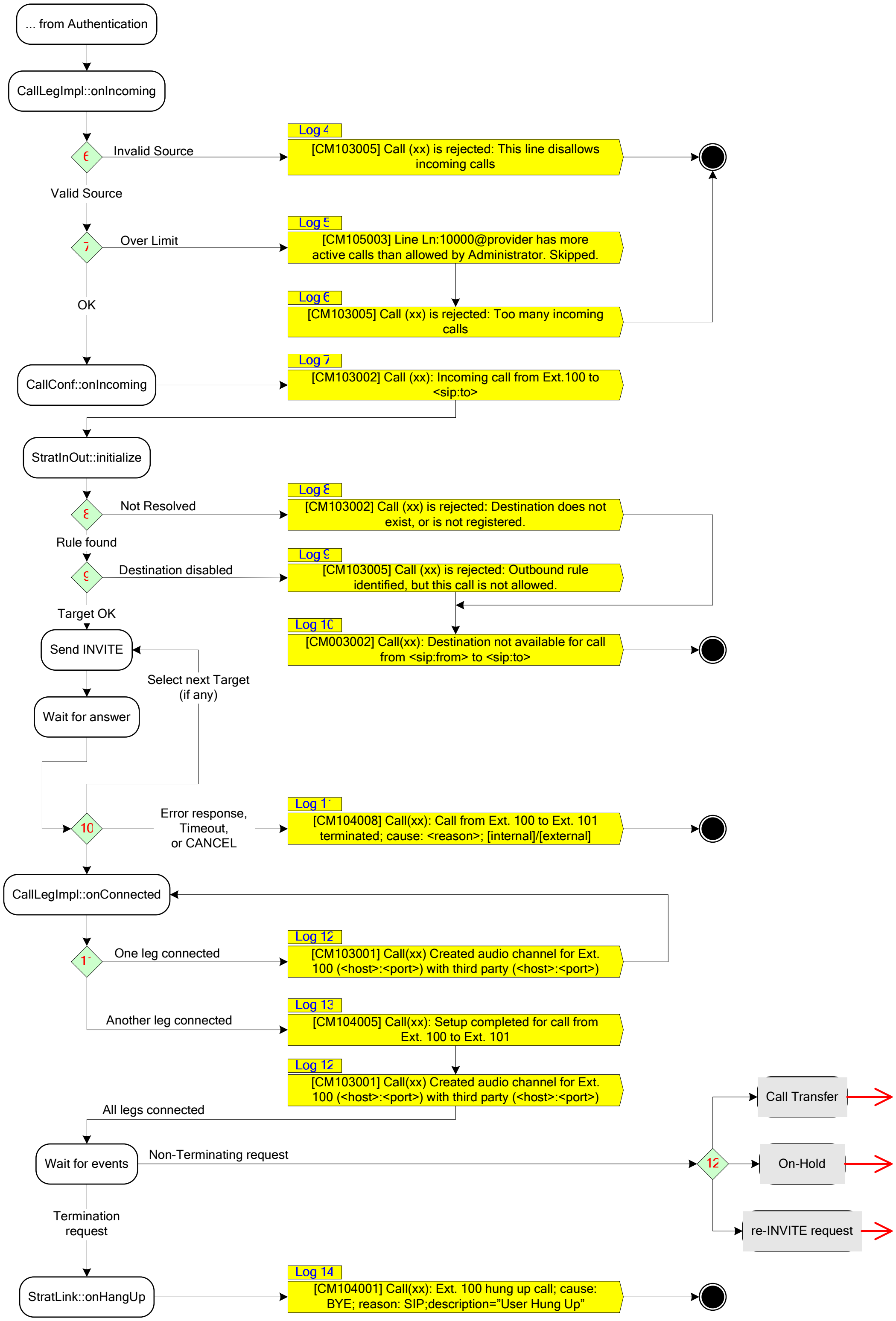
```
AuthMgr::requestCredential    [CM001005] Message <msg.brief> failed authentication!
```

Log: 3

This log entry gives a summary of the authentication request provided to help with identifying the problem.

Call Flow

After authentication has been successfully performed, the call process continues. This process is displayed in the flowchart below.



```
[CM103005] Call (xx) is rejected: This line disallows incoming calls
[CM105003] Line Ln:10000@provider has more active calls than allowed by Administrator. Skipped.
[CM103005] Call (xx) is rejected: Too many incoming calls
[CM103002] Call (xx): Incoming call from Ext.100 to <sip:to>
[CM103002] Call (xx) is rejected: Destination does not exist, or is not registered.
[CM103005] Call (xx) is rejected: Outbound rule identified, but this call is not allowed.
[CM003002] Call(xx): Destination not available for call from <sip:from> to <sip:to>
[CM104008] Call(xx): Call from Ext. 100 to Ext. 101 terminated; cause: <reason>; [internal]/[external]
[CM103001] Call(xx) Created audio channel for Ext. 100 (<host>:<port>) with third party (<host>:<port>)
[CM104005] Call(xx): Setup completed for call from Ext. 100 to Ext. 101
[CM103001] Call(xx) Created audio channel for Ext. 100 (<host>:<port>) with third party (<host>:<port>)
[CM104001] Call(xx): Ext. 100 hung up call; cause: BYE; reason: SIP;description="User Hung Up"
```

Figure 5 - Call flow

Step 6 – SIP server checks whether the line is allowed to receive incoming calls

The SIP server will check whether inbound calls are allowed on this line.

CallConf::onIncoming	[CM103005] Call (xx) is rejected: This line disallows incoming calls

Log: 4

If the line may not receive calls, log entry 4 will be posted, and the call will subsequently be terminated. This means that the call is originating from a line that has been declared “outgoing only” or “disabled”.

To check whether a line has been set to “outgoing only” or “disabled”:

1. In the 3CX Management Console go to Lines -> Manage
2. Click on the External Number of the Line to be examined
3. To enable incoming calls on this line, make sure the checkbox labeled “Answer incoming calls on this line” is ticked.

Step 7 – SIP server checks whether the allowed number of simultaneous calls is exceeded

The SIP server will check whether this new call will exceed the maximum number of calls which can be handled by the line forwarding this call.

CallConf::onIncoming	[CM105003] Line Ln:10000@provider has more active calls than allowed by Administrator. Skipped.

Log: 5

CallConf::Rejected	[CM103005] Call (xx) is rejected: Too many incoming calls

Log: 6

If the call EXCEEDS the set limit of simultaneous calls for this line, log 5 and 6 will be posted. To resolve this issue, you must go the Line Configuration page and increase the value in the “Maximum simultaneous calls” field. Please be aware that sufficient bandwidth must be available to handle the number of calls specified, also taking into account the “Maximum simultaneous calls” settings on other lines.

CallConf::onIncoming	[CM105003] Call(xx): Incoming call from Ext. 100 to <sip:to>

Log: 7

If the call DOES NOT EXCEED the maximum number of calls for this line, log entry 7 will be posted, and the call will proceed will proceed to Step 8.

Step 8 – SIP server checks for a valid destination

The SIP Server will attempt to identify the correct destination for this call – which may be an extension number, a voice mailbox for an extension number, a ring group, a digital receptionist, or an outside number via an outbound rule.

StratInOut::initialize	[CM103002] Call (xx) is rejected: Destination does not exist, or is not registered.

Log: 8

StratInOut::initialize	[CM003002] Call(xx): Destination not available for call from <sip:from> to <sip:to>

Log: 10

If the SIP server CANNOT identify the correct destination for this call, log entries 8 and 10 will be posted, and the call will subsequently be terminated.

The PBX Server has attempted to route the call, but the destination could not be resolved to a registered SIP phone. This message is typically generated (together with Log: 10) when a call needs to be routed to a SIP Phone or gateway line that is not registered with the PBX Server.

Step 9 – SIP server checks if the route to the destination is enabled or disabled

The SIP server will check whether the configured route to the destination of this call is enabled or disabled. This occurs in the case of an incoming call from a gateway or provider line that is destined for an extension which is being forwarded to an outside number.

StratInOut::initialize	[CM103005] Call (xx) is rejected: Outbound rule identified, but this call is not allowed.

Log: 9

StratInOut::initialize	[CM003002] Call(xx): Destination not available for call from <sip:from> to <sip:to>

Log: 10

If the route to the destination can not be used, log entries 9 and 10 will be posted, and the call will subsequently be terminated.

This message is typically generated (together with Log: 10) when the PBX Server has attempted to route the call, but the lines which have been configured as the outbound route for this call are all disabled (or set to incoming only).

Step 10 – SIP server responds to request

The SIP Server has determined that the configured destination is valid, enabled, and reachable, and responds to the request and advises that it will attempt to connect the call, typically by sending a SIP “100 Trying” message to the destination of the call.

If the destination responds with SIP “200 OK”, the SIP Server will proceed to connect the 2 endpoints, or legs, to the call, and the call will proceed to Step 11.

StratInOut::onCancel	[CM104008] Call(xx): Call from Ext. 100 to Ext. 101 terminated; cause: <reason>; [internal]/[external]

Log: 11

If the destination does not respond, or the destination responds with an error message, such as “404 Not Found”, “486 Busy”, or “600 Busy Everywhere”, or the source sends a SIP “CANCEL” message before the destination answers the call, log entry 11 will be posted, together with the reason, and the call will subsequently be terminated.

Step 11 – SIP server starts to connect endpoints

The SIP Server has received a SIP “200 OK” message from the destination, and will connect both endpoints to the call. The first endpoint to be connected will be the destination, posting log entry 12. The second endpoint is also connected, posting log entries 13 and 12. At this point the call signaling for call setup is complete, and the call progresses via the media channels created. The SIP Server will wait for events.

It is important at this stage to understand that firewall issues may still halt audio from reaching one or both endpoints, independently of whether the PBX Server can successfully complete the signaling phase.

CallLegImpl::onConnected	[CM103001] Call(xx): Created audio channel for Ext. 100: (<host>:<port>) with third party (<host>:<port>)

Log: 12

If a non-terminating request is received, such as a request for call transfer, a request to put an endpoint on hold, or a re-invite (to renegotiate a different set of media parameters, for example), the process will move to Step 12, and will eventually be re-routed into the call flow process to reconnect a different set of endpoints, or to re-establish a connection to an endpoint currently on hold – and this continues until there are only 2 remaining endpoints in a call, and one of these last 2 remaining endpoints send a call termination request.

The PBX Server has negotiated an audio channel between two endpoints for a call, and reports host and port information for both endpoints. Typically a call will generate two of these messages for every call – one entry for each endpoint. If the audio channel is established directly between the two endpoints, then the host and port information will be exact mirror images – because the two endpoints are source and destination. If the audio channels are established with Media Server, then the “third party” for both messages will be replaced with “Media Server” – because the first channel will be between source and Media Server, and the second channel will be between destination and Media Server.

This information will be very useful to identify firewall issues in cases of missing audio.

CallConf::onConnected	[CM104005] Call(xx): Setup completed for call from Ext. 100 to Ext. 101

Log: 13

The PBX Server has completed call setup for all endpoints for this call and will wait for new messages from the endpoints to notify hang up, or request transfer to another destination.

Call Termination

StratLink::onHangUp	[CM104001] Call (xx): Ext. 100 hung up call; cause: BYE; reason: SIP;description=”User Hung Up”

Log: 14

If a call termination request is received from either of the endpoints, typically with a SIP “BYE” message, log entry 14 will be posted, and the call is ended. The PBX Server will proceed to notify all other still active endpoints and disconnect them, and will post entries into the call history for future reference.

Troubleshooting

Introduction

When you are experiencing a problem always break the problem down into parts, and establish which part is causing the problem. Then you can troubleshoot the problem clearly and also provide a support agent with clear information about a problem. We suggest the following steps:

1. Check that the extensions or lines are registered. See the section "Analyzing registration" for more information.
2. Check that basic call functionality is working on the IP PBX. You should be able to call another extension and have a conversation.
3. Understand call set-up and how the process is logged in the server status log. Identify the step at which it is failing and use this manual to figure out what could be wrong and what to try.

Troubleshooting phones or lines that are not registering

Successfully registered phones will show up in the Line status with a green lamp. In addition, the phone will notify you that it is registered. The server status log will also indicate a successful registration in the following format:

ServRegs::onAdd	[CM113002] Registered: Ext.102

If registration is not successful, you will not be able to call the extension or call from the extension. Furthermore it will create problems if you have configured inbound calls to route to an unregistered phone.

Analyzing SIP Phone Failed Registration (Internal or External)

Failed registration is caused by any of the following issues

- Wrong user credentials or user credentials are entered into the wrong fields of the SIP phone
- No Network Traffic is received by the PBX Server or by the SIP Phone.

Wrong user credentials or credentials are entered into the wrong fields of the SIP phone

To check whether user credentials were wrong, examine the Server Status page and identify log entries for failed registration. Examples of such entries would be:

AuthMgr::onAuthFailure:	[CM101000] Registration request from sip:115@10.172.0.2 failed. Reason: Credentials don't match, check that authorization-ID and password match the ones in extension settings

AuthMgr::requestCredential	[CM001004] User <auth-id> is not found in database! Hint: Authentication is from <sip:from>
AuthMgr::requestCredential	[CM001001] Authentication ID for CfgExt:<ext.number> doesn't match provided: <req.number> vs <ext.number>

To resolve this, ensure that user credentials were indeed correctly entered, and ensure that you have entered these into the correct fields of the SIP phone. Note that the terminology for authentication ID and authentication password and extension number can differ between vendors, and therefore some trial and error may be required.

No Network Traffic is received by the PBX Server or by the SIP Phone

There may be a number of reasons explaining why expected Network Traffic is not observed at different points in the network:

1. No Network Traffic has actually been generated.

Some SIP Phones require restarting before new settings take effect. Some SIP Phones also have connection profiles disabled by default, and will require the profile to be enabled before registration traffic commences.

Nothing will be shown in the server status, because no data is being sent to 3CX Phone System!

2. Network Traffic leaving the SIP Phone cannot reach the PBX Server.
In the case of a Soft Phone, one explanation may be that the PC running the SoftPhone has firewall software installed that has rules blocking outgoing traffic. Additionally, especially in the case of external extensions, it may be that a border device (NAT/PAT device, router, Firewall) between extension and PBX Server has packet filtering rules that prohibit packets from traveling across the border.

Nothing will be shown in the server status, because no data is being sent to 3CX Phone System!

3. Network Traffic reaches the PBX Server, but responses cannot reach the SIP Phone. The PBX Server may have firewall software installed that has rules blocking outgoing traffic. Additionally, especially in the case of external extensions, it may be that a border device (NAT/PAT device, router, Firewall) between PBX Server and extension has packet filtering rules that prohibit packets from traveling across the border.

These scenarios are best identified using a packet capture tool such as Wireshark or Ethereal – you will be able to check for incoming traffic and ascertain whether it's a firewall issue or not.

Testing authentication with no password

Tip: If you have difficult authenticating the phone, try to authenticate with no username or password. To do this:

SIP Phone Registration without authentication (Internal)

1. Configure 3CX Phone System to accept SIP Phone (Extension 102 in this example)
2. Click “Extensions -> Add” in the Management Console
3. The “Extension Number” will be suggested by the Management Console (e.g.: 102) – leave the suggested number
4. The “Authentication ID” will be suggested by the Management Console to be equal to the “Extension Number” (e.g.: 102) – leave the suggested number
5. The “Authentication Password” should be left blank at this stage since we do not require authentication credentials matching.
6. All fields should be left with default values – scroll down to the bottom of the page and click OK.

Since in section 1.a.iv we left the password blank, the 3CX Phone System will accept any combination of Authentication ID and Authentication Password as being valid – Only the Extension Number will be checked to identify the SIP Phone registration.

One-way audio or missing audio issues

When a call is successfully established at the signaling layer, it may occur that the media layer may not be able to deliver some of the audio. Assuming that the device is correctly configured, then firewall issues must be in play. To assist with identifying which device has experienced this problem, the Media Server will post the following log entry every time a call is ended and no incoming audio packets were received:

MediaServerReporting:: RTPReceiver	[MS105000] Call(xx) Ln:10000@Gateway1: No RTP packets were received on nnnnnnnn@name: remoteAddr=aaa.bbb.ccc.ddd:ppppp, extAddr=aaa.bbb.ccc.ddd:ppppp, localAddr=aaa.bbb.ccc.ddd:ppppp

This will show the IP Address and Port which the caller extension is using to transmit media, and the external or internal IP Address and Port which Media Server is using to transmit media.

This information should be sufficient to identify the firewall configuration changes which need to be effected to resolve this issue.

Description of failsafe settings for most devices

Unsupported devices are untested, and therefore no guarantees are made regarding the functionality or otherwise of such devices. However, the following settings will afford the best possible scenario to get the device to work, providing the broadest possible compatibility, at the expense of features and bandwidth conservation.

- For internal devices, the device MUST NOT HAVE STUN configured – otherwise it will communicate the wrong (external) IP address to the PBX server. For external extensions, the device MUST HAVE STUN configured – otherwise it will communicate the wrong (internal) IP address to the PBX server.

- The device must NOT have outbound proxy configured – otherwise it will send some of the signaling messages to some other entity that is not the PBX server, or will send the signaling messages in an incorrect format, resulting in failed calls, or otherwise unpredictable behavior.
- The device must be configured to use G711a or G711u as its main codec.
- The 3CX Management Console settings for this device should specify a blank password – to bypass authentication issues.
- The 3CX Management Console advanced options settings for this device should have the following settings:
    - Extension/Gateway is external: off (for internal devices) or on (for external devices)
    - PBX delivers audio: on
    - Supports Re-Invite: off
    - Supports 'Replaces' header: off

Codecs

Codecs available for communication with 3CX Phone System are G711a, G711u, GSM, Speex, and iLBC. Incoming calls originating from devices or providers attempting to negotiate other codecs, without providing one of the above as a fallback, will be unable to deliver calls to the DR or VM - typically resulting in a dropped call.

Log Files

Location of logs assuming standard install location:

Setup Log: C:\Program Files\3CX PhoneSystem\install.log

Postgres: C:\Program Files\3CX PhoneSystem\Data\DB\pg_log\*.log

Apache: C:\Program Files\3CX PhoneSystem\Bin\Apache\logs\*.log

CallManager: C:\Program Files\3CX PhoneSystem\Data\Logs\3CXPhoneSystem*.log

MediaServer: C:\Program Files\3CX PhoneSystem\Data\Logs\3CXMediaServer*.log

VoiceMail: C:\Program Files\3CX PhoneSystem\Data\Logs\3CXVoiceBoxManager*.log

IVR: C:\Program Files\3CX PhoneSystem\Data\Logs\3CXIvrServer*.log

IVR: C:\Program Files\3CX PhoneSystem\Data\Logs\IvrPhp.log

After each restart of the 3CXPhoneSystem Service (Call Manager), all logs in C:\Program Files\3CX PhoneSystem\Data\Logs\*.* are backed up into a time-stamped folder in C:\Program Files\3CX PhoneSystem\Data\Logs\Backup so fresh logs are available for each restart.

Setup, Postgres, Apache logs are NOT included in support request file.

DTMF Considerations

DMTF tones are delivered either in-band (as a beep), or out-of-band via SIP or RTP signaling messages. 3CX Phone System supports both, as described hereunder:

1. Incoming stream delivers DTMF signals in-audio independently of codecs - in this case the 3CX Phone System Media Server listens to the audio stream, and will detect DTMF signals.
    a. Delivery to DR or VM: Efficiency of DTMF detection depends on audio quality. Dropped packets will also reduce audio quality.
    b. Delivery to some external party (typically via gateway or provider): DTMF strokes are recognized from the in-audio stream, and delivered to the external party in 2 forms – in-audio (leaving audio content unchanged) and additionally via RFC2833.
    c. Delivery to MS Exchange 2007 IVR: DTMF strokes are recognized from the in-audio stream, and delivered to the external party in 2 forms – in-audio (leaving audio content unchanged) and additionally via RFC2833. Please note that since MS Exchange 2007 IVR does not provide in-audio recognition, it will only use RFC2833 delivery mechanism.
2. Incoming stream delivers DTMF signals out-of-audio using either SIP-INFO or RFC-2833 mechanism, independently of codecs - in this case the DTMF signals are sent separately from the actual audio stream.
    a. Delivery to DR or VM: These are passed through as received.
    b. Delivery to some external party (typically via gateway or provider): These are passed through as received. The external party must support the corresponding delivery mechanism if DTMF strokes are to be recognized. Effectively this means that if DTMF is received with SIP-INFO, it is forwarded also as SIP-INFO. If your VoIP Provider requires RFC2833 DTMF delivery, then it will be necessary to ensure all SIP Phones are configured to deliver DTMF using RFC2833.
    c. Delivery to MS Exchange 2007 IVR: These are passed through as received.

Note:

SIP-INFO is not recommended for DTMF delivery, since it cannot deliver strokes synchronously with the audio stream, introducing timing artifacts (mainly because it's delivered using SIP, which is not a real-time mechanism for delivering media). It is very common for public services to NOT support SIP INFO, and it seems unlikely that such services will improve support for this delivery mechanism.