Oracle® Communications Unified Session Manager

Essentials Guide

Release S-CX6.3.5

Formerly Net-Net SIP Multimedia Xpress

January 2014

ORACLE®

Table of Contents

About This Guide

Overview

Oracle® Communications Unified Session Manager (USM) combines core session control with leading session border control functions to reduce the complexity and cost of delivering high-value, revenue generating SIP multimedia services. Oracle USM can be used to deliver a broad range of SIP services including residential or business voice, GSMA-defined Rich Communication Suite (RCS) services and fixed mobile convergence (FMC) for small subscriber populations or initial service rollouts.

Audience

This guide is written for network administrators and those who configure network devices. This document is meant to be used in conjunction with the Oracle Communications ACLI Configuration guide in order to provide full explanation of all functionality. Strong familiarity with the Oracle SBC is required for use with this guide.

The Oracle Communications Unified Session Manager Essentials Guide provides information related to the features, operation, and maintenance of the Oracle Communications Unified Session Manager Essentials Guide. Only experienced and authorized personnel should perform installation, configuration, and maintenance tasks.

About Oracle Software Releases

Release Version S-CX6.3.5 is supported on the Acme Packet 4500 series platforms.

Related Documentation

The following table lists the members that comprise the documentation set for this release:

Document Name	Document Description
Acme Packet 4500 System Hardware Installation Guide	Contains information about the components and installation of the Acme Packet 4500 system.
Release Notes	Contains information about the current documentation set release, including new features and management changes.
ACLI Configuration Guide	Contains information about the administration and software configuration SBC.
ACLI Reference Guide	Contains explanations of how to use the ACLI, as an alphabetical listings and descriptions of all ACLI commands and configuration parameters.

Document Name	Document Description
Maintenance and Troubleshooting Guide	Contains information about SBC logs, performance announcements, system management, inventory management, upgrades, working with configurations, and managing backups and archives.
MIB Reference Guide	Contains information about Management Information Base (MIBs), Enterprise MIBs, general trap information, including specific details about standard traps and enterprise traps, Simple Network Management Protocol (SNMP) GET query information (including standard and enterprise SNMP GET query names, object identifier names and numbers, and descriptions), examples of scalar and table objects.
Accounting Guide	Contains information about the SBC's accounting support, including details about RADIUS accounting.
HDR Resource Guide	Contains information about the SBC's Historical Data Recording (HDR) feature. This guide includes HDR configuration and system-wide statistical information.
Administrative Security Essentials	Contains information about the SBC's support for its Administrative Security license.

Revision History

This section contains a revision history for this document.

Date	Revision Number	Description
July 27, 2012	1.00	• Initial Release
Aug 30, 2013	1.10	• This document revision coincides with software release 6.3.5M1, which does not include any new features. This release includes changes generated by Net-Net SD software version 6.3.0M5P1.
Jan 14, 2014	1.20	• Removes reference indicating support on the Acme Packet 3800 platform.

1 Diameter Based Net-Net USM

The Net-Net USM functions in an IMS core. It communicates with the HSS to obtain Authorization, Authentication, S-CSCF assignment, and ultimately routing instructions. To accomplish these functions, the Net-Net USM now can perform the SIP registrar role in conjuunction with an HSS.

Message Authentication for SIP Requests

The Net-Net SMX authenticates requests by configuring the *sip authentication profile* configuration element. The *name* of this configuration element is either configured as a parameter in the *sip registrar* configuration element's *authentication profile* parameter or in the *sip interface* configuration element's *sip-authentication-profile* parameter. This means that the Net-Net USM can perform SIP digest authentication either globally, per domain of the Request URI or as received on a SIP interface.

After naming a *sip authentication profile*, the received methods that trigger digest authentication are configured in the *methods* parameter. You can also define which anonymous endpoints are subject to authentication based on the request method they send to the Net-Net SMX by configuring in the *anonymous-methods* parameter. Consider the following three scenarios:

1. By configuring the *methods* parameter with REGISTER and leaving the *anonymous-methods* parameter blank, the Net-Net SMX authenticates only REGISTER request messages, all other requests are unauthenticated.
2. By configuring the *methods* parameter with REGISTER and INVITE, and leaving the *anonymous-methods* parameter blank, the Net-Net SMX authenticates all REGISTER and INVITE request messages from both registered and anonymous endpoints, all other requests are unauthenticated.
3. By configuring the *methods* parameter with REGISTER and configuring the *anonymous-methods* parameter with INVITE, the Net-Net SMX authenticates REGISTER request messages from all endpoints, while INVITES are only authenticated from anonymous endpoints.

User Authorization

In an IMS network, the Net-Net USM requests user authorization from an HSS when receiving a REGISTER message. An HSS is defined on the Net-Net USM by creating a *home subscriber server* configuration element that includes a *name*, *ip address*, *port*, and *realm* as its basic defining data.

UAR/UAA Transaction

Before requesting authentication information, the Net-Net USM sends a User Authorization Request (UAR) to the HSS for the registering endpoint to determine if this user is allowed to receive service. The Net-Net USM populates the UAR's AVPs as follows:

- Public-User-Identity—the SIP AOR of the registering endpoint
- Visited-Network-Identity—the value of the *network-id* parameter from the ingress *sip-interface*.

- Private-User-Identity—the username from the SIP authorization header or the SIP AOR if the *aor-for-puid* parameter is enabled in the home subscriber server configuration element.
- User-Authorization-Type—always set to REGISTRATION_AND_CAPABILITIES (2)

The Net-Net USM expects the UAA to be either:

- DIAMETER_FIRST_REGISTRATION
- DIAMETER_SUBSEQUENT_REGISTRATION

Any of these responses result in the continued processing of the registering endpoint. Any other result code results in an error and a 403 returned to the registering UA. The next step is the authentication and request for the H(A1) hash.

SIP Digest User Authentication

Authentication via MAR/MAA

To authenticate the registering user, the Net-Net USM needs a digest realm, QoP, and the H(A1) hash. It requests these from a server, usually the HSS, by sending it a Multimedia Auth Request (MAR) message. The MAR's AVPs are populated as follows:

- Public-User-Identity—the SIP AOR of the endpoint being registered (same as UAR)
- Private-User-Identity—the username from the SIP authorization header or the SIP AOR if the *aor or puid* parameter is enabled. (Same as UAR)
- SIP-Number-Auth-Items—always set to 1
- SIP-Auth-Data-Item -> SIP-Item-Number—always set to 1
- SIP-Auth-Data-Item -> SIP-Authentication-Scheme—always set to SIP_DIGEST
- Server-Name—the *home-server-route* parameter in the *sip registrar* configuration element. It is the URI (containing FQDN or IP address) used to identify and route to this SMX.

The Net-Net USM expects the MAA to include a SIP-Auth-Data-Item VSA, which includes digest realm, QoP and H(A1) information as defined in RFC2617. The information is cached for subsequent requests. Any result code received from the HSS other than DIAMETER_SUCCESS results in a 403 error response returned for the original request.

The MAR/MAA transaction is conducted with the server defined in the *credential retrieval config* parameter found in the the *sip-authentication profile* configuration element. This parameter is populated with the name of an *home-subscriber-server* configuration element.

SIP Authentication Challenge

When the Net-Net SMX receives a response from the HSS including the hash value for the user, it sends a SIP authentication challenge to the endpoint, if the endpoint did not provide any authentication headers in its initial contact the with Net-Net SMX. If the endpoint is registering, the Net-Net SMX replies with a 401 Unauthorized message with the following WWW-Authenticate header:

```
WWW-Authenticate: Digest realm="atlanta.com",
domain="sip:boxesbybob.com", qop="auth",
nonce="f84f1cec41e6cbe5aea9c8e88d359", opaque="", stale=FALSE,
algorithm=MD5
```

If the endpoint initiates any other request to the Net-Net SMX besides REGISTER, the Net-Net SMX replies with a 407 Proxy Authentication Required message with the following Proxy-Authenticate header:

```
Proxy-Authenticate: Digest realm="atlanta.com", qop="auth",
nonce="f84f1cec41e6cbe5aea9c8e88d359", opaque="", stale=FALSE,
algorithm=MD5
```

Authentication Header Elements

- Domain—A quoted, space-separated list of URIs that defines the protection space. This is an optional parameter for the "WWW-Authenticate" header.
- Nonce—A unique string generated each time a 401/407 response is sent.
- Qop—A mandatory parameter that is populated with a value of "auth" indicating authentication.
- Opaque—A string of data, specified by the Net-Net USM which should be returned by the client unchanged in the Authorization header of subsequent requests with URIs in the same protection space.
- Stale—A flag indicating that the previous request from the client was rejected because the nonce value was stale. This is set to true by the SD when it receives an invalid nonce but a valid digest for that nonce.
- Algorithm—The Net-Net SMX always sends a value of "MD5"

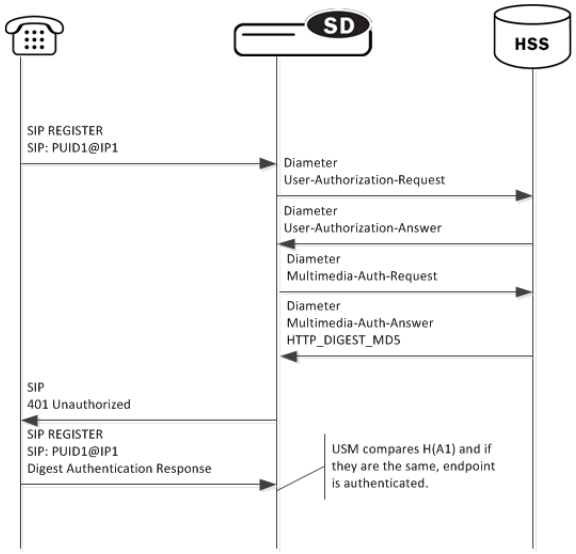
SIP Authentication Response

After receiving the 401/407 message from the Net-Net USM, the UA resubmits its original request with an Authorization: header including its own internally generated MD5 hash.

Net-Net USM Authentication Check

At this point, the Net-Net USM has received an MD5 hash from the HSS and an MD5 hash from the UA. The Net-Net SMX compares the two values and if they are identical, the endpoint is successfully authenticated. Failure to match the two hash values results in a 403 or 503 sent to the authenticating endpoint.

The following image shows the User Authorization and Authentication process:



The Net-Net USM acts as a SIP Registrar and updates an HSS with the state of its registrants.

Net-Net USM as Registrar

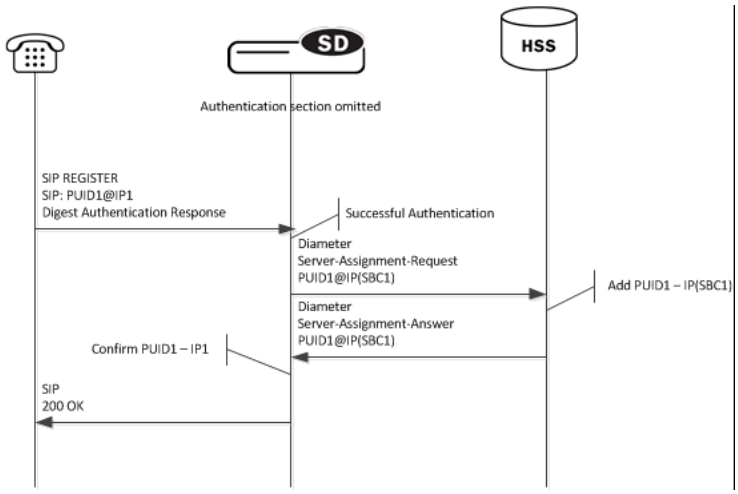
Creating a sip registrar configuration element enables the Net-Net USM to act as a SIP registrar. When registration functionality is enabled, the Net-Net USM actually registers endpoints rather than only caching and forwarding registrations to another device. Net-Net USM registry services are enabled globally per domain, not on individual SIP interfaces or other remote logical entities.

On receiving a REGISTER message, the Net-Net USM checks if it is responsible for the domain contained in the Request-URI as defined by the *domains* parameter and finds the corresponding *sip registrar* configuration. This is a global parameter—all messages are checked against all *sip registrar domains*. Thus you could create one sip registrar configuration element to handle all .com domains and one sip registrar configuration element to handle all .org domains. The Net-Net USM begins registrar functions for all requests that match the configured domain per sip-registrar configuration element.

A UA is considered registered once a SAA assignment is received from the HSS, after which the Net-Net SMX sends a 200 OK message back to the registering UA.

New Registration

The following image shows a simplified call flow for a registering user:



HSS Server Assignment

As the Net-Net USM registers UAs, it requests to assign itself as the S-CSCF for the registering AoR. The Net-Net USM's S-CSCF identity is configured in the *home-server-route* parameter in *sip-registrar* configuration element. This is a entered as a SIP URI (containing FQDN or IP address) and is used to identify and route messages to this Net-Net USM on behalf of the registered user.

Server Assignment Messages

The Net-Net USM sends a Server Assignment Request (SAR) to the HSS requesting to confirm the SIP or SIPS URI of the SIP server that is currently serving the user. The SAR message also serves the purpose of requesting that the Diameter server send the user profile to the SIP server. The SAR's AVPs are populated as follows:

- Public-User-Identity—the SIP AOR of the endpoint being registered (same as UAR)

- Private-User-Identity—the username from the SIP authorization header or the SIP AOR if the *aor-for-puid* parameter in the *home subscriber server* configuration is enabled. (Same as UAR)
- Server-Name—the *home server route* parameter in the sip-registrar configuration element. It is the FQDN or IP address used to identify and route to this Net-Net USM sent as a URI.
- Server-Assignment-Type—the value of this attribute depends upon the registration state:
  - REGISTRATION (1)—for all new and refreshing registrations.
  - Set to TIMEOUT_DEREGISTRATION (4)—when the contact is unregistered due to expiration. This occurs if the *force-unregistration* option is configured in the *sip config*.
  - USER_DEREGISTRATION (5)—when the contact is unregistered by the user (contact parameter expires=0).
- User-Data-Already-Available—always set to USER_DATA_ALREADY_AVAILABLE (1)

Server-Assignment-Response

The Net-Net USM expects a DIAMETER_SUCCESS code in the SAA to indicate that the assignment was successful. Then a 200 OK response is returned to the registering user. Any other Diameter result code is an error and result in an error response for the original REGISTER request (by default 503) and the contacts to be invalidated in the registration cache.

Register Refresh

When a UA sends a register refresh, the Net-Net USM first confirms that the authentication exists for that UA's registration cache entry, and then is valid for the REGISTER refresh. (If a valid hash does not exist for that AoR, then the Net-Net USM sends an MAR to the HSS to retrieve authentication data once again).

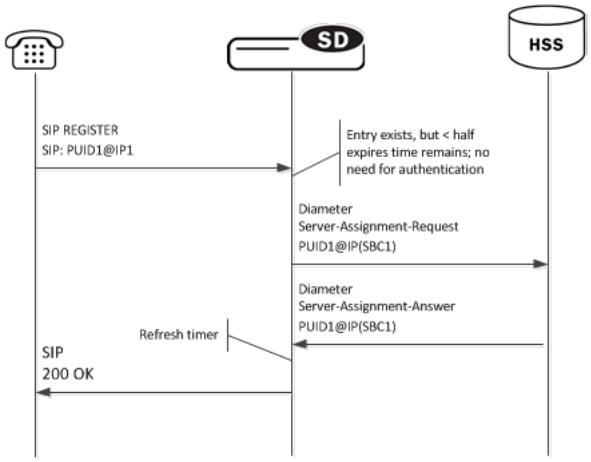
Next, the Net-Net USM determines if the it can perform a local REGSITER refresh or if the HSS needs to be updated. If any of the following 3 conditions exists for the re-registering UA, theNet-Net USM updates the HSS:

1. The *location update interval* timer has expired—This value, configured in the *sip registrar* configuration element ensures that HSS database always has the correct Net-Net USM address by periodically sending SARs for each registered contact.
2. The message's call-id changes while the f**orward-reg-callid-change** option in the *sip config* configuration element is set. This covers the case where the UA changes the Net-Net USMs through which it attaches to the network.
3. The REGISTER message's Cseq has skipped a number. This covers the case in which a user registered with SMX1, moves to SMX2 , and then returns to SMX1.

If the Net-Net USM updates the HSS database because of matching one of the above conditions, the access side expiration timer per contact is reset to the REGISTER message's Expires: header value, and returned in the 200 OK. This happens even in the case when the reREGISTER was received in the first half of the previous Expires period. In addition, the core-side *location update interval* timer are refreshed on both active and standby.

When the above three conditions are not met, the registration expiration proceeds normally.

If the timer has not exceeded half of its lifetime, a 200 OK is returned to the UA. If the timer has exceeded half of its lifetime, the Net-Net USM just refreshes the access-side expiration timer; the registration cache exipiration timer for that AoR begins its count again.



Core-side SAR Lifetime

The Net-Net USM maintains a timer for user registrations per SAR on the core side as specified above. The core-side SAR lifetime timer is configured in the *location update interval* parameter in the *sip registrar* configuration element. This timer ensures that the HSS always has the correct SMX address, by sending SAR messages periodically.

Entry Unregistration

Because AoRs and not contacts are referenced by the HSS, an AoR is valid and should not be removed from HSS until all associated contacts have been removed or expired. If all the contacts are removed for an AoR by receiving REGISTER messages with `Expires: 0` header, then the SAR sent to the HSS includes Server-Assignment-Type of USER_DEREGISTRATION (5).

When the *force-unregister* option in the *sip config* is enabled, then the HSS is explicitly updated when all of the contacts for an AoR have expired. This event prompts the Net-Net USM to send a SAR to the HSS using the Server-Assignment-Type of TIMEOUT_DEREGISTRATION (4).

The HSS can send a Registration-Termination-Request to request removing a registration, which corresponds to entries in the Net-Net USM's registration cache. When an RTR is received, the following AVPs are expected:

- Private-User-Identity—Username of the user, which is being de-registered.
- Associated-Identities—The Private-Id's in the same subscription which need to be de-registered. (optional)
- Public-Identity—One or more public-Id's of the user being de-registered. (optional)

For the AoR specified by the Private-User-Identity AVP, all associated contacts are removed in the registration cache. The Net-Net USM sends a Registration Termination Answer is sent to HSS to indicate success.

User Registration based on Reg-ID and Instance-ID (RFC 5626)

Sometimes a user's device reregisters from a different network than from its original registration. This event should be considered a location update rather that a completely new registration for the Contact. The Net-Net USM can perform this way by considering the endpoint's reg-id and instance-id parameters defined in RFC 5626.

The Net-Net USM identifies new REGISTER received on a different access network as a location update of the existing binding between the Contact and AoR. Without this feature, the Net-Net USM would create a new binding and leave the old binding untouched in the local registration cache/ENUM database. This scenario is undesirable and leads to unnecessary load on various network elements including the Net-Net USM itself.

The following conditions must be matched to equate a newly registering contact as a location update:

For a received REGISTER:

1. The message must not have more than 1 Contact header while 1 of those Contact headers includes a reg-id parameter. (failure to pass this condition prompts the Net-Net USM to reply to the requester with a 400 Bad Request).
2. The Supported: header contains **outbound** value
3. The Contact header contains a **reg-id** parameter
4. The Contact header contains a **+sip.instance** parameter

After these steps are affirmed, the Net-Net USM determines if it is the First hop. If there is only one Via: header in the REGISTER, the Net-Net USM determines it is the first hop and continues to perform Outbound Registration Binding processing.

If there is more than 1 Via: header in the REGISTER message, the Net-Net USM performs additional validation by checking for a Path: header corresponding to the last Via: includes an ob URI parameter, Outbound Registration Binding may continue.

If the Net-Net USM is neither the first hop nor finds an ob URI in Path headers, it replies to the UA's REGISTER with a 439 First Hop Lack Outbound Support reply.

reREGISTER Example

The user (AoR) bob@example.com registers from a device +sip.instance= <urn:uuid:0001> with a reg-id ="1", contact URI = sip:1.1.1.1:5060. A binding is be created for bob@example.com+<urn:uuid:0001>+reg-id=1 at sip:1.1.1.1.:5060.

Next, Bob@example.com sends a reREGISTER with the same instance-id but with a different reg-id = 2 and contact URI = sip:2.2.2.2:5060.

The previous binding is removed. A binding for the new contact URI and reg-id is created. bob@example.com+<urn:uuid:0001>+reg-id=2 at sip:2.2.2.2:5060

Outbound Registration Binding Processing

An outbound registration binding is created between the AoR, instance-id, reg-id, Contact URI, and other contact parameters. This binding also stores the Path: header.

Matching re-registrations update the local registration cache as expected. REGISTER messages are replied to including a Require: header containing the outbound option-tag.

If the SMX receives requests for the same AOR with some registrations with reg-id + instance-id and some without them, the SMX will store them both as separate Contacts for the AOR; The AoR+sip.instance+reg-id combination becomes the key to this entry.

Net-Net USM Licensing

The Net-Net USM requires three licenses: Registration Cache Limit, Cx, SIP Authorization/Authentication.

For CX-based Net-Net USM, the Cx license reveals the *home subscriber server* configuration element and the SIP Authorization/Authentication license reveals the *SIP Authentication Profile* configuration element. Configuring both configuration elements is required to operate a Net-Net USM. Refer to the Licensing and Database Registration Limits (24) section for the third license required for Net-Net USM operation.

Refer to the Net-Net SBC ACLI Configuration guide, Getting Started chapter for how to install licenses in your system.

ACLI Instructions

The following configuration enables the Net-Net USM to authorize and authenticate registering users. In addition it sets the Net-Net USM to request itself as the S-CSCF for the registering users.

home subscriber server

To configure a home subscriber server (HSS):

1. In Superuser mode, type **configure terminal** and press <Enter>.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type **session-router** and press <Enter> to access the session router path.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type **home-subscriber-server** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.

   ```
   ACMEPACKET(session-router)# home-subscriber-server
   ACMEPACKET(home-subscriber-server)#
   ```

4. **name**—Enter the name for this home subscriber server configuration element to reference from other configuration elements.
5. **state**—Set this to **enabled** to use this configuration element.
6. **address**—Enter the IP address of this HSS.
7. **port**—Enter the port which to connect on of this HSS, the default value is 80.
8. **realm**—Enter the realm name where this HSS exists.
9. **aor-for-puid**—Set this to **enabled** to use the message's AoR as the Private User Identity value.
10. Type **done** when finished.

SIP Authentication Profile

To configure the SIP Authentication Profile:

1. In Superuser mode, type **configure terminal** and press <Enter>.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type **session-router** and press <Enter> to access the session router path.

```
ACMEPACKET(configure)# session-router
```

3. Type **sip-authentication-profile** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# sip-authentication-profile
ACMEPACKET(sip-authentication-profile)#
```

You may now begin configuring the SIP Authentication Profile configuration element.

4. **name**—Enter the name of this SIP authentication profile that will be referenced from a SIP registrar (or a SIP interface) configuration element.
5. **methods**—Enter all the methods that should be authenticated. Enclose multiple methods in quotes and separated by commas.
6. **anonymous-methods**—Enter the methods from anonymous users that require authentication. Enclose multiple methods in quotes and separated by commas.
7. **digest-realm**—Leave this blank for Cx deployments.
8. **credential-retrieval-method**—Enter CX.
9. **credential-retrieval-config**—Enter the *home-subscriber-server* name used for retrieving authentication data.
10. Type **done** when finished.

SIP Interface

The full SIP interface should be configured according to your network needs. Please refer to the Net-Net SBC ACLI Configuration Guide.

To configure a SIP Digest Authentication on a specific SIP Interface:

1. In Superuser mode, type **configure terminal** and press <Enter>.

```
ACMEPACKET# configure terminal
```

2. Type **session-router** and press <Enter> to access the session router path.

```
ACMEPACKET(configure)# session-router
```

3. Type **sip-interface** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# sip-interface
ACMEPACKET(sip-interface)#
```

4. Type **select** and choose the number of the pre-configured sip interface you want to configure.

```
ACMEPACKET(sip-interface)# select
<realm-id>:
1: private 192.168.101.17:5060
2: public  172.16.101.17:5060

selection: 1
```

5. **registration-caching**—Set this parameter to **enabled.**
6. **sip-ims-feature**—Set this parameter to **enabled.**
7. **sip-authentication-profile**—Set this to the name of an existing *sip-authentication* profile if you wish to authenticate per SIP interface.
8. Type **done** when finished.

SIP Registrar

To configure the Net-Net USM to act as a SIP Registrar:

1. In Superuser mode, type **configure terminal** and press <Enter>.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type **session-router** and press <Enter> to access the session router path.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type **sip-registrar** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.

   ```
   ACMEPACKET(session-router)# sip-registrar
   ACMEPACKET(sip-registrar)#
   ```

4. **name**—Enter a name for this SIP registrar configuration element.
5. **state**—Set this to **enabled** to use this SIP registrar configuration element.
6. **domains**—Enter one or more domains that this configuration element will invoke SIP registrarion for. Wildcards are valid for this parameter. Multiple entries can be entered in quotes, separated by commas.
7. **subscriber-database-method**—Set this to **CX**.
8. **subscriber-database-config**—Enter the home-subscriber-server configuration element *name* that will handle REGISTER messages for this domain. The HSS configuration element includes the actual IP address of the server that SAR's are sent to.
9. **authentication-profile**—Enter a *sip-authentication-profile* configuration element's *name*. The *sip authentication profile* object referenced here will be looked up for a REGISTER message with a matching *domain* in the request URI. You may also leave this blank for the receiving SIP Interface to handle which messages require authentication if so configured.
10. **home-server-route**—Enter the identification for this Net-Net USM that will be sent as the Server-Name in MAR and SAR messages to the HSS. This value should be entered as a SIP URI.
11. **location-update-interval**—Keep or change from the default of 1400 minutes (1 day). This value is used as the timer lifetime for core-side HSS updates.
12. Type **done** when finished.

Message Routing

Once the HSS assigns the Net-Net USM as the S-CSCF for a registered endpoint, the Net-Net USM is ready to route SIP traffic into the IMS core. The Net-Net USM performs routing in two ways depending on the *routing precedence* parameter in the *sip registrar* as **registrar** (HSS) or **local policy**. Registrar routing uses the configured SIP registrar/HSS for its initial, destination address query, while local policy routing lets you configure routing decisions within the Net-Net USM's local policy routing functionality.

If Net-Net USM fails to route the message, it falls back to trying by the other method. That is, if the system is set to registrar routing and this method fails, the Net-Net USM then attempts to route the call via local policy, and vice versa. A Net-Net USM deployment generally functions in registrar mode, but local policy may be used for more control over routing decisions, performed on the individual Net-Net USM.

Local Policy Routing

To specify an HSS to query, a *policy attribute* with next-hop `cx:<home-subscriber-server-name>` is configured in the *local-policy*. This assumes that the

sip registrar configuration element's *routing precedence* parameter is set to **local-policy**.

If the target endpoint is unreachable or there is no matching local policy, the Net-Net USM falls back to query the subscriber database (HSS/ENUM) for determining the next hop.

If any policy-attributes > next-hop parameters originally stepped through had a next-hop cx: or enum: the fall back to registrar is not performed. This is because the Net-Net USM has already queried the database once and it would be redundant.

Also, when the *routing precedence* parameter is set to **local-policy** and the Net-Net USM uses a local subscriber database, it checks the registration cache for a pre-existing matching contact in the INVITE to find where to forward request.

Registrar Routing

When **route-precidence** parameter is set to **registrar** and the lookup via the registrar fails, you can configure the Net-Net USM to not fallback to make a local policy lookup for the destination. This is configured by adding the **skip-local-policy-lookup** option to the *sip-registrar* configuration element.

Default Egress Realm

The *sip registrar* configuration element should be configured with a default *egress realm id*. This is the name of the *realm config* which defines the IMS control plane through which all Net-Net USMs, HSSs, and other network elements communicate and exchange SIP messaging. It is advisable to configure this parameter in order to ensure well defined reachability among Net-Net USMs.

Originating Net-Net SMX

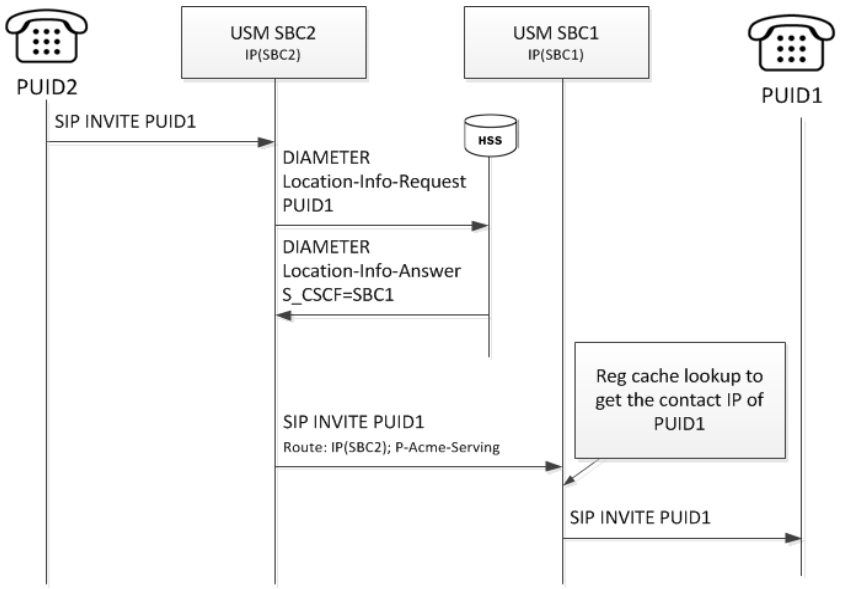
Requests from endpoints registered on a SIP interface are routed via registrar or local-policy routing. The From: address is not checked against HSS. When the Net-Net USM receives an INVITE, for example, it determine how forward the message. The Net-Net USM queries the HSS with an LIR to find the S-CSCF for the received request.

LIR/LIA Transaction

The LIR includes the Public-User-Identity AVP containing UA's actual PUID. The HSS responds with the assigned S-CSCF server (a Net-Net USM) for this PUID. The answer is the form of a Location Info Answer (LIA) and is in the Server Name AVP. If the S-CSCF specified in this AVP is not the current Net-Net USM, then the INVITE is forwarded to the address specified in the LIR.

If the S-CSCF returned in the LIR is this Net-Net USM, then the AoR from the request URI is found in the registration cache and the message is forwarded to that endpoint. When the registration cache entry does not exist or is invalid, local policy processing continues with the next policy-attribute following "stop-recurse" rules. If there are no other routes, then a 404 is sent to the UA who sent the INVITE.

If the HSS returns a remote S-CSCF in the LIA that is not this Net-Net USM, the INVITE is forwarded to that S-CSCF.



Net-Net USM Identification Parameter

When forwarding the message into the core network, the Net-Net USM inserts its egress IP address and a `P-Acme-Serving` parameter into the Route: header. This is configured by setting the *add lookup parameter* parameter to **enabled** in the *home subscriber server* configuration element.

Upon seing that a received request contains the `P-Acme-Serving` parameter, the Net-Net USM interprets that an LIR is unnecessary. It then checks its registration cache and forward the message appropriately to the target.

ACLI Instructions

SIP Registrar

To configure a SIP registrar configuration element for message routing:

1. In Superuser mode, type **configure terminal** and press <Enter>.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type **session-router** and press <Enter> to access the session router path.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type **sip-registrar** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.

   ```
   ACMEPACKET(session-router)# sip-registrar
   ACMEPACKET(sip-registrar)#
   ```

4. Type **select** and choose the number of the pre-configured sip interface you want to configure.
5. **routing-precedence**— Set this to either **registrar** or **local-policy** depending on your deployment.
6. **egress-realm-id**—Enter the default egress realm for Net-Net USM messaging.
7. Type **done** when finished.

Home Subscriber Server

1. In Superuser mode, type **configure terminal** and press <Enter>.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type **session-router** and press <Enter> to access the session router path.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type **home-subscriber-server** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.

   ```
   ACMEPACKET(session-router)# home-subscriber-server
   ACMEPACKET(home-subscriber-server)#
   ```

4. Type **select** and choose the number of the pre-configured sip interface you want to configure.
5. **add-lookup-parameter**—Set this to enabled to insert the P-Acme-Serving parameter into the Route header.
6. Type **done** when finished.

HSS Initiated User Profile Changes

The Net-Net USM can receive Push Profile Request (PPR) messages from an HSS and update the state of the IMS User Profile and associated subscription information it has cached locally. The SIP digest authentication information can also be updated and reassociated with an AoR in case that has changed too. The Net-Net USM expects to receive the following AVPs in a PPR message.

- Private-User-Identity—the username, whose subscription/authentication data has changed.
- SIP-Auth-Data-Item—if present new authentication data is included here.
- User-Data—if present new User data is included here.
- Charging-Information—if present new charging information is included here.

The Net-Net USM replies to an HSS's PPR in a PPA message with the following AVPs:

- Result-Code—indicates Diameter base protocol error. Valid errors for in a PPA are:
  - DIAMETER_SUCCESS—The request succeeded.
  - DIAMETER_ERROR_NOT_SUPPORTED_USER_DATA—The request failed. The SMX SBC informs HSS that the received user information contained information, which was not recognized or supported.
  - DIAMETER_ERROR_USER_UNKNOWN—The request failed because the Private Identity is not found in SMX SBC.
  - DIAMETER_ERROR_TOO_MUCH_DATA—The request failed. The SMX SBC informs to the HSS that it tried to push too much data into the SMX SBC.
  - DIAMETER_UNABLE_TO_COMPLY—The request failed.
- Experimental-Result—indicates diameter application (3GPP/Cx) error if present.

Licensing and Database Registration Limits

The Net-Net USM (and Net-Net SBC) limit the number of unexpired registration cache entries globally. The total number of system registrations is configured with the *registration cache limit* parameter in the *sip config* configuration element.

The Net-Net USM also limits the number of registration cache entries that were obtained from a User Subscriber Database; only REGISTERs that prompted the database query are counted here. As User Subscriber Database entries are added and removed, this counter is updated accordingly. Note that it is the actual number of *SD-contacts* that count against the license limit. Discrete database registration license values range from 20,000 through 500,000 in increments of 20,000.

When a registering contact is rejected because it will exceed one of these limits, the Net-Net USM sends a 503 message to the registering endpoint.

Refer to the Net-Net 4000 ACLI Configuration Guide, chapter 2 “Getting Started”, Software Licensing section for how to install a license.

Database Registration Limit Alarm

By default, a major alarm is enabled when 98% or more of the licensed number of Database Registrations are used. This alarm is cleared when the number of database registrations falls below 90%. You can configure minor and critical alarms when crossing configured thresholds and you can also reassign the major alarm. This is configured in by creating a *system-config* > *alarm-threshold* sub element with *type* of **database-registration**.

3GPP Compliance

P-Asserted-Id in Requests and Dialogs

When an AoR is successfully registered through the Net-Net USM, the list of implicitly registered public IDs is returned from the HSS. The set of implicitly registered public IDs includes the explicitly registered Public-id and may include wild-carded public-ids. If there are no implicitly registered public-ids, then the implicit set returned by HSS will at least contain the explicitly registered public-id.

Based on local configuration and network conditions, the registering UE may or may not be trusted.

When a non-trusted UE sends an initial request for a dialog or a request for a standalone transaction to the Net-Net USM, the P-Asserted-Identity to be inserted in the outgoing request is formed as follows:

1. If the request does not have a P-Preferred-Identity header field or none of the P-Preferred-Identity header fields included in the request match any of the registered public user identities, then the Net-Net USM inserts the default Public-Identity in the outgoing P-Asserted-Identity header.
2. If the request includes one or more P-Preferred-Identity header fields which match the registered public user identities, then the Net-Net USM includes only the first P-Preferred-Identity header field.
3. If the request includes one or more P-Asserted-Identity header fields which do not match the registered public user identities, then the Net-Net USM inserts the default Public-Identity in the outgoing P-Asserted-Identity header.

When a trusted UE sends an initial request for a dialog or a request for a standalone transaction to the Net-Net USM, the P-Asserted-Identity to be inserted in the outgoing request is formed as follows:

1. If the request does not have a P-Preferred-Identity header field or none of the P-Preferred-Identity header fields included in the request match any of the registered public user identities, then the Net-Net USM inserts the default Public-Identity in the outgoing P-Asserted-Identity header.
2. If the request includes one or more P-Preferred-Identity header fields which match the registered public user identities, then the Net-Net USM inserts the first P-Preferred-Identity header field.
3. If the request includes one or more P-Asserted-Identity header fields, then the Net-Net USM will use the first P-Asserted-Identity header field.

- The contents of the From header field do not form any part of this decision process.
- P-Preferred-Identity header fields will always be removed.

The Default Public Identity is the first appearing, non-barred identity in the set of implicitly registered Public User Identities.

To enable this behavior, add the **pai-comply-to-3gpp** option in the *sip config* configuration element.

P-Associated-URI in 200 OK

In a 200 OK response to a UE on a successful registration, the Net-Net USM includes a P-Associated-URI header. This header includes the list of registered, distinct public user identities and the associated set of implicitly registered distinct public user identities.

When there are no associated implicit public identities, only the explicitly registered Public User Identity is included.

Other Diameter Cx Configuration

Origin Host AVP and Origin Realm AVP Configuration for Cx

You can configure the values sent in the origin-host and origin-realm AVPs when the Net-Net USM communicates with a server over the Cx interface.

The configuration parameters are located in the home-subscriber-server configuration element. The two parameters used to configured the AVPs are origin-realm and origin-host-identifier. The AVPs are constructed as follows:

```
Origin Host AVP = <origin-host-identifier>.<origin-realm>
Origin Realm AVP = <origin-realm>
```

If the **origin-realm** is not configured, then the realm parameter in the home-subscriber-server configuration element will be used as the default. If origin-host-identifier is not configured, then the name parameter in the home-subscriber-server configuration element will be used as the default.

If neither parameter is configured, then the AVPs are constructed as follows:

```
Origin Host = <HSS Config name>.<HSS Config realm>.com
Origin Realm AVP = <HSS Config realm>
```

ACLI Instructions

1. In Superuser mode, type **configure terminal** and press <Enter>.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type **session-router** and press <Enter> to access the session router path.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type **home-subscriber-server** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# home-subscriber-server
ACMEPACKET(home-subscriber-server)#
```

4. **origin-realm**—Set this to a string for use in constructing unique Origin Host and Origin Realm AVPs.
5. **origin-host-identifier**—Set this to a string for use in constructing a unique Origin Host AVP.
6. Save your work.

Initial Filter Criteria (IFC)

The Net-Net USM, acting as a S-CSCF, downloads a set of rules known as Initial Filter Criteria (IFC) from the HSS/AS. IFCs are downloaded over the Cx interface.

iFCs are a way for an S-CSCF to evaluate which ASs should be involved in the call flow for a given user agent (UA). iFCs are functionally defined by Boolean statements, whose component parts are expressed in XML; they reference the destination AS(s) where a desired service is provided.

IFC Evaluation

IFCs are evaluated as described in: 3GPP TS 29.228 . The Net-Net USM supports all tags found in the 3GPP initial filter criteria specifications. An IFC is evaluated until its end, after which the call flow continues as expected.

SIP Registration

When the Net-Net USM receives an authenticated REGISTER request from a served UE, it sends an SAR request to the HSS to obtain an SAA which includes iFCs associated with the UE's subscription. Before the Net-Net USM responds to the UE with a 200 OK, all iFCs indicating third party registration procedures are evaluated and performed.

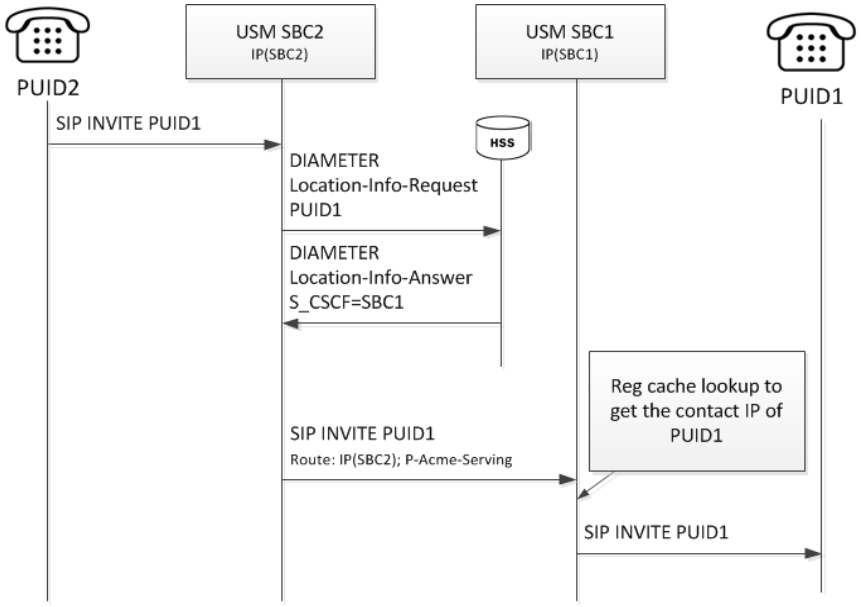
SIP Call

The Net-Net USM evaluates all IFC logic to determine that messages with matching characterists are sent to the proper AS specified in the iFC evlatuation. In this INVITE, the Net-Net USM adds two Route headers. The first (top) route header contains the target AS's URI. The second Route parameter is built using the IP address of the egress SIP interface and contains the ODI as a parameter. For example:

```
INVITE SIP:test@service.com
...
Route:2.2.2.2;lr
Route:1.1.1.1:5060;lr;smx_odi=1
```

If the AS routes the call back to the Net-Net USM, it is expected to include the ODI parameter that it received from the Net-Net USM, unchanged. The presence of the ODI parameter indicates that IFC evaluation needs to continue from where it left off for this call. If this continuation of IFC evaluation results in another AS URI, the Net-Net USM initiates a request towards that AS this time with a new ODI. In this way, the ODI is a state-signifier of Service Point Triggers.

The process continues until IFC evaluation is completed. Below is an example of an IFC evaluation completing after two iterations.



The iFC continues to be evaluated completely which may result in the INVITE being forwarded to additional ASs. At the conclusion of evaluating the iFC, the Net-Net USM checks if the target of the initial request is registered to itself, or not. If the UE is not registered locally the Net-Net USM forwards the request by regular means into the network. If the the target UE is registered locally, the proceeds to obtain iFCs for the target and begin iFC evaluation for the terminating side of the call.

Evaluating Session Case in the P-Served-User Header

The P-served-user header field conveys the identity of the served user, the session case that applies to the particular communication session, and application invocation, as defined in RFC 5502 and TS 24.229. The Session Case (sescase) and Registration State (regstate) parameters are either populated by the UE originating the message or by the Net-Net USM after it determines if a request is originating or terminating, and registered or unregistered

The P-served-user header is created and added to an outgoing request if the next hop is trusted. A trusted next hop is an entity defined by a session agent whose trust-me parameter is enabled. Likewise, the P-served-user header is stripped if the request is forwarded to a next hop that is not known to be trusted.

When the Net-Net USM creates a P-served-User header, the user value in the originating case is the user value found in the P-Asserted-Identity header field. In the terminating case, the user value is taken from the Request URI.

Supported Sessioncase and Registration State

The following cases are suppported for IFC evaluation. Conditions for classifying the calls as such are listed below.

Originating request by a UE, Registered User

When the Net-Net USM receives an Initial request, it is validated as an originating request from a registered user when the following conditions are met:

- The request is a dialog creating request or a standalone request.
- There is no "odi" parameter in the top route of the request.
- The regstate and sescase parameters of the P-served-user indicate for this to be treated as originating request for a registered user OR "The request is received from a registered contact.

Originating request by a UE, Unregistered User

This case is not supported.

Terminating Requests to a UE, Registered User

When the Net-Net USM receives an Initial request, it is validated as a terminating request towards a registered when the following conditions are met:

- The request is a dialog creating request or a standalone request.
- There is no "orig" parameter in the top route of the request.
- There is no "odi" parameter in the top route of the request.
- The regstate and sescase parameters of the P-served-user indicate for this to be treated as terminating request for a registered user OR the request is finished with originating services if applicable and the request is destined to a user who is currently registered with the SMX.

Terminating Requests to a UE, Unregistered User

See the following section IFC Support for Unregistered Users (28) for this case.

Additional Options

- The Net-Net USM can populate the top Route: header with the sescase value for ASs that require it. In such a case, the parameter is created as either call=orig or call=term. This behavior is enabled by configuring the **add-sescase-to-route** option in the *ifc-profile*.
- When the dialog-transparency parameter in the sip-config is set to enabled and your networkincludes multiple ASs, you should add the **dialog-transparency-support** option in the *ifc-profile*.

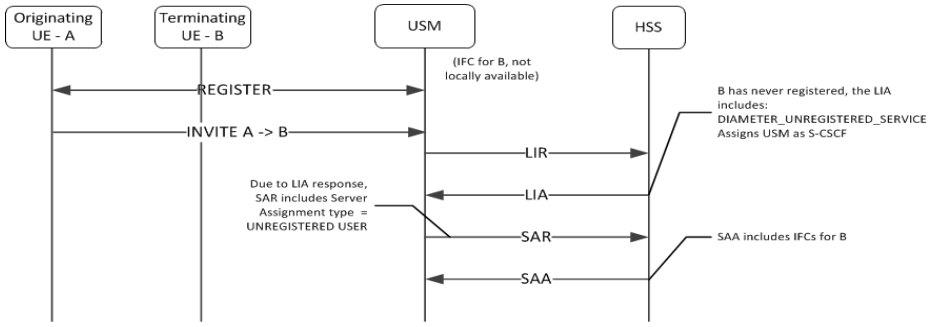
IFC Support for Unregistered Users

The Net-Net USM can downloading Initial Filter Criteria (IFC) from the HSS for unregistered users. Currently, this is performed when the Net-Net USM is terminating a call toward an unregistered user.

UE-terminating requests to an unregistered user

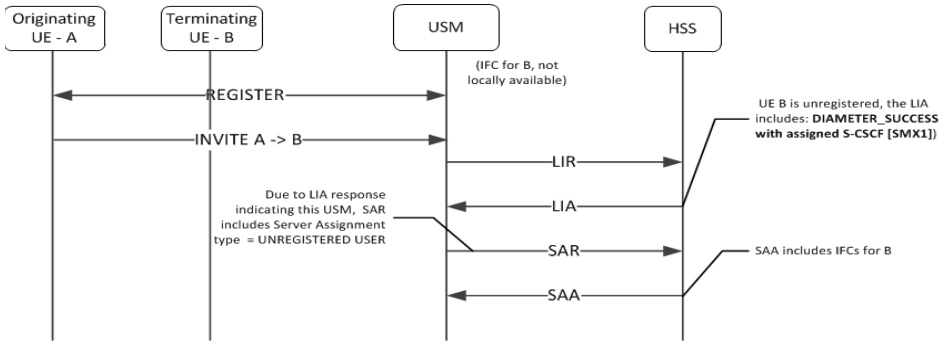
The Net-Net USM downloads and executes IFCs for the terminating end of calls. The following call flows indicate possible cases for the terminating unregistered user.

Terminating UE - Not Registered

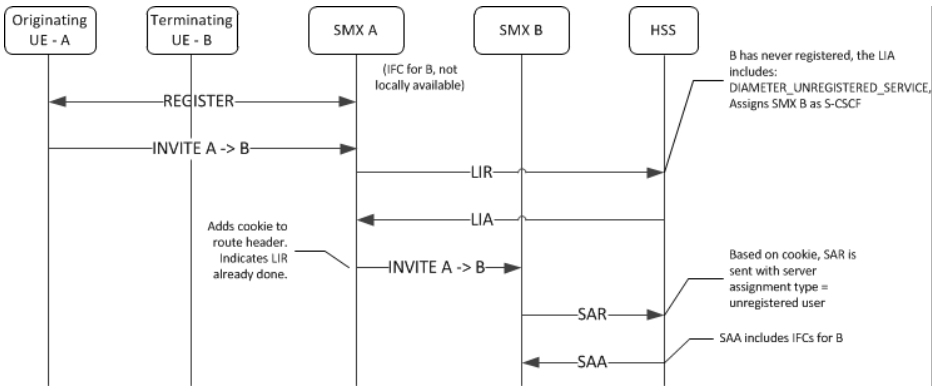


Terminating UE - Not registered

UE originally registered as a consequence of an originating or terminating request or an S-CSCF has stored the user profile.



Terminating UE - Not Registered, Served by other SMX



UE Subsequent Registration

If the Net-Net USM has a cached IFC downloaded for an unregistered UE who later registers to that SMX, the cached IFC will be cleared and updated with the IFC downloaded by the registration process.

Caching the Downloaded IFC

When the Net-Net USM downloads IFCs for unregistered users they are saved to a local cache. If the IFC cache fills up, an older cached IFC for a user is released.

Optimizing IFC Updates

The Net-Net USM aims to reduce the number of IFC updates traversing the network to save bandwidth and transactional overhead. Unless the unregistered UE's IFC entry has been deleted because of exhausting cache space, the following optimizations are performed:

- If IFCs are available locally, then an SAR/SAA operation to download IFCs will not be performed.
- If a previous IFC download operation did not return any IFCs, then subsequent calls to that unregistered user will not invoke the SAR/SAA messaging to download IFCs.

Push Profile Request (PPR) updates

The HSS can push service profile updates for private IDs. The Net-Net USM can process PPR updates for unregistered entities. If the user entry has been deleted because IFC cache space has been exhausted, the PPRs will not be processed.

ACLI Instructions

SIP Registrar

To create an IFC Profile:

1. In Superuser mode, type **configure terminal** and press <Enter>.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type **session-router** and press <Enter> to access the session router path.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type **ifc-profile** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.

   ```
   ACMEPACKET(session-router)# ifc-profile
   ACMEPACKET(ifc-profile)#
   ```

4. **name**—Enter a name for this IFC profile.
5. **state**—Set this to enabled to use this ifc-profile.
6. **options**—Set the options parameter by typing options, a <Space>, the option name with a "plus" sign in front of it, and then press <Enter>.

   If you type the option without the "plus" sign, you will overwrite any previously configured options. In order to append the new options to the options list, you must prepend the new option with a "plus" sign.

   The options included in this section are: **add-sescase-to-route** and **dialog-transparency-support.**
7. Type **done** when finished.

SIP Registrar

To enable IFC support in a SIP Registrar:

1. In Superuser mode, type **configure terminal** and press <Enter>.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type **session-router** and press <Enter> to access the session router path.

```
ACMEPACKET(configure)# session-router
```

3. Type **sip-registrar** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# sip-registrar
ACMEPACKET(sip-registrar)#
```

4. Type **select** and choose the number of the pre-configured SIP registrar configuration element you want to configure.

```
ACMEPACKET(sip-registrar)# select
name:
1: registrar1

selection:1
ACMEPACKET(sip-registrar)#
```

5. **ifc-profile**—Set this parameter to the name of the IFC profile you just created.
6. Type **done** when finished.

2 ENUM Based Net-Net SMX

The Net-Net USM contacts an ENUM server via DNS for two purposes:

- to obtain authentication data for a registering UA
- to query the user subscriber database (ENUM) regarding the registration state of the AoR and update the database with the latest information

Message Authentication for SIP Requests

The Net-Net SMX authenticates requests by configuring the *sip authentication profile* configuration element. The *name* of this configuration element is either configured as a parameter in the *sip registrar* configuration element's *authentication profile* parameter or in the *sip interface* configuration element's *sip-authentication-profile* parameter. This means that the Net-Net USM can perform SIP digest authentication either globally, per domain of the Request URI or as received on a SIP interface.

After naming a *sip authentication profile*, the received methods that trigger digest authentication are configured in the *methods* parameter. You can also define which anonymous endpoints are subject to authentication based on the request method they send to the Net-Net SMX by configuring in the *anonymous-methods* parameter. Consider the following three scenarios:

1. By configuring the *methods* parameter with REGISTER and leaving the *anonymous-methods* parameter blank, the Net-Net SMX authenticates only REGISTER request messages, all other requests are unauthenticated.
2. By configuring the *methods* parameter with REGISTER and INVITE, and leaving the *anonymous-methods* parameter blank, the Net-Net SMX authenticates all REGISTER and INVITE request messages from both registered and anonymous endpoints, all other requests are unauthenticated.
3. By configuring the *methods* parameter with REGISTER and configuring the *anonymous-methods* parameter with INVITE, the Net-Net SMX authenticates REGISTER request messages from all endpoints, while INVITES are only authenticated from anonymous endpoints.

Credential Retrieval

The Net-Net USM requests authentication information from an ENUM server via DNS when it receives a REGISTER or other message from an endpoint. This server, which provides authentication information, is defined on the Net-Net USM in an *enum-config* configuration element that includes an *enum-servers* (the IP addresses of the servers) and *realm* parameters. Together, these two parameters define the DNS/ENUM server(s) which provide authentication data.

The target ENUM server is determined first by setting the *credential retrieval config* parameter to **enum-config** so the Net-Net USM will reference that *enum config*. Next, set the *credential retrieval config* parameter to the name of an *enum config* configuration element which is populated with the ENUM servers' IP addresses.

User Authentication Query

As soon as a request is received on a SIP interface and has been determined to require authentication, the Net-Net SMX attempts to authenticate the endpoint. It sends a DNS TXT query including the UA's AoR to an ENUM database and expects the H(A1) defined in RFC2617 for the user being authenticated.

SIP Digest User Authentication

SIP Authentication Challenge

When the Net-Net SMX receives a response from the ENUM server including the hash value for the user, it sends a SIP authentication challenge to the endpoint, if the endpoint did not provide any authentication headers in its initial contact the with Net-Net SMX. If the endpoint is registering, the Net-Net SMX replies with a 401 Unauthorized message with the following WWW-Authenticate header:

```
WWW-Authenticate: Digest realm="atlanta.com",
domain="sip:boxesbybob.com", qop="auth",
nonce="f84f1cec41e6cbe5aea9c8e88d359", opaque="", stale=FALSE,
algorithm=MD5
```

If the endpoint initiates any other request to the Net-Net SMX besides REGISTER, the Net-Net SMX replies with a 407 Proxy Authentication Required message with the following Proxy-Authenticate header:

```
Proxy-Authenticate: Digest realm="atlanta.com", qop="auth",
nonce="f84f1cec41e6cbe5aea9c8e88d359", opaque="", stale=FALSE,
algorithm=MD5
```

Authentication Header Elements

- Digest Realm—This value is configured in the *digest-realm* parameter in the *sip-registrar* configuration element. This parameter is mandatory when using the "ENUM-TXT" credential retrieval method.
- Domain—A quoted, space-separated list of URIs that defines the protection space. This is an optional parameter for the "WWW-Authenticate" header.
- Nonce—A unique string generated each time a 401/407 response is sent.
- Qop—A mandatory parameter that is populated with a value of "auth" indicating authentication.
- Opaque—A string of data, specified by the Net-Net USM which should be returned by the client unchanged in the Authorization header of subsequent requests with URIs in the same protection space.
- Stale—A flag indicating that the previous request from the client was rejected because the nonce value was stale. This is set to true by the SD when it receives an invalid nonce but a valid digest for that nonce.
- Algorithm—The Net-Net SMX always sends a value of "MD5"

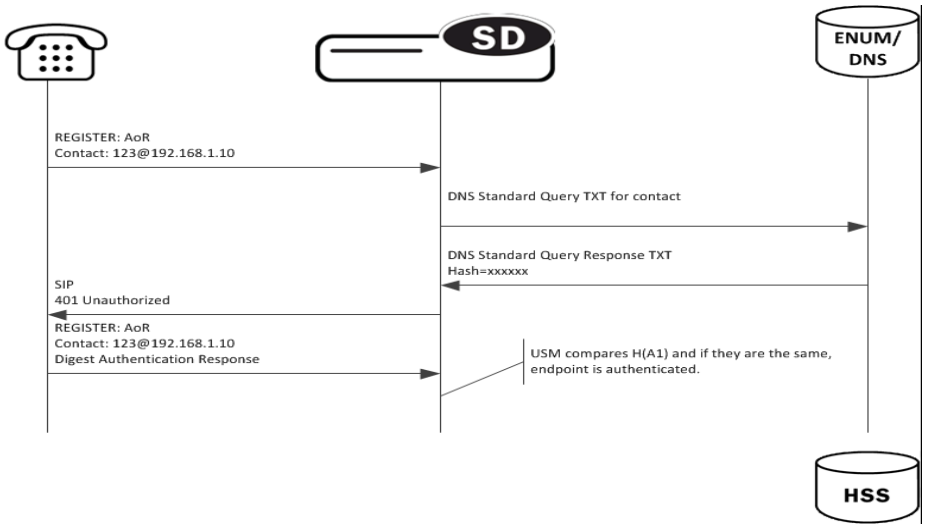
SIP Authentication Response

After receiving the 401/407 message from the Net-Net USM, the UA resubmits its original request with an Authorization: header including its own internally generated MD5 hash.

Net-Net USM Authentication Check

At this point, the Net-Net USM has received an MD5 hash from the ENUM server and an MD5 hash from the UA. The Net-Net SMX compares the two values and if they are identical, the endpoint is successfully authenticated. Failure to match the two hash values results in a 403 or 503 sent to the authenticating endpoint.

The following image shows the user authentication process.



Net-Net USM as Registrar

DDNS Update to User Subscriber Database

As REGISTER messages are received, the Net-Net USM updates the ENUM database via DDNS UPDATE messages as defined by RFC2136. New registrations are added to the database while expired or deleted registrations ones are removed.

The Net-Net USM acts as a registrar by configuring the *sip registrar* configuration element. When registrar functionality is enabled, the Net-Net USM acts as a registrar rather than only caching and forwarding registrations to another device. Net-Net USM registry services are enabled globally per domain, not on individual SIP interfaces or other remote logical entities.

On receiving a REGISTER message, the Net-Net USM checks if it is responsible for the domain contained in the Request-URI as defined by the *domains* parameter and finds the corresponding *sip registrar* configuration. This is a global parameter and all messages are checked against all *sip registrar domains*. Thus you could create one sip registrar configuration element to handle all *.com domains and one sip registrar configuration element to handle all *.org domains. The Net-Net USM begins registrar functions for all requests that match the configured domain per sip-registrar configuration element.

A UA is considered registered after the Net-Net USM updates the ENUM server with a DDNS dynamic update. After this action, the Net-Net SMX sends a 200 OK message back to the registering UA.

TTL

Part of the Net-Net USM architecture includes a local ENUM cache which maintains the results from ENUM queries locally. To enable Net-Net USM, the TTL value in the enum config must be set to 0. This ensures that whenever an ENUM query is required, the Net-Net USM consults the central User Subscriber Database to have the latest network-wide information about the UA it is trying to reach, instead of its ENUM cache.

ENUM Database Correlation

When a UA registers, as the number of associated contacts for an AoR grows or shrinks, the ENUM-based User Subscriber Database is updated in turn with the latest information using a DDNS UPDATE. After a REGISTER including authentication information is received from a UA, the Net-Net USM sends a Standard NAPTR query for the AoR to the ENUM server. The ENUM server replies with a Standard Query response, including the NAPTR records.

After receiving the entries from the ENUM database via the ENUM query, the list must be correlated with the Net-Net USM's view of the AoR's registration state. The differences will be resolved by sending a DDNS UPDATE to add or remove entries from the ENUM server. The database correlation phase only occurs when endpoints register.

Contacts that need to be added are put on the UPDATE add list. Contacts that are being unregistered (contact with Expires=0) or have expired timestamp are added to the UPDATE remove list.

Entry Expiration

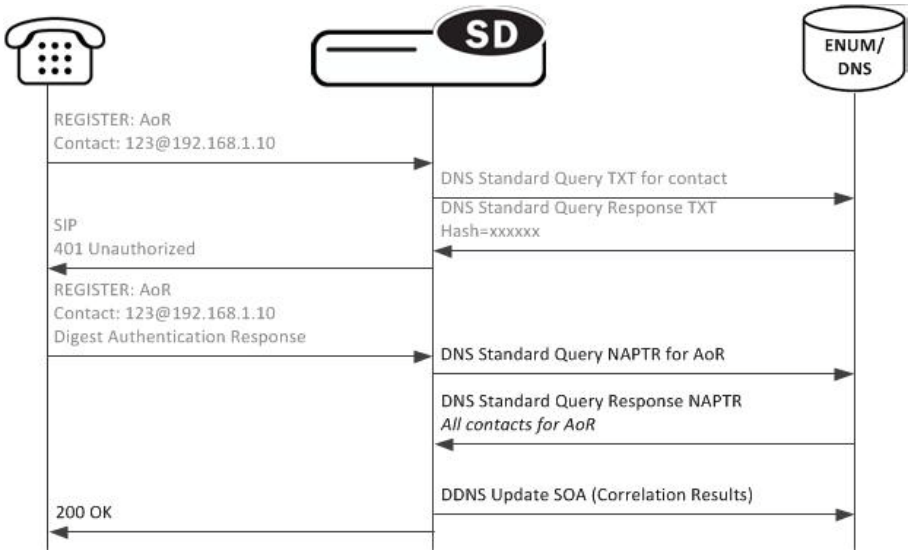
The Net-Net USM employs a process to remove expired contacts from the ENUM database whether they were entered by the active Net-Net USM or other Net-Net USM. After judging if a contact as expired, the Net-Net USM removes the contact in a subsequent DDNS update.

To do this, the Net-Net USM includes an expiration parameter in the Contacts it insert into the ENUM database. Expiration is indicated with a ts= parameter. This parameter's value is set to the initial registration time measured on the Net-Net USM plus the REGISTER message's Expires: header value or Expires parameter in the Contact header value. This value is measured in seconds after the epoch. In a DDNS update, a ts= parameter appears as follows:

```
Regex: "!^.*$!sip:234-hchse6c0d01u2@172.16.101.51:5060;ts=1313493824!"
```

After the Net-Net USM retrieves contacts for an AoR in a NAPTR record that are expired, based on ts= parameters whose have passed, the Net-Net USM's following Dynamic update indicates to remove those contacts from the ENUM database.

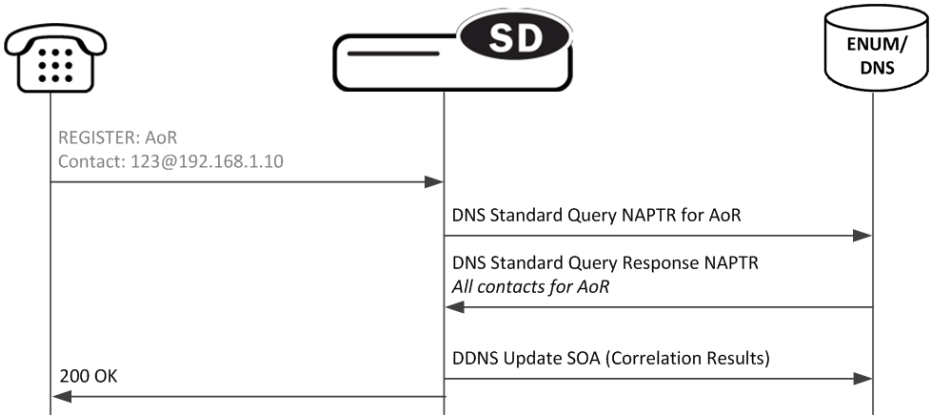
In this way, all Net-Net USM members of a domain may remove expired contacts from the ENUM database.



Register Refresh

When a UA sends a register refresh, the Net-Net USM first confirms that the authentication exists for that UA's registration cache entry, and then is valid for the REGISTER refresh. Then, the lifetime timer (value in Expires: header) for that registration cache entry is checked.

If the timer has not exceeded half of its lifetime, only a 200 OK is sent back to the UA. If the timer has exceeded half of its lifetime, the Net-Net USM sends a NAPTR update to the ENUM database.



In addition to the baseline Net-Net SBC REGISTER refresh conditions, an ENUM database update is required when one of the following conditions is satisfied:

1. The *location update interval* timer has expired—This value, configured in the *sip registrar* configuration element ensures that ENUM database always has the latest user information by periodically sending Standard Queries.
2. The message's call-id changes while the f**orward-reg-callid-change** option in the *sip config* configuration element is set. This covers the case where the UA changes the Net-Net USMs through which it attaches to the network.
3. The REGISTER message's Cseq has skipped a number. This covers the case in which a user registered with SMX1, moves to SMX2, and then returns to SMX1.
4. The REGISTER message's contact list has changed.

After receiving the entries from the ENUM database via the NAPTR query, the list is correlated with the internal registration cache. Appropriate DDNS updates are preformed (see:ENUM Database Correlation (36)).

If the Net-Net USM updates the ENUM database because of matching one of the above conditions, the UA's access-side Expires timer is reset to the REGISTER message's Expires: header value, and returned in the 200 OK. This happens even in the case when the reREGISTER was received in the first half of the previous Expires period. In addition, the core-side *location update interval* timer are refreshed on both active and standby.

When the above four conditions are not met, the registration and reregistration expiration proceeds normally. If the access-side expiration timer has not exceeded half of its lifetime, only a 200 OK is sent back to the UA. If the timer has exceeded half of its lifetime, the Net-Net USM refreshes the registration to the ENUM server.

Note: Upon a Call-id or contact list change, both the registration cache timer and the ENUM database are updated.

User Registration based on Reg-ID and Instance-ID (RFC 5626)

Sometimes a user's device reregisters from a different network than from its original registration. This event should be considered a location update rather that a completely new registration for the Contact. The Net-Net USM can perform this way by considering the endpoint's reg-id and instance-id parameters defined in RFC 5626.

The Net-Net USM identifies new REGISTER received on a different access network as a location update of the existing binding between the Contact and AoR. Without this feature, the Net-Net USM would create a new binding and leave the old binding untouched in the local registration cache/ENUM database. This scenario is undesirable and leads to unnecessary load on various network elements including the Net-Net USM itself.

The following conditions must be matched to equate a newly registering contact as a location update:

For a received REGISTER:

1. The message must not have more than 1 Contact header while 1 of those Contact headers includes a reg-id parameter. (failure to pass this condition prompts the Net-Net USM to reply to the requester with a 400 Bad Request).
2. The Supported: header contains **outbound** value
3. The Contact header contains a **reg-id** parameter
4. The Contact header contains a **+sip.instance** parameter

After these steps are affirmed, the Net-Net USM determines if it is the First hop. If there is only one Via: header in the REGISTER, the Net-Net USM determines it is the first hop and continues to perform Outbound Registration Binding processing.

If there is more than 1 Via: header in the REGISTER message, the Net-Net USM performs additional validation by checking for a Path: header corresponding to the last Via: includes an ob URI parameter, Outbound Registration Binding may continue.

If the Net-Net USM is neither the first hop nor finds an `ob` URI in Path headers, it replies to the UA's REGISTER with a 439 First Hop Lack Outbound Support reply.

reREGISTER Example

The user (AoR) bob@example.com registers from a device +sip.instance= <urn:uuid:0001> with a reg-id ="1", contact URI = sip:1.1.1.1:5060. A binding is be created for bob@example.com+<urn:uuid:0001>+reg-id=1 at sip:1.1.1.1.:5060.

Next, Bob@example.com sends a reREGISTER with the same instance-id but with a different reg-id = 2 and contact URI = sip:2.2.2.2:5060.

The previous binding is removed. A binding for the new contact URI and reg-id is created. bob@example.com+<urn:uuid:0001>+reg-id=2 at sip:2.2.2.2:5060

Outbound Registration Binding Processing

An outbound registration binding is created between the AoR, instance-id, reg-id, Contact URI, and other contact parameters. This binding also stores the Path: header.

Matching re-registrations update the local registration cache as expected. REGISTER messages are replied to including a Require: header containing the outbound option-tag.

If the SMX receives requests for the same AOR with some registrations with reg-id + instance-id and some without them, the SMX will store them both as separate Contacts for the AOR; The AoR+sip.instance+reg-id combination becomes the key to this entry.

ENUM Database Update

When a REGISTER message is received:

1. The ENUM user database is queried for the AoR and any existing entries.
2. If there are any entries with the same instance-id as the current REGISTER request in the ENUM query response, then those entries will be marked for subsequent removal in the ENUM database.
3. The ENUM database is updated with a NAPTR request. This request adds the new Contact URI for that AOR+instance-id and removes any existing entries for the same AOR+instance-id.

NAPTR Update Format

The ENUM database update includes the instance-id and reg-id when those parameters are present in a registration. These values are appended to the regex replacement field. For example:

```
!^.*$!sip:3556-1cdstqjt90hve@172.16.101.62:5060;sip.instance=<urn:uuid:00000000-
0000-1000-8000-000A95A0E128>;reg-id=1;ts=1326568408;!
```

Net-Net USM Licensing

The Net-Net USM connected to an ENUM database requires two licenses: Registration Cache Limit, SIP Authorization/Authentication.

For ENUM-based Net-Net USM, the SIP Authorization/Authentication license reveals the *SIP Authentication Profile* configuration element. Configuring both configuration elements is required to operate a Net-Net USM. Refer to the Licensing and Database Registration Limits (46) section for the third license required for Net-Net USM operation.

Refer to the Net-Net SBC ACLI Configuration guide, Getting Started chapter for how to install licenses in your system.

ACLI Instructions

ENUM Configuration

First the server used for authentication and as the User Subscriber Database is created.

To configure the ENUM Configuration:

1. In Superuser mode, type **configure terminal** and press <Enter>.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type **session-router** and press <Enter> to access the media-related configurations.

```
ACMEPACKET(configure)# session-router
```

3. Type **enum-config** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# enum-config
ACMEPACKET(enum-config)#
```

You may now begin configuring the enum-config configuration element.

4. **name**—Set a name to use to reference this enum configuration from within the Net-Net USM.
5. **top-level-domain**—Enter the domain which this ENUM server(s) services and returns results for.
6. **realm-id**—Enter the realm name where this ENUM server exists.
7. **enum-servers**—Enter the IP address of one or more ENUM servers used for registration. Multiple entries are separated by commas.
8. **service-type**—Leave this as its default.
9. **ttl**—Leave this at the default of 0 to set the TTL value (in seconds) for NAPTR entries as populated when sending a DNS update to the ENUM server.
10. **order**—Enter the value to populate the order field with when sending NAPTR entries to the ENUM server.
11. **preference**—Enter the value to populate the preference field with when sending NAPTR entries to the ENUM server.
12. Type **done** when finished.

SIP Authentication Profile

To configure the SIP Authentication Profile:

1. In Superuser mode, type **configure terminal** and press <Enter>.

```
ACMEPACKET# configure terminal
```

2. Type **session-router** and press <Enter> to access the media-related configurations.

```
ACMEPACKET(configure)# session-router
```

3. Type **sip-authentication-profile** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# sip-authentication-profile
ACMEPACKET(sip-authentication-profile)#
```

You may now begin configuring the SIP Authentication Profile configuration element.

4. **name**—Enter the name of this SIP authentication profile that will be referenced from a SIP registrar (or SIP interface).
5. **methods**—Enter all the methods that should be authenticated. Enclose multiple methods in quotes and separated by commas.
6. **anonymous-methods**—Enter the methods from anonymous users that require authentication. Enclose multiple methods in quotes and separated by commas.
7. **digest-realm**—enter the digest realm sent in an authentication challenge (401/407) sent to a UA. This is required in ENUM/DNS deployments,.
8. **credential-retrieval-method**—Enter **ENUM-TXT**.
9. **credential-retrieval-config**—Enter the *enum-config* name used for retrieving authentication data.

10. Type **done** when finished.

SIP Registrar

To configure the Net-Net USM to act as a SIP Registrar:

1. In Superuser mode, type **configure terminal** and press <Enter>.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type **session-router** and press <Enter> to access the session router path.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type **sip-registrar** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.

   ```
   ACMEPACKET(session-router)# sip-registrar
   ACMEPACKET(sip-registrar)#
   ```

4. **name**—Enter a name for this SIP registrar configuration element.
5. **state**—Set this to **enabled** to use this SIP registrar configuration element.
6. **domains**—Enter one or more domains in the R-URI this configuration element handles. Wildcards are valid for this parameter. Multiple entries can be entered in quotes, separated by commas.
7. **subscriber-database-method**—Set this to **DDNS**.
8. **subscriber-database-config**—Enter the enum-config configuration element *name* that will handle REGISTER messages for this domain. This should be the same element used for requesting authentication data.
9. **authentication-profile**—Enter a *sip-authentication-profile* configuration element's *name*. The *sip authentication profile* object referenced here will be looked up for a REGISTER message with a matching *domain* in the request URI. You may also leave this blank for the receiving SIP Interface to handle which messages require authentication if so configured.
10. **location-update-interval**—Keep or change from the default of 1400 minutes (1 day).
11. Type **done** when finished.

Update to ENUM Database on Endpoint Connection Loss

The Net-Net USM can monitor an endpoint's transport-layer status for loss of connectivity in multiple ways. Then, when the endpoint's connection to the Net-Net USM has been terminated, the Contact is removed from the registration cache, as associated under a registered AoR. In addition, the user database is immediately updated.

These transactions contribute to the registration cache and ENUM user database being updated in real time to retain only reachable contacts for a registered AoR. This feature helps to alleviate:

- unnecessary transactions and system load spent on attempting to reach an unreachable endpoint
- incorrect and out of date statistics

Connection Reuse

The connection between the Net-Net USM and an endpoints must employ the connection reuse mechanism to enable this feature's de-registration and user database updates. Connection reuse is when an endpoint registers to the Net-Net

USM and all subsequent signaling between the Net-Net USM and that endpoint reuses the same socket pair. There are four cases when connection reuse is enabled:

1. Connection reuse is enabled on the SIP interface facing the endpoint(s). This is enabled by adding the **reuse-connections=yes** option on a SIP interface.
2. The endpoint is behind a NAT. Because of the fundamental method that the Net-Net SBC uses for maintaining its connection to an endpoint behind a NAT, this case will always force connection reuse.
3. The endpoint includes the *alias* parameter in the Via: header in its REGISTER message to the Net-Net USM (RFC 5923).
4. If the endpoint is configured as a session agent, the reuse-connections parameter must be set to TCP. When receiving signaling from a remote logical entity such as a session agent defined for an endpoint, if the reuse-connections parameter is set to tcp, the Net-Net USM enables connection reuse between itself and the UA.

Unreachability Determination

There are four ways that the Net-Net USM determines endpoint is not reachable:

1. No CRLF message is returned to the Net-Net USM within the expected time frame: this is based on the Net-Net USM's RFC5626 support.
2. No TCP Keepalive is returned to the Net-Net USM within the expected timeframe. This is based on configuring the *network parameter* configuration element per application interface.
3. The endpoint explicitly terminates its transport-layer connection to the Net-Net USM.
4. The endpoint is otherwise labeled as unreachable from a non-explicit fault condition for a call made to an unreachable endpoint.

RFC 5635 Failure

The Net-Net USM supports the RFC 5635 method of SIP application keepalives, which are endpoint-initiated, i.e., the endpoint starts the mechanism by including the keep parameter in the initial Via: header. Endpoint reachability is determined the receipt or loss of a CR/LF ping-pong message. In the SIP interface configuration element, you set the **register keep alive** parameter to **always** or **bnat** (behind NAT), to enable RFC 5635 functionality. This applicable to TCP or TLS connections. **Always** forces the Net-Net USM to always return a CRLF reply when the *keep* parameter is in the initial Via: header. **bnat** forces the Net-Net USM to replies to RFC 5635 requests when the endpoint is located behind a NAT.

Next, you can accept the Net-Net USM's default keep alive window of 30 seconds, or you may set your own by configuring the **tcp nat interval** or **inactive con timeout** value, both found in the *sip interface* configuration element. The Net-Net USM uses the smaller of the two configured values. The chosen value is inserted into the keep parameter in the 200 OK message returned to the registering endpoint.

> **Note:** The **inactive con timeout** value otherwise disconnects a TCP/TLS connection after the configured value elapses. The **tcp nat interval** value is also inserted into the *expires* parameter in the Contact: header for devices identified as behind a NAT.

In addition, the Net-Net USM maintains a keep timer. The Net-Net USM adds 31 seconds to the keep value it returns and begins counting down. 31 is the chosen margin to account for any network or application delay.

If the endpoint returns the CRLF before the timer expires, the endpoint is considered up and a new CRLF ping is sent to the endpoint; the timer begins counting down again. If the Net-Net USM fails to receive the CRLF ping before the timer expires, then the UA is considered unreachable. Provisions begin to remove that contact from the registration cache and then the ENUM user database.

TCP Keepalive Failure

Endpoint reachability can be determined from TCP keepalives, as enabled per SIP interface. This method of determining connectedness is based on configuring the global *network parameters* configuration element that is enabled on a SIP interface with the tcp-keepalive parameter. See the System TCP Keepalive Settings section of the Net-Net SBC ACLI Configuration guide for how to configure the TCP keepalive feature. When an endpoint fails the TCP keepalive test, provisions begin to remove that contact from the registration cache and then the ENUM user database.

Explicit and undetermined connection termination

Ideally, a UA will gracefully close its TCP connection to the Net-Net USM, and in turn the socket pair will be considered closed with the UA being unreachable. In less ideal cases, the UA goes dark and no response is received when expected. The Net-Net USM considers the unresponsive UA as unreachable as call attempts time out. Provisions then begin to remove that contact from the registration cache and then the ENUM user database.

Registration Cache and User Database Removal

When the Net-Net USM sets up the initial REGISTER request from a UA, an internal binding is created between the contact and the AoR for that user. If a UA is considered unreachable by either of the four conditions explained in the previous section, the following occur:

- The contact's entry in the registration cache is removed. If this is the last contact registered for an AoR, the entire entry for the AoR is removed from the registration cache.
- The Net-Net USM sends an UPDATE to the ENUM user database removing the Contact.

To enable these actions, you must configure the **force unregistration** option *sip config* configuration element and also set the **unregister on connection loss** parameter in the *sip interface* configuration element to **enabled**.

ACLI Instructions

To globally enable force-unregistration at the sip-config level:

1. In Superuser mode, type **configure terminal** and press <Enter>.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type **session-router** and press <Enter> to access the session router path.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type **sip-config** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.

   ```
   ACMEPACKET(session-router)# sip-config
   ACMEPACKET(sip-config)#
   ```

4. Type **select** to continue.

   ```
   ACMEPACKET(sip-interface)# select
   ACMEPACKET(sip-interface)#
   ```

5. **options**—Set the options parameter by typing **options**, a <Space>, **force-unregistration** with a "plus" sign in front of it, and then press <Enter>.

   ```
   ACMEPACKET(sip-interface)# options +force-unregistration
   ```

   If you type the option without the "plus" sign, you will overwrite any previously configured options. In order to append the new options to the realm configuration's options list, you must prepend the new option with a "plus" sign as shown in the previous example.

6. Type **done** and **exit** to complete configuration of this **sip-config** configuration element.

To configure the SIP Interface configuration element portion of the ENUM Database update feature:

1. In Superuser mode, type **configure terminal** and press <Enter>.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type **session-router** and press <Enter> to access the session router path.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type **sip-interface** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.

   ```
   ACMEPACKET(session-router)# sip-interface
   ACMEPACKET(sip-interface)#
   ```

4. Type **select** and choose the number of the pre-configured sip interface you want to configure.

   ```
   ACMEPACKET(sip-interface)# select
   <realm-id>:
   1: private 192.168.101.17:5060
   2: public  172.16.101.17:5060

   selection: 1
   ```

5. u**nregister-on-connection-loss**—Set this parameter to **enabled** for the Net-Net USM to update the ENUM server when an endpoint is deemed failed.

Parameters for determining endpoint reachability:

6. **tcp-keepalive**—You may set this parameter to enabled to enforce the network-parameters configuration element located in the system-config path. See the Net-Net SBC ACLI Configuration guide, System TCP Keepalive Settings section for more information.
7. **register-keep-alive**—Set this parameter to **always** for the Net-Net USM to return the **keep** parameter, with optional value (as configured in the next two steps) in the Via: header to an endpoint including an empty keep value in its initial REGISTER message.

You may set none, one, of both the following for a keep value returned to the initiating endpoint. Read the RFC 5635 Failure (42) section for how the actual value is determined:

8. **inactive con timeout**—Set this parameter value to the value in seconds inserted in the returned keep parameter for RFC 5635 support
9. **tcp nat interval**—Set this parameter value to the value in seconds inserted in the returned keep parameter for RFC 5635 support
10. Type **done** and **exit** to complete configuration of this **sip-interface** configuration element.

Message Routing

In order for the Net-Net USM to forward requests such as INVITEs, the ENUM database is queried first before forwarding the message. The Net-Net USM performs routing in two ways depending on the *routing precedence* parameter in the *sip registrar* as **registrar** or **local policy**. Registrar routing uses the configured SIP registrar/ENUM server for its initial, destination address query, while local policy routing lets you configure routing decisions within the Net-Net USM's local policy routing functionality.

If Net-Net USM fails to route the message, it falls back to trying by the other method. That is, if the system is set to registrar routing and this method fails, the Net-Net USM then attempts to route the call via local policy, and vice versa. A Net-Net USM deployment generally functions in registrar mode, but local policy may be used for more control over routing decisions, performed on the individual Net-Net USM.

Local Policy Routing

To specify an ENUM server to query, a *policy attribute* with next-hop `enum:<enum-config-name>` is configured in the *local-policy*. This assumes that the *sip registrar* configuration element's *routing precedence* parameter is set to **local-policy**.

If the target endpoint is unreachable or there is no matching local policy, the Net-Net USM falls back to query the subscriber database (ENUM) for determining the next hop.

If any policy-attributes > next-hop parameters originally stepped through had a next-hop of enum: the fall back to registrar is not performed. This is because the Net-Net USM has already queried the database once and it would be redundant.

Also, when the *routing precedence* parameter is set to **local-policy** and the Net-Net USM uses a local subscriber database, it checks the registration cache for a pre-existing matching contact in the INVITE to find where to forward request.

Registrar Routing

When **route-precidence** parameter is set to **registrar** and the lookup via the registrar fails, you can configure the Net-Net USM to not fallback to make a local policy lookup for the destination. This is configured by adding the **skip-local-policy-lookup** option to the *sip-registrar* configuration element.

Default Egress Realm

The *sip registrar* configuration element should be configured with a default *egress realm id*. This is the name of the *realm config* which defines the IMS control plane through which all Net-Net USMs, ENUM servers, and other network elements communicate and exchange SIP messaging. It is advisable to configure this parameter in order to ensure well defined reachability among Net-Net USMs.

ACLI Instructions

SIP Registrar

To configure a SIP registrar configuration element for message routing:

1. In Superuser mode, type **configure terminal** and press <Enter>.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type **session-router** and press <Enter> to access the session router path.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type **sip-registrar** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.

   ```
   ACMEPACKET(session-router)# sip-registrar
   ACMEPACKET(sip-registrar)#
   ```

4. Type **select** and choose the number of the pre-configured sip registrar you want to configure.
5. **routing-precedence**— Set this to either **registrar** or **local-policy** depending on your deployment.
6. **egress-realm-id**—Enter the default egress realm for Net-Net USM messaging.
7. Type **done** when finished.

Licensing and Database Registration Limits

The Net-Net USM (and Net-Net SBC) limit the number of unexpired registration cache entries globally. The total number of system registrations is configured with the *registration cache limit* parameter in the *sip config* configuration element.

The Net-Net USM also limits the number of registration cache entries that were obtained from a User Subscriber Database; only REGISTERs that prompted the database query are counted here. As User Subscriber Database entries are added and removed, this counter is updated accordingly. Note that it is the actual number of *SD-contacts* that count against the license limit. Discrete database registration license values range from 20,000 through 500,000 in increments of 20,000.

When a registering contact is rejected because it will exceed one of these limits, the Net-Net USM sends a 503 message to the registering endpoint.

Refer to the Net-Net 4000 ACLI Configuration Guide, chapter 2 “Getting Started”, Software Licensing section for how to install a license.

Database Registration Limit Alarm

By default, a major alarm is enabled when 98% or more of the licensed number of Database Registrations are used. This alarm is cleared when the number of database registrations falls below 90%. You can configure minor and critical alarms when crossing configured thresholds and you can also reassign the major alarm. This is configured in by creating a *system-config* > *alarm-threshold* sub element with *type* of **database-registration**.

Extended ENUM Record Length

In some cases, when a user registers a contact from a UA, the Contact: header's contents are too long to be inserted into the DNS-based user database as presented in the NAPTR record.

To mitigate this, the Net-Net USM stores contact-related metadata, i.e. parameter key and value pairs, in a related TXT record.

If the contents of the Contact: header do not exceed 255 bytes, there is generally no need to extract the metadata to store in an additional TXT record, so the Net-Net USM will not enact this process.

NAPTR and TXT Record Creation and Association

When a user initially registers to the Net-Net USM, the DDNS update sent to the user database will include a NAPTR record and a TXT record. The NAPTR record contains the Contact: header's SIP URI as a regexp replacement. All URI and header parameters from the Contact: header are excluded from the NAPTR record and are inserted into the accompanying TXT record.

The NAPTR record and the TXT record both contain a Net-Net USM-generated key that binds the Contact metadata and the Contact URI sent in the separate records. This common key indicates the data in the NAPTR and TXT record belong to the same registering contact, and is used to recreate the Contact: header for later use. The format of the common key is:

```
p-acme-ckey=<SD-Core-IP>:<integer id>
```

The <SD-Core-IP> is the IP address from which the Net-Net USM communicates with the DNS server. The <integer id> enumerates each contact. Contacts are enumerated in the integer-id element because they can be non-unique when a user behind a NAT registers more than one contact from behind the NAT. For example:

```
TXT "p-acme-ckey=172.16.101.61:1" "$key=value" "^key=value"
NAPTR 1 1 "u" "E2U+sip""!^.*$!sip:642-10u72@172.16.101.61:5060\;p-acme-key=172.16.101.61:1!"
```

In the previous example, the key=value pair represent parameters in the Contact header and Contact SIP URI. A `$key=` indicates the parameter existed in the Contact: SIP URI. A `^key=` indicates the parameter existed in a Contact: header parameter. The Net-Net USM uses this convention to reconstruct the parameters' placement in the original Contact: header.

NAPTR Record Format

Net-Net USMs learn that an associated TXT record exists for the from a NAPTR lookup for the queried SIP URI.

In the DDNS update, the Net-Net USM indicates an associated TXT record by setting the NAPTR resource record with:

- `m` in the flags field (in addition to the `u` flag)

- `E2U+sip:contact` in the service type field

When no additional metadata and no parameters need to stored in ENUM server, the flags field contains `u`.

TXT Record Retrieval

A Net-Net USM performs a separate query to the ENUM server for the TXT metadata records when it detects the 'm' flag in a NAPTR result (and the one-query-txt-naptr option is not configured). You can set the one-query-txt-naptr option to enabled in the enum-config configuration element to force the Net-Net USM to request the TXT and NAPTR records both in one query from the ENUM server.

Once the SMX receives the metadata stored in the TXT records, it restores the header and URI parameters that were present on the original registration.

Requirements

BIND 9.8.1-P1 or later hosting the ENUM server supports this feature.

3 Local Subscriber Tables

Local Subscriber Table

A local subscriber table (LST) is an XML formatted file that contains one or more usernames associated with a hash as encrypted or plaintext. The LST is saved locally on the Net-Net USM's filesystem.

LSTs enable a standalone Net-Net USM node or high-availability (HA) pair forego relying on an external user database. Thus the Net-Net USM does not need to communicate with a server to authenticate users. This can eliminate the operational complexity of deploying a highly available credential storage system.

LST Runtime Execution

The LST is loaded on boot up when the configuration is appropriately set. Incoming messages thereafter can then be authenticated based on the credentials in the LST. If the Net-Net USM can not load an LST file, three things occur:

1. The following log message is recorded at the NOTICE level:

   ```
   LST [table-name] was not loaded - [filename] has error loading XML file
   ```

2. The message stated above is printed on the ACLI.
3. A 503 Response is returned to the UA that sent the initial REGISTER message to the Net-Net USM.

LST Configuration

To configure the Net-Net USM to use LSTs for authentication, you need to create a *local subscriber table* configuration element that identifies that LST. You then need to set the *sip authentication profile* configuration to reference that LST configuration so that when messages requiring authentication are received and processed by a *sip registrar* configuration element, the Net-Net USM will use the identified LST for authentication.

In a *local subscriber table* configuration, you must define an object **name**, identify the specific LST **filename** (and path). If the filename is entered without a path, the Net-Net USM looks in the default LST directory, which is `/code/lst`. If the LST file is located elsewhere on the Net-Net USM, you must specify the filename and absolute path. For example /code/path/01302012lst.xml.

The corresponding *sip authentication profile* must be set to use the **local subscriber table** configuration element you just created. First set **credential retrieval method** to **local**, set the **digest realm** appropriately (this is required for authentication), and finally set the **credential retrieval config** parameter to the **name** of the *local subscriber table* configuration element that you just created. At this point you may save and activate your configuration.

Unencrypted passwords for each user in the table is computed with the MD5 hash function as follows:

```
MD5(username:digest-realm:password)
```

ACLI Instructions

LST Table

To configure the Net-Net USM to use an LST:

1. In Superuser mode, type **configure terminal** and press <Enter>.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type **session-router** and press <Enter> to access the session router path.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type **local-subscriber-table** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.

   ```
   ACMEPACKET(session-router)# local-subscriber-table
   ACMEPACKET(local-subscriber-table)#
   ```

   You may now begin configuring the local subscriber table configuration element.

4. **name**—Enter the name of this local subscriber table configuration element that will be referenced from a SIP registrar configuration element.
5. **filename**—Enter the filename that describes this LST XML file. If no path is given, the Net-Net USM looks in the `/code/lst` directory. You may provide a complete path if the file is located elsewhere.
6. **secret**—Enter the PSK used in encryption and decryption of the passwords in the XML file. Once saved, this value is not echoed back to the screen in plaintext format. See: LST Subscriber Hash and Encryption (52).
7. Type **done** when finished.

SIP authentication profile

To configure the Net-Net USM to utilize an LST, continuing from the previous step:

8. Type **exit** to return to the session router path.
9. Type **sip-authentication-profile** and press <Enter>.
10. Type **select** to choose the existing sip-authentication-profile configuration element you wish to use LST for authentication.

    ```
    ACMESYSTEM(sip-authentication-profile)# select
    <name>:
    1: name=sipAuthSMX1 digest-realm=acme.com credential-retrieval-
    method=loacl

    selection: 1
    ACMESYSTEM(sip-authentication-profile)#
    ```

You may now begin configuring the relevant parameters.

11. **digest-realm**—Enter the digest realm used for authenticating here.
12. **credential-retrieval-method**—Set this parameter to **local** to use an LST.
13. **credential-retrieval-config**—Enter the name of the LST configuration you just configured.
14. Type **done** when finished.

LST Redundancy for HA Systems

LSTs must be synchronized between redundant nodes to ensure that the standby node contains identical LST files. You can either SFTP the same LST file to both the

active and standby node, or you can use the synchronize command. The **synchronize** command is always executed from the active system. It copies the specified file from the active to the standby node placing the copy in the same file location on the standby node. Use the **synchronize lst** command as follows:

```
ACMESYSTEM# synchronize lst file.xml
```

Note: The synchronize command does not reload the LST files.

Reloading the LST

After copying a new LST file to the Net-Net USM (and its standby peer), you can reload this newer file from the ACLI using the **refresh lst** command. For example:

```
ACMEPACKET# refresh lst <local-subscriber-table name>
```

Using the **refresh lst** command selects the LST by name to refresh. Alternatively, saving and activating the configuration will reload the configuration as well and should be used when configuration parameters have also changed.

Note: In an HA pair of Net-Net USMs, you must independently execute the **refresh lst** command on both the active and standby systems.

LST File Compression

To save local disk flash space, you can compress the LST XML file using .gz compression. The resultant file must then have an `.xml.gz` extension.

LST File Format

The LST file format is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<localSubscriberTable encrypt-algo="aes-128-cbc">
<subscriber username="alice@apkt.com" hash="02:5E:78:D8:7E:75:A3:39" encrypted="true"/>
<subscriber username="bob@apkt.com" hash="bc4b2a76b9719d911017c59" encrypted="false"/>
<subscriber username="acme@apkt.com" hash="5d41402abc4b2a76b9719d9" encrypted="false"/>
</localSubscriberTable>
```

The LST file's elements are as follows:

localSubscriberTable

This is the head element in the XML file. Each file can have only one head element. The folloing attribute is found in this element:

- encrypt-algo—This indicates the algorithm type used to encrypt the hash in the XML file. The key for this encryption will be a preshared key and is configurable in the local subscriber table configuration element with the secret parameter.
- The value in this element is for display purposes only.
- Currently AES-128-CBC is the only supported encryption algorithm.

subscriber

This element has the subscriber information. And has the following 3 attributes:

- username—The value given in the username attribute must be same as the username that will be sent in the Authorization Header in the Request message from the users. Refer RFC 2617 Http Authentication for details.

- hash—The hash provided in the XML must be an MD5 hash of the Username, digest-realm and the password of the user. This is same as the H(A1) described in RFC 2617.

  hash = md5(username:digest-realm:password)

- encrypted—The encrypted flag indicates if the "hash" given in the XML file is encrypted or not

LST Subscriber Hash and Encryption

You may additionally use AES-128 CBC to encrypt the hash in the subscriber element in the LST XML file. The PSK used for encryption is configured in the **secret** parameter and an 8-byte pseudo random number is used as the salt. The LST file must set the encrypted attribute per subscriber element to true. To derive the final encrypted data you place in the XML file, three steps are performed according to the following blocks. The output of the last step, Formatting final Encrypted Data (53), is inserted into the LST files, subscriber element's hash value, when the encrypted attribute is set to true.

Key/Initialization Vector

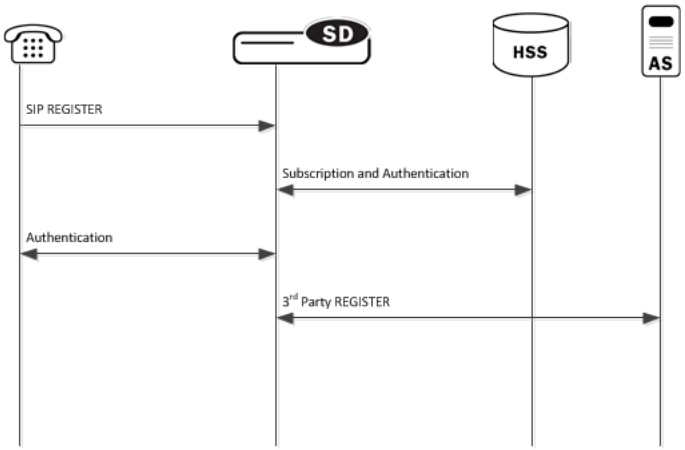
Encryption

Formatting final Encrypted Data

4 Third Party Registration

The Third Party Registration feature is a mechanism for forwarding Registration information to a third party server. An IM (Instant Messaging) server might be the recipient of a third party REGISTER message.

The Net-Net USM accepts incoming REGISTER requests from UAs. After the UA has been authenticated and successfully registered with the registration database (an HSS or User Subscriber Database), the Net-Net USM forwards a modified copy of the REGISTER message to a 3rd party server.



REGISTER Message Generation

The 3rd Party Registration is generated by the Net-Net USM on behalf of the user in the To: header of REGISTER request.

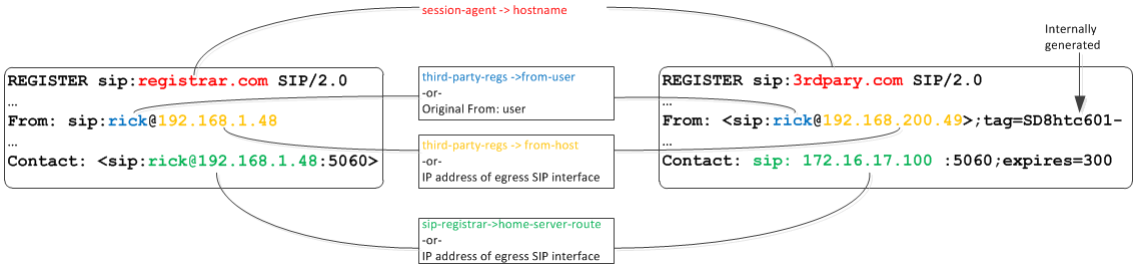
The request URI of the new REGISTER message uses the value of *registrar-host* parameter in the *third party regs* configuration element.

The From: header of the new REGISTER message uses the value of the *from-user* parameter in the *third party regs* configuration element as the user portion of the URI. If the *from-user* parameter is left blank, the Net-Net USM uses the user in the original From: header.

The From: header of the new REGISTER message uses the value of the *from-host* parameter in the *third party regs* configuration element as the host portion of the URI. If the fr*om-host* parameter is left blank, the Net-Net USM uses the IP address of the egress SIP interface as the host portion of the from header.

The Contact: header of the new REGISTER message uses the *home server route* parameter in the *sip registrar* configuration element. If the *home server route* parameter is left blank, the Net-Net USM uses the IP address of the egress interface.

See the following diagram:



3rd Party Registration Expiration

The REGISTER message sent from the Net-Net USM to the Register server uses the Expires: value returned from the User Subscriber Database or HSS. The 3rd party registration server sends a 200 OK message containing Contact bindings and an expires value chosen by the 3rd party server. The Net-Net USM checks each contact address to determine if it created it. For those address it created (as SD-Contacts) the Expires value from the 200 OK is used as the final value.

Once the expires timer has reached half the expires period as returned from the 3rd party registration server, the Net-Net USM refreshes the registration.

If the 3rd Party Registration server responds to a REGISTER Request with a 423 (Interval Too Brief) response, the Net-Net USM updates the contact's expiration interval to the Min-Expires value of the 423 response. It then submits a new REGISTER Request with the updated expires value.

3rd Party Registration Server States

If the 3rd party registration server does not respond to a REGISTER request, the Net-Net USM adheres to standard SIP session agent retransmission/ timeout procedures. If the 3rd party registration server is set to out of service, the Net-Net USM attempts a connectivity retry procedures. The retry procedures dictate that the Net-Net USM periodically send a REGISTER message to the 3rd party registration Server to check if connectivity has come back. The time interval for checking connectivity to a third party server is set with the *retry interval* parameter in the *third party regs* configuration element. Retries continue forever or until the third party server responds. The retry mechanism may be disabled by setting the *retry interval* parameter to 0.

When a 3rd party registration server is out of service, the Net-Net USM maintains a queue of outstanding third party registration requests. When the 3rd party registration server returns to service, the Net-Net USM gracefully flushes the queue of outstanding requests. This prevents a registration flood from being directed at the 3rd party registration server.

Defining 3rd Party Registrar

To send 3rd party registrations to a 3rd Party Registration server, three configuration elements are required. The primary configuration element is the *third party regs*. One or more may be configured in order to send the REGISTER message to multiple registration servers. You need to configure a *name* and set the *state* to enabled. The *registrar* host must be configured to indicate the value to insert into the Net-Net USM-generated request URI in the REGISTER message.

> **Note:** Acme Packet recommends that the list of third party registration server be restricted to a maximum of 3.

A session agent needs to represent the 3rd Party Registrar. Create a session agent as the 3rd Party Registrar server and note its name. Next, configure the *registrar-host* parameter with a session agent *hostname* in the *third-party-reg* configuration element. This specifies the session agent to be used as the registrar.

Finally, the address of the 3rd Party registrar must be added to the *third-party-registrars* parameter in the *sip-registrar* configuration element. This does not supercede any core SMX Registrar functionality. It informs the Net-Net USM of the 3rd Party Registar(s) to send messages to after initial registration. Thus the value configured here must exist in the *third-party-regs* configuration element's *registrar-host* parameter list.

ACLI Instructions

Third Party Registrar

To configure a 3rd Party Registrar:

1. In Superuser mode, type **configure terminal** and press <Enter>.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type **session-router** and press <Enter> to access the session router path.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type **third-party-regs** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.

   ```
   ACMEPACKET(session-router)# third-party-regs
   ACMEPACKET(third-party-regs)#
   ```

4. **name**—Enter a name for this 3rd Party Registrar configuration element.
5. **state**—Set this to enabled to use this configuration.
6. **registrar-host**—Set this value to the complementary session agents' *hostname* parameter to include those session agents as 3rd Party registrars. This parameter may be modified like an options parameter. This value also appears in the request URI of the outgoing REGISTER message being sent to the 3rd Party Registrar.
7. **from-user**—Configure this parameter to be the user portion of the From: header of the outgoing REGISTER message being sent to the 3rd Party Registrar. Leaving this blank sets the user portion that in the original From: header
8. **from-host**—Configure this parameter to be the host portion of the From: header of the outgoing REGISTER message being sent to the 3rd Party Registrar. Leaving this blank sets the host portion to the Net-Net USM's egress SIP interface.
9. **retry-interval**—Enter the number of seconds the Net-Net USM waits before retrying a 3rd Party Registration server after a failed registration. Enter **0** to disable this feature.
10. Type **done** when finished.

SIP Registrar

To indicate to a local SIP Registrar when and what 3rd Party Registrar to send 3rd Party registrations to:

1. In Superuser mode, type **configure terminal** and press <Enter>.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type **session-router** and press <Enter> to access the session router path.

```
ACMEPACKET(configure)# session-router
```

3. Type **sip-registrar** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# sip-registrar
ACMEPACKET(sip-registrar)#
```

4. Type **select** and choose the number of the pre-configured SIP registrar configuration element you want to configure.

```
ACMEPACKET(sip-registrar)# select
name:
1: registrar1

selection:1
ACMEPACKET(sip-registrar)#
```

5. **home-server-route**—Enter the value to insert into the REGISTER message's request URI as sent to the 3rd party registration server. Leaving this blank uses the AoR (or To: header) in the original REGISTER message.
6. **third-party-registrars**—Enter the name of a *third party regs* configuration element *registrar-host* parameter to send third part registrations associated with that SIP registrar.
7. Type **done** when finished.

Session Agent

To create a session agent to represent the 3rd Party Registration Server:

1. In Superuser mode, type **configure terminal** and press <Enter>.

```
ACMEPACKET# configure terminal
```

2. Type **session-router** and press <Enter> to access the session router path.

```
ACMEPACKET(configure)# session-router
```

3. Type **session-agent** and press <Enter>. The system prompt changes to let you know that you can begin configuring individual parameters.

```
ACMEPACKET(session-router)# session-agent
ACMEPACKET(session-agent)#
```

4. **hostname**—Enter the name for this session agent.
5. **ip-address**—Enter the IP address for this session agent. This value must be the same as the registrar-host parameter in the third party regs configuration element to which this session agent definition corresponds.

   Continue configuring this session agent's parameters. Not all session agent functionality is applicable to the Net-Net USM.
6. Type **done** when finished.

5 RADIUS Accounting of REGISTERs

CDR Generation for REGISTER Events

The Net-Net USM can generate RADIUS CDRs, per Contact's event, for registration, refresh registration, and registration removal. A single REGISTER message can generate multiple RADIUS CDRs since that message may contain multiple contacts.

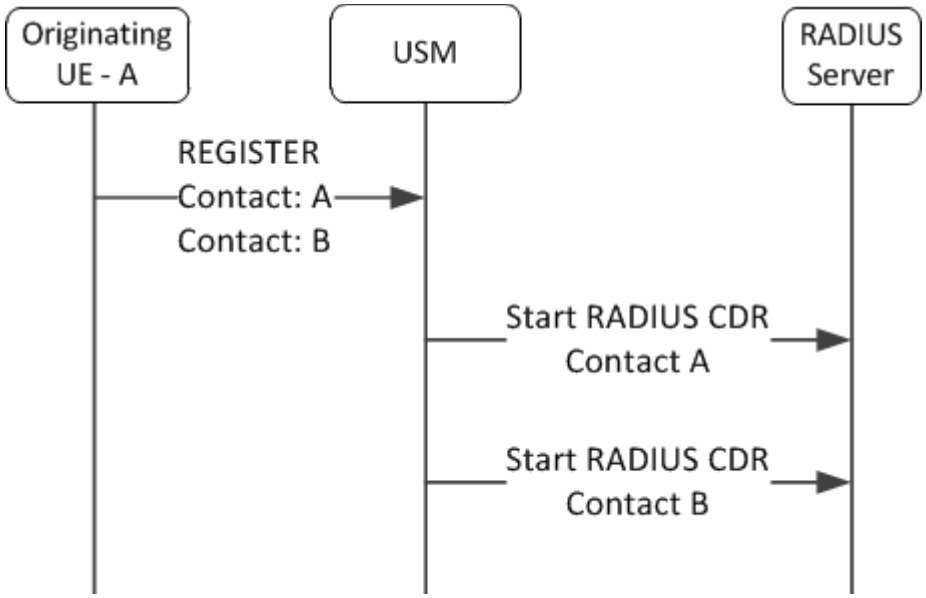
To enable CDR generation for received REGISTERs, you set the **generate event** parameter in the *account-config* configuration element to **register** or **local-register**. The register value may exist with other events such as **invite**.

REGISTER Scenarios

RADIUS CDRs are generated for each registration change per Contact. There are 5 main scenarios which cover CDR creation.

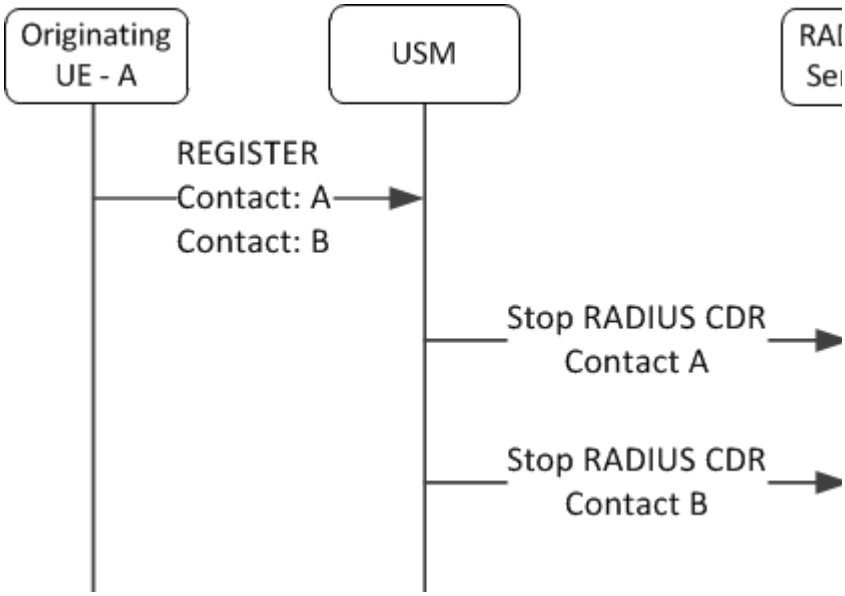
Initial REGISTER

One or more RADIUS Start CDRs are sent to the RADIUS server for each contact in a successful REGISTER message before the Net-Net USM replies to the endpoint with a 200 OK.



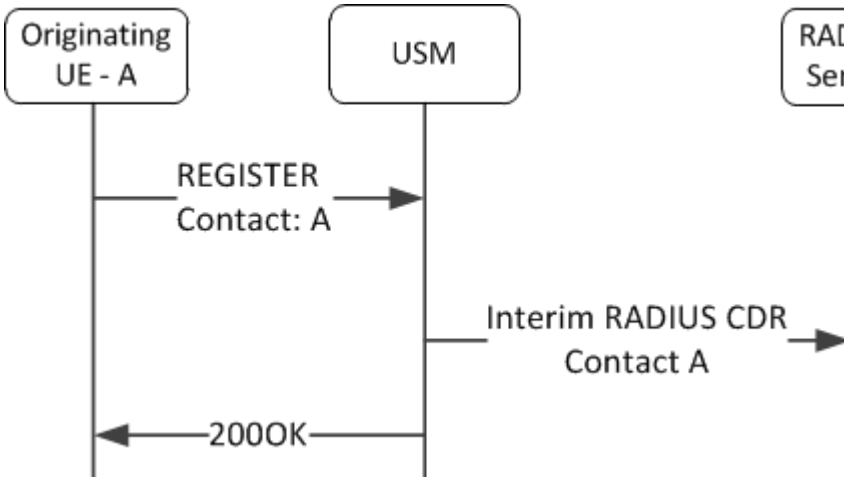
Failed REGISTER

One or more RADIUS Stop CDRs are sent to the RADIUS server for each contact in an unsuccessful REGISTER message before the Net-Net USM replies to the endpoint with a SIP Final Response (4xx or 5xx) message.



REGISTER Refresh

One or more RADIUS Interim CDRs are sent to the RADIUS server for each contact in a successful reREGISTER message before the Net-Net USM replies to the endpoint with a 200 OK. This happens when a database query is made and succeeds.



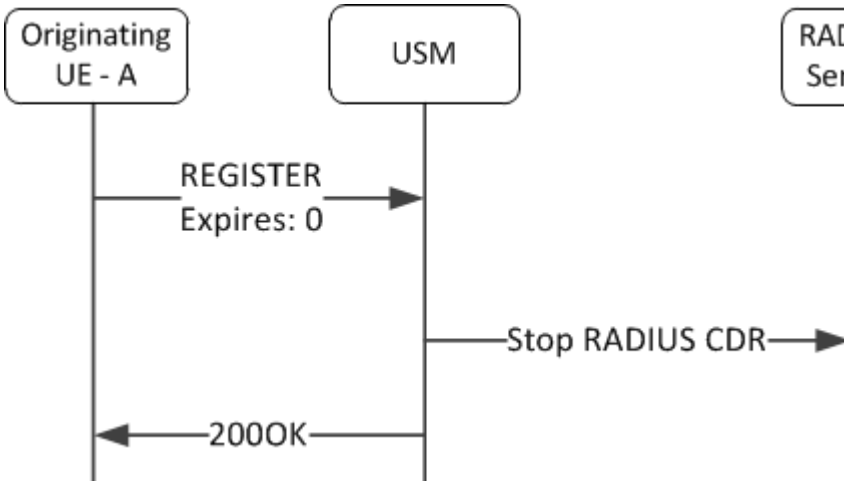
Failed REGISTER Refresh

One or more RADIUS Interim CDRs are sent to the RADIUS server for each contact in an unsuccessful reREGISTER message before the Net-Net USM replies to the endpoint with a SIP Final Response (4xx or 5xx) message.

deREGISTER

One or more RADIUS Stop CDRs are sent to the RADIUS server for the contacts in a deREGISTER message before the Net-Net USM replies to the endpoint with a 200 OK. The Net-Net USM interprets an expires=0 parameter in a Contact: header as

only removing the registration (and sending a corresponding stop record) for that contact, or an Expires: 0 header prompts Stop RADIUS records for all contacts.



Registration Update

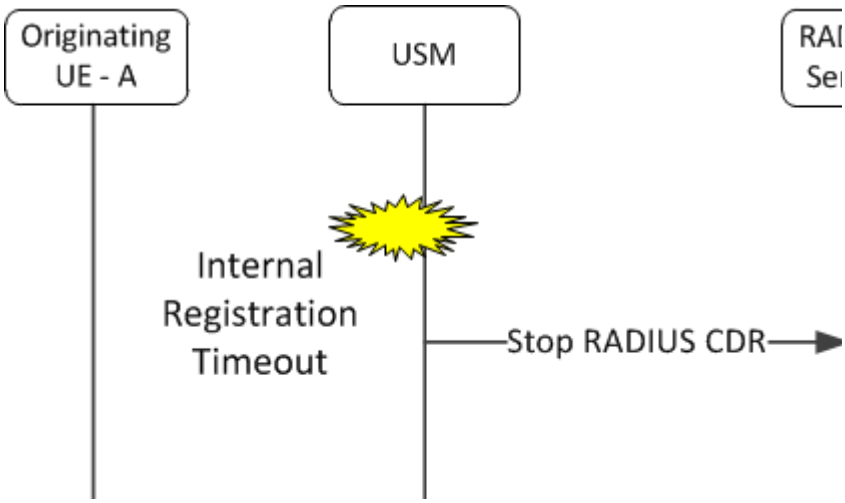
For each Contact's registration update with an existing Instance-ID and AoR, the Net-Net USM sequentially sends a RADIUS Stop CDR for the original contact address and then a RADIUS Start CDR for the new contact address.

Similarly, when the Net-Net USM receives a REGISTER for an existing Contact and AoR but with a different Instance-ID, the registration is updated by the Net-Net USM. The same corresponding stop and start CDRs are sent to the RADIUS server.

Internal Deregistration

When a condition occurring within the Net-Net USM causes a contact's registration to be removed, a RADIUS Stop CDR is sent to the RADIUS server. In such a case,

generally no indication is sent to the UA. Examples are if the registration times out, or if the contact is manually removed at the ACLI.



REGISTER VSA Format

The following table lists new VSAs introduced with this feature.

Parameter Name	Dictionary Index	Data Type	Valid Records	Definition
Acme-SIP-Method-Type	235	String	Start, Interim, Stop	This is the SIP method type that is associated to the CDR. Possible values are INVITE, BYE, or REGISTER.
Acme-Domain-Name	236	String	Start, Interim, Stop	This is domain name of the request URI.
Acme-SIP-Contact	237	String	Start, Interim, Stop	This is contact from SIP message. This is not the entire contact header.
Acme-SIP-Expires	238	Integer	Start, Interim	This is the expires value of the Expires: header from the sip message.
Acme-Reason-Phrase	239	String	Interim, Stop	This is the SMX reason code. This will not be set for all Stop and Interim

The following table lists the new definitions of existing VSAs when CDRs are created on REGISTER messages.

Parameter Name	Dictionary Index	Data Type	Valid Records	Definition
Acme-SIP-Status	71	String	Interim	

Parameter Name	Dictionary Index	Data Type	Valid Records	Definition
Acct-Session-Id	44	String	Start, Interim, Stop	This is the a unique string assigned to the contact. The string is made up of the system name concatenated with a timestamp and an 8 digited hex number. It is of form <system name>-<timestamp>-<number>.
Called-Station-Id		String	Start, Interim, Stop	When in a stop record, this value is populated when an internal reason causes the stop record to be generated. in this case it is the AoR.

The following pairs of Acme-Disconnect-Initiator and Acme-Disconnect-Cause VSAs, as presented in a RADIUS stop CDR that corresponds to a REGISTER message are defined in the Reason column.

Reason	Acme-Disconnect-Initiator	Acme-Disconnect-Cause
UA requested deregistration	1 - CALLING_PARTY_DISCONNECT	1-PW_CAUSE_USER_REQUEST
Contact Times Out	3 - INTERNAL_DISCONNECT	4 -PW_CAUSE_IDLE_TIMEOUT
REGISTER error on establishment.	1 - CALLING_PARTY_DISCONNECT	17-PW - CAUSE_USER_ERROR
RTR from HSS specifies user's private ID	3 - INTERNAL_DISCONNECT	20 - PW_CAUSE_RTR_REQUEST_PRIV
RTR from HSS specifies user's public ID	3 - INTERNAL_DISCONNECT	21 - PW_CAUSE_RTR_REQUEST_PUB
Contact's registration is removed via ACLI command	3 - INTERNAL_DISCONNECT	1-PW_CAUSE_USER_REQUEST
Reuse of ID by a different Contact	3 - INTERNAL_DISCONNECT	22-PW_CAUSE_REUSE_ID

ACLI Instructions

To add CDR generation on REGISTER messages:

1. In Superuser mode, type **configure terminal** and press <Enter>.

   ```
   ACMEPACKET# configure terminal
   ```

2. Type **session-router** and press <Enter>.

   ```
   ACMEPACKET(configure)# session-router
   ```

3. Type **account-config** and press <Enter>.

   ```
   ACMEPACKET(session-router)# account-config
   ```

   From this point, you can reach the individual parameters for duplicate RADIUS attribute prevention and for RADIUS attribute selection.

4. **generate-events**—Set this parameter to register and/or local-register.
5. Save and activate your configuration.

Example CDRs

The following examples list when basic registrations create CDRs.

Initial Registration CDR

A Start CDR is created for an initial REGISTER received on the Net-Net USM's 192.168.101.20 interface from 192.168.12.12.

REGISTER message:

```
REGISTER sip:registrar.biloxi.com SIP/2.0
Via: SIP/2.0/UDP bobspc.biloxi.com:5060;branch=z9hG4bKnashds7
Max-Forwards: 70
To: Bob <sip:bob@biloxi.com>
From: Bob <sip:bob@biloxi.com>;tag=456248
Call-ID: 843817637684230@998sdasdh09
User-Agent: Softphone Beta1.5
Authorization: Digest username="Bob", realm=" biloxi.com",
    nonce="84a4cc6f3082121f32b42a2187831a9e",
    response="7587245234b3434cc3412213e5f113a5432"
CSeq: 1826 REGISTER
Contact: <sip:bob@192.168.12.12>
Expires: 7200
Content-Length: 0
```

Start CDR Message, selected attributes:

```
NAS-Identifier = "abc123"
Acct-Status-Type = Start
NAS-IP-Address = 172.16.101.20
NAS-Port = 5060
Acct-Session-Id = "Iapetus-C00000001"
Acme-Session-Ingress-CallId = "843817637684230@998sdasdh09"
Acme-Session-Protocol-Type = "SIP"
Calling-Station-Id = " Bob <sip:bob@biloxi.com>;tag=456248"
Called-Station-Id = " Bob <sip:bob@biloxi.com>"
Acme-Ingress-Network-Interface-Id = "M00"
Acme-Ingress-Vlan-Tag-Value = 0
Acme-Session-Ingress-Realm = "net192"
Acme-Firmware-Version = "SCX6.3.3 F-1 GA (WS Build 18)"
Acme-Local-Time-Zone = "Time Zone Not Set"
Acme-Ingress-Local-Addr = "192.168.101.20:5060"
Acme-Ingress-Remote-Addr = "192.168.12.12:5060"
Acme-SIP-Method-Type = "REGISTER"
Acme-Domain-Name = "registrar.biloxi.com"
Acme-SIP-Contact = "sip:bob@192.168.12.12"
Acme-SIP-Expires = 7200
Acme-CDR-Sequence-Number = 1
Client-IP-Address = 172.30.70.121
Acct-Unique-Session-Id = "51a15d4381d9fe38"
Timestamp = 1329241213
```

Interim Registration CDR

An interim CDR is created for a REGISTER refresh received on the Net-Net USM's 192.168.101.20 interface from 192.168.12.12.

REGISTER message:

```
REGISTER sip:registrar.biloxi.com SIP/2.0
Via: SIP/2.0/UDP bobspc.biloxi.com:5060;branch=z9hG4bKnashds7
Max-Forwards: 70
To: Bob <sip:bob@biloxi.com>
From: Bob <sip:bob@biloxi.com>;tag=456248
```

```
Call-ID: 843817637684230@998sdasdh09
User-Agent: Softphone Beta1.5
Authorization: Digest username="Bob", realm=" biloxi.com",
    nonce="84a4cc6f3082121f32b42a2187831a9e",
    response="7587245234b3434cc3412213e5f113a5432"
CSeq: 1826 REGISTER
Contact: <sip:bob@192.168.12.12>
Expires: 7200
Content-Length: 0
```

Interim CDR Message, selected attributes:

```
NAS-Identifier = "abc123"
Acct-Status-Type = Interim-Update
NAS-IP-Address = 172.16.101.20
NAS-Port = 5060
Acct-Session-Id = " Iapetus-C00000001"
Acme-Session-Ingress-CallId = "843817637684230@998sdasdh09"
Acme-Session-Protocol-Type = "SIP"
Calling-Station-Id = " Bob <sip:bob@biloxi.com>;tag=456248"
Called-Station-Id = " Bob <sip:bob@biloxi.com>"
Acme-Ingress-Network-Interface-Id = "M00"
Acme-Ingress-Vlan-Tag-Value = 0
Acme-Session-Ingress-Realm = "net192"
Acme-Firmware-Version = "SCX6.3.3 F-1 GA (WS Build 18)"
Acme-Local-Time-Zone = "Time Zone Not Set"
Acme-Ingress-Local-Addr = "192.168.101.20:5060"
Acme-Ingress-Remote-Addr = "192.168.12.12:5060"
Acme-SIP-Method-Type = "REGISTER"
Acme-Domain-Name = "registrar.biloxi.com"
Acme-SIP-Contact = "sip:bob@192.168.12.12"
Acme-SIP-Expires = 7200
Acme-SIP-Status = "200"
Acme-Reason-Phrase = "OK"
Acme-CDR-Sequence-Number = 1
Client-IP-Address = 172.30.70.121
Acct-Unique-Session-Id = "51a15d4381d9fe38"
Timestamp = 1329241213
```

STOP CDR on REGISTER message

Stop CDRs are generated for REGISTER messages with Expires of 0 from a user agent, a failed initial registration, or Net-Net USM removing the registration. A Stop CDR can take one of two forms:

1. The Stop CDR is generated as part of receiving a response to a REGISTER message.
2. The Stop CDR is generated as part of an internal event such as a Contact timing out.

REGISTER message:

The following REGISTER contains an expires of 0 received on interface 192.168.101.20 from 192.168.12.12:

```
REGISTER sip:registrar.biloxi.com SIP/2.0
Via: SIP/2.0/UDP bobspc.biloxi.com:5060;branch=z9hG4bKnashds7
Max-Forwards: 70
```

```
To: Bob <sip:bob@biloxi.com>
From: Bob <sip:bob@biloxi.com>;tag=456248
Call-ID: 843817637684230@998sdasdh09
User-Agent: Softphone Beta1.5
Authorization: Digest username="Bob", realm=" biloxi.com",
    nonce="84a4cc6f3082121f32b42a2187831a9e",
    response="7587245234b3434cc3412213e5f113a5432"
CSeq: 1826 REGISTER
Contact: <sip:bob@192.168.12.12>
Expires: 0
Content-Length: 0
```

Stop CDR Message, selected attributes:

```
NAS-Identifier = "abc123"
Acct-Status-Type = Stop
NAS-IP-Address = 172.16.101.20
NAS-Port = 5060
Acct-Session-Id = " Iapetus-C00000001"
Acme-Session-Ingress-CallId = "843817637684230@998sdasdh09"
Acme-Session-Protocol-Type = "SIP"
Calling-Station-Id = " Bob <sip:bob@biloxi.com>;tag=456248"
Called-Station-Id = " Bob <sip:bob@biloxi.com>"
Acme-Ingress-Network-Interface-Id = "M00"
Acme-Ingress-Vlan-Tag-Value = 0
Acme-Session-Ingress-Realm = "net192"
Acme-Firmware-Version = "SCX6.3.3 F-1 GA (WS Build 18)"
Acme-Local-Time-Zone = "Time Zone Not Set"
Acme-Ingress-Local-Addr = "192.168.101.20:5060"
Acme-Ingress-Remote-Addr = "192.168.12.12:5060"
Acme-Disconnect-Initiator = 1
Acme-Disconnect-Cause = 1
Acme-SIP-Status = "200"
Acme-SIP-Method-Type = "REGISTER"
Acme-Domain-Name = "registrar.biloxi.com"
Acme-SIP-Contact = "sip:bob@192.168.12.12"
Acme-Reason-Phrase = "OK"
Acme-CDR-Sequence-Number = 1
Client-IP-Address = 172.30.70.121
Acct-Unique-Session-Id = "51a15d4381d9fe38"
Timestamp = 1329241213
```

Net-Net USM Initiated Deregistration

Since no SIP Result code is returned to an endpoint when it was internally deregistered by the Net-Net USM, its corresponding VSA is not created and will not appear in the CDR.

No REGISTER message received.

Stop CDR Message, selected attributes:

```
NAS-Identifier = "abc123"
Acct-Status-Type = Stop
NAS-IP-Address = 172.16.101.20
NAS-Port = 5060
Acct-Session-Id = " Iapetus-C00000001"
Called-Station-Id = "sip:bob@biloxi.com"
```

```
Acme-Session-Protocol-Type = "SIP"
Acme-Disconnect-Initiator = 3
Acme-Disconnect-Cause = 4
Acme-SIP-Method-Type = "REGISTER"
Acme-Domain-Name = "registrar.biloxi.com"
Acme-SIP-Contact = "sip:bob@192.168.12.12"
Acme-CDR-Sequence-Number = 1
Client-IP-Address = 172.30.70.121
Acct-Unique-Session-Id = "51a15d4381d9fe38"
Timestamp = 1329241213
```

Local CDR CSV Orientation

This section lists the order of VSAs (and other statistics) in local CDR files.

Start Record

CSV Placement	Attribute Name	ACME VSA ID
1	Acct-Status-Type	
2	NAS-IP-Address	
3	NAS-Port	
4	Acct-Session-Id	
5	Acme-Session-Ingress-CallId	3
6	Acme-Session–Egress-CallId	4
7	Acme-Session-Protocol-Type	43
8	Acme-Session-Forked-Call-Id	171
9	Acme-Session–Generic-Id	40
10	Calling-Station-Id	
11	Called-Station-Id	
12	h323-setup-time	
13	h323-connect-time	
14	Acme-Egress-Network-Interface-Id	139
15	Acme-Egress-Vlan-Tag-Value	140
16	Acme-Ingress-Network-Interface-Id	137
17	Acme-Ingress-Vlan-Tag-Value	138
18	Acme-Session-Egress-Realm	42
19	Acme-Session-Ingress-Realm	41
20	Acme-FlowId_FS1_F	1
21	Acme-FlowType_FS1_F	2
22	Acme-Flow-In-Realm_FS1_F	10
23	Acme-Flow-In-Src-Addr_FS1_F	11
24	Acme-Flow-In-Src-Port_FS1_F	12

CSV Placement	Attribute Name	ACME VSA ID
25	Acme-Flow-In-Dst-Addr_FS1_F	13
26	Acme-Flow-In-Dst-Port_FS1_F	14
27	Acme-Flow-Out-Realm_FS1_F	20
28	Acme-Flow-Out-Src-Addr_FS1_F	21
29	Acme-Flow-Out-Src-Port_FS1_F	22
30	Acme-Flow-Out-Dst-Addr_FS1_F	23
31	Acme-Flow-Out-Dst-Port_FS1_F	24
32	Acme-FlowID_FS1_R	78
33	Acme-FlowType_FS1_R	79
34	Acme-Flow-In-Realm_FS1_R	80
35	Acme-Flow-In-Src-Addr_FS1_R	81
36	Acme-Flow-In-Src-Port_FS1_R	82
37	Acme-Flow-In-Dst-Addr_FS1_R	83
38	Acme-Flow-In-Dst-Port_FS1_R	84
39	Acme-Flow-Out-Realm_FS1_R	85
40	Acme-Flow-Out-Src-Addr_FS1_R	86
41	Acme-Flow-Out-Src-Port_FS1_R	87
42	Acme-Flow-Out-Dst-Addr_FS1_R	88
43	Acme-Flow-Out-Dst-Port_FS1_R	89
44	Acme-FlowID_FS2_F	90
45	Acme-FlowType_FS2_F	91
46	Acme-Flow-In-Realm_FS2_F	92
47	Acme-Flow-In-Src-Addr_FS2_F	93
48	Acme-Flow-In-Src-Port_FS2_F	94
49	Acme-Flow-In-Dst-Addr_FS2_F	95
50	Acme-Flow-In-Dst-Port_FS2_F	96
51	Acme-Flow-Out-Realm_FS2_F	97
52	Acme-Flow-Out-Src-Addr_FS2_F	98
53	Acme-Flow-Out-Src-Port_FS2_F	99
54	Acme-Flow-Out-Dst-Addr_FS2_F	100
55	Acme-Flow-Out-Dst-Port_FS2_F	101
56	Acme-FlowID_FS2_R	112
57	Acme-FlowType_FS2_R	113
58	Acme-Flow-In-Realm_FS2_R	114

CSV Placement	Attribute Name	ACME VSA ID
59	Acme-Flow-In-Src-Addr_FS2_R	115
60	Acme-Flow-In-Src-Port_FS2_R	116
61	Acme-Flow-In-Dst-Addr_FS2_R	117
62	Acme-Flow-In-Dst-Port_FS2_R	118
63	Acme-Flow-Out-Realm_FS2_R	119
64	Acme-Flow-Out-Src-Addr_FS2_R	120
65	Acme-Flow-Out-Src-Port_FS2_R	121
66	Acme-Flow-Out-Dst-Addr_FS2_R	122
67	Acme-Flow-Out-Dst-Port_FS2_R	123
68	Acme-Session-Charging-Vector	54
69	Acme-Session-Charging-Function_Address	55
70	Acme-Firmware-Version	56
71	Acme-Local-Time-Zone	57
72	Acme-Post-Dial-Delay	58
73	Acme-Primary-Routing-Number	64
74	Acme-Originating-Trunk-Group	65
75	Acme-Terminating-Trunk-Group	66
76	Acme-Originating-Trunk-Context	67
77	Acme-Terminating-Trunk-Context	68
78	Acme-P-Asserted-ID	69
79	Acme-Ingress-Local-Addr	74
80	Acme-Ingress-Remote-Addr	75
81	Acme-Egress-Local-Addr	76
82	Acme-Egress-Remote-Addr	77
83	Acme-SIP-Diversion	70
84	Acme-Egress-Final-Routing-Number	134
85	Acme-Session-Ingress-RPH	135
86	Acme-Session-Egress-RPH	136
87	Acme-Custom-VSA-200	200
88	Acme-Custom-VSA-201	201
89	Acme-Custom-VSA-202	202
90	Acme-Custom-VSA-203	203
91	Acme-Custom-VSA-204	204
92	Acme-Custom-VSA-205	205

CSV Placement	Attribute Name	ACME VSA ID
93	Acme-Custom-VSA-206	206
94	Acme-Custom-VSA-207	207
95	Acme-Custom-VSA-208	208
96	Acme-Custom-VSA-209	209
97	Acme-Custom-VSA-210	210
98	Acme-Custom-VSA-211	211
99	Acme-Custom-VSA-212	212
100	Acme-Custom-VSA-213	213
101	Acme-Custom-VSA-214	214
102	Acme-Custom-VSA-215	215
103	Acme-Custom-VSA-216	216
104	Acme-Custom-VSA-217	217
105	Acme-Custom-VSA-218	218
106	Acme-Custom-VSA-219	219
107	Acme-Custom-VSA-220	220
108	Acme-Custom-VSA-221	221
109	Acme-Custom-VSA-222	222
110	Acme-Custom-VSA-223	223
111	Acme-Custom-VSA-224	224
112	Acme-Custom-VSA-225	225
113	Acme-Custom-VSA-226	226
114	Acme-Custom-VSA-227	227
115	Acme-Custom-VSA-228	228
116	Acme-Custom-VSA-229	229
117	Acme-Custom-VSA-230	230
118	Acme-Flow-Calling-Media-Stop-Time_FS1	231
119	Acme-Flow-Called-Media-Stop-Time_FS1	232
120	Acme-Flow-Calling-Media-Stop-Time_FS2	233
121	Acme-Flow-Called-Media-Stop-Time_FS2	234
122	Acme-FlowMediaType_FS1_F	142
123	Acme-FlowMediaType_FS1_R	143
124	Acme-FlowMediaType_FS2_F	144

CSV Placement	Attribute Name	ACME VSA ID
125	Acme-FlowMediaType_FS2_R	145
126	Acme-SIP-Method-Type	235
127	Acme-Domain-Name	236
128	Acme-SIP-Contact	237
129	Acme-SIP-Expires	238
130	Acme-CDR-Sequence-Number	59

Interim Record

CSV Placement	Attribute Name	ACME VSA ID
1	Acct-Status-Type	
2	NAS-IP-Address	
3	NAS-Port	
4	Acct-Session-Id	
5	Acme-Session-Ingress-CallId	3
6	Acme-Session–Egress-CallId	4
7	Acme-Session-Protocol-Type	43
9	Acme-Session-Forked-Call-Id	171
8	Acme-Session–Generic-Id	40
10	Calling-Station-Id	
11	Called-Station-Id	
12	h323-setup-time	
13	h323-connect-time	
14	Acme-Egress-Network-Interface-Id	139
15	Acme-Egress-Vlan-Tag-Value	140
16	Acme-Ingress-Network-Interface-Id	137
17	Acme-Ingress-Vlan-Tag-Value	138
18	Acme-Session-Egress-Realm	42
19	Acme-Session-Ingress-Realm	41
20	Acme-FlowId_FS1_F	1
21	Acme-FlowType_FS1_F	2
22	Acme-Flow-In-Realm_FS1_F	10
23	Acme-Flow-In-Src-Addr_FS1_F	11
24	Acme-Flow-In-Src-Port_FS1_F	12

CSV Placement	Attribute Name	ACME VSA ID
25	Acme-Flow-In-Dst-Addr_FS1_F	13
26	Acme-Flow-In-Dst-Port_FS1_F	14
27	Acme-Flow-Out-Realm_FS1_F	20
28	Acme-Flow-Out-Src-Addr_FS1_F	21
29	Acme-Flow-Out-Src-Port_FS1_F	22
30	Acme-Flow-Out-Dst-Addr_FS1_F	23
31	Acme-Flow-Out-Dst-Port_FS1_F	24
32	Acme-Calling-RTCP-Packets-Lost_FS1	32
33	Acme-Calling-RTCP-Avg-Jitter_FS1	33
34	Acme-Calling-RTCP-Avg-Latency_FS1	34
35	Acme-Calling-RTCP-MaxJitter_FS1	35
36	Acme-Calling-RTCP-MaxLatency_FS1	36
37	Acme-Calling-RTP-Packets-Lost_FS1	37
38	Acme-Calling-RTP-Avg-Jitter_FS1	38
39	Acme-Calling-RTP-MaxJitter_FS1	39
40	Acme-Calling-Octets_FS1	28
41	Acme-Calling-Packets_FS1	29
42	Acme-Calling-R-Factor	151
43	Acme-Calling-MOS	152
44	Acme-FlowID_FS1_R	78
45	Acme-FlowType_FS1_R	79
46	Acme-Flow-In-Realm_FS1_R	80
47	Acme-Flow-In-Src-Addr_FS1_R	81
48	Acme-Flow-In-Src-Port_FS1_R	82
49	Acme-Flow-In-Dst-Addr_FS1_R	83
50	Acme-Flow-In-Dst-Port_FS1_R	84
51	Acme-Flow-Out-Realm_FS1_R	85
52	Acme-Flow-Out-Src-Addr_FS1_R	86
53	Acme-Flow-Out-Src-Port_FS1_R	87
54	Acme-Flow-Out-Dst-Addr_FS1_R	88
55	Acme-Flow-Out-Dst-Port_FS1_R	89
56	Acme-Called-RTCP-Packets-Lost_FS1	46
57	Acme-Called-RTCP-Avg-Jitter_FS1	47
58	Acme-Called-RTCP-Avg-Latency_FS1	48

CSV Placement	Attribute Name	ACME VSA ID
59	Acme-Called-RTCP-MaxJitter_FS1	49
60	Acme-Called-RTCP-MaxLatency_FS1	50
61	Acme-Called-RTP-Packets-Lost_FS1	51
62	Acme-Called-RTP-Avg-Jitter_FS1	52
63	Acme-Called-RTP-MaxJitter_FS1	53
64	Acme-Called-Octets_FS1	44
65	Acme-Called-Packets_FS1	45
66	Acme-Called-R-Factor	153
67	Acme-Called-MOS	154
68	Acme-FlowID_FS2_F	90
69	Acme-FlowType_FS2_F	91
70	Acme-Flow-In-Realm_FS2_F	92
71	Acme-Flow-In-Src-Addr_FS2_F	93
72	Acme-Flow-In-Src-Port_FS2_F	94
73	Acme-Flow-In-Dst-Addr_FS2_F	95
74	Acme-Flow-In-Dst-Port_FS2_F	96
75	Acme-Flow-Out-Realm_FS2_F	97
76	Acme-Flow-Out-Src-Addr_FS2_F	98
77	Acme-Flow-Out-Src-Port_FS2_F	99
78	Acme-Flow-Out-Dst-Addr_FS2_F	100
79	Acme-Flow-Out-Dst-Port_FS2_F	101
80	Acme-Calling-RTCP-Packets-Lost_FS2	104
81	Acme-Calling-RTCP-Avg-Jitter_FS2	105
82	Acme-Calling-RTCP-Avg-Latency_FS2	106
83	Acme-Calling-RTCP-MaxJitter_FS2	107
84	Acme-Calling-RTCP-MaxLatency_FS2	108
85	Acme-Calling-RTP-Packets-Lost_FS2	109
86	Acme-Calling-RTP-Avg-Jitter_FS2	110
87	Acme-Calling-RTP-MaxJitter_FS2	111
88	Acme-Calling-Octets_FS2	102
89	Acme-Calling-Packets_FS2	103
90	Acme-FlowID_FS2_R	112
91	Acme-FlowType_FS2_R	113
92	Acme-Flow-In-Realm_FS2_R	114

CSV Placement	Attribute Name	ACME VSA ID
93	Acme-Flow-In-Src-Addr_FS2_R	115
94	Acme-Flow-In-Src-Port_FS2_R	116
95	Acme-Flow-In-Dst-Addr_FS2_R	117
96	Acme-Flow-In-Dst-Port_FS2_R	118
97	Acme-Flow-Out-Realm_FS2_R	119
98	Acme-Flow-Out-Src-Addr_FS2_R	120
99	Acme-Flow-Out-Src-Port_FS2_R	121
100	Acme-Flow-Out-Dst-Addr_FS2_R	122
101	Acme-Flow-Out-Dst-Port_FS2_R	123
102	Acme-Called-RTCP-Packets-Lost_FS2	126
103	Acme-Called–RTCP-Avg-Jitter_FS2	127
104	Acme-Called–RTCP-Avg-Latency_FS2	128
105	Acme-Called–RTCP-MaxJitter_FS2	129
106	Acme-Called-RTCP-MaxLatency_FS2	130
107	Acme-Called-RTP-Packets-Lost_FS2	131
108	Acme-Called-RTP-Avg-Jitter_FS2	132
109	Acme-Called-RTP-MaxJitter_FS2	133
110	Acme-Called-Octets_FS2	124
111	Acme-Called-Packets_FS2	125
112	Acme-Session-Charging-Vector	54
113	Acme-Session-Charging-Function_Address	55
114	Acme-Firmware-Version	56
115	Acme-Local-Time-Zone	57
116	Acme-Post-Dial-Delay	58
117	Acme-Primary-Routing-Number	64
118	Acme-Originating-Trunk-Group	65
119	Acme-Terminating-Trunk-Group	66
120	Acme-Originating-Trunk-Context	67
121	Acme-Terminating-Trunk-Context	68
122	Acme-P-Asserted-ID	69
123	Acme-Ingress-Local-Addr	74
124	Acme-Ingress-Remote-Addr	75
125	Acme-Egress-Local-Addr	76
126	Acme-Egress-Remote-Addr	77

CSV Placement	Attribute Name	ACME VSA ID
127	Acme-SIP-Diversion	70
128	Acme-Intermediate_Time	63
129	Acct-Session-Time	
130	Acme-Egress-Final-Routing-Number	134
131	Acme-Session-Ingress-RPH	135
132	Acme-Session-Egress-RPH	136
133	Acme-Custom-VSA-200	200
134	Acme-Custom-VSA-201	201
135	Acme-Custom-VSA-202	202
136	Acme-Custom-VSA-203	203
137	Acme-Custom-VSA-204	204
138	Acme-Custom-VSA-205	205
139	Acme-Custom-VSA-206	206
140	Acme-Custom-VSA-207	207
141	Acme-Custom-VSA-208	208
142	Acme-Custom-VSA-209	209
143	Acme-Custom-VSA-210	210
144	Acme-Custom-VSA-211	211
145	Acme-Custom-VSA-212	212
146	Acme-Custom-VSA-213	213
147	Acme-Custom-VSA-214	214
148	Acme-Custom-VSA-215	215
149	Acme-Custom-VSA-216	216
150	Acme-Custom-VSA-217	217
151	Acme-Custom-VSA-218	218
152	Acme-Custom-VSA-219	219
153	Acme-Custom-VSA-220	220
154	Acme-Custom-VSA-221	221
155	Acme-Custom-VSA-222	222
156	Acme-Custom-VSA-223	223
157	Acme-Custom-VSA-224	224
158	Acme-Custom-VSA-225	225
159	Acme-Custom-VSA-226	226
160	Acme-Custom-VSA-227	227

CSV Placement	Attribute Name	ACME VSA ID
161	Acme-Custom-VSA-228	228
162	Acme-Custom-VSA-229	229
163	Acme-Custom-VSA-230	230
164	Acme-Flow-Calling-Media-Stop-Time_FS1	231
165	Acme-Flow-Called-Media-Stop-Time_FS1	232
166	Acme-Flow-Calling-Media-Stop-Time_FS2	233
167	Acme-Flow-Called-Media-Stop-Time_FS2	234
168	Acme-FlowMediaType_FS1_F	142
169	Acme-FlowMediaType_FS1_R	143
170	Acme-FlowMediaType_FS2_F	144
171	Acme-FlowMediaType_FS2_R	145
172	Acme-SIP-Method-Type	235
173	Acme-Domain-Name	236
174	Acme-SIP-Contact	237
175	Acme-SIP-Expires	238
176	Acme-SIP-Status	71
177	Acme-Reason-Phrase	239
178	Acme-CDR-Sequence-Number	59

Stop Record

CSV Placement	Attribute Name	ACME VSA ID
1	Acct-Status-Type	
2	NAS-IP-Address	
3	NAS-Port	
4	Acct-Session-Id	
5	Acme-Session-Ingress-CallId	3
6	Acme-Session–Egress-CallId	4
7	Acme-Session-Protocol-Type	43
8	Acme-Session-Forked-Call-Id	171
9	Acme-Session–Generic-Id	40

CSV Placement	Attribute Name	ACME VSA ID
10	Calling-Station-Id	
11	Called-Station-Id	
12	Acct-Terminate-Cause	
13	Acct-Session-Time	
14	h323-setup-time	
15	h323-connect-time	
16	h323-disconnect-time	
17	h323-disconnect-cause	
18	Acme-Egress-Network-Interface-Id	139
19	Acme-Egress-Vlan-Tag-Value	140
20	Acme-Ingress-Network-Interface-Id	137
21	Acme-Ingress-Vlan-Tag-Value	138
22	Acme-Session-Egress-Realm	42
23	Acme-Session-Ingress-Realm	41
24	Acme-FlowId_FS1_F	1
25	Acme-FlowType_FS1_F	2
26	Acme-Flow-In-Realm_FS1_F	10
27	Acme-Flow-In-Src-Addr_FS1_F	11
28	Acme-Flow-In-Src-Port_FS1_F	12
29	Acme-Flow-In-Dst-Addr_FS1_F	13
30	Acme-Flow-In-Dst-Port_FS1_F	14
31	Acme-Flow-Out-Realm_FS1_F	20
32	Acme-Flow-Out-Src-Addr_FS1_F	21
33	Acme-Flow-Out-Src-Port_FS1_F	22
34	Acme-Flow-Out-Dst-Addr_FS1_F	23
35	Acme-Flow-Out-Dst-Port_FS1_F	24
36	Acme-Calling-RTCP-Packets-Lost_FS1	32
37	Acme-Calling-RTCP-Avg-Jitter_FS1	33
38	Acme-Calling-RTCP-Avg-Latency_FS1	34
39	Acme-Calling-RTCP-MaxJitter_FS1	35
40	Acme-Calling-RTCP-MaxLatency_FS1	36
41	Acme-Calling-RTP-Packets-Lost_FS1	37
42	Acme-Calling-RTP-Avg-Jitter_FS1	38
43	Acme-Calling-RTP-MaxJitter_FS1	39

CSV Placement	Attribute Name	ACME VSA ID
44	Acme-Calling-Octets_FS1	28
45	Acme-Calling-Packets_FS1	29
46	Acme-Calling-R-Factor	151
47	Acme-Calling-MOS	152
48	Acme-FlowID_FS1_R	78
49	Acme-FlowType_FS1_R	79
50	Acme-Flow-In-Realm_FS1_R	80
51	Acme-Flow-In-Src-Addr_FS1_R	81
52	Acme-Flow-In-Src-Port_FS1_R	82
53	Acme-Flow-In-Dst-Addr_FS1_R	83
54	Acme-Flow-In-Dst-Port_FS1_R	84
55	Acme-Flow-Out-Realm_FS1_R	85
56	Acme-Flow-Out-Src-Addr_FS1_R	86
57	Acme-Flow-Out-Src-Port_FS1_R	87
58	Acme-Flow-Out-Dst-Addr_FS1_R	88
59	Acme-Flow-Out-Dst-Port_FS1_R	89
60	Acme-Called-RTCP-Packets-Lost_FS1	46
61	Acme-Called-RTCP-Avg-Jitter_FS1	47
62	Acme-Called-RTCP-Avg-Latency_FS1	48
63	Acme-Called-RTCP-MaxJitter_FS1	49
64	Acme-Called-RTCP-MaxLatency_FS1	50
65	Acme-Called-RTP-Packets-Lost_FS1	51
66	Acme-Called-RTP-Avg-Jitter_FS1	52
67	Acme-Called-RTP-MaxJitter_FS1	53
68	Acme-Called-Octets_FS1	44
69	Acme-Called-Packets_FS1	45
70	Acme-Called-R-Factor	153
71	Acme-Called-MOS	154
72	Acme-FlowID_FS2_F	90
73	Acme-FlowType_FS2_F	91
74	Acme-Flow-In-Realm_FS2_F	92
75	Acme-Flow-In-Src-Addr_FS2_F	93
76	Acme-Flow-In-Src-Port_FS2_F	94
77	Acme-Flow-In-Dst-Addr_FS2_F	95

CSV Placement	Attribute Name	ACME VSA ID
78	Acme-Flow-In-Dst-Port_FS2_F	96
79	Acme-Flow-Out-Realm_FS2_F	97
80	Acme-Flow-Out-Src-Addr_FS2_F	98
81	Acme-Flow-Out-Src-Port_FS2_F	99
82	Acme-Flow-Out-Dst-Addr_FS2_F	100
83	Acme-Flow-Out-Dst-Port_FS2_F	101
84	Acme-Calling-RTCP-Packets-Lost_FS2	104
85	Acme-Calling-RTCP-Avg-Jitter_FS2	105
86	Acme-Calling-RTCP-Avg-Latency_FS2	106
87	Acme-Calling-RTCP-MaxJitter_FS2	107
88	Acme-Calling-RTCP-MaxLatency_FS2	108
89	Acme-Calling-RTP-Packets-Lost_FS2	109
90	Acme-Calling-RTP-Avg-Jitter_FS2	110
91	Acme-Calling-RTP-MaxJitter_FS2	111
92	Acme-Calling-Octets_FS2	102
93	Acme-Calling-Packets_FS2	103
94	Acme-FlowID_FS2_R	112
95	Acme-FlowType_FS2_R	113
96	Acme-Flow-In-Realm_FS2_R	114
97	Acme-Flow-In-Src-Addr_FS2_R	115
98	Acme-Flow-In-Src-Port_FS2_R	116
99	Acme-Flow-In-Dst-Addr_FS2_R	117
100	Acme-Flow-In-Dst-Port_FS2_R	118
101	Acme-Flow-Out-Realm_FS2_R	119
102	Acme-Flow-Out-Src-Addr_FS2_R	120
103	Acme-Flow-Out-Src-Port_FS2_R	121
104	Acme-Flow-Out-Dst-Addr_FS2_R	122
105	Acme-Flow-Out-Dst-Port_FS2_R	123
106	Acme-Called-RTCP-Packets-Lost_FS2	126
107	Acme-Called–RTCP-Avg-Jitter_FS2	127
108	Acme-Called–RTCP-Avg-Latency_FS2	128
109	Acme-Called–RTCP-MaxJitter_FS2	129
110	Acme-Called-RTCP-MaxLatency_FS2	130
111	Acme-Called-RTP-Packets-Lost_FS2	131

CSV Placement	Attribute Name	ACME VSA ID
112	Acme-Called-RTP-Avg-Jitter_FS2	132
113	Acme-Called-RTP-MaxJitter_FS2	133
114	Acme-Called-Octets_FS2	124
115	Acme-Called-Packets_FS2	125
116	Acme-Session-Charging-Vector	54
117	Acme-Session-Charging-Function-Address	55
118	Acme-Firmware-Version	56
119	Acme-Local-Time-Zone	57
120	Acme-Post-Dial-Delay	58
121	Acme-Primary-Routing-Number	64
122	Acme-Originating-Trunk-Group	65
123	Acme-Terminating-Trunk-Group	66
124	Acme-Originating-Trunk-Context	67
125	Acme-Terminating-Trunk-Context	68
126	Acme-P-Asserted-ID	69
127	Acme-Ingress-Local-Addr	74
128	Acme-Ingress-Remote-Addr	75
129	Acme-Egress-Local-Addr	76
130	Acme-Egress-Remote-Addr	77
131	Acme-SIP-Diversion	70
132	Acme-Session-Disposition	60
133	Acme-Disconnect-Initiator	61
134	Acme-Disconnect-Cause	62
135	Acme-SIP-Status	71
136	Acme-Egress-Final-Routing-Number	134
137	Acme-Session-Ingress-RPH	135
138	Acme-Session-Egress-RPH	136
139	Acme-Refer-Call-Transfer-Id	141
140	Acme-Custom-VSA-200	200
141	Acme-Custom-VSA-201	201
142	Acme-Custom-VSA-202	202
143	Acme-Custom-VSA-203	203
144	Acme-Custom-VSA-204	204
145	Acme-Custom-VSA-205	205

CSV Placement	Attribute Name	ACME VSA ID
146	Acme-Custom-VSA-206	206
147	Acme-Custom-VSA-207	207
148	Acme-Custom-VSA-208	208
149	Acme-Custom-VSA-209	209
150	Acme-Custom-VSA-210	210
151	Acme-Custom-VSA-211	211
152	Acme-Custom-VSA-212	212
153	Acme-Custom-VSA-213	213
154	Acme-Custom-VSA-214	214
155	Acme-Custom-VSA-215	215
156	Acme-Custom-VSA-216	216
157	Acme-Custom-VSA-217	217
158	Acme-Custom-VSA-218	218
159	Acme-Custom-VSA-219	219
160	Acme-Custom-VSA-220	220
161	Acme-Custom-VSA-221	221
162	Acme-Custom-VSA-222	222
163	Acme-Custom-VSA-223	223
164	Acme-Custom-VSA-224	224
165	Acme-Custom-VSA-225	225
166	Acme-Custom-VSA-226	226
167	Acme-Custom-VSA-227	227
168	Acme-Custom-VSA-228	228
169	Acme-Custom-VSA-229	229
170	Acme-Custom-VSA-230	230
171	Acme-Flow-Calling-Media-Stop-Time_FS1	231
172	Acme-Flow-Called-Media-Stop-Time_FS1	232
173	Acme-Flow-Calling-Media-Stop-Time_FS2	233
174	Acme-Flow-Called-Media-Stop-Time_FS2	234
175	Acme-FlowMediaType_FS1_F	142
176	Acme-FlowMediaType_FS1_R	143
177	Acme-FlowMediaType_FS2_F	144

CSV Placement	Attribute Name	ACME VSA ID
178	Acme-FlowMediaType_FS2_R	145
179	Acme-SIP-Method-Type	235
180	Acme-Domain-Name	236
181	Acme-SIP-Contact	237
182	Acme-Reason-Phrase	239
183	Acme-CDR-Sequence-Number	59

6 Reference and Debugging

ACLI Configuration Elements

The following sections describe the Net-Net USM's unique configuration elements.

sip-registrar

Parameters

`name`—Configured name of this sip registrar.

Default: empty

`state`—Running status of this policy-director-group.

Default: enabled

Values: enabled | disabled

`domains`—List of registration domains that this Net-Net USM is responsible for. * means all domains. These domains are compared for an exact match with the domain in the request-uri of the REGISTER message. the wildcard '*' can also be entered as part of this parameter. This is entered as the domains separated by a space in quotes. No quotes required if only one domain is being configured. "+" and "-"are used to add to subtract from the list.

Default: empty

`subscriber-database-method`—Protocol used to connect to User Subscriber Database server.

Default: CX

Values: CX | DDNS | local

`subscriber-database-config`—The configuration element that defines the server used for retrieving user subscriber data. For Cx deployments it is a home-subscriber-server name. For ENUM deployments it is an enum-config name.

Default: empty

`authentication-profile`—Name of the sip-authentication-profile configuration used to retrieve authentication data when an endpoint is not authenticated.

Default: empty

`home-server-route`—The value inserted into the Server Name AVP in an MAR message. This should be entered as a SIP URI as per 3gpp TS 24229 & RFC 3261. The host can be FQDN or IPv4 address, and the port portion should be in the 1025 - 65535 range. Examples: `SIP:12.12.12.12:5060`

Default: empty

`third-party-registrars`—The third-party-regs configuration element names where third party REGISTER messages will be forwarded to.

Default: empty

routing-precedence—Indicates whether INVITE routing lookup should use the user database (via the registrar configuration element) or perform local policy lookup immediately.

Default: registrar

Values: registrar | local-policy

egress-realm-id—Indicates the default egress/core realm for SIP messaging.

Default: empty

location-update-interval—Sets the maximum period in minutes in which the core-side user subscriber database is refreshed, per user.

Default: 1440

Values: 0-999999999

ifc-profile—References the ifc-profile configuration element's name that is applied to this sip-registrar.

Path

This sip-registrar configuration element is a element in the session-router path. The full path from the topmost ACLI prompt is: configure terminal > session-router > sip-registrar.

sip-authentication-profile

Parameters

name—Configured name of this sip-authentication profile.

methods—List of SIP methods that prompt authentication. This is entered as the methods separated by a space in quotes. No quotes required if only one method is being configured. "+" and "-"are used to add to subtract from the list.

Default: empty

anonymous-methods—List of SIP methods that prompt authentication when received from anonymous sources. This is entered as the methods separated by a space in quotes. No quotes required if only one method is being configured. "+" and "-" are used to add or subtract from the list.

Default: empty

digest-realm—The value inserted into the digest-realm parameter in an authentication challenge header as sent to UA. (not used for Cx deployments)

Default: empty

credential-retrieval-method—Protocol used to connect to the server providing authentication data.

Default: ENUM-TXT

Values: ENUM-TXT | CX

credential-retrieval-config—The home-subscriber-server name used for retrieving authentication data.

Default: empty

Path

This sip-authentication-profile configuration element is a element in the session-router path. The full path from the topmost ACLI prompt is: configure terminal > session-router > sip-authentication-profile.

home-subscriber-server

Parameters

`name`—Configured name of this home subscriber server.

Default: empty

`state`—Running status of this home subscriber server.

Default: enabled

Values: enabled | disabled

`address`—This home subscriber server's IP address.

Default: none

Values: IP address in dotted decimal notation

`port`—This home subscriber server's port.

Default: 80

Values: 1-65535

`realm`—Net-Net SMX realm-config name where this home subscriber server exists.

Default: none

`origin-host-identifier`—Used to create segement before the dot in the Origin Host AVP.

Default: none

`origin-realm`—Populates the value of the Origin Realm AVP. Populates the segment after the dot in the Origin Host AVP.

Default: none

`watchdog-ka-timer`—

Default: 0

`aor-for-puid`——Sets the Net-Net USM to use the message's AoR as the Private User Identity value.

Default: disabled

Values: enabled | disabled

`add-lookup-parameter`—Sets the Net-Net USM to insert a P-Acme-Serving parameter into a Route: header

Default: disabled

Values: enabled | disabled

Path

This home-subscriber-server configuration element is a element in the session-router path. The full path from the topmost ACLI prompt is: configure terminal > session-router > home-subscriber-server.

third-party-regs

Parameters

`state`—Running status of this third party registration configuration element.

Default: enabled

Values: enabled | disabled

`name`—Configured name of this third party registration configuration element.

Default: none

`registrar-host`—hostname of the configured session agent that will be third party server. This value is also used in the request-uri that is sent to the third party server.

Default: none

`from-user`—The user part of the From URI in the REGISTER Request that is sent to the third party server in the REGISTER message. When this parameter is blank the user part of the From header from the incoming REGISTER Request will be used.

Default: none

`from-host`—The host part of the From URI in the REGISTER Request that is sent the third party server in the REGISTER message. When this parameter is blank the Net-Net USM uses the egress hostname/ IP address as the host.

Default: none

Values: Format this the same as the "registrar-host" in sip-config.

`retry-interval`—number of seconds the Net-Net SMX waits before retrying a 3rd Party Registration server after a failed registration.

Default: 32

Values: 0 - 3600

Path

This third-party-regs configuration element is a element in the session-router path. The full path from the topmost ACLI prompt is: configure terminal > session-router > third-party-regs.

local-subscriber-table

Parameters

`name`—A given name for this local subscriber table element. This name is referenced from the sip-registrar configuration element when the **credential-retrieval-method** is set to **local**.

`filename`—The filename of local subscriber table that this element references. If no path is provided, the default location is `/code/lst`.

`secret`—PSK used for encrypted passwords. This value is not echoed back to the screen upon viewing the configuration element.

Path

The location of this configuration element is: configure terminatl > session-router > local-subscriber-table.

enum-config

Parameters

`name`—Name for this enum-config to be referenced from within the system.

`top-level-domain`—The domain extension used to query the ENUM servers for this configuration.

`realm-id`—The realm-id is used to determine on which network interface to issue an ENUM query.

`enum-servers`—List of IP address that service the top level domain.

`service-type`—The ENUM service types you want supported in this ENUM configuration. Possible entries are E2U+sip and sip+E2U (the default), and the types outlines in RFCs 2916 and 3721.

Default: E2U+sip,sip+E2U

`query-method`—the ENUM query distribution strategy

Values: hunt | round-robin

Default: hunt

`timeout`—The total time, in seconds, that should elapse before a query sent to a server (and its retransmissions) will timeout.

Default: 11

`cacheInactivityTimer`—Enter the time interval, in seconds, after which you want cache entries created by ENUM requests deleted, if inactive for this interval.

Default: 3600

Values: 0-999999999

`max-response-size`—The maximum size in bytes for UDP datagram responses

Default: 512

`health-query-number`—The phone number for the ENUM server health query; when this parameter is blank the feature is disabled.

`health-query-interval`—The interval in seconds at which you want to query ENUM server health.

Values: 0-65535

Default: 0

`failover-to`—Name of the enum-config to which you want to failover.

`cache-addl-records`—Set this parameter to **enabled** to add additional records received in an ENUM query to the local DNS cache.

Default: enabled

Values: enabled | disabled

`include-source-info`—Set this parameter to enabled to send source URI information to the ENUM server with any ENUM queries.

Default: disabled

Values: enabled | disabled

`ttl`—This value ets the TTL value (in seconds) for NAPTR entries in the local ENUM cache and populates when sending a NAPTR entry to the ENUM server.

Default: 0

Values: 1-2592000

`order`—This parameter value populates the order field with when sending NAPTR entries to the ENUM server.

Default: 1

Values: 0-65535

`preference`—This parameter value populates the preference field with when sending NAPTR entries to the ENUM server.

Default: 1

Values: 0-65535

Path

This enum-config configuration element is a element in the session-router path. The full path from the topmost ACLI prompt is: configure terminal > session-router > enum-config.

ifc-profile

Parameters

`name`—A given name for this ifc profile element. This name is referenced from the sip-registrar configuration element's **ifc-support** parameter.

`state`—Running status of this ifc-profile.

Default: enabled

Values: enabled | disabled

Path

The location of this configuration element is: configure terminatl > session-router > ifc-profile.

SNMP MIBs and Traps

The following MIBs and traps are supported for the Net-Net SMX. Please consult the *Net-Net 4000 S-CX6.3.0 MIB Reference Guide* for more SNMP information.

Acme Packet License MIB (ap-license.mib)

The following table describes the SNMP GET query names for the Acme Packet License MIB (ap-license.mib).

SNMP GET Query Name	Object Identifier Name: Number	Description
Object Identifier Name: apLicenseEntry (1.3.6.1.4.1.9148.3.5.1.1.1)		
apLicenseAuthFeature	apLicenseEntry: 1.3.6.1.4.1.9148.3.5.1.1.1.20	If authorization and authentication is allowed for the Net-Net SMX, the value is true. If disabled, the value is false.
apLicenseDatabaseRegFeature	apLicenseEntry: 1.3.6.1.4.1.9148.3.5.1.1.1.21	If the Net-Net SMX is configured as a registrar, the value is true. If registrar functionality is not enabled, this value is false.
apLicenseDatabaseRegCap	apLicenseEntry: 1.3.6.1.4.1.9148.3.5.1.1.1.22	The database registration contact capacity.

Acme Packet System Management MIB (ap-smgmt.mib)

The following table describes the SNMP GET query names for the Acme Packet System Management MIB (ap-smgmt.mib).

SNMP GET Query Name	Object Identifier Name: Number	Description
Object Identifier Name: apSysMgmtMIBObjects (1.3.6.1.4.1.9148.3.2.1)		
Object Identifier Name: apSysMgmtGeneralObjects (1.3.6.1.4.1.9148.3.2.1.1)		
apSysSipStatsActiveDatabaseContacts	apSysMgmtGeneralObjects: 1.3.6.1.4.1.9148.3.2.1.1.24.0	Number of database-type contacts in the registration cache.

Enterprise Traps

The following table identifies the proprietary traps that Net-Net SMX system supports.

Trap Name: OID	Description
apSysMgmtDatabaseRegCacheCapTrap: 1.3.6.1.4.1.9148.3.2.6.0.76	Generated when the number of database-type contacts stored in the registration cache exceeds the license threshold.
apSysMgmtDatabaseRegCacheCapClearTrap: 1.3.6.1.4.1.9148.3.2.6.0.77	Trap is generated when the number of database-type contacts stored in the registration cache falls below the license threshold.

Net-Net USM Show Commands

show sipd endpoint-ip

The show `sipd endpoint-ip <user | IP address>` command displays information about each endpoint. For a supplied AoR, the Net-Net USM displays all

associated contacts (both access and core side), the expiration of each contact entry and associated 3rd Party Registration information. For example:

```
ACMEPACKET# show sipd endpoint-ip 11111
User <sip:111111@172.16.17.100>
  Contact exp=1198
    UA-Contact: <sip:111111@172.16.17.100:5060> UDP keep-acl
         realm=net172 local=172.16.101.13:5060 UA=172.16.17.100:5060
    SD-Contact: <sip:111111-s37q249kvluaa@192.168.101.13:5060> realm=net192
    Call-ID: 1-15822@172.16.17.100'
Third Party Registration:
   Third Party Reg User=<sip:111111@172.16.17.100> state: REGISTERED
   Expire Secs=298 seqNum= 1 refreshInterval=300
   Call-ID: d355a67277d9158e7901e46a12719663@192.168.101.13

   Third Party Reg User=<sip:111111@172.16.17.100> state: REGISTERED
   Expire Secs=178 seqNum= 1 refreshInterval=180
   Call-ID: 07ebbdebfdf64a48985bb82fa8b4c595@192.168.101.13
```

show sipd third-party

The show sipd third-party command displays the current status of third party servers and statistics for messages. The format is:

```
show sipd third-party <all | name>
```

The name argument allows status to be displayed for just the server specified by the name. Not specifying a name results in status being displayed for all third party servers. For example:

```
ACMEPACKET# show sipd third-party-reg all
3rd Party Registrar            SA State  Requests     200OK  Timeouts    Errors
192.168.17.101                     INSV         9         9         0         0
192.168.17.102                     INSV        14        14         0         0
```

Column definitions are as follows:

- IP Address—IP Address of third party server
- Status—Session Agent State
- Requests—Register requests sent
- 200 OK—200 OK Responses received
- Timeouts—Requests timed out
- Error—Error Responses

show registration sipd

The show registration sipd by-user <username> detail reflects user registration information including downloaded IFCs. For example:

```
ACMEPACKET# show registration sipd by-user +19999092907 d

Registration Cache (Detailed View)    MON JUN 25 2012  13:47:46

User: sip:+19999092907@mobile.com
  Registered at: 2012-06-25-13:43:50     Surrogate User: false
```

```
  Contact Information:
    Contact:
      Name: sip:+19999092907@mobile.com
      Valid: true
      Challenged: false
      Registered at: 2012-06-25-13:43:50
      Last Registered at: 2012-06-25-13:47:30
      Expire: 48
      Local expire: 13

      Registrar IP: 0.0.0.0
      Transport: UDP
      Secure: false
      Local IP: 155.212.214.175:5060

      User Agent Info:
        Contact:
sip:+19999092907@50.76.51.62:5762;transport=udp;acme_nat=+19999092907
+50.76.51.62@10.1.10.20:5762
        Realm: access
        IP: 50.76.51.62:5762

      SD Info:
        Contact: sip:+19999092907-rb8tulsbv3u72@108.108.108.108:5060
        Realm: core

      Call-ID: H_yvkgTAAA@10.1.10.20

    Associated URI(s):
      URI: sip:+19999092907@mobile.com

  Filter Criteria:
    Priority: 0
     Filter: ((case == 'Originating Registered') and (method == INVITE)
and ('Accept-Contact'=='+g.app2app')) or
            ((case == 'Originating Registered') and (method == INVITE)
and ('Contact'=='+g.app2app')) or
            ((case == 'Originating Registered') and (method == INVITE)
and ('P-Message-Auth'=='.*')) or
            ((case == 'Originating Registered') and (method == INVITE)
and ('P-Application-ID'=='.*'))
      Application Server: sip:pza.mobile.com:5280
```

Supporting Configuration

The following configuration elements which are not mentioned in this guide are required for the Net-Net USM to function. Please refer to the Net-Net 4000 ACLI Configuration Guide for details about configuring all supporting elements.

- network-interface
- physical-interface
- realm-config

- sip-config
- system-config

The following configuration elements are mentioned in this guide briefly and still require configuration:

- local-policy
- session-agent
- sip-interface

Session Load Balancer Support

In order to rapidly increase the number of supported endpoints, the Net-Net USM can interoperate with the Net-Net SLB. When paired with a Net-Net SLB, the Net-Net USM maintains its ability to function with an HSS or ENUM database.

In addition, the Net-Net USM and Net-Net SLB pair supports Cx or ENUM based registrations.

To communicate with a Net-Net SLB, in addition to all baseline Net-Net SBC SIP functionality, the Net-Net USM advertises its registration capacity to the Net-Net SLB. This value is defined in the SIP interface as the reg cache limit parameter. Since the SMX has a database registrar license, the lower of the two will be advertised to the SLB.

Verify Config

The Net-Net USM performs application specific verification checks when you save a config with the save-config ACLI command. These checks are in addition to baseline Net-Net SBC verification checks.

sip authentication profile (CX)

If “session-router > sip-authentication-profile > credential-retrieval-method” = CX

then confirm

“session-router > sip-authentication-profile > credential-retrieval-config” value =

any existing “session-router > home-subscriber-server configuration > name” value

Error

If the above check fails:

1. A WARNING is displayed on the ACLI.
2. An INFO log message is generated.

sip authentication profile (ENUM)

If “session-router > sip-authentication-profile > credential-retrieval-method” = ENUM-TXT then confirm

“session-router > sip-authentication-profile > credential-retrieval-config” value =

any existing “session-router > enum-config > name” value

Error

If the above check fails:

1. A WARNING is displayed on the ACLI.

2. An INFO log message is generated.

sip authentication profile (Local)

If "session-router > sip-authentication-profile > credential-retrieval-method" = local then confirm

"session-router > sip-authentication-profile > credential-retrieval-config" =

"session-router > local-subscriber-table > ame" Error

If the above check fails:

1. A WARNING is displayed on the ACLI.
2. An INFO log message is generated.

sip-registrar

If "session-router > sip-registrar > subscriber-database-method" = DDNS

then confirm

"session-router > sip-registrar > subscriber-database-config" value =

any existing "session-router > enum-config > name" value

Error

If the above check fails:

1. A WARNING is displayed on the ACLI.
2. An INFO log message is generated.

sip-registrar

If "session-router > sip-registrar > authentication-profile" is configured, then confirm its value is any existing:

"session-router > sip-authentication-profile > name" value

Error

If the above check fails:

1. A WARNING is displayed on the ACLI.
2. An INFO log message is generated.