

CSCF Network Tracing

Call Session Control Function

USER GUIDE

Copyright

© Ericsson AB 2016–2019. All rights reserved. No part of this document may be reproduced in any form without the written permission of the copyright owner.

Disclaimer

The contents of this document are subject to revision without notice due to continued progress in methodology, design and manufacturing. Ericsson shall have no liability for any error or damage of any kind resulting from the use of this document.

Trademark List

All trademarks mentioned herein are the property of their respective owners. These are shown in the document Trademark Information.



Contents

1	Network Tracing Overview	1
2	CSCF Network Tracing Tools Overview	3
2.1	NetTrace	3
2.2	AppTrace and AppLog	10
2.3	NetTraceCollector	11
3	Procedure	11
3.1	Handle a NetTrace Session	11
3.2	Start a NetTrace Session	12
3.3	Handle Machine-Readable Traces	14
3.4	Handle Human-Readable Traces	15
4	File Management	16





1 Network Tracing Overview

The principle of NetTrace is to allow a user the possibility to log SIP and Cx transactions traversing the Call Session Control Function (CSCF) for fault finding and localization purposes.

Using the AppTrace, a trace session can be configured to trace SIP and Cx transactions based on the filtering of the Originating Public User IDs and Terminating Public User IDs for both Min and Max levels.

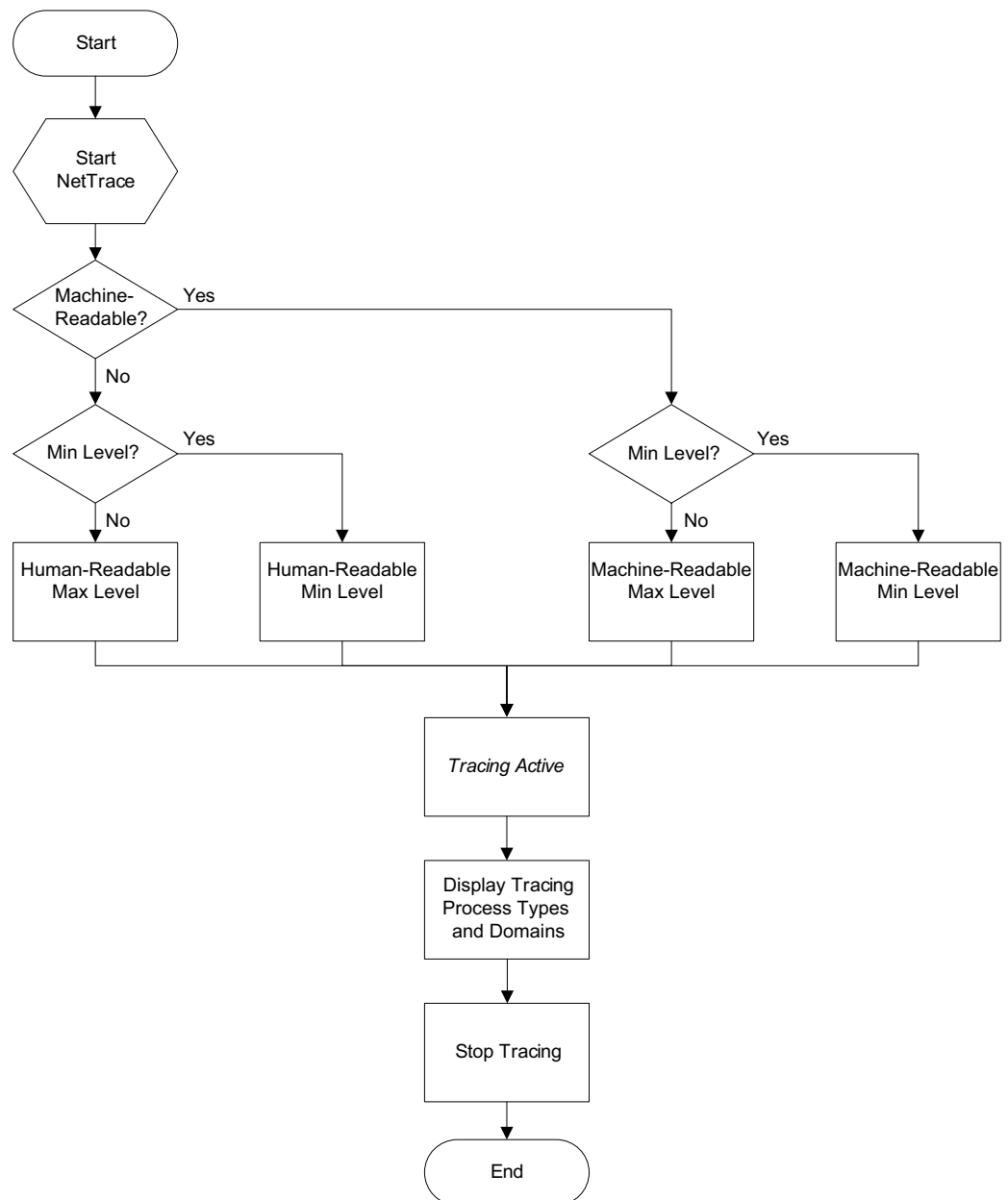


Figure 1 Simplified Flow Configuration and Use of NetTrace



2 CSCF Network Tracing Tools Overview

2.1 NetTrace

Note: The inherent problem with observing the behavior of a system by tracing is the consumed capacity of the tracing itself. If the cost is too high, it can interfere with the primary function of the system, at worst even causing system failure.

NetTrace is a tool that allows the user to trace transactions that traverse the CSCF depending on user-defined filter criteria. These transactions are formatted and output in standardized XML[®] file format (in this document referred to as “machine readable”) according to the 3GPP[®] specification [Telecommunication management; Subscriber and equipment trace; Trace data definition and management](#).

Alternatively, traces can be read directly from the AppLog and AppTrace files (referred to as “human-readable”). Human-readable format of traces is proprietary and not specified by the TS 32.423.

It is possible to trace at two levels; Min (minimum) and Max (maximum) for both machine-readable and human-readable output formats.

When NetTrace is active, tracing is performed on all CSCF-implemented SIP interfaces and Standard Cx/Dx interfaces, except Push-Profile-Request (PPR)/Push-Profile-Answer (PPA) and Registration Termination Request (RTR)/Registration Termination Answer (RTA) when the Public User Identity is not included in the request message.

If CSCF nodes are collocated, such as for Interrogating and Serving CSCF (IS-CSCF), signaling within a physical node is not traced unless there is a transition between the originating CSCF and the terminating CSCF or the other way around.

Descriptions for ieGroup name and ie name can be found in the TS 32.423.

The presence of an information element in the following tables is defined by the P (presence) column as follows:

- M = Mandatory. The element is always present.
- O = Optional. The element can be present.

The terms tracedPublicId1 and tracedPublicId2 used in the following tables see the Public User IDs that triggered the trace. The element tracedPublicId1 is always present as one Public User ID must have triggered the trace.

If the trace is triggered by more than one Public User ID, its output is as tracedPublicId2. For example, if a Public User ID is specified as an OrigPublicId, a different Public User ID is specified as a TermPublicId, and a session is set up between the users, both Public User IDs trigger in the same trace.



Every time a user starts a NetTrace session, the identity of the user and the executed command are logged in the Linux Syslog.

2.1.1 Machine-Readable SIP Output at Min Level

At Min Trace Level, the SIP transactions are represented by several XML® tags. Limited information is output when tracing at this level.

Table 1 Machine-Readable SIP Transaction Output Request Data at Min Level

Request			
ieGroup Name	ie Name	Presence	Comment
tracedPublicIds ⁽¹⁾	tracedPublicId1	M	A Public User ID that triggered tracing of this Request.
tracedPublicIds ⁽¹⁾	tracedPublicId2	O	A Public User ID that triggered tracing of this Request.
-	Request-Line	M	SIP Request line
Message Headers	To	M	SIP To header
Message Headers	From	M	SIP From header
Message Headers	Call-ID	M	SIP Call-ID header
Message Headers	CSeq	M	SIP CSeq header

(1) ieGroup name `tracedPublicIds` is only output once, when the trace session is started.

Table 2 Machine-Readable SIP Transaction Output Response Data at Min Level

Response			
ieGroup Name	ie Name	Presence	Comment
-	Status-Line	M	SIP Status line
Message Headers	To	M	SIP To header
Message Headers	From	M	SIP From header
Message Headers	Call-ID	M	SIP Call-ID header
Message Headers	CSeq	M	SIP CSeq header



2.1.2 Machine-Readable Cx Output at Min Level

At Min Trace Level, the Cx transactions are represented by several XML® tags. Limited information is output when tracing at this level. Post-processing is required on the generated XML files to obtain meaningful trace data.

Table 3 Machine-Readable Cx Transactions Output Command Data at Min Level

Command			
ieGroup Name	ie Name	Presence	Comment
tracedPublicIds ⁽¹⁾	tracedPublicId1	M	A Public User ID that triggered tracing of this Command.

(1) ieGroup name tracedPublicIds is only output once, when the Trace Session is started.

Table 4 Machine-Readable Cx Transactions Output Response Data at Min Level

Response			
ieGroup Name	ie Name	Presence	Comment
-	Result-Code ⁽¹⁾	O	Diameter Result-Code
-	Experimental-Result-Code ⁽¹⁾	O	Diameter Experimental-Result-Code

(1) Either Result-Code or Experimental-Result-Code is present.

2.1.3 Human-Readable SIP Output at Min Level

At Min Trace Level, the SIP transactions are represented in plain-text form within the AppTrace. The Message header and applicable parameters are included in output on individual lines. Limited information is included in output when tracing at this level.

Table 5 Human-Readable SIP Transactions Output Request Data at Min Level

Request		
Message	Presence	Comment
traceSessionRef ⁽¹⁾	M	Indicates the forloop specified used by the operator.
origPublicId ⁽¹⁾	M	The Originating Public User ID derived from this Request.



Request		
Message	Presence	Comment
termPublicId ⁽¹⁾	M	The Terminating Public User ID derived from this Request.
tracedPublicId1 ⁽¹⁾	M	A Public User ID that triggered tracing of this Request.
tracedPublicId2 ⁽¹⁾	O	A Public User ID that triggered tracing of this Request.
Initiator	M	IP/Port/Transport that initiated this Request.
Target	M	IP/Port/Transport that is the intended recipient of this Request.
Request-Line	M	SIP Request line
To	M	SIP To header
From	M	SIP From header
Call-ID	M	SIP Call-ID header
CSeq	M	SIP CSeq header

(1) These fields are only output once, when the trace session is started.

Table 6 Human-Readable SIP Transactions Output Response Data at Min Level

Response		
Message	Presence	Comment
Initiator	M	IP/Port/Transport that initiated this Request.
Target	M	IP/Port/Transport that is the intended recipient of this Request.
Status-Line	M	SIP Status line
To	M	SIP To header
From	M	SIP From header
Call-ID	M	P Call-ID header
CSeq	M	SIP CSeq header



2.1.4 Human-Readable Cx Output at Min Level

At Min Trace Level, the Cx transactions are represented in plain-text form within the AppLog. The Command and Response, and the Result-Code or Experimental-Result-Code Attribute-Value Pair (AVP) are included in output on individual lines. Limited information is included in output when tracing at this level.

Table 7 Human-Readable Cx Transactions Output Command Data at Min Level

Command		
Message	Presence	Comment
Command-Name	M	Diameter Command Name

Table 8 Human-Readable Cx Transactions Output Response Data at Min Level

Response		
Msg	Presence	Comment
Response-Name	M	Diameter Response Name
Result-Code ⁽¹⁾	O	Diameter Result-Code
Experimental-Result-Code ⁽¹⁾	O	Diameter Experimental-Result-Code

(1) Either Result-Code or Experimental-Result-Code is present.

2.1.5 Machine-Readable SIP and Cx Output at Max Level

At Max Trace Level, the SIP and Cx transactions are encoded into hexadecimal and output as raw data in XML[®] file format. In contrast to Min level, the complete contents of each request or command and response are output.

Post-processing is required on the generated XML files to obtain meaningful trace data.

Table 9 Mandatory Machine-Readable Data at Max Level

ieGroup Name	ie Name	Presence	Data Type	Comment
tracedPublicIds ⁽¹⁾	tracedPublicId 1	M	SIP Transaction Output Request	A Public User ID that triggered tracing of this Request.



ieGroup Name	ie Name	Presence	Data Type	Comment
-	Request-Line	M	SIP Transaction Output Request	SIP Request line
Message Headers	To	M	SIP Transaction Output Request	SIP To header
Message Headers	From	M	SIP Transaction Output Request	SIP From header
Message Headers	Call-ID	M	SIP Transaction Output Request	SIP Call-ID header
Message Headers	CSeq	M	SIP Transaction Output Request	SIP CSeq header
-	Status-Line	M	SIP Transaction Output Response	SIP Status line
Message Headers	To	M	SIP Transaction Output Response	SIP To header
Message Headers	From	M	SIP Transaction Output Response	SIP From header
Message Headers	Call-ID	M	SIP Transaction Output Response	SIP Call-ID header



ieGroup Name	ie Name	Presence	Data Type	Comment
Message Headers	CSeq	M	SIP Transaction Output Response	SIP CSeq header
tracedPublicIds ⁽¹⁾	tracedPublicId 1	M	Cx Transactions Output Command	A Public User ID that triggered tracing of this Command.

(1) ieGroup name tracedPublicIds is only output once, when the trace session is started.

2.1.6

Human-Readable SIP and Cx Output at Max Level

At Max Trace Level, the SIP and Cx transactions are represented in plain-text form within the AppLog. In contrast to Min level, the complete contents of each request or command and response are output.

Table 10 Mandatory Human-Readable Data at Max Level

Message	Presence	Data Type	Comment
traceSessionRef ⁽¹⁾	M	SIP Transactions Output Request	Indicates the forlop specified used by the operator.
origPublicId ⁽¹⁾	M	SIP Transactions Output Request	The Originating Public User ID derived from this Request.
termPublicId ⁽¹⁾	M	SIP Transactions Output Request	The Terminating Public User ID derived from this Request.
tracedPublicId1 ⁽¹⁾	M	SIP Transactions Output Request	A Public User ID that triggered tracing of this Request.
Initiator	M	SIP Transactions Output Request	IP/Port/Transport that initiated this Request.
Target	M	SIP Transactions Output Request	IP/Port/Transport that is the intended recipient of this Request.



Message	Presence	Data Type	Comment
Request-Line	M	SIP Transactions Output Request	SIP Request line
To	M	SIP Transactions Output Request	SIP To header
From	M	SIP Transactions Output Request	SIP From header
Call-ID	M	SIP Transactions Output Request	SIP Call-ID header
CSeq	M	SIP Transactions Output Request	SIP CSeq header
Initiator	M	SIP Transactions Output Response	IP/Port/Transport that initiated this Request.
Target	M	SIP Transactions Output Response	IP/Port/Transport that is the intended recipient of this Request.
Status-Line	M	SIP Transactions Output Response	SIP Status line
To	M	SIP Transactions Output Response	SIP To header
From	M	SIP Transactions Output Response	SIP From header
Call-ID	M	SIP Transactions Output Response	P Call-ID header
CSeq	M	SIP Transactions Output Response	SIP CSeq header
Command-Name	M	Cx Transactions Output Command	Diameter Command Name
Response-Name	M	Cx Transactions Output Response	Diameter Response Name

(1) These fields are only output once, when the trace session is started.

2.2 AppTrace and AppLog

vDicos AppTrace is used to realize the NetTrace. For a detailed description of AppTrace functionality, see [AppTrace User Guide](#).

To get machine-readable format or human-readable format, the AppTrace output must be processed with AppLog.



2.3 NetTraceCollector

The `NetTraceCollector` is a Perl-based script that collects trace data from the `AppLog` and outputs the data in XML® format.

3 Procedure

3.1 Handle a NetTrace Session

Prerequisites

— This user guide references the following documents:

- Handling Files
- IMS Common Components Troubleshooting Guide

— The following tools are required:

- NetTrace
- AppTrace
- AppLog
- NetTraceCollector

— The following conditions must apply:

- Certain troubleshooting activities can have an impact on node performance. For example, trace activation can be traffic disturbing and is not recommended without first consulting Ericsson. However, NetTrace can be activated for a few users and sessions without adversely affecting performance (up to 10 users is the recommended limit).
- The user is authorized to access the CSCF with maintenance or `systemtroubleshooter` privileges for the required actions.

Steps

1. Start a NetTrace session, see Section 3.2 Start a NetTrace Session on page 12.
2. Handle the trace output:
 - For machine-readable traces, see Section 3.3 Handle Machine-Readable Traces on page 14.



- For human-readable traces, see Section 3.4 Handle Human-Readable Traces on page 15.

3.2 Start a NetTrace Session

Steps

1. Log on to the System Controller (SC) processor of the CSCF node:

```
> ssh -A <emergency_username>@<OAM-MIP>
```

2. Is the desired NetTrace output type Machine-Readable?

Yes: Continue with the next step.

No: Proceed with Step 4.

3. Start the NetTraceCollector script:

- a. Check if the NetTraceCollector script is already running by using, for example:

```
> ps -ef | grep nettracecollector
```

- b. If the NetTraceCollector script is not running, start the script:

```
> nettracecollector &
```

Note: For more information on how to ensure that the XML[®] files are output correctly, see [IMS Common Components Troubleshooting Guide](#).

4. Choose one of the following options to start NetTrace:

- Machine-Readable Output at Min Level: Proceed with Step 5.
- Machine-Readable Output at Max Level: Proceed with Step 6
- Human-Readable Output at Min Level: Proceed with Step 7
- Human-Readable Output at Max Level: Proceed with Step 8

5. Start NetTrace for machine-readable output at min level:

```
> CscfTrace -ro applog -user sip:user@domain --forlop  
sip:user@domain=<forlop> --net_min_sip_method <sip method(s)>  
CSCFv_NetTraceMinMachineReadable_Trace_Profile
```

Proceed with Step 9.

6. Start NetTrace for machine-readable output at max level:

```
> CscfTrace -ro applog -user sip:user@domain --forlop sip:user@  
domain=<forlop> CSCFv_NetTraceMaxMachineReadable_Trace_Profile
```




Proceed with Step 9.

7. Start NetTrace for human-readable output at min level:

```
> CscfTrace -ro applog -user sip:user@domain --forlop
sip:user@domain=<forlop> --net_min_sip_method <sip method(s)>
CSCFv_NetTraceMinHumanReadable_Trace_Profile
```

Proceed with Step 9.

8. Start NetTrace for human-readable output at max level:

```
> CscfTrace -ro applog -user sip:user@domain --forlop sip:user@
domain=<forlop> CSCFv_NetTraceMaxHumanReadable_Trace_Profile
```

Proceed with Step 9.

9. If needed, display the types and domains that are enabled for the current session by using the following command:

```
> CscfTrace display
```

10. If needed, stop an ongoing trace session by using the following command:

```
> CscfTrace stop
```

11. Log off from the System Controller:

```
exit
```

Note: It is possible to express multiple user IDs and multiple forlop values using command CscfTrace.

Table 11 CscfTrace Command Line Options and Parameters

CscfTrace Option	Description
-h --help	Displays the CscfTrace help page.
-ro <ROUTE_OUTPUT> --route=<ROUTE_OUTPUT>	Mandatory It specifies where the output is routed. The mandatory setting is applog .
-f <PUB_ID=forlop pairs> --forlop <PUB_ID=forlop pairs>	Public ID to forlop hash entries. If this option is not specified, forlop values are allocated.
-mc <MAX_CPU> -max_cpu=<MAX_CPU>	Maximum CPU Threshold Percentage. Valid range: 1–99. By default, it is 70%. If the CPU load in any Payload node exceeds this maximum CPU threshold %, the script stops all traces. This option is used only if the CPU load monitoring is present. skip_cpu_check(sc) and max_cpu(mc) are mutual exclusive options.
-sip <SIP_METHODS> --net_min_sip_method <SIP_METHODS>	If this option is not specified, all SIP methods are applied.



Table 11 CscfTrace Command Line Options and Parameters

CscfTrace Option	Description
-sc skip_cpu_check	This option is used when CPU load in Payload nodes is not to be monitored. By default, this option is disabled. skip_cpu_check(sc) and max_cpu(mc) or stop_timeout(st) are mutual exclusive options.
-st <STOP_TIMEOUT> --stop_timeout=<STOP_TIMEOUT>	Stop time-out. Valid range: 1–1440. By default, it is 20 minutes. While the script is still running, it stops all traces automatically after it has been running for this number of minutes. This option is used only if the CPU load monitoring is present. skip_cpu_check(sc) and stop_timeout(st) are mutual exclusive options.
-user <USER_PUB_IDS> --user_pub_id=<USER_PUB_IDS>	This option specifies the User Public IDs. Examples with a single User Public ID: --user_pub_id=sip:eric.almighty@ericsson25.lab --user_pub_id=sip:alice.almighty@ericsson25.lab --user_pub_id=tel:+49616000014 Example multiple User Public IDs: --user sip:eric.almighty@ericsson25.lab,sip:alice./=> almighty@ericsson25.lab,tel:+49616000014 This option is case-sensitive.
-v, verbose	This option is used to display the result texts of CLU successful commands.
<forlop>	This parameter is a Trace Identity and an integer value. The value is chosen by the operator. Normally, a unique forlop value is assigned to each Public User ID, but this is not mandatory. It is allowed to assign the same forlop value to all traced Public User IDs.
display	After starting CscfTrace, this option displays the types and domains that are enabled for the current session.
stop	This option stops the ongoing network tracing session.

3.3 Handle Machine-Readable Traces

NetTrace XML® files that are completed and ready for post-processing are on the CSCF node in the directory /storage/no-backup/cmco_utils-cxp9020686/nettrace/cscf and have this naming format: A<date in yyyyymmdd format>.<time in hhmm format>-CSCF.<nodeName>.<TraceReference>.<SessionRef>.xml.COMPLETE

Where

- <nodeName> is specified by the environment variable Node_Distinguished_Name. The default node name is ManagedElement1.
- <TraceReference> is the user-defined forlop ID.
- <SessionRef> is the system-defined Trace Session Recording Session Reference.

This is an example output filename:

A20110128.1609-CSCF.ManagedElement1.1111.56032.xml



After an XML file is completed and ready for post-processing, the suffix `.COMPLETE` is appended to the filename.

This is an example output filename of a file that is ready for post-processing:
`A20110128.1609-CSCF.ManagedElement1.1111.56032.xml.COMPLETE`

NetTrace XML files can be fetched using File Management. For more information, see Section 4 on page 16.

Steps

1. Fetch the NetTrace Machine-Readable XML output files through File Management.
2. Post-process the output files through a system that is compliant with the TS 32.423 standard.

3.4 Handle Human-Readable Traces

Human-readable trace outputs are located in the AppLog, `/storage/no-backup/coremw/var/log/saflog/AppTrace_*.log`.

Steps

1. Open a Trace output logfile.
2. Locate the Start message:

— For Min Level tracing:

The start message is a line with message `Msg: "traceSessionRef: <TraceReference>"` and trace domain `ims.cscf.netio.info`.

— For Max Level tracing:

The start messages are lines with message `Msg: "traceSessionRef: <TraceReference>"` and trace domains `ims.sip.netio.rx`, `ims.sip.netio.rx`, `ims.diaif.netio.rx`, and `ims.diaif.netio.rx`.

For example, for Min Level tracing:

```
...pid:581 forlop:237897 id:"ims.cscf.netio.info" \
MsgLength:"123" Msg:"traceSessionRef: 1234"
```

`<TraceReference>` is the user-defined forlop ID. In this example, the `TraceReference` is 1234.

3. Locate the next three lines that contain the `origPublicId`, `termPublicId`, and `tracedPublicIds` information.

For example:



```
...pid:581 id:"ims.cscf.netio.info" forlop:237897 \  
...MsgLength:"102" Msg:"origPublicId: sip:user_a@cscf.com"  
...pid:581 id:"ims.cscf.netio.info" forlop:237897 \  
...MsgLength:"69" Msg:"termPublicId: sip:user_a@cscf.com"  
...pid:581 id:"ims.cscf.netio.info" forlop:237897 \  
...MsgLength:"36" Msg:"tracedPublicId1: sip:user_a@cscf.com"
```

The forlop number represents a session that has been traced that is associated with the TraceReference number. In this example, forlop 237897 is a traced session that is associated with TraceReference 1234.

4. Locate any other line that contains the same forlop.

These lines represent each a header of a traced message or response from the same session.

For example:

```
...pid:581 id:"ims.cscf.netio.sip.register" forlop:237897 \  
...MsgLength:"130" Msg:"REGISTER sip:cscf.com"  
...pid:581 id:"ims.cscf.netio.sip.register" forlop:237897 \  
...MsgLength:"109" Msg:"To: <sip:user_a@cscf.com>"  
...pid:581 id:"ims.cscf.netio.sip.register" forlop:237897 \  
...MsgLength:"84" Msg:"From: <sip:user_a@cscf.com>"  
...pid:581 id:"ims.cscf.netio.sip.register" forlop:237897 \  
...MsgLength:"57" Msg:"Call-ID: 18be-147-67840"  
...pid:581 id:"ims.cscf.netio.sip.register" forlop:237897 \  
...MsgLength:"20" Msg:"CSeq: 12271 REGISTER"  
...pid:186 id:"ims.cscf.netio.dia.cx" forlop:237897 \  
...MsgLength:"85" Msg:" Name: User-Authorization-Answer"  
...pid:186 id:"ms.cscf.netio.dia.cx" forlop:237897 \  
...MsgLength:"62" Msg:" AVP: Result-Code 2001"
```

4 File Management

The CSCF Network Tracing XML® files are exposed by File Management in the following file group structure:

- FileGroup=Cscf
 - FileGroup=CscfNetTraceLogs

For more information on file groups, see [Handling Files](#).