

MTAS Network Tracing

MTAS

USER GUIDE

Copyright

© Ericsson AB 2016–2018. 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	Introduction	1
1.1	Prerequisites	1
2	Tools	3
2.1	NetTrace	3
2.2	AppTrace and AppLog	9
2.3	NetTraceCollector	9
2.4	Machine-readable Min Trace Level	9
2.5	Machine-readable Max Trace Level	10
3	NetTrace Function	19
3.1	NetTrace Procedure	20
3.2	Analysis of Trace Outputs	25





1 Introduction

This document describes the procedure that is to be used to obtain network traces from MTAS.

1.1 Prerequisites

This section describes the prerequisites for this document.

It is assumed that users of this document are familiar with performing operations within the area for O&M in general.

1.1.1 Documents

Before starting this procedure, ensure that the following documents are available:

- AppTrace User Guide
- IMS Common Components Troubleshooting Guide
- MTAS Troubleshooting Guideline

1.1.2 Conditions

Certain troubleshooting activities can have an impact on node performance. For example, trace activation can cause traffic disturbance 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).





2 Tools

This section describes the tools that can be used for network tracing in MTAS.

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 MTAS 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, [3GPP TS 32.423, 8.1.0 :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 [3GPP 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 active, tracing is performed on all MTAS implemented SIP interfaces.

MTAS uses the 3GPP standard XML format to visualize the content of the various SIP messages.

For example, SIP message header will be located in “name” attribute of “ie” XML element. The “ie” elements are grouped into the “ieGroup” elements.

Descriptions for “ieGroup name” and “ie name” can be found in the [3GPP 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 refer to 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 was triggered by more than one Public User Id, it is output as `tracedPublicId2`. One example for this is if a Public User Id were specified as an



OrigPublicId and a different Public User Id were specified as a TermPublicId, and a session was set up between the users. In this case, both Public User Ids are triggering in the same trace.

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.

For the standard XML elements included in the output, refer to the 3GPP specification [3GPP TS 32.423](#).

The MTAS implemented data output is listed in Table 1 and Table 3.

Table 1 Machine-readable SIP 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, with tracedPublicId1, 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 Standard SIP Output Request Data at Min Level

Attribute Name	Description
Standard attributes	Standard Trace Data Definition and Management 3GPP TS 32.423 attributes



Attribute Name	Description
fileHeader: fileFormatVersion	This attribute specification identifies the file format version applied by the sender. For example, “32.423 V8.1.0” vendorName=“Ericsson AB”
traceCollec: beginTime	This attribute specification contains a time stamp that refers to the start of the first trace data that is stored in this file. It is a complete time stamp including day, time, and delta UTC hour. For example, “2010-06-29T08:18:43+01:00”.
traceRecSession: traceSessionRef	Attribute specification that provides a unique trace session identifier as described in Trace concepts and requirements 3GPP TS 32.421 . A user-defined identity of the trace session.
traceRecSession: traceRecSessionRef	Attribute specification that provides a unique trace recording session identifier as described in Trace concepts and requirements 3GPP TS 32.421 , and Trace control and configuration management 3GPP TS 32.422 . This attribute is derived from hashing of the Call-ID.
msg function	Attribute specification that provides the function name associated to the traced message. For example, “SIP”.
msg name	Attribute specification that provides the function name associated with the traced message. For example, “INVITE”.
initiator	Optional element that identifies the Network Element (NE) initiator of the protocol message. For example, [address == 130.100.96.123, port == 50195, transport == Udp].
target	Optional element that identifies the NE target of the protocol message. For example, [address == 10.64.65.10, port == 5082, transport == Udp].
ie	Information elements specific to Ericsson IMS SIP requests.



Attribute Name	Description
tracedPublicId1	A Public User Id that triggered tracing of this request for served user. If tracing is triggered for the originating user only, then it contains the Public Id of originating user. If tracing is triggered for the terminating user only, then it contains the Public Id of the terminating user.
tracedPublicId2	A Public User Id that triggered tracing of this request for terminating user. tracedPublicId2 exists together with the tracedPublicId1 in case when both originating and terminating users involved in a session are traced. This attribute is optional and exists only when both originating and the terminating user in a session are traced.
Request-Line	SIP Request line
To	SIP To header
From	SIP From header
Call-ID	SIP Call-ID header
CSeq	SIP CSeq header

Table 3 Machine-readable SIP 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

Table 4 Machine-readable Standard SIP Output Response Data at Min Level

Attribute Name	Description
Standard attributes, see Table 2	Standard Trace Data Definition and Management 3GPP TS 32.423 attributes
ie	Information elements specific to Ericsson IMS SIP requests



Attribute Name	Description
PublicId1	A Public User Id that triggered tracing of this Response for served user. If tracing is triggered for the originating user only, then it contains the Public Id of originating user. If tracing is triggered for the terminating user only, then it contains the Public Id of the terminating user.
PublicId2	A Public User Id that triggered tracing of this Response for terminating user. tracedPublicId2 exists together with the tracedPublicId1 in case when both originating and terminating users involved in a session are traced. This attribute is optional and exists only when both originating and the terminating user in a session are traced.
Status-Line	SIP Status line
To	SIP To header
From	SIP From header
Call-ID	SIP Call-ID header
CSeq	SIP CSeq header

For more details on the Min level, see Example 1.

2.1.2 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 output on individual lines. Limited information is output when tracing at this level.

The MTAS implemented data output is listed in Table 5 and Table 6.

Table 5 Human-readable SIP Output Request Data at Min Level

Request		
Msg	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		
Msg	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 Output Response Data at Min Level

Response		
Msg	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



2.1.3 Machine-Readable SIP Output at Max Level

At Max Trace Level, the SIP 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, command, or response are output.

For the standard XML elements included in the output, refer to [3GPP TS 32.423, 8.1.0: Telecommunication management; Subscriber and equipment trace; Trace data definition and management](#).

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

For more details on the Max level, see Example 2.

2.1.4 Human-readable SIP Output at Max Level

At Max Trace Level, the SIP transactions are represented in plain-text form within the vDicos AppLog. In contrast to Min level, the complete contents of each request or command/response are output. For more information about vDicos Applog, refer to vDicos Management.

2.2 AppTrace and AppLog

vDicos AppTrace is used to realize the NetTrace.

For a detailed description of AppTrace functionality, refer to AppTrace User Guide.

For a detailed description of Applog, refer to MTAS Logs.

2.3 NetTraceCollector

The NetTraceCollector is a Perl-based tool that collects trace data from the AppLog and outputs the data as in .xml format.

2.4 Machine-readable Min Trace Level

At Min trace level, the SIP transactions are represented by several XML elements. Not all SIP headers are represented when tracing at this level.

SIP tracing at Min level is in plaintext and possible to read without post-processing, although post-processing would normally be undertaken.

An example of the XML file output for the Min trace level is provided in Example 1.

Note: The XML file usually contains more than one message.



In Example 1, “==>” is used to highlight where the output lines have been shifted down to fit the PDF version of this document.

```
<?xml version="1.0" encoding="UTF-8"?>
<trace CollecFile xmlns="http://www.3gpp.org/ftp/specs/archive/32_series/32.423#traceData"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.3gpp.org/ftp/specs/archive/32_series/32.423#traceData">
<fileHeader fileFormatVersion="32.423 V8.1.0" vendorName="Ericsson AB">
<fileSender/><trace Collec beginTime="2018-09-05T15:22:47+02:00"/></fileHeader>
<traceRecSession traceRecSessionRef="8042958"><traceSessionRef>
<TRACE_ID>1320</TRACE_ID></traceSessionRef><ue idType="Private User Id" idValue="0"/>
<msg function="SIP" name="REGISTER" changeTime="0.000" vendorSpecific="false"><initiator>
{address == 192.168.83.254, port == 23757, transport == Udp}</initiator>
<target>{address == 192.168.83.100, port == 5082, transport == Udp}
</target><ieGroup name="Message Headers"><ie name="From">sip:scscf1.network.net</ie>
<ie name="To">sip:A-TC_NCTRL_TRACE0320@ericsson.com</ie><ie name="Call-ID">381623364</ie>
<ie name="CSeq">1 REGISTER</ie></ieGroup><ie name="Request-Line">
REGISTER sip:192.168.83.100 SIP/2.0</ie><ieGroup name="tracedPublicIds">
<ie name="tracedPublicId1">sip:A-TC_NCTRL_TRACE0320@ericsson.com</ie></ieGroup></msg>
<msg function="SIP" name="200" changeTime="0.021" vendorSpecific="false"><initiator>
{address == 192.168.83.100, port == 5082, transport == Udp}</initiator>
<target>{address == 192.168.83.254, port == 5060, transport == Udp}
</target><ie name="Status-Line">SIP/2.0 200 OK</ie><ieGroup name="Message Headers">
<ie name="CSeq">1 REGISTER</ie><ie name="Call-ID">381623364</ie><ie name="From">sip:scscf1.ne
<ie name="To">sip:A-TC_NCTRL_TRACE0320@ericsson.com</ie></ieGroup></msg>
<msg function="SIP" name="REGISTER" changeTime="3.736" vendorSpecific="false"><initiator>
{address == 192.168.83.254, port == 44975, transport == Udp}</initiator>
<target>{address == 192.168.83.100, port == 5082, transport == Udp}
</target><ie name="Request-Line">REGISTER sip:192.168.83.100 SIP/2.0</ie>
<ieGroup name="Message Headers"><ie name="Call-ID">381623364
</ie><ie name="CSeq">2 REGISTER</ie><ie name="From">sip:scscf1.network.net</ie>
<ie name="To">sip:A-TC_NCTRL_TRACE0320@ericsson.com</ie></ieGroup>
</msg><msg function="SIP" name="200" changeTime="3.740" vendorSpecific="false">
<initiator>{address == 192.168.83.100, port == 5082, transport == Udp}
</initiator><target>{address == 192.168.83.254, port == 5060, transport == Udp}
</target><ie name="Status-Line">SIP/2.0 200 OK</ie><ieGroup name="Message Headers">
<ie name="To">sip:A-TC_NCTRL_TRACE0320@ericsson.com</ie>
<ie name="From">sip:scscf1.network.net</ie><ie name="Call-ID">381623364</ie>
<ie name="CSeq">2 REGISTER</ie></ieGroup></msg>
</traceRecSession>
</trace CollecFile>
```

Example 1 XML File Output for Min Trace Level

2.5 Machine-readable Max Trace Level

At Max trace level, the SIP transactions are encoded in hexadecimal format and output as raw data. In contrast to the Min level, the complete contents of each request or response are output.



Post-processing of XML files obtained at Max level is required to obtain human readable SIP traces.

An example of the XML file output (partial trace) for the Max trace level is provided in Example 2.

In Example 2, “==>” is used to highlight where the output lines have been shifted down to fit the PDF version of this document.

```
<?xml version="1.0" encoding="UTF-8"?>
<trace CollecFile xmlns="http://www.3gpp.org/ftp/specs/archive/32_series/32.423#traceData"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.3gpp.org/ftp/specs/archive/32_series/32.423#traceData">
  <fileHeader fileFormatVersion="32.423 V8.1.0" vendorName="Ericsson AB">
  <fileSender/><trace Collec beginTime="2018-09-05T15:33:40+02:00"/>
  </fileHeader><traceRecSession traceRecSessionRef="537955"><traceSessionRef>
  <TRACE_ID>1330</TRACE_ID></traceSessionRef><ue idType="Private User Id" idValue="0"/>
  <msg function="SIP" name="INVITE" changeTime="0.000" vendorSpecific="false"><initiator>
  {address == 192.168.83.254, port == 9132, transport == Udp}</initiator>
  <target>{address == 192.168.83.100, port == 5082, transport == Udp}</target>
  <rawMsg protocol="SIP" version="2.0">
  494E56495445207369703A422D54435F4E4354524C5F545241434530333330406572696373736F6E2E6
  36F6D205349502F322E300D0A416C6C6F773A52454749535445522C494E564954452C41434B2C425945
  2C43414E43454C2C4F5054494F4E532C5550444154452C505241434B2C52454645522C4E4F544946592
  C4D4553534147452C5355425343524942452C5055424C4953480D0A43616C6C2D49443A343033363833
  3539380D0A436F6E746163743A7369703A412D54435F4E4354524C5F545241434530333330406572696
  373736F6E2E636F6D0D0A436F6E74656E742D4C656E6774683A323030D0A436F6E74656E742D547970
  653A6170706C69636174696F6E2F7364700D0A435365713A3120494E564954450D0A46726F6D3A73697
  03A412D54435F4E4354524C5F545241434530333330406572696373736F6E2E636F6D3B7461673D3135
  3934363937323330D0A4D61782D466F7277617264733A37300D0A4D696E2D53453A393030D0A502D4
  1737365727465642D4964656E746974793A7369703A412D54435F4E4354524C5F545241434530333330
  406572696373736F6E2E636F6D0D0A502D4368617267696E672D46756E6374696F6E2D4164647265737
  365733A6363663D6363665265616C6D312E6572696373736F6E2E73653B6363663D6363665265616C6D
  322E6572696373736F6E2E73650D0A502D4368617267696E672D566563746F723A696369642D76616C7
  5653D31323937313932333343B696369642D67656E6572617465642D61743D3139322E302E302E313B
  6F7269672D696F693D6373636641496F692E636F6D0D0A5265636F72642D526F7574653A3C7369703A4
  F4449403139322E3136382E38332E3235343A353036303B6C723E0D0A526F7574653A3C7369703A3139
  322E3136382E38332E3130303A353038323B6C723B6D736973646E3D34363132333435363E2C3C73697
  03A4F4449403139322E3136382E38332E3235343A353036313B6C723E0D0A53657373696F6E2D457870
  697265733A31383030D0A537570706F727465643A74696D65722C31303072656C0D0A546F3A7369703
  A422D54435F4E4354524C5F545241434530333330406572696373736F6E2E636F6D0D0A557365722D41
  67656E743A204D5441535F5454434E2D335F4672616D65776F726B0D0A5669613A5349502F322E302F5
  54450203139322E3136382E38332E3235343A353036303B6272616E63683D7A39684734624B36323534
  39363133320D0A0D0A763D300D0A6F3D757365722036323037343437373220313339323439363730382
  0494E20495034203139322E3136382E38332E3235340D0A733D2D0D0A653D3139322E3136382E38332E
  323534406572696373736F6E2E73650D0A743D3020300D0A6D3D766964656F2032313030205254502F4
  15650203020320D0A633D494E20495034203139322E3136382E302E310D0A623D41533A3132380D0A61
  3D7274706D61703A302050434D552F383030300D0A613D7274706D61703A3220473732312F383030300D0A
  </rawMsg><ieGroup name="tracedPublicIds"><ie name="tracedPublicId1">
  sip:A-TC_NCTRL_TRACE0330@ericsson.com
  </ie></ieGroup></msg><msg function="SIP" name="100" changeTime="0.006"
```



```
vendorSpecific="false">
<initiator>{address == 192.168.83.100, port == 5082, transport == Udp}
</initiator><target>
{address == 192.168.83.254, port == 5060, transport == Udp}
</target><rawMsg protocol="SIP" version="2.0">
5349502F322E302031303020547279696E670D0A5669613A205349502F322E302F554450203139322E3
136382E38332E3235343A353036303B6272616E63683D7A39684734624B3632353439363133320D0A46
726F6D3A207369703A412D54435F4E4354524C5F545241434530333330406572696373736F6E2E636F6
D3B7461673D313539343639373233300D0A546F3A207369703A422D54435F4E4354524C5F5452414345
30333330406572696373736F6E2E636F6D3B7461673D70363535333774313533363135343432306D373
1303234366333343073325F333533323936373632382D313033323235323533300D0A43616C6C2D4944
3A203430333638333539380D0A435365713A203120494E564954450D0A537570706F727465643A20746
96D65720D0A436F6E746163743A207369703A70363535333774313533363135343432306D3731303234
36633334307332403139322E3136382E38332E3130303A353038320D0A5365727665723A20457269637
3736F6E204D544153202D20435850323031303133342F31205231334138330D0A436F6E74656E742D4C
656E6774683A20300D0A0D0A
</rawMsg></msg><msg function="SIP" name="INVITE" changeTime="0.021"
vendorSpecific="false"><initiator>
{address == 192.168.83.100, port == 5082, transport == Udp}
</initiator><target>{address == 192.168.83.254, port == 5061, transport == Udp}
</target><rawMsg protocol="SIP" version="2.0">494E56495445207369703A422D54435F4E435
4524C5F545241434530333330406572696373736F6E2E636F6D205349502F322E300D0A5669613A2053
49502F322E302F554450203139322E3136382E38332E3130303A353038323B6272616E63683D7A39684
734624B333533323938313439332D3639343135323338340D0A526F7574653A203C7369703A4F444940
3139322E3136382E38332E3235343A353036313B6C723E0D0A4D61782D466F7277617264733A2036390
D0A416C6C6F773A2052454749535445522C52454645522C4E4F544946592C5355425343524942452C55
50444154452C505241434B2C5055424C4953482C494E564954452C41434B2C4F5054494F4E532C43414
E43454C2C4259450D0A46726F6D3A207369703A412D54435F4E4354524C5F5452414345303333304065
72696373736F6E2E636F6D3B7461673D70363535333774313533363135343432306D373130323436633
3343073325F333533323938313531392D313334343733383239380D0A546F3A207369703A422D54435F
4E4354524C5F545241434530333330406572696373736F6E2E636F6D0D0A43616C6C2D49443A2070363
535333774313533363135343432306D3731303234366333343073330D0A435365713A203120494E5649
54450D0A4D696E2D53453A203930300D0A53657373696F6E2D457870697265733A20313830300D0A537
570706F727465643A2074696D65722C31303072656C0D0A436F6E746163743A207369703A7036353533
3774313533363135343432306D373130323436633334307332403139322E3136382E38332E3130303A3
53038323B2B672E336770702E696373692D7265663D2275726E25334175726E2D787878253341336770
702D736572766963652E696D732E696373692E6D6D74656C220D0A53657373696F6E2D49443A6162366
336333330346533333662333633633338366533337653335373233390D0A4163636570742D436F6E74
6163743A202A3B2B672E336770702E696373692D7265663D2275726E25334175726E2D7878782533413
36770702D736572766963652E696D732E696373692E6D6D74656C220D0A502D4368617267696E672D46
756E6374696F6E2D4164647265737365733A206363663D6363665265616C6D312E6572696373736F6E2
E73653B6363663D6363665265616C6D322E6572696373736F6E2E73650D0A502D4368617267696E672D
566563746F723A20696369642D76616C75653D313239373139323333343B696369642D67656E6572617
465642D61743D3139322E302E302E313B6F7269672D696F693D6373636641496F692E636F6D0D0A502D
41737365727465642D4964656E746974793A207369703A412D54435F4E4354524C5F545241434530333
330406572696373736F6E2E636F6D0D0A436F6E74656E742D547970653A206170706C69636174696F6E
2F7364700D0A557365722D4167656E743A204572696373736F6E204D544153202D20435850323031303
133342F31205231334138330D0A436F6E74656E742D4C656E6774683A203139380D0A0D0A763D300D0A
6F3D2D2031323934373233353032203335333239383131353820494E20495034203139322E3136382E3
8332E3130300D0A733D2D0D0A653D3139322E3136382E38332E323534406572696373736F6E2E73650D
0A743D3020300D0A6D3D766964656F2032313030205254502F415650203020320D0A633D494E2049503
```




```
4203139322E3136382E302E310D0A623D41533A3132380D0A613D7274706D61703A302050434D552F38
3030300D0A613D7274706D61703A3220473732312F383030300D0A
</rawMsg></msg><msg function="SIP" name="180" changeTime="0.040" vendorSpecific="false">
<initiator>{address == 192.168.83.254, port == 28902, transport == Udp}
</initiator><target>{address == 192.168.83.100, port == 5082, transport == Udp}
</target><rawMsg protocol="SIP" version="2.0">
5349502F322E30203138302052696E67696E670D0A416C6C6F773A52454749535445522C494E564954
452C41434B2C4259452C43414E43454C2C4F5054494F4E532C5550444154452C505241434B2C524546
45522C4E4F544946592C4D4553534147452C5355425343524942452C5055424C4953480D0A43616C6C
2D49443A703635333774313533363135343432306D3731303234366333343073330D0A436F6E7461
63743A7369703A422D54435F4E4354524C5F545241434530333330406572696373736F6E2E636F6D0D
0A436F6E74656E742D4C656E6774683A300D0A435365713A3120494E564954450D0A46726F6D3A7369
703A412D54435F4E4354524C5F545241434530333330406572696373736F6E2E636F6D3B7461673D70
363535333774313533363135343432306D3731303234366333343073325F333533323938313531392D
313334343733383239380D0A5265636F72642D526F7574653A3C7369703A4F4449403139322E313638
2E38332E3235343A353036313B6C723E0D0A5365727665723A204D5441535F5454434E2D335F467261
6D65776F726B0D0A537570706F727465643A74696D65720D0A546F3A7369703A422D54435F4E435452
4C5F545241434530333330406572696373736F6E2E636F6D3B7461673D3236323935323834360D0A56
69613A5349502F322E302F554450203139322E3136382E38332E3130303A353038323B6272616E6368
3D7A39684734624B333533323938313439332D3639343135323338340D0A0D0A
</rawMsg></msg><msg function="SIP" name="180" changeTime="0.044"
vendorSpecific="false">
<initiator>{address == 192.168.83.100, port == 5082, transport == Udp}</initiator><target>
{address == 192.168.83.254, port == 5060, transport == Udp}
</target><rawMsg protocol="SIP" version="2.0">
5349502F322E30203138302052696E67696E670D0A5669613A205349502F322E302F554450203139322
E3136382E38332E3235343A353036303B6272616E63683D7A39684734624B3632353439363133320D0A
5265636F72642D526F7574653A203C7369703A4F4449403139322E3136382E38332E3235343A3530363
03B6C723E0D0A46726F6D3A207369703A412D54435F4E4354524C5F5452414345303333304065726963
73736F6E2E636F6D3B7461673D313539343639373233300D0A546F3A207369703A422D54435F4E43545
24C5F545241434530333330406572696373736F6E2E636F6D3B7461673D703635353337743135333631
35343432306D3731303234366333343073325F333533323936373632382D313033323235323533300D0
A43616C6C2D49443A203430333638333539380D0A435365713A203120494E564954450D0A537570706F
727465643A2074696D65720D0A436F6E746163743A207369703A7036353533377431353336313534343
2306D373130323436633334307332403139322E3136382E38332E3130303A353038320D0A416C6C6F77
3A204E4F544946592C52454645522C505241434B2C5550444154452C4F5054494F4E532C4259452C414
34B2C43414E43454C2C494E564954452C52454749535445520D0A5365727665723A204572696373736F
6E204D544153202D20435850323031303133342F31205231334138330D0A436F6E74656E742D4C656E6
774683A20300D0A0D0A
</rawMsg></msg><msg function="SIP" name="200" changeTime="0.087" vendorSpecific="false">
<initiator>{address == 192.168.83.254, port == 48795, transport == Udp}</initiator>
<target>{address == 192.168.83.100, port == 5082, transport == Udp}
</target><rawMsg protocol="SIP" version="2.0">
5349502F322E3020323030204F4B0D0A416C6C6F773A52454749535445522C494E564954452C41434B2
C4259452C43414E43454C2C4F5054494F4E532C5550444154452C505241434B2C52454645522C4E4F54
4946592C4D4553534147452C5355425343524942452C5055424C4953480D0A43616C6C2D49443A70363
535333774313533363135343432306D3731303234366333343073330D0A436F6E746163743A7369703A
422D54435F4E4354524C5F545241434530333330406572696373736F6E2E636F6D0D0A436F6E74656E7
42D4C656E6774683A3137370D0A436F6E74656E742D547970653A6170706C69636174696F6E2F736470
0D0A435365713A3120494E564954450D0A46726F6D3A7369703A412D54435F4E4354524C5F545241434
530333330406572696373736F6E2E636F6D3B7461673D70363535333774313533363135343432306D37
```



```
31303234366333343073325F333533323938313531392D313334343733383239380D0A502D436861726
7696E672D46756E6374696F6E2D4164647265737365733A6363663D6363665265616C6D312E65726963
73736F6E2E73653B6363663D6363665265616C6D322E6572696373736F6E2E73650D0A502D436861726
7696E672D566563746F723A696369642D76616C75653D313239373139323333343B696369642D67656E
6572617465642D61743D3139322E302E302E313B6F7269672D696F693D6373636641496F692E636F6D3
B7465726D2D696F693D6373636642496F692E636F6D0D0A5265636F72642D526F7574653A3C7369703A
4F4449403139322E3136382E38332E3235343A353036313B6C723E0D0A526571756972653A74696D657
20D0A53657373696F6E2D457870697265733A313830303B7265667265736865723D7561630D0A536573
73696F6E2D49443A61623663363333303465333336623336336333383665333337653335373233390D0
A5365727665723A204D5441535F5454434E2D335F4672616D65776F726B0D0A537570706F727465643A
74696D65720D0A546F3A7369703A422D54435F4E4354524C5F545241434530333330406572696373736
F6E2E636F6D3B7461673D3236323935323834360D0A5669613A5349502F322E302F554450203139322E
3136382E38332E3130303A353038323B6272616E63683D7A39684734624B333533323938313439332D3
639343135323338340D0A0D0A763D300D0A6F3D75736572203130363035323431303020313237383632
3739303820494E20495034203139322E3136382E38332E3235340D0A733D2D0D0A653D3139322E31363
82E38332E323534406572696373736F6E2E73650D0A743D3020300D0A6D3D766964656F203232303020
5254502F41565020300D0A633D494E20495034203139322E3136382E302E320D0A623D41533A3132380
D0A613D7274706D61703A302050434D552F383030300D0A
</rawMsg></msg>
<msg function="SIP" name="200" changeTime="0.127" vendorSpecific="false">
<initiator>{address == 192.168.83.100, port == 5082, transport == Udp}
</initiator><target>{address == 192.168.83.254, port == 5060, transport == Udp}
</target>
rawMsg protocol="SIP" version="2.0">
5349502F322E3020323030204F4B0D0A5669613A205349502F322E302F554450203139322E3136382E3
8332E3235343A353036303B6272616E63683D7A39684734624B3632353439363133320D0A5265636F72
642D526F7574653A203C7369703A4F4449403139322E3136382E38332E3235343A353036303B6C723E0
D0A46726F6D3A207369703A412D54435F4E4354524C5F545241434530333330406572696373736F6E2E
636F6D3B7461673D313539343639373233300D0A546F3A207369703A422D54435F4E4354524C5F54524
1434530333330406572696373736F6E2E636F6D3B7461673D7036353533377431353336313534343230
6D3731303234366333343073325F333533323936373632382D313033323235323533300D0A43616C6C2
D49443A203430333638333539380D0A435365713A203120494E564954450D0A53657373696F6E2D4578
70697265733A20313830303B7265667265736865723D7561730D0A537570706F727465643A2074696D6
5720D0A436F6E746163743A207369703A70363535333774313533363135343432306D37313032343663
3334307332403139322E3136382E38332E3130303A353038323B2B672E336770702E696373692D72656
63D2275726E25334175726E2D787878253341336770702D736572766963652E696D732E696373692E6D
6D74656C220D0A436F6E74656E742D547970653A206170706C69636174696F6E2F7364700D0A5365737
3696F6E2D49443A61623663363333303465333336623336336333383665333337653335373233390D0A
502D4368617267696E672D46756E6374696F6E2D4164647265737365733A206363663D6363665265616
C6D312E6572696373736F6E2E73653B6363663D6363665265616C6D322E6572696373736F6E2E73650D
0A502D4368617267696E672D566563746F723A20696369642D76616C75653D313239373139323333343
B696369642D67656E6572617465642D61743D3139322E302E302E313B6F7269672D696F693D63736366
41496F692E636F6D3B7465726D2D696F693D6373636642496F692E636F6D0D0A416C6C6F773A2052454
749535445522C52454645522C4E4F544946592C5355425343524942452C5550444154452C505241434B
2C5055424C4953482C494E564954452C41434B2C4F5054494F4E532C43414E43454C2C4259450D0A536
5727665723A204572696373736F6E204D544153202D20435850323031303133342F3120523133413833
0D0A436F6E74656E742D4C656E6774683A203137340D0A0D0A763D300D0A6F3D2D20313738323539323
13530203335333330383739343920494E20495034203139322E3136382E38332E3130300D0A733D2D0D
0A653D3139322E3136382E38332E323534406572696373736F6E2E73650D0A743D3020300D0A6D3D766
964656F2032323030205254502F41565020300D0A633D494E20495034203139322E3136382E302E320D
0A623D41533A3132380D0A613D7274706D61703A302050434D552F383030300D0A
```



```

</rawMsg>
</msg><msg function="SIP" name="ACK" changeTime="0.217" vendorSpecific="false">
<initiator>
{address == 192.168.83.254, port == 41098, transport == Udp}
</initiator><target>
{address == 192.168.83.100, port == 5082, transport == Udp}
</target><rawMsg protocol="SIP" version="2.0">
41434B207369703A70363535333774313533363135343432306D37313032343663333430733240313932
2E3136382E38332E3130303A35303832205349502F322E300D0A43616C6C2D49443A3430333638333539
380D0A436F6E746163743A7369703A412D54435F4E4354524C5F54524143453033333040657269637373
6F6E2E636F6D0D0A436F6E74656E742D4C656E6774683A300D0A435365713A312041434B0D0A46726F6D
3A7369703A412D54435F4E4354524C5F545241434530333330406572696373736F6E2E636F6D3B746167
3D313539343639373233300D0A4D61782D466F7277617264733A37300D0A526F7574653A3C7369703A31
39322E3136382E38332E3130303A353038323B6C723B6D736973646E3D34363132333435363E2C3C7369
703A4F4449403139322E3136382E38332E3235343A353036313B6C723E0D0A546F3A7369703A422D5443
5F4E4354524C5F545241434530333330406572696373736F6E2E636F6D3B7461673D7036353533377431
3533363135343432306D3731303234366333343073325F333533323936373632382D3130333232353235
33300D0A557365722D4167656E743A204D5441535F5454434E2D335F4672616D65776F726B0D0A566961
3A5349502F322E302F554450203139322E3136382E38332E3235343A353036303B6272616E63683D7A39
684734624B323132343437313138320D0A0D0A
</rawMsg></msg>
<msg function="SIP" name="ACK" changeTime="0.227" vendorSpecific="false">
<initiator>
{address == 192.168.83.100, port == 5082, transport == Udp}
</initiator><target>{address == 192.168.83.254, port == 5061, transport == Udp}
</target><rawMsg protocol="SIP" version="2.0">
41434B207369703A422D54435F4E4354524C5F545241434530333330406572696373736F6E2E636F6D20
5349502F322E300D0A5669613A205349502F322E302F554450203139322E3136382E38332E3130303A35
3038323B6272616E63683D7A39684734624B333533333138383239372D313337343239313138320D0A52
6F7574653A203C7369703A4F4449403139322E3136382E38332E3235343A353036313B6C723E0D0A4D6
1782D466F7277617264733A2036390D0A46726F6D3A207369703A412D54435F4E4354524C5F545241434
530333330406572696373736F6E2E636F6D3B7461673D70363535333774313533363135343432306D373
1303234366333343073325F333533323938313531392D313334343733383239380D0A546F3A207369703
A422D54435F4E4354524C5F545241434530333330406572696373736F6E2E636F6D3B7461673D3236323
935323834360D0A43616C6C2D49443A2070363535333774313533363135343432306D373130323436633
3343073330D0A435365713A20312041434B0D0A436F6E746163743A207369703A7036353533377431353
3363135343432306D373130323436633334307332403139322E3136382E38332E3130303A353038323B2
B672E336770702E696373692D7265663D2275726E25334175726E2D787878253341336770702D7365727
66963652E696D732E696373692E6D6D74656C220D0A557365722D4167656E743A204572696373736F6E2
04D544153202D20435850323031303133342F31205231334138330D0A436F6E74656E742D4C656E67746
83A20300D0A0D0A
</rawMsg></msg><msg function="SIP" name="BYE" changeTime="0.272" vendorSpecific="false">
<initiator>{address == 192.168.83.254, port == 51330, transport == Udp}
</initiator><target>{address == 192.168.83.100, port == 5082, transport == Udp}
</target><rawMsg protocol="SIP" version="2.0">425945207369703A703635353337743135333
63135343432306D373130323436633334307332403139322E3136382E38332E3130303A353038322053
49502F322E300D0A416C6C6F773A52454749535445522C494E564954452C41434B2C4259452C43414E4
3454C2C4F5054494F4E532C5550444154452C505241434B2C52454645522C4E4F544946592C4D455353
4147452C5355425343524942452C5055424C4953480D0A43616C6C2D49443A3430333638333539380D0
A436F6E746163743A7369703A412D54435F4E4354524C5F545241434530333330406572696373736F6E
2E636F6D0D0A436F6E74656E742D4C656E6774683A300D0A435365713A32204259450D0A46726F6D3A7

```



```
369703A412D54435F4E4354524C5F545241434530333330406572696373736F6E2E636F6D3B7461673D
313539343639373233300D0A4D61782D466F7277617264733A37300D0A526F7574653A3C7369703A313
9322E3136382E38332E3130303A353038323B6C723B6D736973646E3D34363132333435363E2C3C7369
703A4F4449403139322E3136382E38332E3235343A353036313B6C723E0D0A546F3A7369703A422D544
35F4E4354524C5F545241434530333330406572696373736F6E2E636F6D3B7461673D70363535333774
313533363135343432306D3731303234366333343073325F333533323936373632382D3130333232353
23533300D0A557365722D4167656E743A204D5441535F5454434E2D335F4672616D65776F726B0D0A56
69613A5349502F322E302F554450203139322E3136382E38332E3235343A353036303B6272616E63683
D7A39684734624B3139313934323333300D0A0D0A</rawMsg>
</msg><msg function="SIP" name="BYE" changeTime="0.278" vendorSpecific="false">
<initiator>{address == 192.168.83.100, port == 5082, transport == Udp}</initiator>
<target>{address == 192.168.83.254, port == 5061, transport == Udp}
</target><rawMsg protocol="SIP" version="2.0">
25945207369703A422D54435F4E4354524C5F545241434530333330406572696373736F6E2E636F6D20
5349502F322E300D0A5669613A205349502F322E302F554450203139322E3136382E38332E3130303A3
53038323B6272616E63683D7A39684734624B333533333233393430392D3137373937313731360D0A52
6F7574653A203C7369703A4F4449403139322E3136382E38332E3235343A353036313B6C723E0D0A4D
1782D466F7277617264733A2036390D0A416C6C6F773A2052454749535445522C52454645522C4E4F5
44946592C5355425343524942452C5550444154452C505241434B2C5055424C4953482C494E5649544
52C41434B2C4F5054494F4E532C43414E43454C2C4259450D0A46726F6D3A207369703A412D54435F4
E4354524C5F545241434530333330406572696373736F6E2E636F6D3B7461673D70363535333774313
533363135343432306D3731303234366333343073325F333533323938313531392D313334343733383
239380D0A546F3A207369703A422D54435F4E4354524C5F545241434530333330406572696373736F6
</rawMsg></msg>
<msg function="SIP" name="200" changeTime="0.299" vendorSpecific="false">
<initiator>{address == 192.168.83.254, port == 13007, transport == Udp}
</initiator><target>{address == 192.168.83.100, port == 5082, transport == Udp}
</target><rawMsg protocol="SIP" version="2.0">5349502F322E3020323030204F4B0D0A416
C6C6F773A52454749535445522C494E564954452C41434B2C4259452C43414E43454C2C4F5054494F
4E532C5550444154452C505241434B2C52454645522C4E4F544946592C4D4553534147452C5355425
343524942452C5055424C4953480D0A43616C6C2D49443A7036353533377431353336313534343230
6D3731303234366333343073330D0A436F6E74656E742D4C656E6774683A300D0A435365713A32204
259450D0A46726F6D3A7369703A412D54435F4E4354524C5F54524143453033333040657269637373
6F6E2E636F6D3B7461673D70363535333774313533363135343432306D37313032343663333430733
25F333533323938313531392D313334343733383239380D0A5365727665723A204D5441535F545443
4E2D335F4672616D65776F726B0D0A537570706F727465643A74696D65720D0A546F3A7369703A422
D54435F4E4354524C5F545241434530333330406572696373736F6E2E636F6D3B7461673D32363239
35323834360D0A5669613A5349502F322E302F554450203139322E3136382E38332E3130303A35303
8323B6272616E63683D7A39684734624B333533333233393430392D3137373937313731360D0A0D0A
</rawMsg></msg><msg function="SIP" name="200" changeTime="0.300"
vendorSpecific="false">
<initiator>{address == 192.168.83.100, port == 5082, transport == Udp}
</initiator><target>{address == 192.168.83.254, port == 5060, transport == Udp}
</target><rawMsg protocol="SIP" version="2.0">
5349502F322E3020323030204F4B0D0A5669613A205349502F322E302F554450203139322E313638
2E38332E3235343A353036303B6272616E63683D7A39684734624B313931393432333330300D0A4
6726F6D3A207369703A412D54435F4E4354524C5F545241434530333330406572696373736F6E2E
636F6D3B7461673D313539343639373233300D0A546F3A207369703A422D54435F4E4354524C5F5
45241434530333330406572696373736F6E2E636F6D3B7461673D70363535333774313533363135
343432306D3731303234366333343073325F333533323936373632382D313033323235323533300
D0A43616C6C2D49443A203430333638333539380D0A435365713A2032204259450D0A537570706F
```



```
727465643A2074696D65720D0A416C6C6F773A2052454749535445522C52454645522C4E4F54494
6592C5355425343524942452C5550444154452C505241434B2C5055424C4953482C494E56495445
2C41434B2C4F5054494F4E532C43414E43454C2C4259450D0A5365727665723A204572696373736
F6E204D544153202D20435850323031303133342F31205231334138330D0A436F6E74656E742D4C
656E6774683A20300D0A0D0A
</rawMsg></msg>
</traceRecSession>
</traceCollecFile>
```

Example 2 XML File Output (Partial Trace) for Max Trace Level





3 NetTrace Function

The principle of NetTrace is to allow a user the possibility to log SIP transactions traversing the MTAS for fault finding and localization purposes.

Using the vDicos AppTrace, a trace session can be configured to trace SIP transactions based on the filtering of the Originating Public User Ids or Terminating Public User Ids (for both Min and Max levels), or both. The flow of actions required to obtain trace outputs for SIP is shown in Figure 1.

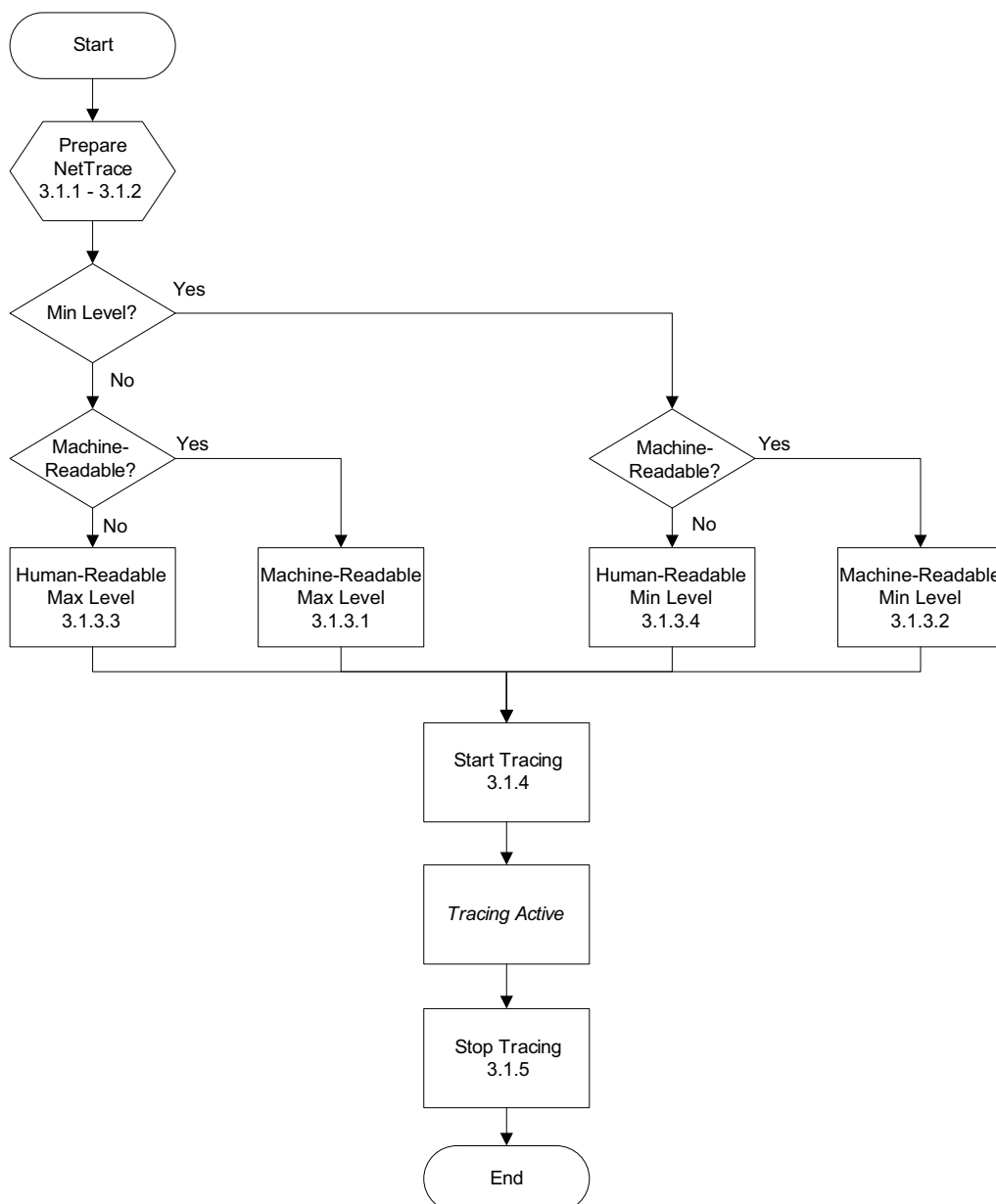


Figure 1 Simplified Flow Configuration and Use of NetTrace



3.1 NetTrace Procedure

This section describes the NetTrace procedure.

3.1.1 Prepare NetTrace

The following sequence is applicable when preparing NetTrace for any type of trace output. “Terminal1” and “Terminal2” refer to two different shells that must be opened.

This procedure is valid when the user has direct access to the node.

“Terminal1” and “Terminal2” refer to two separate System Controllers (shells).

To prepare NetTrace for any type of trace output:

1. For more information on how to ensure that the .xml files (if applicable) are output correctly, refer to [IMS Common Components Troubleshooting Guide](#).
2. Log on to the primary System Controller SC (from Terminal1), then from the SC log on to one of the Payloads (PLs).

```
> ssh -Y root@<OAM VIP>
```

```
> ssh root@<PL>
```

3. Open a new shell (Terminal2), log on to one of the SCs. From the SC, log on to one of the Payloads (PLs) and start the NetTrace collector (applicable for machine-readable only):

```
> ssh -Y root@<OAM VIP>
```

```
> ssh root@<PL>
```

```
> cd /opt/ericsson/cmco/nettrace/bin
```

Check if the nettracecollector.pl script is already running by using, for example:

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

If the nettracecollector.pl is not running, start the script:

```
> ./nettracecollector.pl &
```

4. From Terminal1, enter the AppTrace-CLU directory:

```
> ssh root@<PL>
```

```
> cd /opt/lpmsv/bin/appttrace
```




For more information about AppTrace commands, refer to AppTrace User Guide.

5. From Terminal1, gather Trace Domains:

```
> ./collect_domains.sh
```

6. From Terminal1, verify Trace Domains:

```
> ./verify_domains.sh
```

The following MTAS trace domains must be present in the domain tree:

```
— ims.mtas.nettrace.init
— ims.mtas.nettrace.info
— ims.mtas.nettrace.sip
— ims.mtas.netio.rx
— ims.mtas.netio.tx
— ims.mtas.nettrace.rx
— ims.mtas.nettrace.tx
```

If any of the listed trace domains do not exist, consult the next level of maintenance support.

7. From Terminal1, create a trace session:

```
> ./begin_session.sh
```

8. From Terminal1, include all processors in the trace session:

```
> ./include_processors.sh -a
```

9. From Terminal1, add process types:

```
> ./add_process_type.sh ApplicationProcess.1060633
> ./add_process_type.sh SipDistributorProcessNew.1126603
```

3.1.2 Set up the Traced User

This section describes how to specify the public users to be traced.

The parameter “forlop” is a “Trace Identity” and is a positive integer value from 0 through 1048575 (20 bits). The default value of forlop is zero, which is predefined to mean “anonymous forlop”. The value is chosen by the operator.

To specify the public users to be traced on Terminal 1:

1. Trace the originating user:



```
> ./insert_expression.sh 'ims.mtas.nettrace.init($OrigPublicId
=="<PublicIdToBeTraced>") => $forlop = <forlop>'
```

Examples:

```
— > ./insert_expression.sh 'ims.mtas.nettrace.init($OrigPublicId == "sip:userA@domain.x") => $forlop =12345'
```

```
— > ./insert_expression.sh 'ims.mtas.nettrace.init($OrigPublicId == string(sip:userA@domain.x)) => $forlop =12345'
```

2. Trace the terminating user:

```
> ./insert_expression.sh 'ims.mtas.nettrace.init($TermPublicId
=="<PublicIdToBeTraced>") => $forlop = <forlop>'
```

Examples:

```
— > ./insert_expression.sh 'ims.mtas.nettrace.init($TermPublicId == "sip:userA@domain.x") => $forlop =12345'
```

```
— > ./insert_expression.sh 'ims.mtas.nettrace.init($TermPublicId == string(sip:userB@domain.x)) => $forlop =12345'
```

3. It is possible to trace multiple users by expressing their Public Ids within the same expression using the OR and AND operators.

Examples:

```
— > ./insert_expression.sh 'ims.mtas.nettrace.init(($OrigPublicId == "sip:userA@domain.x") && ($TermPublicId == "sip:userB@domain.y")) =>$forlop = 12345'
```

```
— > ./insert_expression.sh 'ims.mtas.nettrace.init(($OrigPublicId == "sip:userA@domain.x") || ($TermPublicId == "sip:userB@domain.y")) => $forlop = 12345'
```

Note: For more information on combining logical expressions using OR and AND operators, refer to [AppTrace User Guide](#).

Note: Setting up the traced user is a common step that must be performed for all trace level combinations, that is machine/human readable at max/min level.

3.1.3 Set up the Trace Level

This section describes how to set up the applicable trace levels.

3.1.3.1 Machine-readable Max Level

The following sequence is applicable when specifying the trace output is to be machine readable (.xml) at Max level.



To specify the trace output machine readable (.xml) at Max level:

1. The setup of the traced user is performed according to Section 3.1.2 Set up the Traced User on page 21.
2. From Terminal1, insert expressions:

```
> ./insert_expression.sh 'ims.mtas.nettrace.info =>L
($processorname, $pid, $date, $time, $id, $forlop, $MiId,
$MiVer, $Length, $Msg)'
```

Note: This is mandatory for both Min and Max levels.

3. From Terminal1, insert expressions for incoming or outgoing SIP traffic, or both:

```
> ./insert_expression.sh 'ims.mtas.nettrace.rx =>L
($processorname, $pid, $date, $time, $id, $forlop, $MiId,
$MiVer, $Length, $Msg)'
```

and or

```
> ./insert_expression.sh 'ims.mtas.nettrace.tx =>L
($processorname, $pid, $date, $time, $id, $forlop, $MiId,
$MiVer, $Length, $Msg)'
```

Note: For machine-readable Max level trace depth, it is possible to record incoming SIP traffic (using `ims.mtas.nettrace.rx` domain) or outgoing SIP traffic (using `ims.mtas.nettrace.tx` domain) or both (using both `ims.mtas.nettrace.rx` and `ims.mtas.nettrace.tx` domains).

3.1.3.2 Machine-readable Min Level

The following sequence is applicable when specifying the trace output is to be machine readable (.xml) at Min level.

To specify the trace output machine readable (.xml) at Min level:

1. The setup of the traced user is performed according to Section 3.1.2 Set up the Traced User on page 21.
2. From Terminal1, insert expressions:

```
> ./insert_expression.sh 'ims.mtas.nettrace.info =>L
($processorname, $pid, $date, $time, $id, $forlop, $MiId,
$MiVer, $Length, $Msg)'
```

Note: This is mandatory for both Min and Max levels.

3. From Terminal1, insert expressions for incoming and outgoing SIP traffic:



```
> ./insert_expression.sh 'ims.mtas.nettrace.sip =>L  
($processorname, $pid, $date, $time, $id, $forlop, $MiId,  
$MiVer, $Length, $Msg)'
```

3.1.3.3 Human-readable Max Level

The following sequence is applicable when the trace output is to be human-readable at Max level.

To specify the trace output human-readable at Max level:

1. The setup of the traced user is performed according to Section 3.1.2 Set up the Traced User on page 21.
2. From Terminal1, insert expressions:

```
> ./insert_expression.sh 'ims.mtas.netio.info =>L  
($processorname, $pid, $date, $time, $id, $forlop, $MiId,  
$MiVer, $Length, $Msg)'
```

Note: This is mandatory for both Min and Max levels.

3. From Terminal1, insert expressions for incoming or outgoing SIP traffic, or both:

```
> ./insert_expression.sh 'ims.mtas.netio.rx => L($processorname,  
$pid, $date, $time, $id, $forlop, $MiId, $MiVer, $Length,  
$Msg)'
```

or

```
> ./insert_expression.sh 'ims.mtas.netio.tx => L($processorname,  
$pid, $date, $time, $id, $forlop, $MiId, $MiVer, $Length,  
$Msg)'
```

or both

Note: For human readable Max level trace depth, it is possible to record incoming SIP traffic (using `ims.mtas.netio.rx` domain) or outgoing SIP traffic (using `ims.mtas.netio.tx` domain) or both (using both `ims.mtas.netio.rx` and `ims.mtas.netio.tx` domains).

3.1.3.4 Human-readable Min Level

The following sequence is applicable when the trace output is to be human-readable at Min level.

To specify the trace output human-readable at Min level:

1. The setup of the traced user is performed according to Section 3.1.2 Set up the Traced User on page 21.
2. From Terminal1, insert expressions:



```
> ./insert_expression.sh 'ims.mtas.netio.info => L
($processorname, $pid, $date, $time, $id, $forlop, $MiId,
$MiVer, $Length, $Msg)'
```

Note: This is mandatory for both Min and Max levels.

3. From Terminal1, insert expressions for incoming and outgoing SIP traffic:

```
> ./insert_expression.sh 'ims.mtas.netio.sip => L
($processorname, $pid, $date, $time, $id, $forlop, $MiId,
$MiVer, $Length, $Msg)'
```

3.1.4 Start NetTrace

To start NetTrace:

1. From Terminal1, direct output to the AppLog:

```
> ./route_output.sh applog
```

2. From Terminal1, upload the trace session:

```
> ./upload_session.sh
```

3. From Terminal1, start the trace session:

```
> ./start_trace.sh 24
```

The trace session is now active and tracing starts when the trace criteria are fulfilled.

3.1.5 Stop NetTrace

To stop NetTrace:

1. From Terminal1, stop the trace session:

```
> ./stop_trace.sh
```

2. From Terminal1, unload (uninstall) the trace session:

```
> ./unload_session.sh
```

3. From Terminal1, end the trace session:

```
> ./end_session.sh
```

3.2 Analysis of Trace Outputs

This section describes the analysis of trace outputs.



3.2.1 Machine-readable Traces

NetTrace .xml files are accessed from the PL, and are located in:

```
/cluster/storage/no-backup/cmco_utils-cxp9020686/nettrace/mtas
```

The naming convention is as follows:

```
A<date>.<time>-MTAS.<TraceSessionRef>.<TraceRecSessionRef>.xml
```

Where:

- <date> is in the form `yyyymmdd`.
- <time> is in the form `hhmm`.
- <TraceSessionRef> (TSR) is the user-defined forlop Id. User can be OSS-RC or any troubleshooter.
- <TraceRecSessionRef> (TRSR) is the system-defined Trace Recording Session Ref.

MTAS calculates the value of hashed Call-ID and assigns it to TRSR.

For example:

```
A20110128.1609-MTAS.jambala.1111.56032.xml
```

Post-processing is required by a system compliant with [3GPP TS 32.423](#).

3.2.2 Human-readable Traces

Human readable traces are accessed from the PL, and are located as AppTrace files in the following directory:

```
/cluster/storage/no-backup/coremw/var/log/saflog/MTASAppLogs/vdi  
cos/
```