

# CSCF User Tracing

## Call Session Control Function

---

### USER GUIDE

**Copyright**

© Ericsson AB 2016, 2017. 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

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Prerequisites	1
<b>2</b>	<b>UserTrace Function</b>	<b>3</b>
2.1	Execute UserTrace without CscfTrace Script	3
2.2	Execute UserTrace with CscfTrace Script	6
<b>3</b>	<b>Trace Profiles</b>	<b>11</b>





# 1 Introduction

This document describes the procedure to obtain user traces from the Call Session Control Function (CSCF) and the Emergency Access Transfer Function (EATF) nodes.

## 1.1 Prerequisites

This section provides information on the documents and conditions that apply to the procedure.

### 1.1.1 Documents

Before starting this procedure, make sure that the following documents have been read:

- *AppTrace User Guide*
- *CSCF AppTrace User Guide*
- *CSCF Network Tracing*
- *CSCF Troubleshooting Guideline*
- *IMS Common Components Troubleshooting Guide*

### 1.1.2 Conditions

Certain troubleshooting activities can have impact on node performance. For example, trace activation can be traffic disturbing and is not recommended without first consulting Ericsson.

When UserTrace is activated, even for only one user, node performance is affected. To limit the performance drop, it is recommended that UserTrace is enabled for one Public User Identity and one trace profile. The trace profiles are a set of predefined AppTrace domains and process types that are grouped by functional areas. For a list of trace profiles, see Section 3 on page 11. One case that requires special mention is the Wildcarded Public User Identity (wIMPU). A wIMPU is typically used for the aggregation of SIP end users not individually known to the IP Multimedia Subsystem (IMS) system. A wIMPU gives a common service profile to all aggregated end users. Since the wIMPU can represent many end users, a wIMPU is not recommended as a specified Public User Identity for tracing owing to the high performance impact.





## 2 UserTrace Function

**Note:** The inherent problem with observing the behavior of a system by tracing is the consumed capacity of the tracing itself. UserTrace must be limited to tracing on one user and only on specific Trace Profiles.

UserTrace is a tool that enables logging of trace points that traverse the CSCF and the EATF applications for a defined Public User Identity.

UserTrace can be set to trace on a user initiating an originating SIP request or a user receiving a terminating SIP request. To trace on the same user initiating an originating SIP request or receiving a terminating SIP request, it must be defined as both `OrigPublicId` and `TermPublicId`. UserTrace is not supported for incoming Diameter messages Registration Termination Request (RTR), when the Public User Identity is not included in the message, or Push-Profile-Request (PPR).

Trace Profiles are a set of predefined AppTrace domains and process types that can be used in a live system. UserTrace output is to be directed to `rawconsole` or `applog`.

Tracing on all CSCF trace domains for a Public User Identity is not recommended owing to the impact on the processor load. Therefore tracing on a subset of available CSCF trace domains is recommended.

Specific trace domains have been preconfigured in the `CscfTrace` script into CSCF trace profiles for a live system. They have been verified and they run with a performance level degradation depending on the CSCF trace profile.

### 2.1 Execute UserTrace without CscfTrace Script

The following sequence is applicable when preparing UserTrace for any type of trace output. To limit the performance drop in a live network, it is recommended that it is activated only after receiving the exact trace input parameters required for troubleshooting from Ericsson support. This means that the trace domains and the process types to be used during the trace session are specified by the Ericsson support team. This is valid when the user has direct access to the node.

The following sequence is applicable when specifying the public users to trace and the trace output to `rawconsole`.

The parameter `forlop` is a “Trace Identity” and is an integer value. The value is chosen by user input and must be in the range of 1–1048575 and is assigned to the specified Public User Identity.



## 2.1.1 Prepare UserTrace

To prepare UserTrace for any type of trace output:

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

```
>ssh -A <user>@<SC address>
```

For example:

```
>ssh -A user1@192.168.10.1
```

2. Log on to the CSCF Payload (PL) node:

```
>ssh -A <user>@<PL-X address>
```

For example:

```
>ssh -A user1@192.168.10.2
```

3. If there is an ongoing trace session, clear the previous trace session, see Section 2.1.3 Stop UserTrace on page 6.

4. Clear the console logs:

```
>cdclsv-clear
```

5. Create a trace session:

```
>cd /opt/lpmsv/bin/apptrace
```

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

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

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

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

6. List and add process types for one trace profile.

Verify with Ericsson personnel for the recommended process types.  
List the process type, as the numerical value after the process name is system-dependent.

```
>./ls_processtypes.sh
```

```
>./add_process_type.sh <PT1> [<PT2> ...]
```

For example:

```
>./add_process_type.sh CscfAppProc.1041756
```





7. Insert expressions specifying originating Public User Identity or terminating Public User Identity to trace:

```
>./insert_expression.sh 'ims.cscf.nettrace.init($OrigPublicId ==\"sip:user@domain\") => $forlop = <forlop value>'
```

or:

```
>./insert_expression.sh 'ims.cscf.nettrace.init($TermPublicId ==\"sip:user@domain\") => $forlop = <forlop value>'
```

**Note:** It is possible to express multiple user identities within the same expression using OR and AND operators. However, for UserTrace, only one Public User Identity is recommended.

8. Activate UserTrace (this is the domain to activate UserTrace):

```
>./insert_expression.sh ims.usertrace
```

**Note:** The `ims.usertrace` profile is important. Not adding it results in enabling AppTrace for every user, which is a risk in overloading the CSCF node.

9. Insert CSCF AppTrace trace domains for one trace profile:

List the CSCF AppTrace trace domains in CSCF:

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

```
>./ls_domains.sh
```

Verify with Ericsson personnel for the recommended trace domains.

For CX/DX, for example:

```
>./insert_expression.sh Domains/ims.cscf.cxdx
```

## 2.1.2 Start UserTrace

To start UserTrace:

1. Direct output to the `rawconsole`:

```
>./route_output.sh rawconsole
```

2. Upload the trace session:

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

3. Start the trace session:



```
>./start_trace.sh 58
```

58 is the recommended trace level.

The trace session is now active and tracing begins when the trace criteria is fulfilled. The trace files can be found in one of the following locations:

For console log: `/cluster/storage/no-backup/cdclsv/log/lpm  
sv/`

Applog location: `/cluster/storage/no-backup/coremw/var/lo  
g/saflog`

### 2.1.3 Stop UserTrace

To stop UserTrace:

1. Stop the trace session:

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

2. Unload (uninstall) the trace session:

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

3. End the trace session:

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

## 2.2 Execute UserTrace with CscfTrace Script

The `CscfTrace` is a script that has been developed to simplify the use of UserTrace. It contains the predefined CSCF trace profile and also verifies the CPU load on the Traffic Processors.

### 2.2.1 Start UserTrace

To execute UserTrace with the `CscfTrace` script:

1. Log on to the SC processor of the CSCF node:

```
>ssh -A user1@<IP-address>
```

2. If there is an ongoing trace session, clear the previous trace session, see Section 2.2.2 Stop UserTrace on page 9.

3. Clear the console logs:

```
>cdclsv-clear
```

4. To start the tracing, execute the tracing script:



```
>CscfTrace {-orig <OrigPublicId> {-term <TermPublicId>}}
<CSCF Trace Profile>
```

See Section 3 on page 11 for the list of preconfigured CSCF trace profiles.

The following is a list of options to the CscfTrace script:

**-h** Prints CscfTrace list of options

**-ro <ROUT\_OUTPUT>**

Alters the destination and capacity of output. By default, rawconsole is selected as output destination.

**- applog** The output is sent to the AppLog stream under the AppTrace log label.

**- rawconsole** Sends the output directly (from the application process) to the console log of the processor.

**-l <TRACE\_LEVEL>**

Sets a different trace level. By default, trace level is set to DEBUG.

The following four trace levels are available for <TRACE\_LEVEL>:

DEBUG: only intended for function testing and debugging.

MINOR: intended for detailed subsystem live site tracing.

MAJOR: intended for broad, system-wide, live site tracing.

CRITICAL: intended for live site tracing of only exceptional events.

**-orig <ORIG\_PUB\_ID(s)>, --orig\_pub\_id=<ORIG\_PUB\_ID(s)>**

Originating Public ID, permits tracing for an originating user.

Examples:

```
-orig sip:eric.almighty@ericsson25.lab,s
ip:alice.almighty@ericsson25.lab,tel:+15
143457900
--orig_pub_id=sip:eric.almighty@ericsson25.lab
--orig_pub_id=sip:alice.almighty@ericsson25.lab
--orig_pub_id=tel:+49616000014
```

**Note:** This option is CASE SENSITIVE.

**-term <TERM\_PUB\_ID(s)>, --term\_pub\_id=<TERM\_PUB\_ID(s)>**

Terminating Public IDs, permits tracing for a specific user or users.

**Examples:**

```
-term sip:eric.almighty@ericsson25.lab,s
ip:alice.almighty@ericsson25.lab,tel:+49
616000014
--term_pub_id=sip:eric.almighty@ericsson25.lab
--term_pub_id=sip:alice.almighty@ericsson25.lab
--term_pub_id=tel:+49616000014
```

**Note:** This option is CASE SENSITIVE.

**-user <USER\_PUB\_ID(s)>, --user\_pub\_id=<USER\_PUB\_ID(s)>**

User Public IDs. This option is the equivalent of setting both Originating and Terminating Public IDs to the same value.

```
-user sip:user1@domain is equal to -orig
sip:user1@domain -term sip:user1@domain
```

**Examples:**

```
-user sip:eric.almighty@ericsson25.lab,s
ip:alice.almighty@ericsson25.lab,tel:+49
616000014
--user_pub_id=sip:eric.almighty@ericsson25.lab
--user_pub_id=sip:alice.almighty@ericsson25.lab
--user_pub_id=tel:+49616000014
```

**Note:** This option is CASE SENSITIVE.

**-f <PUB\_ID=forlop pair(s)> --forlop <PUB\_ID=forlop pair(s)>**

Public ID to Forlop hash entries. Forlop number (integer, valid range: 1–1048575) can be assigned to any specified Public ID (string).

If this option is omitted, one forlop number is assigned to each Public ID starting with 1234.

If forlop numbers are only specified for certain Public IDs but not all using this option, one forlop number is assigned to each unspecified Public ID starting with 1234.

If the Public ID in this option equals “all”, the same forlop number is assigned to all Public IDs.

**Examples:**

```
-f sip:user1@domain1=9999 -f tel:+4961600
0014=8888 --forlop all=4444
```



**-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.

**Note:** `skip_cpu_check(sc)` and `max_cpu(mc)` or `stop_timeout(st)` are mutual exclusive options.

**-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

**Note:** `skip_cpu_check(sc)` and `max_cpu(mc)` 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.

**Note:** `skip_cpu_check(sc)` and `stop_timeout(st)` are mutual exclusive options.

**-v, verbose**

This option is used to display the result texts of CLU successful commands.

5. The following is an example of what is displayed on the Terminal while the CscfTrace script is running:

```
Starting CPU load monitoring...
Maximum CPU threshold to stop trace is 70%
This script will automatically stop the trace after 20 minute(s) at 19-09-14 11:48:05
Please press CTRL-C if you want to stop the trace immediately
```

Date	Time	Load	PL-3	PL-4	PL-5	PL-6	PL-7	PL-8
19-09-14	11:29:06	CPUload %	3.78	2.61	2.11	2.42	5.26	3.53

## 2.2.2 Stop UserTrace

To stop the CscfTrace script, and ensure that all processes are terminated correctly:

1. If the CPU monitoring is present, press **Ctrl+C** to stop the CscfTrace script immediately.
2. If the CPU monitoring is absent, run:

```
>CscfTrace STOP
```



## 3 Trace Profiles

It is recommended that UserTrace is enabled for one Public User Identity and one trace profile.

The preconfigured trace profiles in the `CscfTrace` script listed in Table 1 have been verified and run with acceptable performance degradation when running for a limited duration of time to collect traces for troubleshooting. The trace profiles are included in files that are packaged with the `CscfTrace` script.

**Note:** The trace profile files are not to be modified without first consulting Ericsson. Modifying the profile files can have unexpected consequences, including system failures.

The trace profile files can be found in `/opt/ericsson/cscf/trace/etc`

The trace profiles listed in Table 1 are a set of predefined AppTrace domains grouped by functional area.

*Table 1 Trace Profiles*

Trace Profile	Description
CSCFv_All_Trace_Profile	This trace profile outputs trace information for all domains except Diameter ones.
CSCFv_DNS_ICMP_Trace_Profile	This trace profile outputs trace information related to DNS queries/answers from or to the CSCF.
CSCFv_General_Session_Trace_Profile	This trace profile outputs general trace information about a received user request. It is typically used initially to determine if more specific trace profiles are required to get more detailed information in some specific area.



Trace Profile	Description
CSCF_Registration_Trace_Profile	This trace profile outputs trace information related to the registration and authentication between the CSCF and the Subscriber Location Function (SLF)/Home Subscriber Server (HSS). It is typically used initially to retrieve more detailed information about the user registration and authentication.
CSCFv_SIP_Framework_Trace_Profile	This trace profile outputs trace information related to SIP framework.