

IMS Common Components Troubleshooting Guide

TROUBLESHOOTING

Copyright

© Ericsson AB 2015. 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
3	Troubleshooting Functions	5
3.1	AppTrace	5
3.2	NetTrace	6
4	Troubleshooting Procedure	9
5	CMCO Trace Domains	11
6	CMCO Processes	13
7	CMCO Components and Dependencies	15





1 Introduction

The purpose of this document is to provide information on how to perform troubleshooting procedures of the IP Multimedia Subsystem (IMS) Common Component (CMCO) products. This document is to be used together with the troubleshooting guide of a node, for example, Call Session Control Function (CSCF), Multimedia Telephony Application Server (MTAS), where the node is using all or parts of the components that are included in the CMCO products.

1.1 Prerequisites

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

1.1.1 Documents

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

- *IMS Common Components Initial Configuration*
- *AppTrace User Guide*

1.1.2 Tools

The following tools can be used for troubleshooting:

- vDicos App Trace
- NetTrace Collector
- Trace Profile Manager

For more information about the tools, see Section 2 on page 3.

1.1.3 Conditions

Certain troubleshooting activities can have impact on the node performance. For example, trace or log activation can be traffic disturbing and it is not recommended without prior consulting of Ericsson.





2 Tools

vDicos App Trace

Application Trace (AppTrace) is a service, available in the vDicos execution environment. AppTrace is used for tracing vDicos applications.

For more information, refer to *AppTrace User Guide*.

NetTrace Collector

The NetTrace Collector is a utility for converting AppTrace output to XML files according to a schema defined by 3GPP, refer to [3GPP TS 32.423](#). Much like a sniffing tool, it collects information about network transmissions to and from the CMCO components and any supported application.

Trace Profile Manager

The Trace Profile Manager is a generic tool for enabling trace in a system. A profile is defined along with commands and arguments to be executed; basically this can be used to define commonly used AppTrace session settings.





3 Troubleshooting Functions

This section provides a general overview of the functions that can be used to resolve problems that can arise when using the CMCO components.

3.1 AppTrace

An AppTrace session must be set up and configured according to specific requirements for each application. For general information about setting up an AppTrace session, refer to *AppTrace User Guide*.

The following sections describe what must be configured in the AppTrace session in addition to the application-specific settings.

3.1.1 Trace Domains

Each common component has a trace domain under the top-level domain IMS. This trace domain is abstract and is further divided into a number of concrete subdomains. Depending on the component and problem area suitable and appropriate set of CMCO trace domains to be used.

Using `collect_domains.sh` and `ls_domains.sh` AppTrace commands filter out proper set of trace domains to be used. Usually, include all domains of the component which is under troubleshooting and its dependent components. For example sip component troubleshooting use all domains of sip, along with its dependent icmp, parsers, and dns components.

For full list of CMCO domains see Section 5 on page 11, for the list of CMCO components and its dependencies, see Section 7 on page 15.

3.1.2 Defining the Scope of the Session

Usually CMCO Interfaces are executed in an application process and hence processors and processes names are application-dependent. However there are a few CMCO-specific processes which must be used depending on the problem area.

Include all processors in the trace scope. Check all the processes using `ls_processtypes.sh` and prepare list of the processes to be include in the trace session.

For full list of CMCO process names, see Section 6 on page 13.

3.1.3 AppTrace Log Expressions

For normal or initial troubleshooting, the following AppTrace log expression is enough:

```
L($time, $pid, $Method, $Instance, $Msg)
```

For detailed troubleshooting use the following log expression:

```
L($date, $time, $forlop, $pid, $processname, $id, $level,  
$file, $line, $Unit, $Ver, $Method, $Instance, $Length,  
$Msg)
```

3.1.4 Trace Level

At least, use level 23 (DICOS_AT_MAJOR) and, whenever possible level 55 (DICOS_AT_DEBUG), to get reasonably good CMCO traces.

3.2 NetTrace

NetTrace relies on vDicos AppTrace feature for producing raw data, which is then formatted into XML files according to the 3GPP standard [3GPP TS 32.423](#) by NetTrace Collector Tool.

CMCO IMS Framework and AA Answer (AAA) Framework have support for NetTrace and capable of outputting complete protocol messages for the “maximum” NetTrace levels. This data on its own is not sufficient to produce the standard XML files. Application support for NetTrace is also required and each application provides its own specific details on how to set up and use NetTrace. This document only describes the specific details for CMCO components.

3.2.1 Trace Domains

In addition to the application-specific trace domains, the following CMCO trace domains must be used:

- `ims.sip.nettrace.rx`
- `ims.sip.nettrace.tx`
- `ims.sip.netio.rx`
- `ims.sip.netio.tx`
- `ims.diaif.nettrace.rx`
- `ims.diaif.nettrace.tx`
- `ims.diaif.netio.rx`



- `ims.diaif.netio.tx`

3.2.2 Defining the Scope of the Session

In addition to the application-specific process, the following CMCO process must be used:

- `> add_process_type ProcessTypes/SIP_MsgParsingProc.SWI_SIP_MsgParsingProc`
- `> add_process_type ProcessTypes/SIP_TcpInboundSessionProc.SWI_SIP_TcpInboundSessionProc`
- `> add_process_type ProcessTypes/SIP_TcpOutboundSessionProc.SWI_SIP_TcpOutboundSessionProc`
- `> add_process_type ProcessTypes/SIP_UdpServerProc.SWI_SIP_UdpServerProc`
- `> add_process_type ProcessTypes/SIP_TcpServerProc.SWI_SIP_TcpServerProc`

3.2.3 AppTrace Log Expressions

For NetTrace, use the following log expression:

```
L($date, $time, $forlop, $pid, $id, $Length, $Msg)
```

3.2.4 Trace Level

At least level 23 (DICOS_AT_MAJOR) must be used to get proper NetTraces.





4 Troubleshooting Procedure

Troubleshooting a problem in the CMCO components can require the use of one or more functions described in Section 3 on page 5.

To assure an efficient location of the fault:

1. If there is a crash or a failure in a processor or a process, error dump files are generated.

For more information, refer to *Error Dump and Log User Guide*.

2. Check for AppTrace and NetTrace logs for any interesting information about the problem, see Section 3.1 AppTrace on page 5 and Section 3.2 NetTrace on page 6.
3. Check the relevant Initial Configuration parameters related to the problem, refer to *IMS Common Components Initial Configuration*.
4. Check already reported troubles in the specific area.
5. If the problem can't be solved using this document, it must be reported to the next level of maintenance support. This report can result in either a Trouble Report (TR) or a Customer Service Report (CSR).
6. If writing a TR, check the CMCO software versions used, refer to *Software Inventory Management*.





5 CMCO Trace Domains

The CMCO trace domains are listed in Table 1.

Table 1 CMCO Trace Domains

Component	Trace Domains
dicosutils	ims.cmco_dicosutils.api, ims.cmco_dicosutils.core
diafw	ims.charging.diafw.msg, ims.charging.diafw.install
diaif	ims.diaif.core, ims.diaif.m essage, ims.diaif.netio.rx, ims.diaif.netio.tx, ims.diaif.nettrace.rx, ims.diaif.nettrace.tx
dns	ims.dns.api, ims.dns.cache, ims.dns.message, ims.dns.oa m, ims.dns.transport
icmp	ims.icmp.control, ims.icmp.messageLib, ims.icmp.lookup, ims.icm p.oam, ims.icmp.perfmgmt, ims.icmp.traffic, ims.icmp.provisioni ng, ims.icmp.util, ims.icmp.blacklist, ims.icmp.subscription
numana	ims.numana.api, ims.numana. core, ims.numana.mgr, ims.n umana.oam, ims.numana.util
parsers	ims.parsers.global, ims.parsers.http, ims.parsers.mime, ims.pa rsers.sdp, ims.parsers.sip, ims.parsers.xml
diabh	ims.charging.backup



Component	Trace Domains
diaon	ims.charging.online.fsm, ims.charging.online.mgr, ims.charging.online.diase nd, ims.charging.online.d iareceive, ims.charging.o nline.capsule, ims.chargi ng.online.callterminator, ims.charging.online.netio
numnorm	ims.numnorm.global
carsel	ims.carsel.cps, ims.carsel. cs, ims.carsel.dsa, ims.car sel.oam, ims.carsel.util
http	ims.http.api, ims.http.con trol, ims.http.transport, ims.http.util
sip	ims.sip.management, ims.sip.control, ims.sip .txn, ims.sip.behavior, ims.sip.transport, ims.sip.netio.rx, ims.sip.netio.tx, ims.sip.nettrace.rx, ims.sip.nettrace.tx, ims.sip.traceid
netmon	ims.netmon.sipApplication, ims.netmon.monitoring, ims.netmon.state, ims.netmon.configuration, ims.netmon.common, ims.netmon.faultMgmt



6 CMCO Processes

The Table 2 lists only the process name. The full name, which is used in the AppTrace, is for most process types written according to the following format: `ProcessName.SWI_ProcessName`. The process types, which do not follow this standard, are listed with their full name.

Table 2 CMCO Processes Names

Component	Process Name
diafw	DiaFw_InstallerProc
dns	DnsTransportProc, DnsApplInstallProc
icmp	ICMP_OamInstallerProc, ICMP_ControllerProc, ICMP_Traffic_Proc
numana	NUMANA_Proc, NUMAN A_ConfigSyncProc, NUMANA_OamInstallerProc
numnorm	IpmmNumNormOamInsta llerProc, Ipmm_Numb erNormalization_Proc, IpmmNumNormConfigSyncProc
carsel	CARSEL_CsProc, CARSEL_DsaPr oc, CARSEL_ConfigSyncProc, CARSEL_OamInstallerProc
http	HTTP_TransportProc
sip	SIP_ServerControllerProc, SIP_ProcessorCensorProc, SIP_TcpServerProc, SIP_MsgParsingProc, SIP_UdpServerProc, SIP_TcpInboundSessionProc, SIP_TcpOutboundSessionProc
netmon	NETMON_MonitorProc





7 CMCO Components and Dependencies

The CMCO components and the dependencies are listed in Table 3.

Table 3 CMCO Components and Dependencies

Component	Dependencies
diabh	diafw
diaon	diafw
numnorm	parsers
carsel	parsers, numana
http	parsers, dns, icmp
sip	parsers, dns, icmp
netmon	parsers, dns, icmp, sip