

# Data Collection Guideline for IPWorks

OPERATING INSTRUCT

**Copyright**

© Ericsson AB 2017, 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

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Prerequisites	1
1.2	Related Information	1
<b>2</b>	<b>Workflow</b>	<b>3</b>
<b>3</b>	<b>Mandatory Data</b>	<b>5</b>
3.1	Data Collection	5
<b>4</b>	<b>Data Collected Based on Specific Problem Types</b>	<b>9</b>
4.1	Backup	9
4.2	Storage Server	10
4.3	Server Manager	12
4.4	DNS and ASDNS	14
4.5	ENUM	17
4.6	MySQL NDB Cluster	20
4.7	Radius AAA	23
4.8	EPC AAA	27
4.9	DHCPv4	29
4.10	SS7	31
	<b>Reference List</b>	<b>33</b>





# 1 Introduction

The purpose of this document is to instruct what troubleshooting data is to be collected and enclosed in a Customer Service Request (CSR) in case a problem is experienced with IPWorks.

This document is applicable for the following product release:

- IPWorks 1

The IPWorks is running on the following reference platform/component:

- Ericsson Cloud Execution Environment (CEE) based on BSP 8100

## 1.1 Prerequisites

This section describes the prerequisites for performing the data collection procedure.

### 1.1.1 Personnel

The personnel performing the troubleshooting must fulfill the following prerequisites:

- Possess IP network knowledge on an advanced level
- Have practical experience in the operation of the system
- Have administrator level access to the network elements

### 1.1.2 Documents

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

- IPWorks Troubleshooting Guideline

## 1.2 Related Information

Definition and explanation of acronyms and terminology, trademark information, and typographic conventions can be found in the following documents:

- Glossary of Terms and Acronyms
- Trademark Information



## — Typographic Conventions



## 2 Workflow

The workflow for collecting troubleshooting data is as follows:

1. Collect mandatory data that is needed in connection with any problems experienced. Go to Section 3 on page 5.
2. Collect specific data based on the type of problem that is experienced. Go to Section 4 on page 9.
3. Send the CSR to the next level of support for further investigation.







## 3 Mandatory Data

The data described in this section must always be included in a CSR.

The following items must be considered as mandatory when composing a CSR:

- Hardware platform and specifications
- Cluster configuration deployment
- Software information
- Reproduction steps

### 3.1 Data Collection

This section contains the information that is required to troubleshoot all general problems of a component.

**Note:** This section assumes that logging is enabled by ECLI. IPWorks data collection function will collect the logs about DNS, ASDNS, ENUM, SS, AAA, and MySQL. If logging is not enabled, the log information collected is useless for troubleshooting. When it is possible to reproduce steps, enable logging by ECLI first and collect the log information after reproducing steps. To get the detail about enabling logging by ECLI, refer to the document IPWorks Initial Configuration.

---

---

#### Caution!

The commands specified in this section must only be used after consulting Ericsson Technical Support personnel.

---

---

To collect data, perform the following steps by using **root** user:

1. Determine the data that needs to be collected.
  - **Which application?** For example, DNS.
  - **Which type of information?** For example, configuration information, log information.

2. Collect the data.

IPWorks provides two methods for data collection.

- Collect the data automatically by using a data collection tool at the SCs.



```
SC-X:~ # cd /opt/ipworks/common/scripts
```

```
SC-X:~ # bash ipwdatacollection.sh all|component1[,component2,...componentN]|-h
```

For example:

```
SC-X:~ #bash ipwdatacollection.sh
```

```
SC-X:~ #bash ipwdatacollection.sh all
```

```
SC-X:~ #bash ipwdatacollection.sh ipw,ss,dns
```

```
SC-X:~ #bash ipwdatacollection.sh ipw,ss,aaa
```

```
SC-X:~ #bash ipwdatacollection.sh cc,os,ipw,ss,dns,enum,ndb
```

Where:

- all: Collects information for all components.
- componentN: Specifies following components:
  - CC (Base software)
  - OS (Operation System info and disk/mem/networks etc)
  - SS (Storage Server)
  - DNS (Domain Name System and ASDNS)
  - ENUM (E.164 telephone Number Mapping)
  - AAA (Authentication, Authorization, Accounting)
  - DHCP (Dynamic Host Configuration Protocol)
  - NDB (MySQL Ndb)
  - IPW (All IPWorks applications log and configuration)
- -h: Prints manual. The output is the online help for this script.

The result file named as `ipworks-<component-list|all>-<Timestamp>.tar.gz` is stored in the folder `/cluster/storage/no-backup/ipworks/datacollection/`.

- Collect the data manually.

Follow Section 4 on page 9.

Check the disk usage, it is recommended to move the data collection result to other place if needed.



3. Send the files to the next level of support.





## 4 Data Collected Based on Specific Problem Types

This section describes the data to be included in a CSR, depending on what type of problem is experienced. It can be sent to Ericsson for further investigation in case that the problem cannot be solved locally. The General Problems section contains the information that is required to troubleshoot all problems of a component. The Specific Problems section contains the information that is required to troubleshoot a specific problem in addition to the general information in section General Problems.

**Note:** This section assumes that logging is enabled by ECLI. IPWorks data collection function will collect the logs about DNS, ASDNS, ENUM, SS, AAA, and MySQL. If logging is not enabled, the log information collected is useless for troubleshooting. When it is possible to reproduce steps, enable logging by ECLI first and collect the log information after reproducing steps. To get the detail about enabling logging by ECLI, refer to the document IPWorks Initial Configuration.

---



---

### Caution!

The commands specified in this section must only be used after consulting Ericsson Technical Support personnel.

---

## 4.1 Backup

### 4.1.1 General Problems

#### — Configuration information

```
dn ManagedElement=<Node Name>,SystemFunctions=1,BrM=1,BrmBackup
Manager=USER_DATA
```

For more information, refer to the section Managed Object Model in [Backup and Restore](#).

#### — Log information

Collect the log for ipwbrf to debug backup fail. The log is stored at:

```
/cluster/storage/no-backup/ipworks/logs/<hostname>/ipwbrf.log
```

If ipwss is running on SC-1, ipwbrf will be backup at SC-2 and above log will also be generated at SC-2, and vice versa.



If ipwss is in shutdown status, the issue is decided by DRBD. If DRBD role in SC-1 is primary, above log will be generated at SC-1, and vice versa.

## 4.2 Storage Server

### 4.2.1 General Problems

If you want to debug issues about ipwss, collect below configuration and log files.

#### — Configuration information

- Configuration in ECIM:

```
# ssh <username>@<MIP_OAM_IP> -t -s cli
```

```
>dn ManagedElement=<Node Name>,IpworksFunction=1,IpworksCommonRoot=1,StorageServer=1
```

```
(IPWorksAAACCommonRoot=1)>show -v -r
```

- Configuration files:

```
/opt/ipworks/ss/etc/ipworks_ss.conf
```

#### — Log information

```
/cluster/storage/no-backup/ipworks/logs/<hostname>/ipworks_ss_<hostname>.log
```

```
/cluster/storage/no-backup/ipworks/logs/<hostname>/ss_amf_wrapper.log
```

#### — IPWorks CLI log information

```
/var/ipworks/logs/admin_ipworks_cli_<hostname>_<Timestamp>.cmdlog
```

**Note:** To retrieve the IPWorks CLI log, the CLI logging must be enabled before executing commands in the current session.

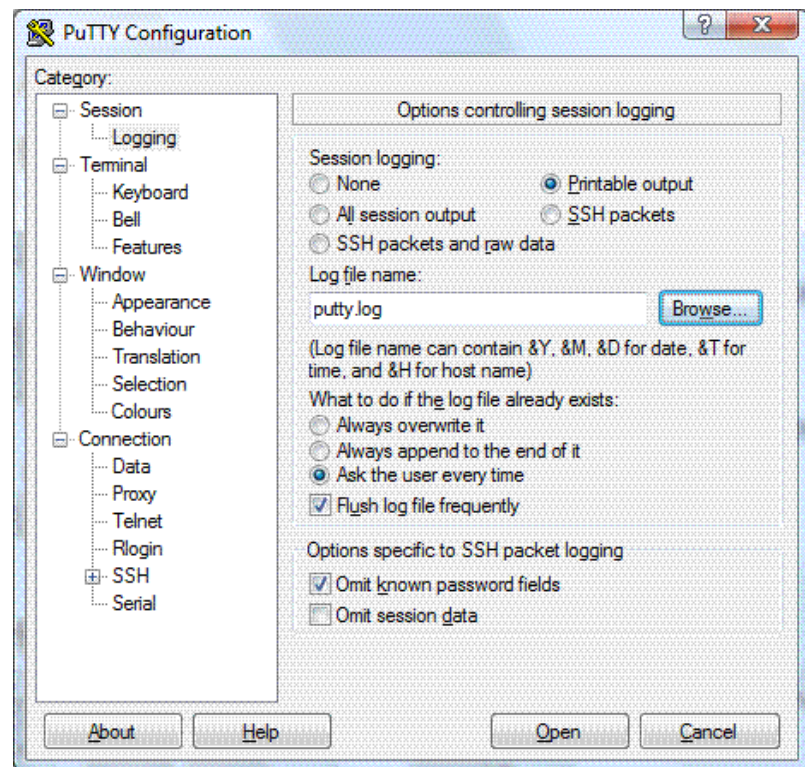
```
IPWorks> enablelog all
```

### 4.2.2 Specific Problems

#### 4.2.2.1 Function Failure

##### — Operation session log

For example, operation session log can be retrieved by enabling session logging in PuTTY. Configure PuTTY as shown in the following figure.



3

#### 4.2.2.2 Process Suspended

1. Get the SS process ID.

```
SC-X:~ # ps -ef |grep ipwss
```

```
0 S root      12096      1 0  80   0 - 1657895 futex_ 13:26 ?
00:00:47 java -DTCPPORT=9701 -DTCPPORT=9708 -DMULTICASTADDRESS=224.0.0.1
-DMULTICASTPORT=15663 -DBIND_INTERFACE_ADDRESS=169.254.100.23 -Djboss.server.name=ipwss
0 S root      24765 18143  0  80   0 - 2650 pipe_w 16:27 pts/0    00:00:00 grep ipwss
```

The process ID is 12096.

2. Use the gdb tool to collect information of the process.

```
SC-X:~ # gdb -p <PID>
```

Where: <PID> represents the process ID got from the previous step.

3. Display CPU usage, memory usage and runtime.

```
SC-X:~ # top
```



### 4.2.2.3 Memory Leak

Monitor and collect memory information. Issue the command periodically once the problem occurs.

- SC-X:~ # **top**
- or
- SC-X:~ # **top -p <PID>**

## 4.3 Server Manager

### 4.3.1 General Problems

If you want to debug the issues about server manager, collect below configuration and log files.

- Configuration information

/etc/ipworks/ipworks\_<\*>sm.conf

**Where:** <\*> is dns , asdnsmon, aaa or dhcpv4.

- Log information

For Server Manager of DNS and ASDNS, the following log files are included:

- /cluster/storage/no-backup/ipworks/logs/<PL hostname>/<\*>sm.log

Where: The <\*> is the dns or asdnsmon.

The log is disabled by default, to enable the SM logging, use the following command in ECLI:

```
# ssh <username>@<MIP_OAM_IP> -t -s cli
>ManagedElement=<Node Name>,IpworksFunction=1,IpworksDnsRoot=1,
<*>Server=1,<*>Sm=1,<*>SmLog=1
(<*>SmLog=1)> config
(config-<*>SmLog=1)>level=LOG_LEVEL_DEBUG
(config-<*>SmLog=1)>commit
```

**Where:** <\*> is Dns or Asdns (case sensitive).

- /cluster/storage/no-backup/ipworks/logs/<PL hostname>/<\*>\_sm\_wrapper.log
- /cluster/storage/no-backup/ipworks/logs/<PL hostname>ipworks\_<\*>\_sm\_wrapper.log





**Where:** <\*> is asdnsmom or dns.

For Server Manager of AAA, the following log files are included:

- /cluster/storage/no-backup/ipworks/logs/<PL hostname>/aaa sm.log

The log is disabled by default, to enable the SM logging, use the following command in ECLI:

```
# ssh <username>@<MIP_OAM_IP> -t -s cli
>config
(config)>ManagedElement=<Node Name>,IpworksFunction=1,IPWorksAAARoot=1,
IPWorksAAACommonRoot=1,AAAServerManager=1,level=LOG_LEVEL_DEBUG
(config)>commit
```

For Server Manager of DHCP, the following log files are included:

- /cluster/storage/no-backup/ipworks/logs/<PL hostname>/dhc pv4sm.log

The log is disabled by default, to enable the SM logging, use the following command in ECLI:

```
# ssh <username>@<MIP_OAM_IP> -t -s cli
>config
(config)>ManagedElement=<Node Name>,IpworksFunction=1,IPWorksDHCPRoot=1,
DHCPManager=1,level=LOG_LEVEL_DEBUG
(config)>commit
```

#### — Storage Server log information

```
/cluster/storage/no-backup/ipworks/logs/<hostname>/ipworks_ss_
<SC hostname>.log
```

## 4.3.2 Specific Problems

### 4.3.2.1 Process Suspended

1. Get the SM process ID.

- For DNS SM: PL-X:~ # ps -ef |grep ipwdnssm
- For ASDNS SM: PL-X:~ # ps -ef |grep ipwasdnsmomsm
- For AAA SM: PL-X:~ # ps -ef |grep ipwaaasm
- For DHCP SM: PL-X:~ # ps -ef |grep ipwdhcpv4sm

2. Use the gdb tool to collect information of the process.

```
PL-X:~ # gdb -p <PID>
```



3. Display CPU usage, memory usage and runtime.

```
PL-X:~ # top
```

#### 4.3.2.2

#### Memory Leak

Monitor and collect memory information. Issue the command periodically once the problem occurs.

```
— PL-X:~ # top
```

```
— PL-X:~ # top -p <PID>
```

## 4.4

## DNS and ASDNS

### 4.4.1

#### General Problems

If you want to debug the issues about DNS and ASDNS, collect the below configuration and log files.

— Configuration information:

- Configuration in ECIM:

```
# ssh <username>@<MIP_OAM_IP> -t -s cli
```

```
>dn ManagedElement=<Node Name>,IpworksFunction=1,Ipworks
DnsRoot=1,DnsServer=1
```

```
(DnsServer=1)>show -v -r
```

```
(DnsServer=1)>..,AsdnsServer=1
```

```
(AsdnsServer=1)>show -v -r
```

- Configuration files:

```
/etc/ipworks/<PL hostname>/*
```

— Log information for DNS:

- /cluster/storage/no-backup/ipworks/logs/<PL hostname>/ipworks\_dns.log

The log file is generated by BIND service. If the log is disabled, enable it through ECLI.



```
# ssh <username>@<MIP_OAM_IP> -t -s cli
>config
(config)>ManagedElement=<Node Name>,IpworksFunction=1,IpworksDnsRoot=1,
BindService=1,DnsLog=1,level=DNS_LOG_LEVEL_DEBUG
(config)>commit
```

- /cluster/storage/no-backup/ipworks/logs/<PL hostname>/ipworks\_dns\_sm\_wrapper.log

The log file is generated by the service dns-sm. If the log is disabled, enable it through ECLI.

```
# ssh <username>@<MIP_OAM_IP> -t -s cli
>config
(config)>ManagedElement=<Node Name>,IpworksFunction=1,IpworksDnsRoot=1,
DnsSm=1,DnsSmLog=1,Level=LOG_LEVEL_DEBUG
(config)>commit
```

- /cluster/storage/no-backup/ipworks/logs/<PL hostname>/ipworks\_dns\_trans.log

The log file is generated by BIND service. To enable the log, do the following:

```
# ssh <username>@<MIP_OAM_IP> -t -s cli
>config
(config)>ManagedElement=<Node Name>,IpworksFunction=1,IpworksDnsRoot=1,
BindService=1,DnsTransLog=1
(config)>commit
```

- /cluster/storage/no-backup/coremw/var/log/<PL hostname>/dns\_wrapper.log

The log file is generated by the service dns-wrapper by ipworks\_dns\_wrapper.sh script, and only logs when DNS is instanced and cleaned up.

- /cluster/storage/no-backup/coremw/var/log/<PL hostname>/ipworks\_dns\_cmw.log

The log file is generated by the service named to record debug logs of the service dns regarding CoreMW events/failures.

#### — Log Information for ASDNS:

- /cluster/storage/no-backup/ipworks/logs/<PL hostname>/ipworks\_asdnsmon.log

The log file is generated by BIND service. If the log is disabled, enable it through ECLI.



```
# ssh <username>@<MIP_OAM_IP> -t -s cli
>config
(config)>ManagedElement=<Node Name>,IpworksFunction=1,IpworksDnsRoot=1,
AsdnsServer=1,AsdnsMonitor=1,AsdnsMonLog=1,level=LOG_LEVEL_INFO
(config)>commit
```

- /cluster/storage/no-backup/ipworks/logs/<PL hostname>/ipworks\_asdnsmon\_trans.log

The log file is generated by asdnsmon. To enable the log, do the following:

```
# ssh <username>@<MIP_OAM_IP> -t -s cli
>config
(config)>ManagedElement=<Node Name>,IpworksFunction=1,IpworksDnsRoot=1,
AsdnsServer=1,AsdnsMonitor=1,AsdnsMonTransLog=1
(config)>commit
```

- /cluster/storage/no-backup/coremw/var/log/<PL hostname>/asdns\_coremw.log

The log file is generated by the service asdnsmon to record the debug log of the service asdns regarding CoreMW events/failures.

#### — System log information

<PL hostname>: /var/log/messages

#### — QPS

#### — Dump database

```
SC-X:~# /opt/ipworks/mysql/mysql/bin/mysqldump -P 3307
--protocol=tcp --no-create-info --opt ipworks > db_dump.sql
```

## 4.4.2 Specific Problems

### 4.4.2.1 Function Failure

For example: Dig arecord to port 5300

Get use information of port 5300 through the following command:

```
PL-X:~# netstat -apn|grep 5300
```

### 4.4.2.2 Process Suspended

1. Get the process ID of DNS and ASDNS.

- For DNS: PL-X:~# ps -ef |grep named| grep -v grep

```
4 S root      5780      1  2  80   0 - 662629 sigsus 11:48 ?    00:02:34 /opt/ipworks/dns/u
```



- For ASDNS: PL-X:~# **ps -ef |grep asdnsmon | grep -v grep**

```
root      4042      1  0 13:37 ?        00:00:00 /opt/ipworks/asdnsmon_sm/bin/amf_wrapp
root      4079      1 99 13:37 ?        00:00:04 java -DApp=ipwasdnsmonsm
...
```

The process about asdnsmon is 4079.

2. Use gdb tool to collection information of the process.

PL-X:~# **gdb -p <PID>**

Where: <PID> represents the process ID got from the previous step.

3. Display CPU usage, memory usage and runtime.

PL-X:~# **top**

#### 4.4.2.3 Memory Leak

Monitor and collect memory information. Issue the command periodically once the problem occurs.

— PL-X:~# **top**

— PL-X:~# **top -p <PID>**

#### 4.4.2.4 Crash

Core dump file: cluster/dumps/<service>.<PID>.<hostname>.core

Use the gdb tool to check the stack contents. For example:

```
# gdb /opt/ipworks/dns/usr/bin/named named.9898.PL-4.core
```

```
(gdb) info threads
```

```
(gdb) thread <thread No.>
```

```
(gdb) bt
```

```
.....
```

```
(gdb) q
```

## 4.5 ENUM

### 4.5.1 General Problems

If you want to debug the issues about ENUM, collect the below configuration and log files.



#### — Configuration information

- Configuration in ECIM:

```
# ssh <username>@<MIP_OAM_IP> -t -s cli

>dn ManagedElement=<Node Name>,IpworksFunction=1,Ipworks
DnsRoot=1,IpworksEnumRoot=1

(IpworksEnumRoot=1)>show -v -r
```

- Configuration files:

Below folder can be fetched from all blades. They are linked to a same folder in /cluster.

```
/etc/ipworks/enum/*
```

#### — Log information for ENUM and ERH over LDAP

- /cluster/storage/no-backup/ipworks/logs/<PL hostname>/ipwen  
um.log

The log file is generated by ENUM service. If the log is disabled, enable it through ECLI. The below example shows how to enable log generation in case of EnumServer=1.

```
# # ssh <username>@<MIP_OAM_IP> -t -s cli
>config
(config)>ManagedElement=<Node Name>,IpworksFunction=1,IpworksDnsRoot=1,
IpworksEnumRoot=1,EnumServer=1,Log=1,level=LOG_LEVEL_DEBUG
(config)>commit
```

#### — Log information for ERH over SS7

- /cluster/storage/no-backup/ipworks/logs/<PL hostname>/ipwe  
rh.log

The log file is generated by ERH. If the log is disabled, enable it through ECLI. The below example shows how to enable log generation in case of EnumServer=1.

```
# ssh <username>@<MIP_OAM_IP> -t -s cli
>config
(config)>ManagedElement=<Node Name>,IpworksFunction=1,IpworksDnsRoot=1,
IpworksEnumRoot=1,EnumServer=1,Erh=1,ErhSs7=1,Log=1,level=LOG_LEVEL_INFO
(config)>commit
```

#### — Log information for ENUM FE Sync

- /cluster/storage/no-backup/ipworks/logs/<PL hostname>/ipwor  
ks\_enumfe\_wrapper.log

The log file is generated by AMF wrapper of ENUM FE Sync.



- `/cluster/storage/no-backup/ipworks/logs/<PL hostname>/ipworks_fesync.log`

The log file is generated by ENUM FE Sync. If the log is disabled, enable it through ECLI as below:

```
# ssh <username>@<MIP_OAM_IP> -t -s cli
> config
(config)> ManagedElement=<Node Name>,IpworksFunction=1,IpworksDnsRoot=1,
IpworksEnumRoot=1,EnumFE=1,ENUMFELog=1,level=LOG_LEVEL_DEBUG
(config)>commit
```

— System log information

<PL hostname>: `/var/log/messages`

## 4.5.2 Specific Problems

### 4.5.2.1 Function Failure

For example: Dig arecord to port 53

Get use information of port 53 through the following command:

PL-X:~# `netstat -apn|grep 53`

### 4.5.2.2 Process Suspended

1. Get process ID of the ENUM or ENUM FE Sync.

- For ENUM: PL-X:~ # `ps -ef |grep enum`
- For ENUM FE Sync: PL-X:~ # `ps -ef | grep axis2`

2. Use gdb tool to get process information.

PL-X:~ # `gdb -p <PID>`

3. Display CPU usage, memory usage and runtime.

PL-X:~ # `top`

### 4.5.2.3 Memory Leak

Monitor and collect memory information. Issue the command periodically once the problem occurs.

— PL-X:~ # `top`

— PL-X:~ # `top -p <PID>`



#### 4.5.2.4 Crash

ENUM core dump file: `/cluster/dumps/enum.<PID>.<hostname>.core`

ENUM FE Sync core dump file: `/cluster/dumps/java.<PID>.<hostname>.core`

Check the stack contents through the gdb tool. For example:

```
PL-X:~ # gdb /opt/ipworks/enum/bin/ipwenum enum.7768.PL-3.core
```

```
(gdb)info threads
```

```
(gdb)thread <thread No.>
```

```
(gdb)bt
```

```
.....
```

```
(gdb)q
```

## 4.6 MySQL NDB Cluster

### 4.6.1 General Problems

If you want to debug the issues about NDB, collect the below configuration and log files

- Configuration information

```
/cluster/home/ipworks/mysql/confs
```

- Log information for NDB cluster node

**Note:** Collect the information on each SC.

Log for Management Node:

```
/local/ipworks/mysql-cluster/mgmnode/mgm.log.<fileNo>
```

Log for Data Node:

```
/local/ipworks/mysql-cluster/datanode/ndb_<nodeId>_out.log.<fileNo>
```

Log for SQL Node:

```
/local/ipworks/mysql-cluster/sqlnode/sqlnode.err.<fileNo>
```

- System log information

```
<SC hostname>: /var/log/messages
```





## 4.6.2 Specific Problems

### 4.6.2.1 Check Process Use Condition

```
SC-X:~ # top -b | head -n15
```

### 4.6.2.2 Get NDB Error Report

```
SC-X:~ # /opt/ipworks/mysql/mysql/bin/ndb_error_reporter
/cluster/home/ipworks/etc/mysql/confs/ipworks_mgm.conf root
```

You will get a tar file (named as "ndb\_error\_report\_<Timestamp>.tar.bz2") in the current directory.

### 4.6.2.3 Get "ndb\_mgmd" Process Information

**Note:** Get the "ndb\_mgmd" process id first and then use gdb to get the threads trace.

1. Check whether "ndb\_mgmd" is running:

```
SC-X:~ # ps -ef |grep "ndb_mgmd" |grep -Ev "grep" | awk '{print $2}'
```

For example:

```
SC-X:~ #ps -ef |grep "ndb_mgmd" |grep -Ev "grep" | awk '{print $2}'
```

```
19266
```

2. Get the "gdb" information from " ndb\_mgmd ":

```
SC-X:~#gdb /opt/ipworks/mysql/mysql//sbin/ndb_mgmd <PID>
```

For example:

```
SC-X:~#gdb /opt/ipworks/mysql/mysql//sbin/ndb_mgmd 19266
```

3. Enter "thread apply all bt" under (gdb) prompt:

```
(gdb) thread apply all bt
```

```
.....
```

```
(gdb)q
```

```
(gdb)y
```



#### 4.6.2.4 Print "ndb\_mgmd" Full Config

```
SC-X:~ # /opt/ipworks/mysql/mysql/sbin/ndb_mgmd --print-full-config  
--ndb-nodeid=<nodeid>
```

#### 4.6.2.5 Get "ndbmtd" Process Information

**Note:** Get the "ndbmtd" process id first and then use gdb to get the threads trace.

1. Check whether "ndbmtd" is running:

```
SC-X:~ #ps -ef | grep ndbmtd | grep -Ev grep | awk '{if( $3 !=  
1) {print $0} }' | awk '{print $2}'
```

For example:

```
SC-X:~ #ps -ef | grep ndbmtd | grep -Ev grep | awk '{if( $3 !=  
1) {print $0} }' | awk '{print $2}'
```

21050

2. Get the "gdb" information from "ndbmtd":

```
SC-X:~ #gdb /opt/ipworks/mysql/mysql/sbin/ndbmtd <PID>
```

For example:

```
SC-X:~ #gdb /opt/ipworks/mysql/mysql/sbin/ndbmtd 21050
```

3. Enter "thread apply all bt" under (gdb) prompt:

```
(gdb) thread apply all bt
```

.....

```
(gdb)q
```

```
(gdb)y
```

#### 4.6.2.6 Get "mysqld" Process Information

**Note:** Get the "mysqld" process id first and then use gdb to get the threads trace.

1. Get the "mysqld" process id:

```
SC-X:~ #ps -ef | grep mysqld | grep -Ev grep | awk '{if( $3 !=  
1) {print $0} }' | awk '{print $2}'
```

For example:

```
SC-X:~ #ps -ef | grep mysqld | grep -Ev grep | awk '{if( $3 !=  
1) {print $0} }' | awk '{print $2}'
```



32683

2. Get the "gdb" information from "mysqld":

```
SC-X:~ #gdb /opt/ipworks/mysql/mysql//sbin/mysqld <PID>
```

For example:

```
SC-X:~ #gdb /opt/ipworks/mysql/mysql//sbin/mysqld 32683
```

3. Enter "thread apply all bt" under (gdb) prompt:

```
(gdb)thread apply all bt
```

```
.....
```

```
(gdb)q
```

```
(gdb)y
```

## 4.7 Radius AAA

### 4.7.1 General Problems

If you want to debug the issues about AAA, collect the below configuration and log files.

— Configuration information

- Configuration in ECIM:

```
# ssh <username>@<MIP_OAM_IP> -t -s cli
```

```
>dn ManagedElement=<Node Name>,IpworksFunction=1,IPWorksAAARoot=1,IPWorksAAACCommonRoot=1 (IPWorksAAACCommonRoot=1)>show -v -r
```

```
(IPWorksAAACCommonRoot=1)>..,IPWorksRadiusAAARoot=1
```

```
(IPWorksRadiusAAARoot=1)>show -v -r
```

- Configuration files:

The following folders can be fetched from all blades. They are linked to a same folder in /cluster.

— Radius AAA Configuration File:

```
/etc/ipworks/aaa_radius/*
```

```
/etc/ipworks/<PL hostname>/aaa_radius/*
```



— CSV Engine Configuration Files:

/etc/ipworks/aaa\_acct\_engine.conf

— Log information

- Log information for Radius AAA:

— Radius Stack Logs:

/cluster/storage/no-backup/ipworks/logs/<PL  
hostname>/aaa\_radius\_stack.log

— Radius Backend logs:

/cluster/storage/no-backup/ipworks/logs/<PL  
hostname>/aaa\_radius\_backend.log

The log file is generated by AAA RADIUS service. If the log is disabled, enable it through ECLI. The below example shows how to enable log generation if AAAServer=PL-3.

```
# ssh <username>@<MIP_OAM_IP> -t -s cli
```

```
>configure
```

```
(config)>dn ManagedElement=<Node Name>,IpworksFunction=1,IPWorksAAARoot=1, IPWorksAAACCommonRoot=1,AAAServer=PL-3,LogManagement=1
```

```
(config-LogManagement=1)>IPWorksLog=AAA_RADIUS_STACK,level=LOG_LEVEL_DEBUG
```

```
(config-LogManagement=1)>IPWorksLog=AAA_RADIUS_BACKEND,level=LOG_LEVEL_DEBUG
```

```
(config-LogManagement=1)>commit
```

- Log information for CSV Engine:

CSV Engine Logs:

/cluster/storage/no-backup/ipworks/logs/<SC hostname>/aaa\_radius\_csvengine.log

The log file is generated by CSV Engine. If the log is disabled, enable it through ECLI. The below example shows how to enable log generation.

```
# ssh <username>@<MIP_OAM_IP> -t -s cli
```

```
>configure
```



```
(config)>dn ManagedElement=<Node Name>,IpworksFunction=1,IP
WorksAAARoot=1,IPWorksRadiusAAARoot=1,RadiusAAAService=1,Ac
countingService=1,CSVEngineCommon=1
```

```
(config-CSVEngineCommon=1)>level=LOG_LEVEL_DEBUG
```

```
(config-CSVEngineCommon=1)>commit
```

— Accounting CSV Files:

```
/cluster/ipworks/cdr/payment/*
```

```
/cluster/ipworks/cdr/temp/*
```

**Note:** If the process a3csvengine is started, the default directory of the CSV files is /cluster/ipworks/cdr/payment/ and the default temp directory is /cluster/ipworks/cdr/temp/. Then they could be changed through ECLI.

```
# ssh <username>@<MIP_OAM_IP> -t -s cli
```

```
>configure
```

```
(config)>dn ManagedElement=<Node Name>,IpworksFunction=1
,IPWorksAAARoot=1,IPWorksRadiusAAARoot=1,RadiusAAAServic
e=1,AccountingService=1,CsvGenerateMethod=1
```

```
(config-CsvGenerateMethod=1)>csvGeneratorFileDir=<Dir
ectory of CSV Files>
```

```
(config-CsvGenerateMethod=1)>csvPaymentTempFileDir=<Dir
ectory of CSV Temp Files>
```

```
(config-CsvGenerateMethod=1)>commit
```

— Dump database on SC blade:

```
# /opt/ipworks/mysql/mysql/bin/mysqldump -P 3307 -h ipw_sql
--protocol=tcp --no-create-info --opt ipworks > db_dump.sql
```

— Show NDB status on SC blade:

```
# /etc/init.d/ipworks.mysql show-status
```

— SS7 configuration and logs.

Refer to Section 4.10 SS7 on page 31.



## 4.7.2 Specific Problems

### 4.7.2.1 Process Suspended

1. Get process ID of the RADIUS AAA and CSV Engine.

```
SC-X:~ # ps -ef |grep a3csvengine
```

```
PL-X:~ # ps -ef |grep a3radiusd
```

```
PL-X:~ # ps -ef |grep a3backend
```

2. Use gdb tool to get process information.

```
PL-X:~ # gdb -p <PID>
```

3. Display CPU usage, memory usage and runtime.

```
PL-X:~ # top
```

### 4.7.2.2 Memory Leak

Monitor and collect memory information. Issue the command periodically once the problem occurs.

— PL-X:~ # top

— PL-X:~ # top -p <PID>

### 4.7.2.3 Crash

Radius Stack dump file: `/cluster/dumps/a3radiusd.<PID>.<hostname>.core`

Radius Backend dump file: `/cluster/dumps/a3backend.<PID>.<hostname>.core`

CSV Engine dump file: `/cluster/dumps/a3csvengine.<PID>.<hostname>.core`

For example, AAA Radius Stack with the PID 1996 on PL-3 crashed. A core file `a3radiusd.1996.PL-3.core` is created in the above directory.

Use the gdb tool to check the stack contents:

```
PL-X:~ # gdb /opt/ipworks/aaa_radius/stack/bin/a3radiusd  
a3radiusd.1996.PL-3.core
```

```
(gdb)info threads
```

```
(gdb)thread <thread No.>
```

```
(gdb)bt
```



.....

(gdb)q

Similarly, execute the following commands to check the stack contents for backend and csvengine core dump:

```
PL-X:~ # gdb /opt/ipworks/aaa_radius/backend/bin/a3backend
a3backend.<PID>.<hostname>.core
```

```
SC-X:~ # gdb /opt/ipworks/aaa_radius/csvengine/bin/a3csvengine
a3csvengine.<PID>.<hostname>.core
```

## 4.8 EPC AAA

### 4.8.1 General Problems

If you want to debug the issues about AAA, collect the below configuration and log files.

#### — Configuration information

- Configuration in ECIM:

```
# ssh <username>@<MIP_OAM_IP> -t -s cli
```

```
>dn ManagedElement=<Node Name>,IpworksFunction=1,IPWorksAAARoot=1,IPWorksAAACCommonRoot=1
```

```
(IPWorksAAACCommonRoot=1)>show -v -r
```

```
(IPWorksAAACCommonRoot=1)>..,IPWorksDiameterAAARoot=1
```

```
(IPWorksDiameterAAARoot=1)>show -v -r
```

- Configuration files:

The following folders can be fetched from all blades. They are linked to a same folder in /cluster.

C-Diameter Configuration File:

```
/storage/system/config/lde/csm/finalized/config/initial/CDiameter-CXP9034135/cdiameter_cdfmodel.xml
```

AAA Server Configuration Files:

```
/etc/ipworks/aaa_diameter/*
```

#### — Log information



- Log information for EPC AAA:

```
/cluster/storage/no-backup/ipworks/logs/<PL hostname>/aaa_diameter_server.log
```

The log file is generated by AAA service. If the log is disabled, enable it through ECLI. The below example shows how to enable log generation if AAAServer=PL-3.

```
# ssh <username>@<MIP_OAM_IP> -t -s cli
>config
(config)>ManagedElement=<Node Name>,IpworksFunction=1,IPWorksAAARoot=1,
IPWorksAAACCommonRoot=1,AAAServer=PL-3,LogManagement=1,IPWorksLog=AAA_DIA
(config)>commit
```

- Log information for C-diameter:

Log files shall be collected from any SC or PL blade. Log on to the blade and get the log files:

```
# ssh <username>@<MIP OAM IP>

# cd /storage/no-backup/diaacc/log
```

- System log information:

```
# ssh <username>@<MIP OAM IP>

# ssh <PL hostname>

<PL hostname>: /var/log/messages
```

## 4.8.2 Specific Problems

### 4.8.2.1 Process Suspended

1. Get process ID of the AAA.

```
PL-X:~ # ps -ef |grep ipwa3d
```

2. Use gdb tool to get process information.

```
PL-X:~ # gdb -p <PID>
```

3. Display CPU usage, memory usage and runtime.

```
PL-X:~ # top
```





#### 4.8.2.2 Memory Leak

Monitor and collect memory information. Issue the command periodically once the problem occurs.

- PL-X:~ # **top**
- PL-X:~ # **top -p <PID>**

#### 4.8.2.3 Crash

AAA dump file: /cluster/dumps/ipwa3d.<PID>.<hostname>.core

Check the stack contents through the gdb tool. For example:

```
PL-X:~ # gdb /opt/ipworks/aaa_diameter/bin/ipwa3d ipwa3d.15947.PL-3.core

(gdb)info threads

(gdb)thread <thread No.>

(gdb)bt

.....

(gdb)q
```

### 4.9 DHCPv4

#### 4.9.1 General Problems

If you want to debug the issues about DHCP, collect the below configuration and log files.

- Configuration information

- Configuration in ECIM:

```
# ssh <username>@<MIP_OAM_IP> -t -s cli

>dn ManagedElement=<Node Name>,IpworksFunction=1,IPWorksDHCPRoot=1

(IPWorksDHCPRoot=1)>show -v -r
```

- Configuration files:

Below folder can be fetched from all blades. They are linked to a same folder in /cluster.



```
/etc/ipworks/<PL hostname>/dhcp/*
```

#### — Log information

- Log information for DHCPv4:

```
/cluster/storage/no-backup/ipworks/logs/<PL hostname>/ipworks_dhcpv4.log
```

The log file is generated by DHCP service. If the log is disabled, enable it through ECLI. The below example shows how to enable log generation in case of DHCPv4Server=PL-3.

```
# ssh <username>@<MIP_OAM_IP> -t -s cli
>config
(config)>ManagedElement=<Node Name>,IpworksFunction=1,IPWorksDHCPv4Root=1,DHCPv4Server=PL-3
(config)>commit
```

- Log information for DHCPv4ServerManager:

```
/cluster/storage/no-backup/ipworks/logs/<PL hostname>/dhcpv4sm.log
```

The log file is generated by DHCPv4ServerManager. If the log is disabled, enable it through ECLI. The below example shows how to enable log generation in case of DHCPv4ServerManager=1.

```
# ssh <username>@<MIP_OAM_IP> -t -s cli
>config
(config)>ManagedElement=<Node Name>,IpworksFunction=1,IPWorksDHCPv4Root=1,DHCPv4ServerManager=1
(config)>commit
```

- System log information:

```
<PL hostname>: /var/log/messages
```

## 4.9.2 Specific Problems

### 4.9.2.1 Function Failure

Get use information of port 67 through the following command:

```
PL-X:~ # netstat -apn|grep 67
```

### 4.9.2.2 Process Suspended

Get process ID of the DHCP.

```
For DHCP: PL-X:~ # ps -ef |grep dhcpcd
```



Use gdb tool to get process information.

```
PL-X:~ # gdb -p <PID>
```

Display CPU usage, memory usage and runtime.

```
PL-X:~ # top
```

#### 4.9.2.3 Memory Leak

Monitor and collect memory information. Issue the command periodically once the problem occurs.

```
PL-X:~ # top
```

```
PL-X:~ # top -p <PID>
```

#### 4.9.2.4 Crash

DHCP core dump file: /cluster/dumps/dhcpd.<PID>.<hostname>.core

Check the stack contents through the gdb tool. For example:

```
PL-X:~ # gdb /opt/ipworks/dhcp/usr/bin/dhcpd dhcpd.7768.PL-3.core
```

```
(gdb)info threads
```

```
(gdb)thread <thread No.>
```

```
(gdb)bt
```

```
.....
```

```
(gdb)q
```

### 4.10 SS7

#### 4.10.1 General Problems

If you want to debug the issues about SS7, collect the below configuration and log files.

— Configuration information

The following folders can be fetched from all blades. They are linked to a folder in /cluster.

```
/storage/system/config/ss7caf-ana90137/etc/*
```



— Log information

`/storage/no-backup/ss7caf-ana90137/log/*`



## Reference List

### IPWorks Documents

- [1] Glossary of Terms and Acronyms
- [2] Trademark Information
- [3] Typographic Conventions
- [4] IPWorks Troubleshooting Guideline
- [5] Backup and Restore
- [6] IPWorks Initial Configuration, 5/1553-AVA 901 33/3 Uen
- [7] Data Collection Guideline for BSP, 6/1543-APP 111 01 Uen