

# IPWorks Application Counters in CUDB

## USER GUIDE

**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>Term Definitions</b>	<b>3</b>
<b>3</b>	<b>Measurements</b>	<b>5</b>
<b>4</b>	<b>Auto Installation of IPWorks Application Counters on CUDB Nodes</b>	<b>7</b>
<b>5</b>	<b>Manual Installation of IPWorks Application Counters on CUDB Nodes</b>	<b>11</b>
5.1	Installing Counters of IPWENUMFE	11
5.2	Installing Counters of IPWAAAFE	17
5.3	Installing Counters of IPWPKIFE	21
<b>6</b>	<b>Uninstallation of Application Counters on CUDB Nodes</b>	<b>27</b>
	<b>Reference List</b>	<b>29</b>





# 1 Introduction

This document describes the IPWorks application counters (also named as "measurements") that are collected in Centralized User Database (CUDB), and provides instructions to apply the IPWorks application counters to CUDB. The value for each counter is obtained through the subscriber data stored in CUDB that fulfill filtering criteria.

Two methods to apply the IPWorks application counters:

- Automatic installation of IPWorks application counters on CUDB (recommended), see Section 4 on page 7.
- Manual installation of IPWorks application counters on CUDB, see Section 5 on page 11.

## 1.1 Prerequisites

The personnel must have good knowledge in:

- Intermediate Linux and UNIX
- Concepts, terminologies, and telecommunication abbreviations, such as TCP/IP, PDN (Packet Data Network)

The following conditions must be met:

- The operator has the permission to the following server:
  - IPWorks
  - Jumpstart
  - CUDB
- IPWorks Front-End schema files have been loaded to the CUDB master node.

You can refer to the CUDB CPI CUDB Schema Conversion Tool.

## 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 Term Definitions

The terms used in the following sections are defined as follows:

<b>Measurement Name</b>	The name of the measurement being described.
<b>Measurement Type</b>	The four types of 3GPP standard measurements (counter, gauge, status inspection, and discrete event registration) used to report data.
<b>Counter</b>	<p>A counter is a measurement type that is used to report cumulative, incremental integer variables. An occurrence of an event increases the counter.</p> <p>For the details about the measurement type <b>Collection Method</b> and <b>Aggregation</b>, refer to the section Measurement Parameters in <i>IPWorks Performance Measurements</i>.</p>
<b>Performance Measurement Group</b>	Represents a logical grouping of Measurement Type objects so that these objects can be referred to as a group instead of individual objects.







### 3 Measurements

The measurements are identified by the group and the name of the counter. All measurements are status counters, which means that they show the value at a specific moment, and can be stepped up or down.

Four application counters for IPWorks Front End (FE) are listed in the following tables :

- IPWENUMFE including the attribute values for ENUMNAPTRCNT and ENUMFQDNCNT

Table 1 ENUMNAPTRCNT

Description	The number of total ENUM NAPTR Record provisioned.
Condition	The value is incremented for every ENUM NAPTR Record provisioned.
Aggregation	LAST_UPDATE
Collection	GAUGE
Result Type	Integer
Measurement Object Class	IPWENUMFE

Table 2 ENUMFQDNCNT

Description	The number of total ENUM FQDN Record provisioned.
Condition	The value is incremented for every ENUM FQDN Record provisioned.
Aggregation	LAST_UPDATE
Collection	GAUGE
Result Type	Integer
Measurement Object Class	IPWENUMFE

- IPWAAAFE

Table 3 AAAUSERCNT

Description	The number of AAA Radius user.
Condition	The counter is incremented for every aaa user added and decremented due to aaa users gone.
Aggregation	LAST_UPDATE
Collection	GAUGE



Result Type	Integer
Measurement Object Class	IPWAAAFE

— IPWPKIFE

Table 4 NSDSUSERCNT

Description	The number of PKI user (nsds user).
Condition	The counter is incremented for every nsds user added and decremented due to nsds users gone.
Aggregation	LAST_UPDATE
Collection	GAUGE
Result Type	Integer
Measurement Object Class	IPWPKIFE



## 4 Auto Installation of IPWorks Application Counters on CUDB Nodes

IPWorks provides the script `app_counters_ipworks.pl` to facilitate the installation of IPWorks application counters on CUDB.

For more information about the general guide of installing application counters, refer to CUDB Performance Guide, Reference [5].

To install IPWorks application counters by using the script, do the following:

1. Download the IPWorks Utility package from SW Gateway. For specific information, see the product release notes.
2. Unpack the Utility package to fetch the IPWorks application counters installation package `IPW_App_Counters.tar.gz`.

```
# tar zxvf <Utility package name>
```

For example:

```
# tar zxvf 19010-CXP9029034_2_Ux_<Revision Number>.tar.gz
```

The package `IPW_App_Counters.tar.gz` is included in the `frontend/app_counters` folder.

3. Find out all the CUDB nodes.

```
# ssh root@<CUDB_VIP_OAM>
```

```
<CUDB Node Name>#cudbSystemStatus
```

For example:

```
Node | 31 | 32
```

```
=====
```

```
PLDB ___|___M___|___S2_
```

```
DSG 1 ___|___M___|___S2_
```

```
Printing Detailed Replication Status for the Slave Replicas:
```

```
Node 31:
```

```
There are no Slave clusters
```

```
Node 32:
```

```
Replication in DSG0(Chan=2) .... Up -- Delay = 0.0 seconds, no. of pending
```

```
Replication in DSG1(Chan=2) .... Up -- Delay = 0.0 seconds, no. of pending
```

The command output shows that two CUDB nodes CUDB31 and CUDB32 exist.



The IPWorks application counters must be installed on all CUDB nodes. The following procedure guides you to install the counters on the CUDB nodes CUDB31 and CUDB32 .

4. Copy the package `IPW_App_Counters.tar.gz` to Jumpstart server, and then copy the package to the folder `/cluster` in all CUDB nodes.
5. Unpack the package `IPW_App_Counters.tar.gz` on all CUDB nodes (for example, on CUDB31 and CUDB32).

```
# ssh root@<CUDB31 OAM IP>
```

```
CUDB31# cd /cluster/
```

```
CUDB31# tar zvxf IPW_App_Counters.tar.gz
```

```
CUDB31# ssh root@<CUDB32 OAM IP>
```

```
CUDB32# cd /cluster/
```

```
CUDB32# tar zvxf IPW_App_Counters.tar.gz
```

6. Execute the application counter script on all CUDB nodes (for example, on CUDB31 and CUDB32).

```
# ssh root@<CUDB31 OAM IP>
```

```
CUDB31# cd /cluster/IPW_App_Counters
```

```
CUDB31# ./app_counters_ipworks.pl -i
```

You will be prompted for the application counter installation. During the installation, you can choose to install the IPWorks FE related counters.

For example:



IPWorks Applications Counters Installation, version Rev. 1.0.1

```

* PLDB detected...
-----
hostname is SC_2_1
* Checking Active OAM blade... [ ok ]
-----
* Reading System Hosts info... [ ok ]
-----
* Reading System config... [ ok ]
-----
* Checking Active Alarms... [warn]
* Alarms exist in the node...Are you sure you want to continue ? (y/n)y
* Proceeding... [ ok ]
-----
* Checking the CUDB System Status.. [ ok ]
-----
* Checking IPWorks Front-End Schema Status .. [ ok ]
-----
* IPWorks Front-End application counter installation option:
* Installation of IPWorks ENUM FE application counters. Yes or No? (y/n)y
* Installation of IPWorks AAA FE application counters. Yes or No? (y/n)y
* Installation of IPWorks PKI FE application counters. Yes or No? (y/n)y
-----
* Copying IPWorks Files.. [ ok ]
-----
* Checking Mastership of PLDB... [ ok ]
* This node HAS the Mastership of the PLDB !!!
* Proceeding with Master installation...
* Creating IPWENUMFE procedures in DS1_0... [ ok ]
* Creating IPWAAAFE procedures in DS1_0... [ ok ]
* Creating IPWPKIFE procedures in DS1_0... [ ok ]
* Creating IPWENUMFE procedures in DS1_1... [ ok ]
* Creating IPWAAAFE procedures in DS1_1... [ ok ]
* Creating IPWPKIFE procedures in DS1_1... [ ok ]
* Creating IPWENUMFE procedures in PL0... [ ok ]
* Creating IPWENUMFE tables in PL0.. [ ok ]
* Creating IPWAAAFE procedures in PL0... [ ok ]
* Creating IPWAAAFE tables in PL0.. [ ok ]
* Creating IPWPKIFE procedures in PL0... [ ok ]
* Creating IPWPKIFE tables in PL0.. [ ok ]
* Creating IPWENUMFE procedures in PL1... [ ok ]
* Creating IPWAAAFE procedures in PL1... [ ok ]
* Creating IPWPKIFE procedures in PL1... [ ok ]
-----
* Reading Counters config files.. [ ok ]
-----
* INFORMATION: You are going to be asked if creation of the Pm Counter Job file for the active application counter
* will be handled by either an Ericsson OSS-RC ("yes") or by this program ("no")!
* WARNING : if you choose "yes" be sure OSS-RC is intergated with CUDB
* : if Ericsson OSS-RC is not integrated but you choose "yes", Pm Counter Job file will not be defined
* : and output application counter files will not be generated
* WARNING : Will an Ericsson OSS-RC be used to handle the Pm Counter Job file for application counters ? (y/n)n
* The Pm Counter Jobs file for the active application counters will be created by this program! [warn]
-----
* Creating Jobs File for the active counters... [ ok ]
-----
* Reloading Jobs..
* Stopping PmAgent in node 10.22.0.1 ... OK
* ESA PmAgent has been successfully stopped.
* Starting PmAgent in node 10.22.0.1 ... OK
* Stopping PmAgent in node 10.22.0.2 ... OK
* ESA PmAgent has been successfully stopped.
* Starting PmAgent in node 10.22.0.2 ... OK
-----
CUDB_31 SC_2_1#

```

Repeat the step on CUDB32 to install the counters.

7. Show the information of MySQL tables, indexes, and stored procedures that are related to the application counters in the local CUDB node (PL and DS).



For example in CUDB31:

```
CUDB31# ./app_counters_ipworks.pl -p
```

For example:

UDC IPWorks Applications Counters Installation, version Rev. 1.0.1

```
-----
* Reading System Hosts info... [ ok ]
-----
* Reading System config... [ ok ]
-----
* Printing Tables, Procedures and Indexes info in app_print_2017-10-24_152306.log .. [ ok ]
-----
```

You can check log files for detailed information. The log file is stored in the directory where the script is executed.



## 5 Manual Installation of IPWorks Application Counters on CUDB Nodes

This section describes how to manually install the counters of IPWorks ENUM-FE, AAA-FE, and PKI-FE based on the user requirements.

### 5.1 Installing Counters of IPWENUMFE

This section describes how to install the counters of IPWorks ENUM-FE.

#### 5.1.1 Log on SC as Root

Log on the SC as root:

```
ssh root@<SYSMGMT-SC-2-X>
```

**Note:** <SYSMGMT-SC-2-X> is the IP of the SCs in the SYSMGMT virtual network.

#### 5.1.2 Create Application Counters Configuration File

The configuration file must be created in the following location in all CUDB nodes:

```
/home/cudb/oam/performanceMgmt/appCounters/config/
```

```
#vi /home/cudb/oam/performanceMgmt/appCounters/config/GET_IPWENUM
FQDN_COUNTERS.conf
```

```
*PL_COUNTER*
GET_IPWENUMFQDN_COUNTERS
GRP_IPWENUMFE
ENUMFQDNCNT
```

```
#vi /home/cudb/oam/performanceMgmt/appCounters/config/GET_IPWENUMN
APTR_COUNTERS.conf
```

```
GET_IPWENUMNAPTR_COUNTERS
GRP_IPWENUMFE
ENUMNAPTRCNT
```

#### 5.1.3 Create Table in PLDB

The tables must be filled with initial values and created in the following location in a single CUDB node on which PLDB runs as Master:



```
/home/cudb/oam/performanceMgmt/appCounters/schema/
```

```
#vi /home/cudb/oam/performanceMgmt/appCounters/schema/create_group_counters_IPWENUMFE.sql
```

```
CREATE DATABASE IF NOT EXISTS cudb_application_counters;
USE cudb_application_counters;
DROP TABLE IF EXISTS GRP_IPWENUMFE;
CREATE TABLE GRP_IPWENUMFE(ENUMNAPTRCNT integer,ENUMFQDNCNT integer,PRIMARY KEY(ENUMNAPTRCNT)) ENGINE=NDB;
INSERT INTO GRP_IPWENUMFE VALUES (0,0);
```

Provision the new table to a single PLDB master replica and slave replica:

```
shell> mysql -h<pl0|pl1> -P15000 --user=<user_name>
--password=<password> -e " source /home/cudb/oam/performanceMgmt/a
ppCounters/schema/create_group_counters_IPWENUMFE.sql"
```

**Note:** All tables must be created only from a single PLDB (PL0 or PL1), because these tables depend on the database cluster and are visible from other PLDBs.

#### 5.1.4 Create Stored procedure

This section describes how to create stored procedure in all the replication channels.

To create stored procedure, do:

1. Check the replication channels with the command:

```
cudbSystemStatus
```

The following example is Node 31 as Master replica and Node 32 as Slave replica.

```
Node      | 31 | 32
=====
PLDB      | M  | S2
DSG 1     | M  | S2
```

Printing Detailed Replication Status for the Slave Replicas:

Node 31:

There are no Slave clusters

Node 32:

```
Replication in DSG0(Chan=2) .... Up -- Delay = 0.0 seconds, no. of pending changes = 0
Replication in DSG1(Chan=2) .... Up -- Delay = 0.0 seconds, no. of pending changes = 0
```

2. Create the procedure into all the replication channels.

The scripts with the stored SQL procedures must be kept in the following location in all CUDB nodes:

```
/home/cudb/oam/performanceMgmt/appCounters/procedures/
```





```
#vi /home/cudb/oam/performanceMgmt/appCounters/procedures/create_stored_procedures_IPWENUMFQDN.sql
```

```
USE cudb_user_data;
DELIMITER //
DROP PROCEDURE IF EXISTS GET_IPWENUMFQDN_COUNTERS //
CREATE PROCEDURE GET_IPWENUMFQDN_COUNTERS (OUT ENUMFQDNCNT INT)
BEGIN
SELECT COUNT(*) INTO ENUMFQDNCNT FROM EnumFQDN;
END //
DELIMITER ;
```

```
#vi /home/cudb/oam/performanceMgmt/appCounters/procedures/create_stored_procedures_IPWENUMNAPTR.sql
```

```
USE cudb_user_data;
DELIMITER //
DROP PROCEDURE IF EXISTS GET_IPWENUMNAPTR_COUNTERS //
CREATE PROCEDURE GET_IPWENUMNAPTR_COUNTERS (OUT ENUMNAPTRCNT INT)
BEGIN
SELECT COUNT(*) INTO ENUMNAPTRCNT FROM EnumNAPTRRecord;
END //
DELIMITER ;
```

3. Execute the command on the database cluster access servers of Master PLDB replica and Slave PLDB replica:

```
shell> mysql -hPL0 -P15000 --user=<user_name> --password=<password> -e "source /home/cudb/oam/performanceMgmt/appCounters/procedures/create_stored_procedures_IPWENUMFQDN.sql"
```

```
shell> mysql -hPL1 -P15000 --user=<user_name> --password=<password> -e "source /home/cudb/oam/performanceMgmt/appCounters/procedures/create_stored_procedures_IPWENUMFQDN.sql"
```

4. Execute the command on the database cluster access servers of Master DSG replicas and slave DSG replicas.

```
shell> mysql -hdsX_0 -P<15000+10*X> --user=<user_name> --password=<password> -e "source /home/cudb/oam/performanceMgmt/appCounters/procedures/create_stored_procedures_IPWENUMNAPTR.sql"
```

```
shell> mysql -hdsX_1 -P<15000+10*X> --user=<user_name> --password=<password> -e "source /home/cudb/oam/performanceMgmt/appCounters/procedures/create_stored_procedures_IPWENUMNAPTR.sql"
```

### 5.1.5 Create Scripts to Fetch Application Counters to ESA

The scripts must be stored in the following location in all CUDB nodes:

```
/home/cudb/oam/performanceMgmt/appCounters/scripts/
```

```
#vi /home/cudb/oam/performanceMgmt/appCounters/scripts/APCount_IPWENUMFE.sh
```



```
#!/bin/bash
if [ -f /home/cudb/oam/performanceMgmt/appCounters/scripts/appCountersLib.sh ]
then
. /home/cudb/oam/performanceMgmt/appCounters/scripts/appCountersLib.sh
else
. /home/cudb/oam/performanceMgmt/appCounters/config/appCountersLib.sh
fi
QUERY="select ENUMNAPTRCNT,ENUMFQDNCNT from GRP_IPWENUMFE;"
#DEBUG=1
mysql_query "$QUERY"
# Simple solution is to exit on failure
# More critical counters could follow a retry policy
if [ $? != 0 ]
then
logging_system 'error' "Query \"$QUERY\" failed with \"$ROWS\""
exit 1
fi
GROUPID=IPWENUMFE
COUNTERID=IPWENUMFE
format_counter $GROUPID $COUNTERID
exit $?
```

Execute the permission command:

```
chmod 777 /home/cudb/oam/performanceMgmt/appCounters/scripts/AP
Count_IPWENUMFE.sh
```

### 5.1.6 Configure ESA to Use Scripts

An XML file must be created in the following location in all CUDB nodes:

```
/home/cudb/oam/performanceMgmt/config/PmCounters/
```

```
#vi /home/cudb/oam/performanceMgmt/config/PmCounters/GroupIPWENUM
FECounters.xml
```



```
<?xml version="1.0" encoding="UTF-8"?>
<pmCntGroup xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.ericsson.com/esa" xsi:schemaLocation="http://www.ericsson.com/esa http://www.ericsson.com/esa/PMCounter.xsd">
  <identification>
    <groupId>IPWENUMFE</groupId>
  </identification>
  <description>
    <groupDescr>IPWorks ENUM FE counters</groupDescr>
    <groupInfo>These counters permits collect number of data records for ENUM-FE.</groupInfo>
  </description>
  <!--Counter collection with data source SCRIPT-->
  <!-- G1-->
  <cntCollection active="yes" activeSnm="yes">
    <identification>
      <collectionId>IPWENUMFE</collectionId>
    </identification>
    <cntDefinition cntType="Gauge">
      <identification>
        <counterId>ENUMNAPTRCNT</counterId>
      </identification>
      <description>
        <counterDescr>Number of data records for ENUM NAPTR Records.</counterDescr>
        <counterInfo/>
      </description>
    </cntDefinition>
    <cntDefinition cntType="Gauge">
      <identification>
        <counterId>ENUMFQDNCNT</counterId>
      </identification>
      <description>
        <counterDescr>Number of data records for ENUM FQDN.</counterDescr>
        <counterInfo/>
      </description>
    </cntDefinition>
    <dataSource interval="900">
      <script>
        <location>/cluster/home/cudb/oam/performanceMgmt/appCounters/scripts/APCount_IPWENUMFE</location>
      </script>
    </dataSource>
  </cntCollection>
</pmCntGroup>
```

### 5.1.7 Configure CRON Task

To generate application counters, tasks must be introduced in the cron of the SCs in all CUDB nodes.

Modify the /home/cudb/oam/performanceMgmt/appCounters/scripts/appCounters.cron file to include the new task.

```
#vi /home/cudb/oam/performanceMgmt/appCounters/scripts/appCounters.cron
```

```
LIST_APP_COUNTERS[11]="/opt/ericsson/cudb/OAM/bin/cudbApplicationCounters -C /cluster/home/cudb/oam/performanceMgmt/appCounters/scripts/appCounters.cron"
LIST_APP_COUNTERS[12]="/opt/ericsson/cudb/OAM/bin/cudbApplicationCounters -C /cluster/home/cudb/oam/performanceMgmt/appCounters/scripts/appCounters.cron"
```

**Note:** If the number 11 and 12 exist, then use (the max. number +1) to confirm the number is unique.

### 5.1.8 Configure Jobs to Generate Output Files

Application counters output files are configured using XML-based job files placed in the following folder in all CUDB node:



```
/home/cudb/oam/performanceMgmt/config/PmJobs
```

Modify the /home/cudb/oam/performanceMgmt/config/PmJobs/APP\_COUNT\_JOB.xml file to include the new PmJob.

```
#vi /home/cudb/oam/performanceMgmt/config/PmJobs/APP_COUNT_JOB.xml
```

```
<jobDefinition><jobId>JobIPWENUMFE</jobId><groupId>IPWENUMFE</groupId><startTime>201707241715</startTime><granularity>
```

**Note:** The number '201707241715' is the time for fetching the counter.

The number '15' is the default period for generating PmJob file (minutes).  
The number can be modified, if needed.

### 5.1.9 Reload CUDB PmJob

To make the jobs configuration effective, restart the Performance Management Agent (pmagent) with following command in all CUDB node.

```
cudbPmJobReload
```

When execute the command, an output must be displayed similar to the below example:

```
Stopping PmAgent in node 10.22.27.10 ... OK
Waiting for ESA PmAgent to go off line.
ESA PmAgent has been successfully stopped.
Starting PmAgent in node 10.22.27.10 ... OK
```

### 5.1.10 Activate/Deactivate the New Planned Tasks

To start the new planned tasks in the operating system in both SCs on all CUDB nodes of the CUDB system, execute following command:

```
# /etc/init.d/cudbappCounters start
```

To stop the scheduled cron tasks in the operating system in both SCs on all CUDB nodes of the CUDB system, execute the following command:

```
# /etc/init.d/cudbappCounters stop
```

**Note:** This command removes all the .cron files, as cron tasks, from cron. The result can be reviewed by using the crontab -l command on both SCs.

.cron file entries will be added into crontab after reboot. To remove files from crontab permanently, remove the files from their location.



## 5.2 Installing Counters of IPWAAAFE

This section describes how to install the counters of IPWorks AAA-FE.

### 5.2.1 Log on SC as Root

Log on the SC as root:

```
ssh root@<SYSMGMT-SC-2-X>
```

**Note:** <SYSMGMT-SC-2-X> is the IP of the SCs in the SYSMGMT virtual network.

### 5.2.2 Create Application Counters Configuration File

The configuration file must be created in the following location in all CUDB nodes:

```
/home/cudb/oam/performanceMgmt/appCounters/config/
```

```
#vi /home/cudb/oam/performanceMgmt/appCounters/config/GET_IPWAAAFE_COUNTERS.conf
```

```
GET_IPWAAAFE_COUNTERS
GRP_IPWAAAFE
AAAUSERCNT
```

### 5.2.3 Create Table in PLDB

The tables must be filled with initial values and created in the following location in a single CUDB node on which PLDB runs as Master:

```
/home/cudb/oam/performanceMgmt/appCounters/schema/
```

```
#vi /home/cudb/oam/performanceMgmt/appCounters/schema/create_group_counters_IPWAAAFE.sql
```

```
CREATE DATABASE IF NOT EXISTS cudb_application_counters;
USE cudb_application_counters;
DROP TABLE IF EXISTS GRP_IPWAAAFE;
CREATE TABLE GRP_IPWAAAFE(AAAUSERCNT integer,PRIMARY KEY(AAAUSERCNT)) ENGINE=NDB;
INSERT INTO GRP_IPWAAAFE VALUES (0);
```

Provision the new table to a single PLDB master replica and slave replica:

```
shell> mysql -h<pl0|pl1> -P15000 --user=<user_name>
--password=<password> -e " source /home/cudb/oam/performanceMgmt/a
ppCounters/schema/create_group_counters_IPWAAAFE.sql "
```



**Note:** All tables must be created only from a single PLDB (PL0 or PL1), because these tables depend on the database cluster and are visible from other PLDBs.

## 5.2.4 Create Stored Procedure

This section describes how to create stored procedure in all the replication channels.

To create stored procedure, do:

1. Check the replication channels with the command:

**cudbSystemStatus**

The following example is Node 31 as Master replica and Node 32 as Slave replica.

```
Node      | 31 | 32
=====
PLDB      | M  | S2
DSG 1     | M  | S2
```

Printing Detailed Replication Status for the Slave Replicas:

Node 31:

There are no Slave clusters

Node 32:

```
Replication in DSG0(Chan=2) .... Up -- Delay = 0.0 seconds, no. of pending changes = 0
Replication in DSG1(Chan=2) .... Up -- Delay = 0.0 seconds, no. of pending changes = 0
```

2. Create the procedure into all the replication channels.

The scripts with the stored SQL procedures must be kept in the following location in all CUDB nodes:

`/home/cudb/oam/performanceMgmt/appCounters/procedures/`

**#vi /home/cudb/oam/performanceMgmt/appCounters/procedures/create\_stored\_procedures\_IPWAAAFE.sql**

```
USE cudb_user_data;
DELIMITER //
DROP PROCEDURE IF EXISTS GET_IPWAAAFE_COUNTERS //
CREATE PROCEDURE GET_IPWAAAFE_COUNTERS (OUT AAAUSERCNT INT)
BEGIN
SELECT COUNT(*) INTO AAAUSERCNT FROM AAPProfile;
END //
DELIMITER ;
```

3. Execute the command on the database cluster access servers of Master DSG replicas and slave DSG replicas.

```
shell> mysql -hdsX_0 -P<15000+10*X> --user=<user_name>
--password=<password> -e "source /home/cudb/oam/performanceMgmt
/appCounters/procedures/create_stored_procedures_IPWAAAFE.sql"
```



```
shell> mysql -hdsX_1 -P<15000+10*X> --user=<user_name>
--password=<password> -e "source /home/cudb/oam/performanceMgmt
/appCounters/procedures/create_stored_procedures_IPWAAAFE.sql"
```

### 5.2.5 Create Scripts to Fetch Application Counters to ESA

The scripts must be stored in the following location in all CUDB nodes:

/home/cudb/oam/performanceMgmt/appCounters/scripts/

```
#vi /home/cudb/oam/performanceMgmt/appCounters/scripts/APCount_I
PWAAAFE.sh
```

```
#!/bin/bash
if [ -f /home/cudb/oam/performanceMgmt/appCounters/scripts/appCountersLib.sh ]
then
. /home/cudb/oam/performanceMgmt/appCounters/scripts/appCountersLib.sh
else
. /home/cudb/oam/performanceMgmt/appCounters/config/appCountersLib.sh
fi
QUERY="select AAAUSERCNT from GRP_IPWAAAFE;"
#DEBUG=1
mysql_query "$QUERY"
# Simple solution is to exit on failure
# More critical counters could follow a retry policy
if [ $? != 0 ]
then
logging_system 'error' "Query \"$QUERY\" failed with \"$ROWS\""
exit 1
fi
GROUPID=IPWAAAFE
COUNTERID=IPWAAAFE
format_counter $GROUPID $COUNTERID
exit $?
```

### 5.2.6 Configure ESA to Use Scripts

An XML file must be created in the following location in all CUDB nodes:

/home/cudb/oam/performanceMgmt/config/PmCounters/

```
#vi /home/cudb/oam/performanceMgmt/config/PmCounters/GroupIPWAAA
FECounters.xml
```



```
<?xml version="1.0" encoding="UTF-8"?>
<pmCntGroup xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.ericsson.com/esa" xsi:schemaLocation="http://www.ericsson.com/esa http://www.ericsson.com/esa/PMCounter.xsd">
  <identification>
    <groupId>IPWAAAFE</groupId>
  </identification>
  <description>
    <groupDescr>IPWorks AAA-FE counters</groupDescr>
    <groupInfo>These counters permits collect number of data records for AAA-FE.</groupInfo>
  </description>
  <!--Counter collection with data source SCRIPT-->
  <!-- G1-->
  <cntCollection active="yes" activeSnp="yes">
    <identification>
      <collectionId>IPWAAAFE</collectionId>
    </identification>
    <cntDefinition cntType="Gauge">
      <identification>
        <counterId>AAAUUSERCNT</counterId>
      </identification>
      <description>
        <counterDescr>Number of data records for aauser.</counterDescr>
        <counterInfo/>
      </description>
    </cntDefinition>
    <dataSource interval="900">
      <script>
        <location>/cluster/home/cudb/oam/performanceMgmt/appCounters/scripts/APCount_IPWAAAFE.sh</location>
      </script>
    </dataSource>
  </cntCollection>
</pmCntGroup>
```

Execute the permission command:

```
chmod 777 /home/cudb/oam/performanceMgmt/appCounters/scripts/APCount_IPWAAAFE.sh
```

## 5.2.7 Configure CRON Task

To generate application counters, tasks must be introduced in the cron of the SCs in all CUDB nodes.

Modify the `/home/cudb/oam/performanceMgmt/appCounters/scripts/appCounters.cron` file to include the new task.

```
#vi /home/cudb/oam/performanceMgmt/appCounters/scripts/appCounters.cron
```

```
LIST_APP_COUNTERS[13]="/opt/ericsson/cudb/OAM/bin/cudbApplicationCounters -C /cluster/home/cudb/oam/performanceMgmt/oam/performanceMgmt/appCounters/scripts/appCounters.cron"
```

**Note:** If the number 13 exists, then use (the max. number +1) to confirm the number is unique.

## 5.2.8 Configure Jobs to Generate Output Files

Application counters output files are configured using XML-based job files placed in the following folder in all CUDB node:

```
/home/cudb/oam/performanceMgmt/config/PmJobs
```





Modify the /home/cudb/oam/performanceMgmt/config/PmJobs/APP\_COUNT\_JOB.xml file to include the new PmJob.

```
#vi /home/cudb/oam/performanceMgmt/config/PmJobs/APP_COUNT_JOB.xml
```

```
<jobDefinition><jobId>JobIPWAAAFE</jobId><groupId>IPWAAAFE</groupId><startTime>201707241715</startTime><granulari
```

**Note:** The number '201707241715' is the time for fetching the counter.

The number '15' is the default period for generating PmJob file (minutes).  
The number can be modified, if needed.

### 5.2.9 Reload CUDB PmJob

To make the jobs configuration effective, restart the Performance Management Agent (pmagent) with following command in all CUDB node.

```
cudbPmJobReload
```

When execute the command, an output must be displayed similar to the below example:

```
Stopping PmAgent in node 10.22.27.10 ... OK
Waiting for ESA PmAgent to go off line.
ESA PmAgent has been successfully stopped.
Starting PmAgent in node 10.22.27.10 ... OK
```

### 5.2.10 Activate/Deactivate the New Planned Tasks

To start the new planned tasks in the operating system in both SCs on all CUDB nodes of the CUDB system, execute following command:

```
# /etc/init.d/cudbappCounters start
```

To stop the scheduled cron tasks in the operating system in both SCs on all CUDB nodes of the CUDB system, execute the following command:

```
# /etc/init.d/cudbappCounters stop
```

**Note:** This command removes all the .cron files, as cron tasks, from cron. The result can be reviewed by using the crontab -l command on both SCs.

.cron file entries will be added into crontab after reboot. To remove files from crontab permanently, remove the files from their location.

## 5.3 Installing Counters of IPWPKIFE

This section describes how to install the counters of IPWorks PKI-FE.



### 5.3.1 Log on SC as Root

Log on the SC as root:

```
ssh root@<SYSMGMT-SC-2-X>
```

**Note:** <SYSMGMT-SC-2-X> is the IP of the SCs in the SYSMGMT virtual network.

### 5.3.2 Create Application Counters Configuration File

The configuration file must be created in the following location in all CUDb nodes:

```
/home/cudb/oam/performanceMgmt/appCounters/config/
```

```
#vi /home/cudb/oam/performanceMgmt/appCounters/config/GET_IPWPKIFE_COUNTERS.conf
```

```
GET_IPWPKIFE_COUNTERS
GRP_IPWPKIFE
NSDSUSERCNT
```

### 5.3.3 Create Table in PLDB

The tables must be filled with initial values and created in the following location in a single CUDb node on which PLDB runs as Master:

```
/home/cudb/oam/performanceMgmt/appCounters/schema/
```

```
#vi /home/cudb/oam/performanceMgmt/appCounters/schema/create_group_counters_IPWPKIFE.sql
```

```
CREATE DATABASE IF NOT EXISTS cudb_application_counters;
USE cudb_application_counters;
DROP TABLE IF EXISTS GRP_IPWPKIFE;
CREATE TABLE GRP_IPWPKIFE(NSDSUSERCNT integer,PRIMARY KEY(NSDSUSERCNT)) ENGINE=NDB;
INSERT INTO GRP_IPWPKIFE VALUES (0);
```

Provision the new table to a single PLDB master replica and slave replica:

```
shell> mysql -h<pl0|pl1> -P15000 --user=<user_name>
--password=<password> -e " source /home/cudb/oam/performanceMgmt/appCounters/schema/create_group_counters_IPWPKIFE.sql"
```

**Note:** All tables must be created only from a single PLDB (PL0 or PL1), because these tables depend on the database cluster and are visible from other PLDBs.



### 5.3.4 Create Stored Procedure

This section describes how to create stored procedure in all the replication channels.

To create stored procedure, do:

1. Check the replication channels with the command:

**cudbSystemStatus**

The following example is Node 31 as Master replica and Node 32 as Slave replica.

```
Node      | 31 | 32
=====
PLDB      | M  | S2
DSG 1     | M  | S2
```

Printing Detailed Replication Status for the Slave Replicas:

Node 31:

There are no Slave clusters

Node 32:

```
Replication in DSG0(Chan=2) .... Up -- Delay = 0.0 seconds, no. of pending changes = 0
Replication in DSG1(Chan=2) .... Up -- Delay = 0.0 seconds, no. of pending changes = 0
```

2. Create the procedure into all the replication channels.

The scripts with the stored SQL procedures must be kept in the following location in all CUDB nodes:

`/home/cudb/oam/performanceMgmt/appCounters/procedures/`

**#vi /home/cudb/oam/performanceMgmt/appCounters/procedures/create\_stored\_procedures\_IPWPKIFE.sql**

```
USE cudb_user_data;
DELIMITER //
DROP PROCEDURE IF EXISTS GET_IPWPKIFE_COUNTERS //
CREATE PROCEDURE GET_IPWPKIFE_COUNTERS (OUT NSDSUSERCNT INT)
BEGIN
SELECT COUNT(*) INTO NSDSUSERCNT FROM nsduser;
END //
DELIMITER ;
```

3. Execute the command on the database cluster access servers of Master DSG replicas and slave DSG replicas.

```
shell> mysql -hdsX_0 -P<15000+10*X> --user=<user_name>
--password=<password> -e "source /home/cudb/oam/performanceMgmt
/appCounters/procedures/create_stored_procedures_IPWPKIFE.sql"
```

```
shell> mysql -hdsX_1 -P<15000+10*X> --user=<user_name>
--password=<password> -e "source /home/cudb/oam/performanceMgmt
/appCounters/procedures/create_stored_procedures_IPWPKIFE.sql"
```



### 5.3.5 Create Scripts to Fetch Application Counters to ESA

The scripts must be stored in the following location in all CUDb nodes:

`/home/cudb/oam/performanceMgmt/appCounters/scripts/`

**#vi /home/cudb/oam/performanceMgmt/appCounters/scripts/APCount\_IPWPkIFE.sh**

```
#!/bin/bash
if [ -f /home/cudb/oam/performanceMgmt/appCounters/scripts/appCountersLib.sh ]
then
. /home/cudb/oam/performanceMgmt/appCounters/scripts/appCountersLib.sh
else
. /home/cudb/oam/performanceMgmt/appCounters/config/appCountersLib.sh
fi
QUERY="select NSDSUSERCNT from GRP_IPWPkIFE;"
#DEBUG=1
mysql_query "$QUERY"
# Simple solution is to exit on failure
# More critical counters could follow a retry policy
if [ $? != 0 ]
then
logging_system 'error' "Query \"$QUERY\" failed with \"$ROWS\""
exit 1
fi
GROUPID=IPWPkIFE
COUNTERID=IPWPkIFE
format_counter $GROUPID $COUNTERID
exit $?
```

Execute the permission command:

**chmod 777 /home/cudb/oam/performanceMgmt/appCounters/scripts/APCount\_IPWPkIFE.sh**

### 5.3.6 Configure ESA to Use Scripts

An XML file must be created in the following location in all CUDb nodes:

`/home/cudb/oam/performanceMgmt/config/PmCounters/`

**#vi /home/cudb/oam/performanceMgmt/config/PmCounters/GroupIPWPkIFECounters.xml**



```
<?xml version="1.0" encoding="UTF-8"?>
<pmCntGroup xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.ericsson.com/esa" xsi:schemaLocation="http://www.ericsson.com/esa http://www.ericsson.com/esa/ericsson-cmdb.xsd">
  <identification>
    <groupId>IPWPKIFE</groupId>
  </identification>
  <description>
    <groupDescr>IPWorks PKI-FE counters</groupDescr>
    <groupInfo>These counters permits collect number of data records for PKI-FE.</groupInfo>
  </description>
  <!--Counter collection with data source SCRIPT-->
  <!-- G1-->
  <cntCollection active="yes" activeSnmp="yes">
    <identification>
      <collectionId>IPWPKIFE</collectionId>
    </identification>
    <cntDefinition cntType="Gauge">
      <identification>
        <counterId>NSDSUSERCNT</counterId>
      </identification>
      <description>
        <counterDescr>Number of data records for nsdsuser.</counterDescr>
        <counterInfo/>
      </description>
    </cntDefinition>
    <dataSource interval="900">
      <script>
        <location>/cluster/home/cudb/oam/performanceMgmt/appCounters/scripts/APCount_IPWPKIFE</location>
      </script>
    </dataSource>
  </cntCollection>
</pmCntGroup>
```

### 5.3.7 Configure CRON Task

To generate application counters, tasks must be introduced in the cron of the SCs in all CUDB nodes.

Modify the `/home/cudb/oam/performanceMgmt/appCounters/scripts/appCounters.cron` file to include the new task.

```
#vi /home/cudb/oam/performanceMgmt/appCounters/scripts/appCounters.cron
```

```
LIST_APP_COUNTERS[14]="/opt/ericsson/cudb/OAM/bin/cudbApplicationCounters -C /cluster/home/cudb/oam/performanceMgmt/appCounters/scripts/appCounters.cron
```

**Note:** If the number 14 exists, then use (the max. number +1) to confirm the number is unique.

### 5.3.8 Configure Jobs to Generate Output Files

Application counters output files are configured using XML-based job files placed in the following folder in all CUDB node:

```
/home/cudb/oam/performanceMgmt/config/PmJobs
```

Modify the `/home/cudb/oam/performanceMgmt/config/PmJobs/APP_COUNT_JOB.xml` file to include the new PmJob.

```
#vi /home/cudb/oam/performanceMgmt/config/PmJobs/APP_COUNT_JOB.xml
```



```
<jobDefinition><jobId>JobIPWPKIFE</jobId><groupId>IPWPKIFE</groupId><startTime>201707241715</startTime><granularity>P
```

**Note:** The number '201707241715' is the time for fetching the counter.

The number '15' is the default period for generating PmJob file (minutes).  
The number can be modified, if needed.

### 5.3.9 Reload CUDB PmJob

To make the jobs configuration effective, restart the Performance Management Agent (pmagent) with following command in all CUDB node.

**cudbPmJobReload**

When execute the command, an output must be displayed similar to the below example:

```
Stopping PmAgent in node 10.22.27.10 ... OK
Waiting for ESA PmAgent to go off line.
ESA PmAgent has been successfully stopped.
Starting PmAgent in node 10.22.27.10 ... OK
```

### 5.3.10 Activate/Deactivate the New Planned Tasks

To start the new planned tasks in the operating system in both SCs on all CUDB nodes of the CUDB system, execute following command:

**# /etc/init.d/cudbappCounters start**

To stop the scheduled cron tasks in the operating system in both SCs on all CUDB nodes of the CUDB system, execute the following command:

**# /etc/init.d/cudbappCounters stop**

**Note:** This command removes all the .cron files, as cron tasks, from cron. The result can be reviewed by using the crontab -l command on both SCs.

.cron file entries will be added into crontab after reboot. To remove files from crontab permanently, remove the files from their location.



## 6 Uninstallation of Application Counters on CUDB Nodes

After the counters are installed, if you need to uninstall the IPWorks application counters from ALL the CUDB nodes, execute the following command on all CUDB nodes:

```
<CUDB Node Name># ./app_counters_ipworks.pl -u
```

For example:

```
IPWorks Applications Counters Installation, version Rev. 1.0.1
* PLDB detected...
-----
hostname is SC_2_1
* Checking Active OAM blade... [ ok ]
-----
* Reading System Hosts info... [ ok ]
-----
* Reading System config... [ ok ]
-----
* Checking Active Alarms... [warn]
* Alarms exist in the node...Are you sure you want to continue ? (y/n)y
* Proceeding... [ ok ]
-----
* Checking the CUDB System Status.. [ ok ]
-----
* Checking IPWorks Front-End Schema Status .. [ ok ]
-----
* IPWorks Front-End application counter uninstallation option:
* Uninstallation of IPWorks ENUM FE application counters. Yes or No? (y/n)y
* Uninstallation of IPWorks AAA FE application counters. Yes or No? (y/n)y
* Uninstallation of IPWorks PKI FE application counters. Yes or No? (y/n)n
-----
* Deleting previous counter installation crontab files.. [ ok ]
-----
* Deleting previous counter installation files... [ ok ]
-----
* Reading Counters config files.. [ ok ]
-----
* Reloading Jobs..
* Stopping PmAgent in node 10.22.0.1 ... OK
* ESA PmAgent has been successfully stopped.
* Starting PmAgent in node 10.22.0.1 ... OK
* Stopping PmAgent in node 10.22.0.2 ... OK
* ESA PmAgent has been successfully stopped.
* Starting PmAgent in node 10.22.0.2 ... OK
-----
* Checking Mastership of PLDB... [ ok ]
* This node HAS the Mastership of the PLDB !!!
* Proceeding with Master uninstallation...
* Deleting procedures in DS1_0.. [ ok ]
* Deleting procedures in DS1_1.. [ ok ]
* Deleting procedures in PL0.. [ ok ]
* Dropping tables cudb_application_counters in PL0.. [ ok ]
* Deleting procedures in PL1.. [ ok ]
-----
* Clearing Application Counter Alarms, 0 alarms found [ ok ]
-----
CUDB_31 SC_2_1#
```







## Reference List

- [1] Glossary of Terms and Acronyms
- [2] Trademark Information
- [3] Typographic Conventions
- [4] IPWorks Performance Measurements
- [5] CUDB Performance Guide, 4/1553-HDA 104 03 Uen
- [6] CUDB Schema Conversion Tool, 2/1553-CNH 160 6161/10 Uen