

# MTAS XDMS Management Guide

## MTAS

---

### USER GUIDE

## **Copyright**

© Ericsson AB 2016. 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>Overview</b>	<b>3</b>
<b>3</b>	<b>Configure Diameter Stack</b>	<b>5</b>
<b>4</b>	<b>Configure Sh Interface</b>	<b>7</b>
<b>5</b>	<b>Optional XDMS Function Parameters Configuration</b>	<b>9</b>
<b>6</b>	<b>Access Ut Interface</b>	<b>11</b>
<b>7</b>	<b>Access CAI3G Interface</b>	<b>13</b>
<b>8</b>	<b>CAI3G Interface Configuration</b>	<b>15</b>
8.1	MTAS XDMS Keystore Modifications	15
8.2	Apply the Modified MTAS XDMS Keystore	17
8.3	CAI3G Logging	18
8.4	CAI3G Logging Configuration	18
<b>9</b>	<b>Access CCMP Interface</b>	<b>19</b>





# 1 Introduction

This document describes how to configure the XML Document Management Server (XDMS) function in the MTAS.

## 1.1 Prerequisites

It is assumed that the user of this document is familiar with the O&M area, in general.

### 1.1.1 Documents

Before any of the procedures in this document are done, the following documents must be read and understood:

- *Diameter Management*
- *Ericsson Command-Line Interface User Guide*
- *Managed Object Model (MOM)*

### 1.1.2 Conditions

The following conditions must apply:

An Ericsson Command-Line Interface (ECLI) session in Exec mode is in progress.

For configuring the CAI3G interface, the user must be familiar with and be entitled to use the services of a trusted Certificate Authority. The user must also know the password for the users required for the different steps described in this document.

For information on the different users and the corresponding roles, restrictions, and privileges, refer to *Certificate Management* and *Security Management for ECLI, NETCONF, and SFTP Users*.





## 2 Overview

The XDMS function supports the CAI3G interface to allow the operator to provision and update the PSTN/ISDN Simulation Services data for subscribers and an Ut interface to allow the subscriber to manipulate their own PSTN/ISDN Simulation Services data. To achieve this, the XDMS function also supports a Sh interface to fetch and update the data in the Home Subscriber Server (HSS). All service data XML instance files have normalized entries, refer to *Managed Object Model (MOM)*.

The configuration of the XDMS function involves defining Diameter stack attributes, and defining the realm to which the HSS node belongs. Optionally, the configuration involves defining the hostname of the HSS node or the Subscriber Location Function (SLF) node.

The *MtasXdsm* Managed Object (MO) controls the XDMS function for a complete MTAS node.

The configuration of the Diameter stack and the Sh interface of the XDMS function is shared with the subscriber data function.

The configuration of the Number Normalization data of the XDMS function is shared with the subscriber data function, for more information, refer to *Managed Object Model (MOM)*.





## 3 Configure Diameter Stack

Several the MTAS-specific parameter values must be configured in the Diameter stack.

To configure the Diameter stack instance for the XDMS function:

- Configure the Diameter stack instance, referring to *MTAS Subscriber Data Management Guide*.





## 4 Configure Sh Interface

To route Sh messages correctly, it is necessary to specify which realm the HSS nodes belong to. The Sh configuration attributes of the XDMS function are shared with the subscriber data function.

To configure the Sh parameters:

- Configure the applicable attributes, `mtasShIfDestinationRealm`, and `mtasShIfDestinationHost`, refer to *MTAS Subscriber Data Management Guide*.

The `mtasShIfMmtelServiceInd` is set by default during MTAS installation.





## 5 Optional XDMS Function Parameters Configuration

The `MtasXdmsData` MO makes it possible to configure other parameters, than the ones that are described in this document, and which are related to the XDMS function. For a complete description of all parameters relating to the configuration of the XDMS function MO, refer to *Managed Object Model (MOM)*.





## 6 Access Ut Interface

The XDMS function supports an Ut interface for MMTel AS, on either IPv4, IPv6 or both, to allow the subscriber to manipulate their PSTN/ISDN Simulation Services data, for more information, refer to [Configuration Access Protocol \(XCAP\) over the Ut interface for Manipulating NGN PSTN/ISDN Simulation Services](#).

To enable the Ut interface, it must be unlocked using the `mtasXdmsUtAdministrativeState` parameter on the `MtasXdms` MO.

For further details, refer to *Managed Object Model (MOM)*.

The XCAP root for the XDMS function is as follows:

```
http://<hostname>:8090/mtasxdms
```

The parameter `<hostname>` is the hostname of the MTAS node running the XDMS function, this can be defined during maiden installation or modified later.

**Note:** If `<hostname>` is a numeric IPv6 address, the address must be enclosed in brackets (for example: `[2000::4:66]`).

The PSTN/ISDN Simulation Services application use has an AUID of “simservs.ngn.etsi.org”. The document name for configuration of an individual subscriber is “simservs.xml”. This means that the URL to access a document for a particular user has the following form:

```
http://<hostname>:8090/mtasxdms/simservs.ngn.etsi.org/users/<subscriber_uri>/simservs.xml
```

The parameter `<subscriber_uri>` is the URI of the subscriber.

All requests on this interface must be valid XCAP requests and must have either the X-3GPP-Asserted-Identity or the X-XCAP-Asserted-Identity headers to prove that the proxy has authenticated them.

For information on access at element or attribute granularity requiring an extra node selector, refer to [The Extensible Markup Language \(XML\) Configuration Access Protocol \(XCAP\)](#).

This means a URL of the following form:

```
http://<hostname>:8090/mtasxdms/simservs.ngn.etsi.org/users/<subscriber_uri>/simservs.xml/~/<node_selector>
```

The parameter `<node_selector>` equals the selected extra node selector.



The XDMS function also supports the XCAP server capabilities application use with AUID “xcap-caps”. The URL to access the XCAP server capabilities has the following form:

`http://<hostname>:8090/mtasxdms/xcap-caps/global/index`



## 7 Access CAI3G Interface

The XDMS function supports a CAI3G interface to allow the operator to manage subscriber data. The CAI3G interface is a Web Services interface.

To enable the CAI3G interface, it must be unlocked using the `mtasXdmsCai3gAdministrativeState` parameter on the `MtasXdms` MO.

It is also necessary to create at least one user account – an instance of `MtasXdmsCai3gUser` MO with its associated `mtasXdmsCai3gUserPassword` parameter. For further details, refer to *Managed Object Model (MOM)*.

**The URI for the CAI3G interface on the MMTel AS is as follows:**

HTTP:

`http://<platform-vip>:8095/axis2/services/CAI3G`

HTTPS:

`https://<platform-vip>:8443/axis2/services/CAI3G`

**The URI for the CAI3G interface on the ST AS is as follows:**

HTTP:

`http://<platform-vip>:8095/mtasstas`

HTTPS:

`https://<platform-vip>:8443/mtasstas`

If there is a dedicated VIP for CAI3G traffic, use CAI3G VIP address `cai3g-vip` instead of `platform-vip`. For more information, refer to the MTAS SW installation instruction and *Virtual IP Address Management*.

Use of HTTPS/SSL needs configuration, see Section 8 on page 15 for details.

Ports must be made available in the Hardening of the node.





## 8 CAI3G Interface Configuration

The XDMS function supports a secured CAI3G interface to allow the operator to manage subscriber data in an encrypted and authenticated way. The authentication of the MTAS is enabled by a trusted certificate.

The operator has the possibility to perform the following operations on the CAI3G certificate:

- Deletion of the CAI3G certificate
- Creation of a New Self-Signed CAI3G certificate
- Importing a trusted certificate
- Listing the stored certificates
- Logging

Without a valid CAI3G certificate, the secured CAI3G interface of the MTAS cannot operate. For further `keytool` parameters, refer to [Keytool - Key and Certificate Management Tool](#).

### 8.1 MTAS XDMS Keystore Modifications

This section describes how to delete, create, import, and list a certificate.

**Note:** Once the MTAS XDMS keystore is modified, the changes must be applied, see Section 8.2 Apply the Modified MTAS XDMS Keystore on page 17.

#### 8.1.1 Delete CAI3G Certificate

To delete the CAI3G certificate:

1. From an SSH client, log on to the platform-vip:

```
ssh<emergency user>@<platform-vip>
```

2. Set the Java environment:

- `cd /cluster/storage/system/software/jdk*/bin`
- `export PATH=$PATH: `pwd``

3. Delete the old CAI3G certificate:



- `sudo keytool -delete -alias CAI3G -keypass xdmspass -storepass xdmspass -keystore /cluster/storage/system/config/mtas/.xdmskeystore`

## 8.1.2 Create New Self-Signed CAI3G Certificate

If there is a CAI3G certificate already stored in the XDMS keystore, remove it, see Section 8.1.1 Delete CAI3G Certificate on page 15. For listing the available certificates in the XDMS keystore, see Section 8.1.4 List Stored Certificates on page 17.

To create a new self-signed CAI3G certificate:

1. From an SSH client, log on to the platform-vip:

```
ssh<emergency user>@<platform-vip>
```

2. Set the Java environment:

- `cd /cluster/storage/system/software/jdk*/bin`
- `export PATH=$PATH:`pwd``

3. Generate a new self-signed CAI3G certificate:

- `sudo keytool -genkey -alias CAI3G -storepass xdmspass -keypass xdmspass -keystore /cluster/storage/system/config/mtas/.xdmskeystore`

4. Enter the certificate data.

## 8.1.3 Import Trusted Certificate

If there is a CAI3G certificate already stored in the XDMS keystore, remove it, see Section 8.1.1 Delete CAI3G Certificate on page 15. For listing the available certificates in the XDMS keystore, see Section 8.1.4 List Stored Certificates on page 17.

To import a trusted certificate:

1. Copy the trusted certificate to the cluster:

- `sftp <emergency user>@<platform-vip>`
- `lcd<local path of the certificate file>`
- `cd /cluster/storage/system/config/mtas`
- `put <cacert file>`
- `exit`

2. From an SSH client, log on to the platform-vip:



```
ssh<emergency user>@<platform-vip>
```

3. Set the Java environment:

- `cd /cluster/storage/system/software/jdk*/bin`
- `export PATH=$PATH:`pwd``

4. Import the trusted CAI3G certificate:

- `sudo keytool -import -alias CAI3G -storepass xdmspass -keypass xdmspass -keystore /cluster/storage/system/config/mtas/.xdmskeystore -file /cluster/storage/system/config/mtas/<cacert file>`

5. Examine certificate data, enter `yes` if trusted.

### 8.1.4 List Stored Certificates

To list the stored certificates:

1. From an SSH client, log on to the platform-vip:

```
ssh<emergency user>@<platform-vip>
```

2. Set the Java environment:

- `cd /cluster/storage/system/software/jdk*/bin`
- `export PATH=$PATH:`pwd``

3. List the stored certificate:

- `keytool -list -v -storepass xdmspass -keystore /cluster/storage/system/config/mtas/.xdmskeystore`

## 8.2 Apply the Modified MTAS XDMS Keystore

To apply the modified MTAS XDMS keystore:

1. Restart the MMAS traffic instances:

a. Check the DN of the MMAS instances:

- `immfind safSg=SG-traffic,safApp=ERIC-MMAS-APP | grep ^safComp`

For example:

```
safComp=ERIC-MMAS-COMP-0,safSu=ERIC-MMAS-SU-0, safSg=SG-traffic,safApp=ERIC-MMAS-APP
```



```
safComp=ERIC-MMAS-COMP-0,safSu=ERIC-MMAS-SU-1
,safSg=SG-traffic,safApp=ERIC-MMAS-APP
```

- b. Restart the instances one by one on each Payload Node, where **N** is the number of the corresponding Payload Node, and **DN** is the identity of the MMAS instance, as queried in Step a.

- `ssh<emergency user>@PL- [N]`

For example:

```
sshmtasuser01@PL-3
```

- `echo ':reload'> /tmp/mmase.txt`
- `cd /opt/mmase/instance`
- `sudo ./run_cli_command -n "[DN]" -f /tmp/mmase.txt -o`

For example:

```
sudo ./run_cli_command -n
```

```
safComp=ERIC-MMAS-COMP-0,safSu=ERIC-MMAS-SU-0,safSg=SG-traffic,safApp=ERIC-MMAS-APP" -f /tmp/mmase.txt -o
```

```
{
  "outcome" => "success",
  "result" => undefined
}
```

2. Restart the MTAS software. For further details, refer to *MTAS Node Management Guide*.

## 8.3 CAI3G Logging

Limitation: currently CAI3G logging is not available.

## 8.4 CAI3G Logging Configuration

Limitation: currently CAI3G logging is not available.



## 9 Access CCMP Interface

The XDMS function supports a Centralized Conferencing Manipulation Protocol (CCMP) interface. It is used to administer (create, retrieve, and delete) scheduled conferences per user.

To enable the CCMP interface, it must be unlocked using the `mtasXdmsCcmpAdministrativeState` parameter on the `MtasXdms` MO.

The URI for the CCMP interface on the MTAS XDMS is as follows:

```
http://<hostname>:8096/mtasccmp/<service-number>
```

The parameter `<hostname>` is the hostname of the MTAS node running the XDMS function. If a dedicated VIP address is used for Ut traffic, then use Ut VIP address as hostname. The `<service-number>` is the tel URI of the service number of the scheduled conference service.