

MTAS XDMS Management Guide

MTAS

USER GUIDE

Copyright

© Ericsson AB 2016, 2017. All rights reserved. No part of this document may be reproduced in any form without the written permission of the copyright owner.

Disclaimer

The contents of this document are subject to revision without notice due to continued progress in methodology, design and manufacturing. Ericsson shall have no liability for any error or damage of any kind resulting from the use of this document.

Trademark List

All trademarks mentioned herein are the property of their respective owners. These are shown in the document Trademark Information.



Contents

1	Introduction	1
1.1	Prerequisites	1
2	Overview	3
3	Configuration	5
3.1	Diameter Stack Configuration	5
3.2	Sh Interface Configuration	5
4	Optional XDMS Function Parameters Configuration	7
5	Interface Accesses	9
5.1	XDMS Setups	9
5.2	XCAP Handling	10
5.3	Secure CAI3G Interface	10
6	Logging	19





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 available:

- *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 *User Management*.





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 Configuration

3.1 Diameter Stack Configuration

Several of the MTAS-specific parameter values must be configured in the Diameter stack. To configure the Diameter stack instance for the XDMS function, refer to *MTAS Subscriber Data Management Guide*.

3.2 Sh Interface Configuration

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`. For more information, refer to *MTAS Subscriber Data Management Guide*.

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





4 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)*.

If a parameter is changed, it is a delay of 5 seconds until the new value is reflected.





5 Interface Accesses

The XDMS function supports CAI3G and Ut interfaces, which includes the protocol XML Configuration Access Protocol (XCAP).

CAI3G

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)*.

Note: If Service Profile is used, the MMTel AS Voice Base license must be installed.

Ut, protocol

XCAP

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)*.

To use the Ut interface, the MMTel AS Voice Base license must be installed.

Note: If Service Profile is used, the MMTel AS Voice Base license must be installed.

The XCAP protocol for MMTel Telephony AS allows the subscriber to manipulate their PSTN/ISDN Simulation Services data. For more information, refer to [RFC 4825](#).

5.1 XDMS Setups

The XCAP Root URI does not correspond to an actual resource on an XCAP server. Actual resources are created by appending additional path information to the XCAP Root URI. For more information on XCAP Root, refer to [RFC 4825](#).

The URI for the XCAP Root (interface) on the MMTel AS is as follows:



```
http://<platform-vip>:8090/mtasxdms
```

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

HTTP

```
http://<cai3g-vip4>:8095/axis2/services/CAI3G
```

HTTPS

```
https://<cai3g-vip4>:8443/axis2/services/CAI3G
```

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

HTTP

```
http://<cai3g-vip4>:8095/mtasstas
```

HTTPS

```
https://<cai3g-vip4>:8443/mtasstas
```

Note: If IPv6 is used, <ut-vip6> and <cai3g-vip6> must be used. That is, a numeric IPv6 address, the address must be enclosed in brackets (for example: [2000::4:66]).

5.2 XCAP Handling

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://<ut-vip4>:8090/mtasxdms/simservs.ngn.etsi.org/users/  
<subscriber_uri>/simservs.xml
```

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://<ut-vip4>:8090/mtasxdms/xcap-caps/global/index
```

Note: If IPv6 is used, <ut-vip6> must be used. That is, a numeric IPv6 address, the address must be enclosed in brackets (for example: [2000::4:66]).

5.3 Secure CAI3G Interface

The XDMS function supports to secure the operator CAI3G interface (HTTPS).



The default setting in `xdms.properties` is as follows:

```
[mySsl]
ciphers: HIGH:!aNULL:!eNULL:!EXPORT:!DES:!3DES:!RC4:!MD5:kRSA:!DHE
clientAuth: false
keyPass: xdmsspass
keystoreFile: /opt/mtas/xdms/config/.xdmskeystore
sslEnabledProtocols: TLSv1.1,TLSv1.2
sslProtocol: TLS
```

Note: The set values are valid both for IPv4 and IPv6 and must be set in the `xdms.properties`.

The following handlings are described:

- Update SSL Protocols
- Handling of Certificates
- Update Ciphers
- Update keyPass
- Update other Parameters
- Apply Modification

5.3.1 Update SSL Protocols

The default value is `sslEnabledProtocols: TLSv1.1, TLSv1.2`.

The SSL protocols indicate which protocol can be used for communicating with clients. Acceptable values are `SSLv2`, `SSLv3`, `TLSv1`, `TLSv1.1`, `TLSv1.2`, and any combination of these protocols is comma-separated.

Both `SSLv2` and `SSLv3` protocols are inherently unsafe and the BEAST problem exist in `TLSv1`.

If another value is required for the SSL protocol, the value must be changed.

To set SSL protocols:

1. Log on to controller as maintenance user using Secure Shell (SSH):

```
ssh <maintenance_user>@<OAM VIP>
```

2. Change to directory:

```
cd /cluster/storage/system/config/mtas/
```

3. Verify that the `xdms.properties` file exists:

```
ls xdms.properties
```



If the file does not exist, perform the procedure in Section 5.3.4 Add Default XDMS Properties on page 15.

4. Update the `xdms.properties` file and save it.

Example if only SSL protocol `TLSv1.2` is allowed:

```
sslEnabledProtocols: TLSv1.2
```

5. For the changes to take effect, Tomcat must be restarted on all payloads, see Section 5.3.7 Apply Modification on page 17.

5.3.2 Handling of Certificates

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:

- Delete CAI3G Certificate
- Create a new self-signed CAI3G Certificate
- Import a trusted Certificate
- List stored Certificates
- Validating Certificate

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.

Note: Default pass phrase is `xdmypass`. If customer has generated own key, passphrase can be different, In this case, check "keyPass" in file `xdms.properties`.

5.3.2.1 Delete CAI3G Certificate

To delete the CAI3G certificate:

1. Log on to controller as maintenance user using Secure Shell (SSH):

```
ssh <maintenance_user>@<OAM VIP>
```

2. Delete the old CAI3G certificate:

```
sudo /usr/java/latest/bin/keytool -delete -alias  
CAI3G -keypass xdmypass -storepass xdmypass -keystore  
/cluster/storage/system/config/mtas/.xdmskeystore
```




3. For the changes to take effect, Tomcat must be restarted on all payloads, see Section 5.3.7 Apply Modification on page 17.

5.3.2.2 Create a New Self-Signed CAI3G Certificate

If there is a CAI3G certificate already stored in the XDMS keystore, remove it, see Section 5.3.2.1 Delete CAI3G Certificate on page 12. To list the available certificates in the XDMS keystore, see Section 5.3.2.4 List Stored Certificates on page 14.

To create a new self-signed CAI3G certificate:

1. Log on to controller as maintenance user using Secure Shell (SSH):

```
ssh <maintenance_user>@<OAM VIP>
```

2. Generate a new self-signed CAI3G certificate:

```
sudo /usr/java/latest/bin/keytool -genkey -alias  
CAI3G -storepass xdmypass -keypass xdmypass -keystore  
/cluster/storage/system/config/mtas/.xdmskeystore
```

3. Enter the certificate data.
4. For the changes to take effect, Tomcat must be restarted on all payloads, see Section 5.3.7 Apply Modification on page 17.

5.3.2.3 Import a Trusted Certificate

If there is a CAI3G certificate already stored in the XDMS keystore, remove it, see Section 5.3.2.1 Delete CAI3G Certificate on page 12. For listing the available certificates in the XDMS keystore, see Section 5.3.2.4 List Stored Certificates on page 14.

To import a trusted certificate:

1. Copy the trusted certificate to the cluster:

```
sftp <maintenance_user>@<OAM VIP>  
lcd<local path of the certificate file>  
cd /cluster/storage/system/config/mtas  
put <cacert file>  
exit
```

2. Log on to controller as maintenance user using Secure Shell (SSH):

```
ssh <maintenance_user>@<OAM VIP>
```

3. Import the trusted CAI3G certificate:

```
sudo /usr/java/latest/bin/keytool -import -alias  
CAI3G -storepass xdmypass -keypass xdmypass -keystore
```



```
/cluster/storage/system/config/mtas/.xdmskeystore -file
/cluster/storage/system/config/mtas/<cacert file>
```

4. Examine certificate data, enter **yes** if trusted.
5. For the changes to take effect, Tomcat must be restarted on all payloads, see Section 5.3.7 Apply Modification on page 17.

5.3.2.4 List Stored Certificates

To list the stored certificates:

1. Log on to controller as maintenance user using Secure Shell (SSH):

```
ssh <maintenance_user>@<OAM VIP>
```

2. List the stored certificate:

```
/usr/java/latest/bin/keytool -list -v -storepass
xdmsspass -keystore /cluster/storage/system/config/mta
s/.xdmskeystore
```

5.3.2.5 Validate Certificate

To validate the certificate:

1. Log on to controller as maintenance user using Secure Shell (SSH):

```
ssh <maintenance_user>@<OAM VIP>
```

2. Validate the certificate:

The following information is extracted from file:

```
sudo /usr/java/latest/bin/keytool -printcert -v -file
/cluster/storage/system/config/mtas/xdms.crt
```

For example: Valid... until: Sat Oct 31 13:18:28 CET 2026

Check the date to ensure that the certificate is valid.

5.3.3 Update Ciphers

The default value is:

```
ciphers: HIGH:!aNULL:!eNULL:!EXPORT:!DES:!3DES:!RC4:!M
D5:kRSA:!DHE
```

Note: It is not recommended to use 3DES cipher because of SWEET32 problem.

Java treats the order in which ciphers are defined as an order of preferences.



If another value is needed for ciphers, the value must be changed.

To set ciphers:

1. Log on to controller as maintenance user using Secure Shell (SSH):

```
ssh <maintenance_user>@<OAM VIP>
```

2. Change to directory:

```
cd /cluster/storage/system/config/mtas/
```

3. Verify that the `xdms.properties` file exists:

```
ls xdms.properties
```

If the file does not exist, perform the procedure in Section 5.3.4 Add Default XDMS Properties on page 15.

4. Update the `xdms.properties` file and save it.

Example if only ciphers AES256-GCM-SHA384 is allowed:

```
ciphers: AES256-GCM-SHA384
```

5. For the changes to take effect, Tomcat must be restarted on all payloads, see Section 5.3.7 Apply Modification on page 17.

5.3.4 Add Default XDMS Properties

1. Copy the file `/opt/mtas/xdms/config/xdms.properties` if not existing from one on each Payload Node, where x is the number of the corresponding Payload Node.

```
scp <PL-x>:/opt/mtas/xdms/config/xdms.properties \
/cluster/storage/system/config/mtas/
```

Example from PL-3:

```
scp <PL-3>:/opt/mtas/xdms/config/xdms.properties \
/cluster/storage/system/config/mtas/
```

2. Change the mod:

```
chmod 644 \
/cluster/storage/system/config/mtas/xdms.properties
```

5.3.5 Update keyPass

The default value is:

keyPass = "xdmsspass"

If another keyPass is needed, the value must be changed.



To set the keyPass:

1. Log on to controller as maintenance user using Secure Shell (SSH):

```
ssh <maintenance_user>@<OAM VIP>
```

2. Change to directory:

```
cd /cluster/storage/system/config/mtas/
```

3. Verify that the `xdms.properties` file exists:

```
ls xdms.properties
```

If the file does not exist, perform the procedure in Section 5.3.4 Add Default XDMS Properties on page 15.

4. Update `xdms.properties` file and save it.

Example if keyPass is updated to “newpass”:

```
keyPass: "newpass"
```

5. For the changes to take effect, Tomcat must be restarted on all payloads, see Section 5.3.7 Apply Modification on page 17.

5.3.6 Update Other Parameters

The following parameters have default values set according to Apache Tomcat documentation refer to <http://tomcat.apache.org/tomcat-8.0-doc/>, but can be modified:

- algorithm
- allowUnsafeLegacyRenegotiation
- clientCertProvider
- crlFile
- keyAlias
- keystorePass
- keystoreProvider
- keystoreType
- sessionCacheSize
- sessionTimeout
- trustMaxCertLength
- truststoreAlgorithm



- `truststoreFile`
- `truststorePass`
- `truststoreProvider`
- `truststoreType`

If any of these parameters need another value, it must be added in `xdms.properties` file. An example follows, but it is the same procedure for the other parameters.

To add “algorithm”:

1. Log on to controller as maintenance user using Secure Shell (SSH):

```
ssh <maintenance_user>@<OAM VIP>
```

2. Change to directory:

```
cd /cluster/storage/system/config/mtas/
```

3. Verify that the `xdms.properties` file exists:

```
ls xdms.properties
```

If the file does not exist, perform the procedure in Section 5.3.4 Add Default XDMS Properties on page 15.

4. Update the `xdms.properties` file and save it.

Example: To set algorithm with value “new value”.

5. For the changes to take effect, Tomcat must be restarted on all payloads, see Section 5.3.7 Apply Modification on page 17.

5.3.7 Apply Modification

To restart the MMAS traffic instances:

1. Check the DN of the MMAS instances:

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

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
```



2. Restart the instances one by one on each Payload Node, where x is the number of the corresponding Payload Node, and DN is the identity of the MMAS instance, as queried in Step 1.

```
ssh <emergency user>@<PL-x>  
sudo amf-adm restart <DN>
```

Example:

```
ssh mtasuser01@PL-3  
sudo amf-adm restart safComp=ERIC-MMAS-COMP-0,safSu  
=ERIC-MMAS-SU-0,safSg=SG-traffic,safApp=ERIC-MMAS-APP
```

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



6 Logging

The XDMS logging is described in *MTAS Troubleshooting Guideline*. For information about how to collect the logs, refer to *Data Collection Guideline for MTAS*.