

Operation and Maintenance MAS

Operation Directions

Copyright

© Ericsson AB 2000-2008 - All Rights Reserved

Disclaimer

The information in this document is subject to change without notice.

Ericsson makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Ericsson shall not be liable for errors contained herein nor for incidental or consequential damages in connection with the furnishing, performance or use of this material.

Contents

1	Introduction	1
1.1	Scope	1
1.2	Audience	1
1.3	Related Documents	1
1.4	Document Conventions	2
2	About the Component	3
2.1	Internal Architecture	3
2.2	SNMP Environment	4
3	Operation	5
3.1	Start Component	8
3.2	Stop Component	8
3.3	Restart Component	8
3.4	Check Component Status	9
3.5	Enable/Disable Autostart	10
3.6	Lock/Unlock/Shutdown Component	12
3.7	Viewmib	13
3.8	Version	15
3.9	Reloadconfig	15
3.10	MCR registration (or update)	16
3.11	MCR unregistration	16
3.12	Check that MAS is registered in the SSP	16
3.13	Monitor tool	16
3.14	Testnumber	21
3.15	Installapp	22
3.16	Viewapp	22
3.17	Uninstallapp	22
3.18	Installmcp	23
3.19	Viewmcp	23
3.20	Uninstallmcp	23
4	Process Supervision	25
5	Service Administration	27

5.1	Configure Release Cause Mappings	27
5.2	Configure user provisioning	28
5.3	Call Trace	28
5.4	Configure media codec	28
5.5	Configure silence detection	29
6	Preventive Maintenance	33
6.1	Back Up	33
6.2	View Static Data in MIB	33
6.3	View Dynamic Data in MIB	35
6.4	Manage Disk Space	36
7	Corrective Maintenance	37
7.1	Monitor Traps and Correct Faults	37
7.2	Restore Data	37
7.3	Inspect and Manage Log Files	38
7.4	Change Configuration Parameters in MIBs	42
7.5	Change Configuration Parameters in File	43

1 Introduction

1.1 Scope

This document provides information on operation and maintenance specific for the Media Access Server (MAS).

1.2 Audience

The document is primarily written for O&M personnel.

Part of the document may also be useful for personnel responsible for software installation.

1.2.1 Prerequisite Knowledge

The readers of this document should have a good understanding of Messaging-over-IP and good knowledge and experience of:

- IP based networks
- Protocols SIP, HTTP, IMAP, LDAP, RADIUS, RADIUS-MA, SMTP, SNMP
- Messaging Telephony Gateway, MTG
- UNIX

Note: Operation and maintenance of the system is only to be performed by personnel authorized through having attended the corresponding courses.

1.3 Related Documents

The following documents also contain information related to the component, or are referred to from this document:

- Messaging-over-IP Overview
- Installation Guide MAS
- Solaris Documentation (<http://docs.sun.com>) *System Administration Guide, Solaris 10*

1.4 Document Conventions

1.4.1 Notational Conventions

This document uses the following notational conventions:

Bold font style is used for emphasis, to indicate keywords and buttons.

Italic font style is used for references, window/page/menu titles and specific terms.

Bold Monospace font is used to describe user input.

Monospace font is used for code, paths and on-screen computer output.

Square brackets “[]” are used to enclose parameters that are optional.

Curly brackets “{ }” are used to enclose parameter values given as examples.

Less than and greater than characters “< >” are used to enclose variable names.

Backslash “\” at the end of a line means that this line is continued onto the next line.

The dollar sign “\$” is the UNIX Korn (and Bourne) shell prompt.

The percent sign “%” is the UNIX C shell prompt.

The number sign “#” is the superuser prompt.

The number sign “#” is also used for comments.

A star “*” matches zero or more characters.

A question mark “?” matches one character.

A tilde “~” means the home directory of the current user.

<MAS_HOME> is the installation directory for the MAS, for example /apps/mas.

2 About the Component

For an overview of the MAS component, see *Messaging-over-IP Overview*.

The MAS component provides a VoiceXML and CCXML platform with SIP and XMP interfaces.

2.1 Internal Architecture

The MAS consists of several parts as shown in the figure below.

Media Content Manager and Media	Handles all media content packages such as voice and video files and various textual information.
Execution Engine	Implements the VoiceXML and CCXML executor. It also serves as an access to platform functions via the Platform Access interface.
Media Translation Manager	Implements the interface to external TTS and ASR services using XMP (via Service Request Manager) and RTSP/MRCP.
Call Manager and Stream	Implements the interfaces to MTG via SIP and RTP.
Service Request Manager	Handles inbound and outbound service requests using XMP.
Profile Manager	Handles subscriber profiles retrieved from external user directories using LDAP.
Provision Manager	Handles creation of subscribers in the system via the CAI interface.
Mailbox	Manages messages for a subscriber using IMAP.
Message Sender	Sends messages to a mail server using SMTP.
Number Analysis	Handles rule based (regular expressions) number analysis.
External Component Register	Retrieves information from an external component register using LDAP.

Traffic Event Sender	Send traffic related events using Radius-MA or by SMTP (via Message Sender).
Log Manager and Configuration Manager	Handles logging and configuration.
O&M Manager	Implements various O&M related functionality such as SNMP agent, backup-restore, start-restart etc.

2.2 SNMP Environment

The MAS-MIB has the following OID:

enterprises.24261.1.1

For instructions on how to change a MIB parameter value, see the documentation supplied with MIB- or SNMP management tool.

MIB parameters can be viewed through EMANATE's web interface using an URL like: "http://<hostname>:<port>/subtree/mas"

where <hostname> is the host where the MAS is running, and <port> (default 20280) is defined by EMANATE. User ID and password (default admin/admin) are also defined by EMANATE.

3 Operation

The MAS component uses an initialization and termination script named “/etc/init.d/rc.mas”. The “rc.mas” script can be executed with the following arguments.

start

stop

restart

status

enableautostart

disableautostart

lock

unlock

shutdown

viewmib

version

reloadconfig

monitor

testnumber

register

unregister

installapp

viewapp

uninstallapp

installmcp

viewmcp

uninstallmcp

A blank or illegal argument will print out the list of valid arguments

The “start” argument will start the MAS component. The command will be ignored if the component is already started.

The “stop” argument will stop the MAS component. The command will be ignored if the component is already stopped.

The “restart” argument will restart the MAS component. If the component is already running, the component will first be stopped and then started.

The “status” argument will return the status of the MAS component.

The “enableautostart” argument will enable automatic restart of the MAS component at boot and process termination.

The “disableautostart” argument will disable automatic restart of the MAS application at boot and process termination.

The “lock” argument will lock the MAS component. The MAS unregisters from the SSP (if an SSP is used in the deployment) and all end users will be immediately disconnected. In state locked, the component will not perform any of its traffical tasks.

The “unlock” argument will unlock the MAS component. In state unlocked the component performs its assigned tasks.

The “shutdown” argument will shutdown the MAS component. Shutdown is a temporary state. MAS waits for all callers to end their calls and then changes the state to locked. A time out value can be supplied to the shutdown argument. The MAS will then only wait for the calls to disconnect until the time out is reached, then it will be forced to change the state to locked.

The “viewmib” argument will display the MAS MIB.

The “version” argument will show the version name of the MAS component.

The “reloadconfig” argument will re-load the configuration parameters during run time.

The "testnumber" argument will test numberanalyzer rules.

The “register” argument will register the MAS component, the application package and the media content packages to the MCR.

The “unregister” argument will unregister the MAS component, the application package and the media content packages from the MCR.

The "monitor" argument will start the monitor tools.

The "installapp" argument will install an application package on the MAS.

The "viewapp" argument will show which application package is installed on the MAS.

The "uninstallapp" argument will uninstall the application package.

The "installmcp" argument will install a media content package on the MAS.

The "viewmcp" argument will show which media content packages are installed on the MAS.

The "uninstallmcp" argument will uninstall a media content package.

Operation includes tasks as described in:

- Section 3.1 on page 8
- Section 3.2 on page 8
- Section 3.3 on page 8
- Section 3.4 on page 9
- Section 3.5 on page 10
- Section 3.6 on page 12
- Section 3.7 on page 13
- Section 3.8 on page 15
- Section 3.9 on page 15
- Section 3.10 on page 16
- Section 3.11 on page 16
- Section 3.13 on page 16
- Section 3.14 on page 21
- Section 3.15 on page 22
- Section 3.16 on page 22
- Section 3.17 on page 22
- Section 3.18 on page 23
- Section 3.18 on page 23
- Section 3.20 on page 23

3.1 Start Component

To start the MAS component:

1. Become superuser (root).

```
#su -
```

2. Run the "rc.mas " script with "start" as argument.

```
#/etc/init.d/rc.mas start
```

The result should be the following:

```
"mas:<host name>:<action> ok"
```

The example below shows a typical output.

```
# /etc/init.d/rc.mas start
mas:mashost1:result ok started
```

Example 1 Output, rc.mas start

3.2 Stop Component

To stop the MAS component:

1. Become superuser (root).

```
#su -
```

2. Run the "rc.mas "script with "stop" as argument.

```
#/etc/init.d/rc.mas stop
```

The result should be the following:

```
"mas:<host name>:<action> ok"
```

The example below shows a typical output.

```
# /etc/init.d/rc.mas stop
mas:mashost1:result ok stopped
```

Example 2 Output, rc.mas stop

3.3 Restart Component

To restart the MAS component:

1. Become superuser (root).

```
#su -
```

2. Run the "rc.mas" script with "restart" as argument.

```
#/etc/init.d/rc.mas restart
```

The result should be the following:

```
"mas:<host name>:<action> ok"
```

The example below shows a typical output.

```
# /etc/init.d/rc.mas restart
mas:mashost1:result ok started
```

Example 3 Output, rc.mas restart

3.4 Check Component Status

To check the status of the MAS component:

1. Become superuser (root).

```
#su -
```

Run the "rc.mas" script with "status" as argument.

```
#/etc/init.d/rc.mas status
```

The result can be one of the following:

```
"mas:<host name>:status starting (autostart) "
```

or

```
"mas:<host name>:status started (autostart) (unlocked) (enabled)"
```

or

```
"mas:<host name>:status started (not autostart) (locked) (enabled)"
```

or

```
"mas:<host name>:status stopped (autostart)"
```

or

```
"mas:<host name>:status stopped (not autostart)"
```

Syntax description

```
mas:<host name>:status <status> (<autostart>) ([opstate]) ([admstate])
```

- Status can have one of the following values

starting	The MAS process is starting up.
started	The MAS process is started.
stopped	The MAS process is not running.
<ul style="list-style-type: none"> autostart can have one of the following values. 	
autostart	Indicates that autostart is enabled
not autostart	Indicates that autostart is disabled
<ul style="list-style-type: none"> admstate can have one of the following values. 	
locked	The MAS administrative state is locked.
unlocked	The MAS administrative state is unlocked.
shutdown	The MAS administrative state is shutdown.
<ul style="list-style-type: none"> opstate can have one of the following values. 	
enabled	The MAS operational state is in enabled state.
disabled	The MAS operational state is in disabled state.

The examples below show some typical outputs.

```
/etc/init.d/rc.mas status
mas:mashost1:status started (autostart) (unlocked) (enabled)
```

Example 4 Output, rc.mas status (after start)

```
/etc/init.d/rc.mas status
mas:mashost1:status stopped (autostart)
```

Example 5 Output, rc.mas status (after stop)

```
/etc/init.d/rc.mas status
mas:mashost1:status started (autostart) (unlocked) (enabled)
```

Example 6 Output, rc.mas status (after enable autostart)

```
/etc/init.d/rc.mas status
mas:mashost1:status started (not autostart) (unlocked) (enabled)
```

Example 7 Output, rc.mas status (after disable autostart)

3.5 Enable/Disable Autostart

To make the watchdog monitor and restart the MAS process the autostart mode must be enabled.

If autostart is **enabled**:

- The MAS process will be restarted if MAS process is terminated.

- The MAS process will be restarted if host is rebooted

If autostart is **disabled**

- The MAS process will **not** be restarted if the MAS process is terminated.
- The MAS process will **not** be restarted if host is rebooted

The recommended setting is that autostart is enabled.

3.5.1 Enable Autostart

1. Become superuser (root).

```
#su -
```

2. Run the “rc.mas” script with “enableautostart” as argument.

```
#/etc/init.d/rc.mas enableautostart
```

The result should be the following:

```
“mas:<host name>:result ok”
```

The example below shows a typical output.

```
mas:mashost1:result ok
```

Example 8 Output, rc.mas enableautostart

3.5.2 Disable Autostart

1. Become superuser (root).

```
#su -
```

2. Run the “rc.mas” script with “disableautostart” as argument.

```
#/etc/init.d/rc.mas disableautostart
```

The result should be the following:

```
“mas:<host name>:result ok”
```

The example below shows a typical output.

```
mas:mashost1:result ok
```

Example 9 Output, rc.mas disableautostart

3.6 Lock/Unlock/Shutdown Component

The following parameter in the MAS-MIB controls the behavior of the started component.

Table 1

Parameter	Description	Value
MASAdministrativeState	In state unlocked the component performs its traffic related tasks.	1=unlocked
In state locked, the component will not perform all its traffic related tasks, even if it is up and running.	2=locked	
Shutdown is a temporary state. The MAS waits for all callers to end their calls and then changes the state to locked.	3=shutdown	

3.6.1 Lock

When the MAS is locked, it will immediately disconnect all end users. To lock the MAS component:

1. Become superuser (root).

```
#su -
```

2. Run the "rc.mas" script with "lock" as argument.

```
#!/etc/init.d/rc.mas lock
```

The result should be the following:

```
"mas:mashost1:result ok"
```

The example below shows a typical output.

```
mas:mashost1:result ok
```

Example 10 Output, rc.mas lock

3.6.2 UnLock

To unlock the MAS component:

1. Become superuser (root).


```
# su -
```

2. Run the “rc.mas” script with “unlock” as argument.

```
#/etc/init.d/rc.mas unlock
```

The result should be the following:

```
“mas:mashost1:result ok”
```

The example below shows a typical output.

```
mas:mashost1:result ok
```

Example 11 Output, rc.mas unlock

3.6.3

Shutdown

To shutdown the MAS component:

1. Become superuser (root).

```
#su -
```

2. Run the “rc.mas” script with “shutdown” and a timeout value as arguments.

```
#/etc/init.d/rc.mas shutdown 900
```

The MAS will now wait 900 seconds for all calls to disconnect and then change the state to locked. The script will not return before all calls are disconnected. The result should be the following:

```
“mas:mashost1:result ok”
```

The time out value is optional. If no timeout value is given, the MAS will wait for the calls to disconnect without time limit and return when all calls is disconnected.

The example below shows a typical output.

```
mas:mashost1:result ok
```

Example 12 Output, rc.mas shutdown

3.7

Viewmib

To view the mib of the MAS component:

1. Become superuser (root).

```
#su -
```

2. Run the “rc.mas” script with “viewmib” as argument.

#/etc/init.d/rc.mas viewmib

The example below shows a typical output.

```
# /etc/init.d/rc.mas viewmib
masName.0 = mas@mashost1.domain
masVersion.0 = MAS_R4A.066
masOperationalState.0 = enabled(1)
masAdministrativeState.0 = unlocked(1)
masInstallDate.0 = 2007-Jun-4,17:37:21.0,+1:0
masCurrentUpTime.0 = 1 day, 19:47:23.54
masAccumulatedUpTime.0 = 2 days, 03:03:45.57
masConfigurationState.0 = ok(2)
masLastConfigurationUpdateTime.0 = 2007-Jun-4,17:37:21.0,+1:0
masLastConfigurationUpdateTicks.0 = 1 day, 19:47:23.70
masServiceEnablerStatisticsIndex.1 = 1
masServiceEnablerStatisticsIndex.2 = 2
masServiceEnablerStatisticsProtocol.1 = sip:mashost1.domain:5060
masServiceEnablerStatisticsProtocol.2 = xmp:mashost1.domain:8080
masServiceEnablerStatisticsMaxConnections.1 = 65
masServiceEnablerStatisticsMaxConnections.2 = 20
masConnectionStatisticsIndex.1.1 = 1
masConnectionStatisticsIndex.1.2 = 2
masConnectionStatisticsIndex.1.3 = 3
masConnectionStatisticsIndex.1.4 = 4
masConnectionStatisticsIndex.2.1 = 1
masConnectionStatisticsIndex.2.2 = 2
masConnectionStatisticsIndex.2.3 = 3
masConnectionStatisticsIndex.2.4 = 4
masConnectionStatisticsType.1.1 = voice
masConnectionStatisticsType.1.2 = video
masConnectionStatisticsType.1.3 = service_request
masConnectionStatisticsType.1.4 = unknown
masConnectionStatisticsType.2.1 = voice
masConnectionStatisticsType.2.2 = video
masConnectionStatisticsType.2.3 = service_request
masConnectionStatisticsType.2.4 = unknown
masConnectionStatisticsConnections.1.1 = 0
masConnectionStatisticsConnections.1.2 = 0
masConnectionStatisticsConnections.1.3 = 0
masConnectionStatisticsConnections.1.4 = 0
masConnectionStatisticsConnections.2.1 = 0
masConnectionStatisticsConnections.2.2 = 0
masConnectionStatisticsConnections.2.3 = 0
masConnectionStatisticsConnections.2.4 = 0
masConnectionStatisticsPeakConnections.1.1 = 20
masConnectionStatisticsPeakConnections.1.2 = 0
masConnectionStatisticsPeakConnections.1.3 = 0
masConnectionStatisticsPeakConnections.1.4 = 20
masConnectionStatisticsPeakConnections.2.1 = 0
masConnectionStatisticsPeakConnections.2.2 = 0
masConnectionStatisticsPeakConnections.2.3 = 0
masConnectionStatisticsPeakConnections.2.4 = 0
masConnectionStatisticsPeakTime.1.1 = 2007-Jun-4,17:40:34.1,+1:0
masConnectionStatisticsPeakTime.1.2 = 2007-Jun-4,17:37:21.0,+1:0
masConnectionStatisticsPeakTime.1.3 = 2007-Jun-4,17:37:21.0,+1:0
masConnectionStatisticsPeakTime.1.4 = 2007-Jun-4,17:40:08.8,+1:0
masConnectionStatisticsPeakTime.2.1 = 2007-Jun-4,17:37:21.0,+1:0
masConnectionStatisticsPeakTime.2.2 = 2007-Jun-4,17:37:21.0,+1:0
masConnectionStatisticsPeakTime.2.3 = 2007-Jun-4,17:37:21.0,+1:0
masConnectionStatisticsPeakTime.2.4 = 2007-Jun-4,17:37:21.0,+1:0
masConnectionStatisticsTotalConnections.1.1 = 27
masConnectionStatisticsTotalConnections.1.2 = 0
masConnectionStatisticsTotalConnections.1.3 = 0
masConnectionStatisticsTotalConnections.1.4 = 13
masConnectionStatisticsTotalConnections.2.1 = 0
masConnectionStatisticsTotalConnections.2.2 = 0
masConnectionStatisticsTotalConnections.2.3 = 0
masConnectionStatisticsTotalConnections.2.4 = 0
masConnectionStatisticsAccumulatedConnections.1.1 = 98
masConnectionStatisticsAccumulatedConnections.1.2 = 0
masConnectionStatisticsAccumulatedConnections.1.3 = 0
masConnectionStatisticsAccumulatedConnections.1.4 = 75
masConnectionStatisticsAccumulatedConnections.2.1 = 0
masConnectionStatisticsAccumulatedConnections.2.2 = 0
```

```

masConnectionStatisticsAccumulatedConnections.2.3 = 0
masConnectionStatisticsAccumulatedConnections.2.4 = 0
masProvidedServiceIndex.1 = 1
masProvidedServiceIndex.2 = 2
masProvidedServiceIndex.3 = 3
masProvidedServiceName.1 = Undefined
masProvidedServiceName.2 = OutdialNotification
masProvidedServiceName.3 = MWINotification
masProvidedServiceStatus.1 = up(1)
masProvidedServiceStatus.2 = up(1)
masProvidedServiceStatus.3 = up(1)
masProvidedServiceHostName.1 = mashost1.domain
masProvidedServiceHostName.2 = mashost1.domain
masProvidedServiceHostName.3 = mashost1.domain
masProvidedServicePort.1 = 5060
masProvidedServicePort.2 = 8080
masProvidedServicePort.3 = 8080
masProvidedServiceZone.1 = unspecified
masProvidedServiceZone.2 = unspecified
masProvidedServiceZone.3 = unspecified

```

Example 13 Output, rc.mas viewmib

3.8 Version

To show the version of the MAS component:

1. Become superuser (root).

```
#su -
```

2. Run the “rc.mas” script with “version” as argument.

```
#/etc/init.d/rc.mas version
```

The result should be the following:

```
“version: <version>”
```

The example below shows a typical output.

```
version: MAS_R2A.026
```

Example 14 Output, rc.mas version

3.9 Reloadconfig

Reloads configuration parameters.

Section 7.5.1 on page 43 describes reloadable parameters.

1. Become superuser (root).

```
#su -
```

2. Run the “rc.mas” script with “reloadconfig” as argument.

```
#/etc/init.d/rc.mas reloadconfig
```

The result should be the following:

```
"mas:mashost1:status ok"
```

The example below shows a typical output.

```
/etc/init.d/rc.mas reloadconfig
mas:mashost1:status ok
```

Example 15 Output, rc.mas reloadconfig

3.10 MCR registration (or update)

To register the MAS component, the application package and the media content packages in the MCR, make sure that the MCR values in <MAS_HOME>/etc/mas.cfg are correct, and the correct MCR credentials exist in mas.xml, see Section 7.5.1.7 on page 57.

1. Become superuser (root).

```
#su -
```

2. Run the "rc.mas" script with "register" as argument.

```
#!/etc/init.d/rc.mas register
```

3.11 MCR unregistration

To unregister the MAS component, the application package and the media content packages from the MCR.

1. Become superuser (root).

```
#su -
```

2. Run the "rc.mas" script with "unregister" as argument.

```
#!/etc/init.d/rc.mas unregister
```

3.12 Check that MAS is registered in the SSP

If an SSP is used, check in the Operate and Maintenance Guide for the SSP regarding how to verify that the MAS is registered in the SSP.

3.13 Monitor tool

This tool is used to monitor sessions and shows call statistics.

Run the "rc.mas" script with "monitor" as argument to start the monitor.

```
#!/etc/init.d/rc.mas monitor
```

The monitor has two views, the Statistics view and the Connections view.

To navigate between these views, press F5 for the Connections view and F6 for the Statistics view.

To scroll up and down in a view, use the up arrow key and the down arrow key.

If the monitor for some reason loses its connection to the MAS, press F9 to restore the connection. The Connection Status in the statusbar shows information about the connection state.

To exit the monitor, press the escape key.

3.13.1 Statistics view

The statistic view contains information about counters for each service enabler registered

Figure 1

For each service enabler, the current and total connections will be presented. For the XMP service enabler, "connection" is equivalent to service request, and for the SIP service enabler, "connection" is equivalent to call.

- Current connections
- Total connections

Number of connections since the last restart

Current and total connections present statistics for the following connection types.

- Voice

Shows the number of active voice connections over SIP.
- Video

Shows the number of active video connections over SIP.
- Service Request

Shows the number of active service request connections over XMP.
- Unknown

Shows the number of active Unknown connections over SIP.

For each connection type, the following counters can be displayed.

For current connections.

- Inbound

The number of inbound connections per type.

- Outbound

The number of outbound connections per type.

For total connections

- Total connections

Number of connections since the last restart (with the same definition as Current connections)

- Total inbound connections

Out of Total connections the number of inbound connections.

- Total outbound connections

Out of Total connections the number of outbound connections

- Total connected connections

Out of Total connections the number of connections that have been answered since the last restart.

- Total connected inbound connections

Out of Total answered connections the number of inbound connections

- Total connected outbound connections

Out of Total answered connections the number of outbound connections

- Total abandoned connections

Number of abandoned connections since restart.

A connection is considered "far end abandoned" when no RTP-data has been received on an inbound media stream during a certain time (configured in the stream group, streamabandonedms).

A connection is considered "near end abandoned" when the application has not answered an inbound call within a certain time (configured in the callmanager group, callnotacceptedtimer).

- Total abandoned inbound connections

Out of Total abandoned connections the number of inbound connections.

- Total abandoned outbound connections
Out of Total abandoned connections the number of outbound connections.
- Total failed connections
Numbers of connections since restart that where rejected by the near or far end for any reason.
- Total failed inbound connections
Out of Total rejected connections the number of inbound connections.
- Total failed outbound connections
Out of Total rejected connections the number of outbound connections.

3.13.2 Connection view

The connection view contains information about current connections

This view has two pages. Press F2 for page 1 and F3 for page 2.

Figure 2

For each connection, the following values is displayed.

- Session Id (Sid)
The identity of the session using this connection. This information can be used to see which connections are used for a particular session.
This id can also be found in the logfile.
- Service (Service)
The Service using this connection
- Session initiator (SessInit)
The initiator of the session that created this connection. One of:
 - "xmp <xmpclientid>" for Service Requests
 - "<session id>" when initiated by another session
 - "sip" when initiated by a telephony session.
- Connection type (Tp)
One of Voice or Video

- Connection State (St)

The state of the connections.

The connection state can have the following values

- ALERTING (a). This is an inbound call from the MAS perspective. The call has not yet been answered by the MAS.
- PROGRESSING (p). This is an outbound call from the MAS perspective. The call has not yet been answered by the remote end.
- CONNECTED (c). This is either an inbound or an outbound call from the MAS perspective. The call is connected.
- FAILED (f). This is either an inbound or an outbound call from the MAS perspective. The call failed to reach connected state.
- DISCONNECTED (d). This is either an inbound or an outbound call from the MAS perspective. The call is disconnected.
- ERROR (e). This is either an inbound or an outbound call from the MAS perspective. An abnormal condition has been detected on the call.
- UNKNOWN (""). This call has unknown connection state. This state should be very temporary.

- Direction (Di)

One of Inbound(in) or Outbound (out)

- Outbound activity (O)

One of Idle(i) or Play(p).

- Inbound activity (I)

One of Idle(i) or Record(r).

- ANI

The telephone number of the calling party

- DNIS

The telephone number of the called party

- RDNIS

The telephone number of the redirected called party

- Far end connection properties (FarECP)

A list of the far end IP address and ports (for the outbound traffic) used for this connection and protocol. E.g. "RTP:123.4.56.127:2080:2082, SIP:123.4.56.24:5060"

Page 1 has the following columns.

- Session Id
- Service
- Connection Type
- ConnectionState
- Direction
- Outbound activity
- Inbound activity
- ANI
- DNIS
- RDNIS

Page 2 has the following columns

- Session Id
- Session Initiator
- Far end connection properties

3.14 Testnumber

The testnumber is used to test that number analysis is working correctly.

See the Application Documentation for more details on number analyzer.

```
#/etc/init.d/rc.mas testnumber <filename> <rule> <number>
<aninumber>
```

The options are explained here:

- <filename>

The name and location of the file containing the rule to be tested (normally numberanalyzer.xml)

- <rule>

The name of the rule to test

- <number>
The number to test
- <aninumber>
The ANI number including region code (optional)

3.15 Installapp

To install an application package on the MAS component:

- a Become superuser (root):

```
# su -
```

- b Install the application package by issuing the command:

```
# /etc/init.d/rc.mas installapp <app_file>
```

For a more detailed instruction, see the MAS Installation Guide.

3.16 Viewapp

To view which application package that is installed on the MAS component:

- Become super-user (root):
- ```
su -
```
- View the installed application package by issuing the command:

```
/etc/init.d/rc.mas viewapp
```

For a more detailed instruction, see the MAS Installation Guide.

## 3.17 Uninstallapp

To uninstall an application package from the MAS component:

- a Become super-user (root):

```
su -
```

- b Uninstall the application package by issuing the command:

```
/etc/init.d/rc.mas uninstallapp <id>
```

For a more detailed instruction, see the MAS Installation Guide.

## 3.18 Installmcp

To install an media content package on the MAS component:

- a Become super-user (root):

```
su -
```

- b Install the media content package by issuing the command:

```
/etc/init.d/rc.mas installmcp <mcp_file>
```

For a more detailed instruction, see the MAS Installation Guide.

## 3.19 Viewmcp

To view which media content packages that are installed on the MAS:

- a Become super-user (root):

```
su -
```

- b View the installed media content packages by issuing the command:

```
/etc/init.d/rc.mas viewmcp
```

For a more detailed instruction, see the MAS Installation Guide.

## 3.20 Uninstallmcp

To uninstall a media content package from the MAS component:

- a Become super-user (root):

```
su -
```

- b Uninstall the media content package by issuing the command:

```
/etc/init.d/rc.mas uninstallmcp <id>
```

For a more detailed instruction, see the MAS Installation Guide.



## 4 Process Supervision

The MAS uses the Solaris 10 Service Management Facility (SMF) for process supervision. The SMF will check if any of the MAS processes is down, if so it starts the process immediately.

See Section 3.5 on page 10 how to enable/disable process supervision.

For more information about SMF services, refer to the Solaris 10 Documentation.



## 5 Service Administration

### 5.1 Configure Release Cause Mappings

When an outdial call cannot be established, the MAS receives a SIP Final Response. A SIP Final Response may or may not contain a Q.850 cause and location for the release in the SIP header Reason. The MAS uses the configuration in the callmanager section to map these responses to Network Status Codes as used by Messaging-over-IP. If a Q.850 cause/location value is found in the SIP Final Response, it is used when mapping, otherwise the SIP response code is used. With an application installed, the Network Status Code is used to inform other components in the system of the result of an outbound call. This information can be used for example to map the result of an outdial call to a specific outdial notification retry scheme.

The default configuration of release cause mappings is shown below:

```
<releasecausemappings defaultnetworkstatuscode="621">
 <mapping name="suppressed" networkstatuscode="614"
 sipresponsecodeintervals="600" q850causeintervals="17"
 q850locationintervals="0"/>
 <mapping name="busy" networkstatuscode="603"
 sipresponsecodeintervals="486" q850causeintervals="17"/>
 <mapping name="noreply" networkstatuscode="610"
 sipresponsecodeintervals="408,480"
 q850causeintervals="18-19"/>
 <mapping name="notreachable" networkstatuscode="613"
 sipresponsecodeintervals="301,403-404,410,484,501-502,603"
 q850causeintervals="1-9,20-23,25-31"/>
 <mapping name="congestion" networkstatuscode="620"
 sipresponsecodeintervals="503"
 q850causeintervals="39-44,46"/>
</releasecausemappings>
```

In the above example, the SIP response code 486, which can be seen as a busy signal in the SIP network is mapped to the Network Status Code 603 and given the name busy.

The Network Status Code for the scenarios described in the example should not be changed and are the default values after an installation, i.e. the Network Status Code for the congestion scenario should always be set to 620 although it is possible to change which SIP Final Responses values and Q.850 cause/location values that generate that code.

see Page 50 for more information.

## 5.2 Configure user provisioning

By default, MCR is used to retrieve the user provisioning host(s) and normally does not need to be manually configured. This section describes how user provisioning can be configured manually, overriding the default MCR setting.

The configuration for user provisioning host(s) is located under the services group in the externalcomponentregister group.

In order to override the MCR settings for user provisioning, set the corresponding `overrideMCR` attribute to `true`. Configure the user provisioning host name and port by using the instance tag located in the provisioning group.

**Note:** To provide redundancy, additional provisioning hosts can be configured. This is done by adding an instance tag for each host, in the provision service group, see Section 7.5.1.7 on page 57. For example:

```
<provisioning overrideMCR="true">
 <instance hostname="provisioninghost1.domain"
 port="2400"/>
 <instance hostname="provisioninghost2.domain"
 port="2400"/>
</provisioning>
```

A provisioning administrator must also be configured in the provisioning group. This is located under the provision group located in the profilemanager group. Set the password to use for all provisioning administrators in the password attribute.

## 5.3 Call Trace

The call trace is a log function that filters the log to show all logs for a specified number/numbers.

See Section 7.5.3.2 on page 85 how to configure call trace.

## 5.4 Configure media codec

The MAS is by default configured for the G.711 u-law codec. This section describes which steps that are needed in order to configure a non-default media codec such as AMR or a-law.

- The `defaultInboundPTTime` and `defaultInboundMaxPTTime` configuration must be adjusted according to Section 7.5.1.15 on page 70.
- The `inboundAudioMedia` and `requiredOutboundAudioMedia` must be adjusted according to Section 7.5.1.4 on page 47.
- The codec part of `contentTypeMapper` must be adjusted according to Section 7.5.1.5 on page 55



- The vva.xml must be updated according to the Operation and Maintenance guide for VVA.
- Make sure that the media content packages that are installed supports the codec that should be used.

**Note:** The above list only describes necessary changes for MAS/VVA. Configuration of other parts in the system, such as gateways may also need to be modified.

**Note:** Configuring a new codec requires a restart of MAS, reload of configuration is not enough.

## 5.5 Configure silence detection

Silence detection in MAS is default disabled for both audio and video calls. When silence detection is enabled it gets enabled for audio calls. Silence detection is only supported for pure audio calls and only for G.711  $\mu$ -law and A-law codecs. It is necessary that the silent audio packets are sent by the gateway or end-point all the time (e.g. in case the user mutes his audio), silence detection will not work if MAS does not receive an audio stream. If no audio packets are received there will be no recording.

MAS supports three silence detection modes:

- No Silence Detection mode
- Static Silence Detection mode
- Dynamic Silence Detection mode

### 5.5.1 No Silence Detection mode

To turn off the Silence Detection, the SilenceDetectionMode must be set to 0. When the Silence Detection is turned off, the recording is continued until one of the following occurs:

- Maximum recording duration is reached.
- Caller has hung up
- Caller has pressed the specified DTMF to interrupt.

### 5.5.2 Static Silence Detection mode

To set the Silence Detection to Static Silence Detection mode, the SilenceDetectionMode parameter must be set to 1. This mode uses the configuration attribute "SilenceThreshold" to decide on the silence frames. Every PCM sample value in an audio frame is compared with this threshold

value. A PCM sample value is deemed to be silent if it is below the threshold. A complete frame is deemed silent if 90% of the samples are silent.

### 5.5.3 Dynamic Silence Detection mode

To set the Silence Detection mode to Dynamic Silence Detection mode, the “SilenceDetectionMode” parameter must be set to 2. In this option, the silence detection works on an adaptive threshold value. Here the threshold value is not for each PCM sample; rather it is for the average energy level of a frame used for detection. In this mode the following configuration parameters are used:

- InitialSilenceFrames
- DetectionFrames
- SilenceDeadband
- SignalDeadband

The initial threshold value is calculated by averaging the energy level of first incoming audio frame(s) of length (in ms) as specified by the attribute “InitialSilenceFrames” at the start of recording. The subsequent incoming frames of length as specified by the attribute “DetectionFrames” are averaged for energy and compared with the current threshold. The default detection frame size is chosen as 10 ms since it gives a better granularity while averaging for energy and better zero crossing rate for voice signal. The threshold value is lowered when a transition occurs from speech to silence and vice versa. The increment in threshold value is less monotonic than the decrement since we want to quickly come back to better silence threshold in the event of transition from noise or speech burst. In order to make silence transition more conservative, we keep a silence dead band as specified by the attribute “SilenceDeadBand”. A transition from speech to silence occurs when consecutive silent (as decided by comparing with the current threshold value) audio frames cross the “SilenceDeadBand” value. The default value is 150 ms. The default value is chosen since the inter word silence and hysteresis on an average speaker is less than 150 ms. Similarly, a “SignalDeadBand” value is used to decide on transition from silence to speech. The speech transition is not constrained and the default speech dead band is 10 ms (that is 1 detection frame size). The transition from silence to speech occurs when we detect consecutive speech audio frames equal to “SignalDeadBand”.

### 5.5.4 Description of Silence Detection Parameters

- The silencedetectionmode is used to set the Silence Mode. MAS supports the following three Silence Modes:
  - Mode 0 - No Silence Detection.
  - Mode 1 - Static Silence Detection mode.
  - Mode 2 - Dynamic Silence Detection mode. Default is 0.

- The `silencethreshold` is the initial threshold value for the energy of a detection frame. The range of this attribute is from 0 to 1200. The unit is in short value equivalent to a 16-bit linear PCM sample. This value is not used in the case of dynamic silence detection. Default is 0.
- The `initialsilenceframes` is used to calculate the initial silence threshold. The value should always be 40.
- The `silencedeadband` is used to decide transition from speech to silence. Default is 150 ms. This value should be a non-zero value. This value determines the number of silence frames to accumulate before we make a transition from speech to silence. Also, after we make this transition, the threshold value changes to a lower value based on the maximum silence level during the period of accumulation of silence packets in the deadband. When fine tuning silence detection, this parameter is likely to be the first candidate to adjust.
- The `signaldeadband` is used to decide transition from silence to speech. Default is 10 ms. It has to be a non zero value and less than the "Silence-Deadband" value above. This is same as silence dead band applicable for silence to speech transition. The lower value of this being same as the detection frame size makes the silence detector more sensitive to speech transition.
- The `detectionframes` determines the number of samples for calculation of average energy in an RTP packet. If an RTP packet is bigger than 10 ms (40 ms as in the case of MTG), it is divided into four packets and the average energy is calculated for each one. Default is 10.
- The `silencedetectiondebuglevel` provides different levels of debug information for the Silence Detection functionality. This is information that is provided in addition to the normal debug log information for Stream. See the appropriate section for enabling debug logging for Stream. Default is `silencedetectiondebuglevel=0`. The parameter can have the following values:
  - `silencedetectiondebuglevel=0`: Will indicate that Silence Detection has occurred and print Silence Detection configuration parameters and attribute values (VoiceXML) received from the application.
  - `silencedetectiondebuglevel=1`: Provides information on e.g. energy levels per RTP packet.
  - `silencedetectiondebuglevel=2`: Provides some additional information when Silence has been detected.
  - `silencedetectiondebuglevel=3`: This level is only used for `SilenceDetectionMode=1` (static) and provides additional information on Sample level.

**Note:** Stream debug log must also be enabled in order for this attribute to have any effect.



## 6 Preventive Maintenance

Preventive maintenance means regularly performed measures to prevent faults from occurring and to reduce the time needed to correct faults.

### 6.1 Back Up

The backup script delivered together with the MAS is intended to be used during System backup. The backup script collects all files that should be backed up into the `/apps/backup/MAS_<date>_backup.tar` file and passes the file location to the System backup which proceeds the backup process. The backup script file `bckmasconfig.sh` is stored in the `/apps/backup` directory

Follow this procedure to back up the MAS component:

1. Login to the host that should be backed up as root.
2. Change to the `/apps/backup` directory.
3. Run `bckmasconfig.sh`

```
./bckmasconfig.sh
```

4. The script will end with path to a tar file containing all backed up files.

```
RESULTFILE /apps/backup/MAS_<date>_backup.tar
```

The following table describes the files that gets backed up with the backup script.

*Table 2 Backup files*

File	Contents
<code>&lt;MAS_HOME&gt;/cfg/*.xml</code>	All installation specific Information, e.g. all hostnames of for example MCR and MUR and application configurations.
<code>/apps/logs/mas/*</code>	All logfiles.

### 6.2 View Static Data in MIB

Static parameter values are not updated by the component during operation. The MAS MIB defines the following static parameters:

Table 3 Static parameters in MAS MIB

Parameter	Type	Description
masName	String	The name of the component as registered in MCR i.e. <component_instancename> @<hostname> e.g. mas1@host1.com
masVersion	String	The version of the component as registered in MCR.
masInstallDate	Date and time	The date the component was installed.
<i>For each Service Enabler, the following parameters are shown</i>		
masServiceEnabler\StatisticsMax\Connection	INTEGER	The maximum number of channels (this is also referred to as the high watermark).
masServiceEnabler\StatisticsProtocol	String	The protocol used by this Service Enabler i.e. a literal name with the name of the protocol e.g. "sip", "xmp" etc.
<i>For each Provided Service, the following parameters are shown</i>		
masProvided\ServiceName	String	The name of the registered service.
masProvided\ServiceHostName	String	The host providing the service.
masProvided\ServicePort	INTEGER	The server port providing the service.
masProvided\ServiceZones	String	The name of the logical zone this host is installed in.

For instructions on how to view static data, see the documentation supplied with the SNMP Management tool.

The following information can also be viewed by using *rc.mas viewmib* command, see Section 3.7 on page 13

## 6.3 View Dynamic Data in MIB

Dynamic parameter values may be updated by the component during operation. The MAS MIB defines the following dynamic parameters:

*Table 4 Dynamic parameters in MAS MIB*

Parameter	Type	Description
masOperationalState	INTEGER	Indicates if the component is in operation. The component may be disabled if it cannot perform its tasks.  1=enabled 2=disabled
masAdministrativeState	INTEGER	The Administrative state of the component.  1=unlocked2=locked3=shutdown
masCurrentUpTime	TimeStamp	The time the component has been running since start.
masAccumulatedUpTime	TimeStamp	The time the component has been running since installation.
masConfigurationState	INTEGER	The state of the configuration.  load(1) = The configuration is loading.  If this value is set to 1, the component is triggered to reread the configuration.  ok(2) = The configuration is loaded ok.  nok(3) = The configuration was not loaded ok, Loaded backup configuration successfully.  failed(4) = The configuration was not loaded ok and no backup configuration could be loaded.
masLastConfiguration\UpdateTime	Date and time	The date and time when the last successful reload configuration was performed.
masLastConfigurationUpdateTicks	INTEGER	This variable shows the time in number of hundredths of a second since the last time the configuration was loaded.
<i>For each Type, The following parameters are shown.</i>		

Parameter	Type	Description
masConnectionStatisticsType	String	The type of this connection, one of Voice, Video, Service Request or Unknown.
masConnectionStatisticsConnections	Integer	The current number of connection(s).
masConnection\StatisticsPeakConnections	Integer	The peak number of connection(s) since the last restart.
masConnectionStatisticsPeakTime	Date and time	The time when "masConnectionStatisticPeakConnections" occurred. Until the MAS has received a call, this parameter will equal the MAS start time.
masConnection\StatisticsTotalConnections	Integer	The total number of connection(s) since the last restart.
masConnection\StatisticsAccumulatedConnections	Integer	The total number of connection(s) since installation time.

For instructions on how to view dynamic data, see the documentation supplied with the SNMP Management tool.

The following information can also be viewed by using *rc.mas viewmib* command, see Section 3.7 on page 13

## 6.4 Manage Disk Space

No files need to be manually cleared out to free disk space. This is done automatically.



## 7 Corrective Maintenance

Corrective maintenance means fault tracing and correction.

### 7.1 Monitor Traps and Correct Faults

The traps that are defined by the MAS MIB are started and stopped.

#### 7.1.1 **MASStarted**

The “MASStarted” trap is issued when the MAS component is started. It reports:

- The operational state of the MAS (MASOperationalState).
- The administrative state of the MAS (MASAdministrativeState).

#### 7.1.2 **MASStopped**

The “MASStopped” trap is issued when the MAS component is stopped. It reports:

- The operational state of the MAS (MASOperationalState).
- The administrative state of the MAS (MASAdministrativeState).

### 7.2 Restore Data

Restore of the MAS component can be performed by using the backed up tar file (made in section: Section 6.1 on page 33 ), e.g. `<RESTORE_HOME>/MAS_<date>_backup.tar.gz`, where `<RESTORE_HOME>` is where the backup file is located.

The MAS component can either be totally restored with all backed up files, or partially restored with some of the files.

1. If the operating system is reinstalled, make sure that the MAS component is installed first, according to the MAS Installation Guide.
2. Make sure that any application and media content packages are reinstalled.
3. Copy the compressed tar file to the home directory of the MAS component (`<MAS_HOME>`)(e.g `/apps/mas`)

```
cp <RESTORE_HOME>/MAS_<date>_backup.tar <MAS_HOME>
```

4. Move to the <MAS\_HOME> directory.

```
cd <MAS_HOME>
```

5. Uncompress the file.

```
gunzip MAS_<date>_backup.tar.gz
```

6. To see which files that are backed up in the tar file, type the following command:

```
tar tf MAS_<date>_backup.tar
```

7. Extract files from the container file. Either all, for a full restore, or just a few files for partial restore.

For full restore, type the following command:

```
tar xvf MAS_<date>_backup.tar
```

For partial restore, type the following command (with mas.cfg as example):

```
tar xvf MAS_<date>_backup.tar cfg/mas.xml
```

8. Restart the MAS. See: Section 3.3 on page 8
9. Verify that the MAS handle the traffic correctly, e.g. by making a test call.

## 7.3 Inspect and Manage Log Files

### 7.3.1 Logfile location

All the MAS logs can be found in the /apps/logs/mas directory. A brief description of the content of each log file is provided in this section.

#### 7.3.1.1 Installation Log

The installation log file will named according to the following convention:

```
mas@<Host name>.<YYYY_MM_DD_HHMM>.installog
```

Example:

```
mas@mashost1.domain.2006_01_27_1606.installog
```

This log file is not removed when the component is uninstalled.

#### 7.3.1.2 MAS Log

The MAS logfiles will be named mas.log and mas\_agent.log.

In the default configuration, the following log format is used:

<timestamp><source><loglevel>[SID:<sid>]<trace>

**Timestamp** Time of the log entry conformant with ISO 8601:2000 with the format as yyyy-mm-dd hh:mm:ss,msec timezone. Time zone is specified as +-hhmm offset from UTC.

**Source** Reference to the component responsible for the Log Entry.

**Loglevel** Level of this log event signifying the severity. The loglevel is one of:

- debug
- info
- warn
- error
- fatal

For information about how to set the Loglevel for different parts of the MAS, see Section 7.5 on page 43.

**Sid** End-user session Id.

**Trace** This is the textual description of the log entry.

**Tid** The thread making the log request.

```
2006-09-28 09:43:42,535 MET com.mobeon.masp.externalcomponentregister.algorithm.LogicalHostM
WARN [TID:main] [SID:] - Unknown property: replicationid. Could not retrieve
replicationid. Return replicationid <65535>.
```

#### *Example 16 Call related log entry*

The MAS log files are configured to have the maximum size of 10MB. When the logfile reaches this limit, the log files are renamed with a number suffix. The maximum number of logfiles is 10, where the oldest file have number 10 as suffix.

Example:

```
mas.log
mas.log.1
mas.log.2
.....
```

These log files are removed when the component is uninstalled.

### 7.3.1.3 Stream Log

The stream log handles output from the low level RTP streaming part of MAS. By default the maximum log file size is 10MB. When the log file reaches this limit the file will be renamed with a number suffix. The maximum number of saved log files are 2. The current log file is named stream.log and the oldest saved file will be named stream.log.2.

```
stream.log
stream.log.1
stream.log.2
```

These log files are removed when the component is uninstalled.

### 7.3.1.4 Process Log

The Process Log handles output from the MAS process, i.e. output from the Java Virtual Machine.

The process log is configured to have the maximum size of 5 MB by default. When the logfile reaches this limit, the process.log file is renamed with a number suffix. The maximum number of logfiles is 2, where the oldest file have number 1 as suffix

Example:

```
process.log
process.log.1
```

These log files are removed when the component is uninstalled.

## 7.3.2 Trace errors in Logfiles

Normally, there should be no errors or warnings logged in the log files. If there are, it can indicate a problem with the configuration, with other related components (gateways, SSP's), or with the MAS instance itself. Refer to the table below for a description of some of the most common types of log entries.

Table 5 Common errors and warnings in the runtime log file

Severity	Instance	Message	Description
WARNING	com.mobeon.masp .externalcomponent\register.*	Unknown property: replicationid. Could not retrieve replicationid. Return replicationid <65535>.	The MCR entry of the MUR does not contain a replication ID. Therefore there is no way to determine if the MAS is operating in a MUR multi-master deployment or not.
ERROR	com.mobeon.masp.mailbox or com.mobeon.masp.message_sender	Read timed out	A read operation on the SMTP or IMAP protocol towards the MS component has timed out, probably due to a temporary high load on the MS.
WARNING	com.mobeon.masp .callmanager.*	The INVITE is rejected since the Service is temporarily unavailable due to current administrative state: Locked state. A SIP "Temporarily Unavailable" response is sent.	The MAS is in LOCKED state.  Action: Unlock the MAS
ERROR	com.mobeon.masp .callmanager.*	Exception occurred when registering. The backoff timer is scheduled for a REGISTER retry later. (State=Registering, SSP=ssphost: 5060)	The SSP instance is not reachable.  Action: This may indicate that the SSP instance is down, ssphost attribute is not set in mas.xml or the IP address for the ssphost can not be resolved.

Severity	Instance	Message	Description
ERROR	com.mobeon.masp .\mediatrans\lation manager.*	Unknown ASR engine protocol: [<protocol>]	The protocol specified is not recognized by the MAS.  Action: Update the mas.xml configuration file with the correct ASR protocol.
ERROR	com.mobeon.masp .\mediatrans\lation manager.	Unknown TTS engine protocol: [<protocol>]	The protocol specified is not Recognized by the MAS.  Action: Update the mas.xml configuration file with the correct TTS protocol.
FATAL	Any	Amount of free memory is too small	The amount of free physical memory is too small. The MAS is disabled.  Action: Locate process that is using a large amount of memory and take the appropriate actions to free up memory. If the MAS process itself has used too much memory, please forward all log-files to support for further analysis.  This state is unrecoverable.

Common errors and warnings in the runtime log file.

## 7.4 Change Configuration Parameters in MIBs

There are no configuration parameters in the MAS MIB that can be changed.

## 7.5 Change Configuration Parameters in File

The MAS uses the configuration file described in this chapter. Changed configuration parameters will generally be read when restarting the MAS component. All parameters, except those marked with **NR**, will be reread during a reloadconfig operation. When modified, the log configuration files will be automatically reloaded within a few seconds.

The configuration file for the MAS is XML based and is easily edited with an XML editor. Either one that can handle XML schemes or one that can check start and ending tags.

All configuration files can be found under <MAS\_HOME>/cfg.

The following configuration files exist:

- a mas.xml. Main MAS configuration file.
- b logmanager.xml and logmanageragent.xml. Log configuration.
- c stream.log.properties. Log configuration (the log4cxx logging from Stream C++ which is directed to process.log).
- d numberanalyzer.xml. This file is not affecting the MAS. See the application documentation for more information about this file.
- e trafficevents.xml. This file is not affecting the MAS. See the application documentation for more information about this file.
- f chargingaccountmanager.xml. Configuration file for the charging account function.

### 7.5.1 Main Configuration File

The configuration file consists of a set of root tags, which group the configuration attributes for the different components inside the MAS together. It has the following syntax:

```
<configuration>
 <trafficeventsender>
 </trafficeventsender>
 <callmanager>
 </callmanager>
 <contenttypemapper>
 </contenttypemapper>
 <executionengine>
 </executionengine>
 <externalcomponentregister>
 </externalcomponentregister>
 <mediacontentmanager>
 </mediacontentmanager>
```

```

 <mediatranslationmanager>
 </mediatranslationmanager>
 <mailbox>
</mailbox>
 <messagesender>
</messagesender>
 <operateandmaintainmanager>
</operateandmaintainmanager>
 <profilemanager>
</profilemanager>
 <servicerequestmanager>
</servicerequestmanager>
 <stream>
</stream>
 <provisionmanager>
</provisionmanager>
</configuration>

```

#### 7.5.1.1 The mediacontentmanager group

The mediacontentmanager group holds the configuration for the component in the MAS that handles the media content packages.

```

<configuration>
 <mediacontentmanager
 resourcepath="applications/mediacontentpackages">
 <mediaobjectcache
 policy="lfu"
 maxsize="100"
 elementtimeout="600"
 memorysensitive="false"/>
 </mediacontentmanager>
 </configuration>

```

The mediacontentmanager group has one attribute, the default value set is usually adequate.

- The resourcepath tells the MAS where to look for the media content packages. The path is relative to the MAS installation directory.

#### 7.5.1.2 The mediaobjectcache group

When a Media Object is requested, the Media Content Manager can keep the object in a cache to speed up access to frequently used Media Objects. The mediaobject group holds the configuration for this cache. If the group is not configured, no cache will be used by the Media Content Manager.

The default values of the attributes in the mediaobjectcache group are usually adequate.

```

<mediaobjectcache policy="lfu" maxsize="100"

```



```
elementtimeout="600" memorysensitive="false"/>
```

- The policy attribute determines which caching policy that should be used. Possible values are "lfu" (default), "lru" and "fifo". "lfu" is Least Frequently Used, which means that the cache will remove objects starting with those least frequently used. "lru" is Least Recently Used. The cache will remove objects starting with those least recently used. "fifo" is First In First Out. The cache will remove objects starting with the oldest in the cache. **NR**
- The maxsize attribute determines the maximum number of objects that will be kept in the cache. When the cache is full and a new object is added, an old object will be removed according to the policy set. **NR**
- The elementtimeout attribute determines the life time in seconds of an object in the cache. When the timeout expires for an object, it is removed from the cache. **NR**
- The memorysensitive attribute determines if the objects added to the cache should be available for garbage collection. If "true", the objects are available for the garbage collector. Default is "false". **NR**

### 7.5.1.3

#### The trafficeventsender group

The trafficeventsender group holds the configuration for the component in the MAS that handles traffic events.

```
<trafficeventsender>
 <emailconfig/>
 <radiusconfig>
</radiusconfig>
</trafficeventsender>
```

#### emailconfig group

This group contains parameters used for sending the traffic events via E-mail.

```
<emailconfig templatespath="lib"/>
```

- The templatespath is a file system path to where the template files are located. Relative to <MAS\_HOME>.

#### radiusconfig group

This group contains parameters used for sending the traffic events via Radius-MA.

The group can have these attributes:

```
<radiusconfig servicename="eventreporting" queuesize="2000"
 delay="5000">
 <attribute name="sessionid" number="50" type="0"
```

```

 datatype="S"/>
 <attribute name="username" number="1" type="0"
 datatype="S"/>
 <attribute name="callednumber" number="30" type="0"
 datatype="S"/>
 <attribute name="callingnumber" number="31" type="0"
 datatype="S"/>
 <attribute name="operation" number="1" type="1"
 datatype="E"/>
 <attribute name="messagesize" number="2" type="1"
 datatype="I"/>
 <attribute name="terminatecause" number="3" type="1"
 datatype="E"/>
 <attribute name="accesstype" number="4" type="1"
 datatype="E"/>
 <attribute name="objecttype" number="6" type="1"
 datatype="E"/>
 <attribute name="objectid" number="7" type="1"
 datatype="S"/>
 <attribute name="description" number="8" type="1"
 datatype="S"/>
 <attribute name="messagetype" number="12" type="1"
 datatype="E"/>
 <attribute name="messageencoding" number="13" type="1"
 datatype="E"/>
 <attribute name="ownername" number="16" type="1"
 datatype="S"/>
 <attribute name="sssporttype" number="11" type="1"
 datatype="E"/>
 <attribute name="accountcurrency" number="30" type="1"
 datatype="S"/>
 <attribute name="accountmoney" number="31" type="1"
 datatype="S"/>
 <attribute name="accounttype" number="32" type="1"
 datatype="E"/>
 <attribute name="accountreason" number="33" type="1"
 datatype="S"/>
</radiusconfig>

```

- The servicename is the name of the service to use to get service instances from the External Component Register.
- The queuesize is maximum size of the event queue.
- The delay is the time in milliseconds to wait when no service instance is available before trying to get a new service instance and send events again.

The radius group can have one or many attribute groups:

### essconfig group

This group contains parameters used for sending the traffic events via the event service.

```
<essconfig queuesize="1000" essservicename="EventSubscription"/>
```

- The essservicename is the name of the service to use to get service instances from the External Component Register.
- The queuesize is maximum size of the event queue.

### attribute group

This group contains parameters used to configure each attribute that can be sent via Radius-MA. The group can have these attributes:

```
<attribute name="sessionid" number="50" type="0" datatype="S"/>
```

- The lower case name of the attribute in the traffic event.
- The number that identifies an attribute in the Radius-MA protocol.
- The type is a boolean value that indicates if this attribute should be a Vendor-Specific attribute. If true ("1") it will be set as a Vendor-Specific attribute.
- The datatype is an identifier of which datatype this attribute should be set. Can be I (Integer), S (String) or E (Enumeration)

#### 7.5.1.4

### The callmanager group

The callmanager group holds the configuration for the component in the MAS that handles SIP calls.

```
<callmanager
callnotacceptedtimer="32000"
registerbackofftimer="120000"
registerbeforeexpirationtime="5000"
blacklisttimer="60000"
inboundaudiomedia="pcmu"
inboundvideomedia="h263"
registeredname=""
defaultcalltype="video"
outboundcallcallingparty=""
outboundcallconnecttimeout="30000"
disconnectonsiptimeout="true"
sendprovisionalresponsesreliable="sdponly"
supporttestinput="false"
useragentwithphoneinsipuributnouserparameter=
"cisco,radvision,eyebeam,express talk,mirial"
outboundcallserverport="5060"
```

```

contacturioverride="">

<requiredoutboundaudiomedia> </requiredoutboundaudiomedia>

<requiredoutboundvideomedia> </requiredoutboundvideomedia>

<releasecausemappings> </releasecausemappings>

<remoteparty> </remoteparty>

<siptimers> </siptimers>

<loadregulation> </loadregulation>

<restrictedoutboundheaders> </restrictedoutboundheaders>
</callmanager>

```

The callmanager group consists of a set of attributes. By default media configurations is set to G.711 u-Law for audio and H.263 for video. If other media settings are wanted configuration has to be updated, otherwise the default settings are usually adequate. Attributes should normally not be edited unless there is a specific reason to do so.

- The callnotacceptedtimer is the amount of time in milliseconds within which an inbound call must be accepted by the MAS. If the call has not been accepted within that time, the call will be rejected.
- The registerbackofftimer is the amount of time in milliseconds the MAS will wait before it tries to reregister with an SSP (if an SSP is used in the deployment) in case of a registration failure.
- The registerbeforeexpirationtime indicates how long before the registration expiration time the MAS should reregister to the SSP (if an SSP is used in the deployment). The value is in milliseconds.
- The blacklisttimer is the amount of time in milliseconds a non-SSP (if an SSP is used in the deployment) host is black listed if a timeout occurred when setting up a new outbound call towards that host. No new calls are setup to that host during the that time.
- The inboundaudiomedia is the audio codec to use when recording SIP calls. Valid values are: "pcmu", "pcma" and "amr". Default is "pcmu". This attribute shall be modified if a non-default codec should be used.
- The inboundvideomedia is the video codec to use when recording SIP calls. Currently the only valid value is: "h263".
- The registeredname tells the MAS which name to use when registering to the SSP (if an SSP is used in the deployment) and in the SIP user agent header. This attribute is prompted for during installation of MAS. Should normally not be modified after installation.

- The defaultcalltype indicates which call type to setup for outbound calls that are created without a call type and for inbound calls that lacks a media offer in the initial SIP INVITE. Valid values are: "voice" or "video". Default value is: "video".
- outboundcallcallingparty: If the application has not provided a calling party number when an outbound call is created, this value will be used as calling party for the outbound call.
- outboundcallconnecttimeout: If the application has not provided a connect timeout value when an outbound call is created, this value will be used as connect timeout for the outbound call.
- The disconnectonsiptimeout determines if the call should be disconnected if a timeout occurs for a SIP INFO request. The normal case is to disconnect the call, but since some SIP clients are known to not respond to SIP INFO requests, it is possible to override this behavior. If disconnectonsiptimeout is set to false, the MAS will not disconnect the call if a SIP INFO request times out. This applies only to SIP INFO, on timeout for other requests the call is still disconnected.
- The sendprovisionalresponsesreliable determines whether a provisional SIP response should be sent reliably or not. The available values are "yes", "no" and "sdponly". The normal situation is to send provisional responses reliably only when containing SDP information. Default value: "sdponly".
- The supporttestinput indicates whether or not to support test input given over SIP in a non-standardized way. The available values are "true" or "false". Default value is "false". This attribute should only be modified for test purposes and should always be "false" in a normal deployment.
- The useragentwithphoneinsipuri but nouserparameter indicates for which user-agents MAS will treat the user part of the SIPURI as a phone number despite that user=phone is missing. Specify a comma-separated list of user-agents. The user-agents are case insensitive. There are 2 special values that can be used: "all" means that all URI will be treated as phone number, "empty" means that URIs in a request without an user-agent will be treated as phone number.
- The outboundcallserverport defines what port number to use when contacting an outbound call server. An outbound call server is contacted on behalf of the application for example when doing outbound calls. If the application has not supplied an outbound call server, the SSP/SIP Proxy will be contacted. The outboundcallserverport is used only if the application has not supplied a port.
- The contacturioverride redefines what is sent in the SIP Contact header. By default (when contacturioverride is empty) the SIP Contact header is composed using the configured hostname of the MAS (refer to Section 7.5.1.6 on page 56). That hostname is normally not a fully qualified domain name (FQDN), but this normally works well in a gateway deployment since the hostname is routable within the private network. However in e.g. a VoIP

or IMS deployment where the SIP endpoints are in another network than the MAS, the configured MAS hostname may not be possible to use for routing the SIP messages. In that case it is necessary to define another address in the `contacturioverride`. That address need to be the fully qualified domain name of the MAS, routable from the external IMS or VoIP network.

When specifying that a contact header override is required, the format is as follows:

`contacturioverride="sip:mas@<MAS FQDN>"` Where `<MAS FQDN>` refers to the Fully Qualified Domain Name (FQDN) of the MAS.

To disable Contact URI override (the default), specify an empty string:

`contacturioverride=""`

Example:

`contacturioverride="sip:mas@mas01.mynetwork.com"`

### The `requiredoutboundaudiomedia` group

This group tells the MAS which audio media that shall be possible to stream out. By default `pcmu` and `telephone-event` is configured. “`telephone-event`” should always be present while “`pcmu`” must be modified if a non-default codec is to be used.

```
<requiredoutboundaudiomedia>
 <mediatype encoding="pcmu"></mediatype>
 <mediatype encoding="telephone-event"></mediatype>
</requiredoutboundaudiomedia>
```

Currently the only supported encoding values are `pcmu` (G.711 u-Law codec), `pcma` (G.711 a-law codec) and `amr` (AMR-NB codec), together with `telephone-event` (DTMF according to RFC2833).

### The `requiredoutboundvideomedia` group

This group tells the MAS which video media that shall be possible to stream out. The default value is normally adequate.

```
<requiredoutboundvideomedia>
 <mediatype encoding="h263"></mediatype>
</requiredoutboundvideomedia>
```

The only supported values in this version is `h263`, which is the H.263 codec.

### The `releasecausemappings` group

This group tells the MAS how to map a release cause to a network status code. The release cause information can be received either as a Q.850 cause/location or as a SIP response code. If a Q.850 cause/location is available it is chosen

as first alternative when doing the release cause mapping to a network status code. The default mapping is usually adequate for a standard system.

```
<releasecausemappings defaultnetworkstatuscode="621">
 <mapping name="suppressed" networkstatuscode="614"
 sipresponsecodeintervals="600"
 q850causeintervals="17" q850locationintervals="0"/>
 <mapping name="busy" networkstatuscode="603"
 sipresponsecodeintervals="486"
 q850causeintervals="17"/>
 <mapping name="noreply" networkstatuscode="610"
 sipresponsecodeintervals="408,480"
 q850causeintervals="18-19"/>
 <mapping name="notreachable" networkstatuscode="613"
 sipresponsecodeintervals="301,403-404,410,484,501-502,603"
 q850causeintervals="1-9,20-23,25-31"/>
 <mapping name="congestion" networkstatuscode="620"
 sipresponsecodeintervals="503"
 q850causeintervals="39-44,46"/>
</releasecausemappings>
```

The SIP responses can be any value between 300 and 699 supported by the SIP standard. The Q.850 cause can be any value between 0 and 127. The Q.850 location can be any value between 0 and 15. The network status codes are as defined by the Messaging-over-IP system.

The configuration order is important. The first line has highest priority in case the configuration overlaps.

### The remoteparty group

This group defines how the MAS will behave for outdials and registering towards the SSP:s or an external SIP proxy.

```
<remoteparty>
 <ssplist>
 <address host="ssp1.domain" port="5060"/>
 <address host="ssp2.domain" port="5060"/>
 </ssplist>
</remoteparty>
```

When using SSP:s the remoteparty group contains a list of SSP:s as shown above.

The hostname and port is the name of the SSP host and the port where it responds to SIP requests.

MAS registers to all of the defined SSP:s in the list and chooses one of them randomly when making an outdial.

```
<remoteparty>
```

```
<sipproxy>
 <address host="ssp1.domain" port="5060"/>
</sipproxy>
</remoteparty>
```

When using a SIP proxy instead of SSP:s the remoteparty group contains one SIP proxy as shown above.

The hostname and port is the name of the SIP proxy host and the port where it responds to SIP requests.

No registration is done to the remote SIP proxy. In this configuration the MAS always uses the SIP proxy when making an outdial.

### The siptimers group

This group contains the SIP timer values to use. It is read at startup of the MAS and is not reloadable. **NR**

```
<siptimers t2="8" t4="10" timerb="6" timerc="360"
timerd="64" timerf="18" timerh="64" timerj="64">
</siptimers>
```

The SIP timer values are given as multiples of the T1 timer which is 500 milliseconds. If timer T4 should be 4 seconds, it is configured to the value 8, i.e. 8 times 500 milliseconds.

The default values for the timers are:

- T2: 8, i.e. 4 seconds
- T4: 10, i.e. 5 seconds
- Timer B: 6, i.e. 3 seconds
- Timer C: 360, i.e. 3 minutes
- Timer D: 64, i.e. 32 seconds
- Timer F: 18, i.e. 9 seconds
- Timer H: 64, i.e. 32 seconds
- Timer J: 64, i.e. 32 seconds

The definition of these timers can be found in RFC 3261.

### The loadregulation group

This group holds the configuration for load regulation during startup of the MAS.

```
<loadregulation
```



```

 initialramphwm="1"
 channelstoincrease="1"
 numberofincrements="1">
</loadregulation>

```

From a the MAS perspective, a state of high load happens when the number of active SIP calls exceeds a threshold, i.e. high watermark. The state ceases to be high load when the number of calls goes below another threshold, i.e. the low watermark. As soon as the MAS reaches the state of high load it will unregister from the SSP (if an SSP is used in the deployment). When the high load state has ceased the MAS will register to the SSP again.

During startup, the MAS component cannot take the maximum amount of load immediately and needs a slow ramp up. This is since the MAS is mainly implemented in Java, so the the Just-In-Time compiler of the Java Virtual machine must be invoked for several calls until the the compiler has optimized the Java code appropriately.

When the MAS is started, the operate and maintain manager sets the threshold from the configuration values "serviceinitthreshold", "serviceenablerhighwatermark" and "serviceenablerlowwatermark" in the operate and maintain group. In addition to these values, the MAS uses a rampThreshold, rampHighWatermark and rampLowWatermark internally which is the limit currently used by the MAS during the ramp up. The ramp-limits will move towards the configured limits by incrementing by a ramp rate determined by "channelstoincrease" and "numbersofincrements" when the high load state ceases.

- initialramphwm is the initial value for rampHighWatermark when the MAS is started. If set to 0, the slow ramp up is disabled and the MAS uses the "serviceinitthreshold" from operate and maintain manager directly. **NR**
- channelstoincrease is used to determine the rate for the ramp up. The rate is calculated as channelstoincrease / numberofincrements. **NR**
- numberofincrements is used to determine the rate for the ramp up. The rate is calculated as channelstoincrease / numberofincrements. **NR**

### The ramp up algorithm

In this section the ramp up algorithm is described. This gives the details required to fine tune the load regulation.

Startup:

- a threshold, HWM and LWM is given by the operate and maintain manager configuration.
- b initial rampHWM is given by initialramphwm in the loadregulation group.
- c The rampLWM is given by rampHWM - ( HWM - LWM ), with zero as lowest value.

d The rampThreshold is given by  $\text{rampHWM} + (\text{threshold} - \text{HWM})$ , with zero as lowest value.

e The ramp rate is given by  $\text{channelstoincrease} / \text{numberofincrements}$ .

When the number of channels reaches rampHWM:

a The MAS unregisters from the SSP (if an SSP is used in the deployment).

If the number of channels reaches rampThreshold:

a The MAS starts redirecting incoming calls to the SSPs (if an SSP is used in the deployment).

When the number of calls reaches rampLWM (after recently have reached rampHWM):

a rampHWM is increased by the ramp rate. If rampHWM reaches HWM, HWM is used.

b rampLWM and rampThreshold is re-calculated as in Startup. If rampThreshold reaches threshold, threshold is used.

c The MAS registers to the SSP (if an SSP is used in the deployment).

The following is an example illustrating how the values involved in the rampup algorithm are increased until threshold, LWM and HWM are reached.

In the example, the following configuration is used:

- initialramphwm is 1.
- channelstoincrease is 5.
- numberofincrements is 1.
- threshold is 65.
- LWM is 48.
- HWM is 60.

Initially, the MAS will use the following values: rampThreshold=6, rampHWM=1, rampLWM=0. When the first call arrives, the MAS will consider itself in high load, and this call must disconnect before rampLWM is reached. At this point, the values are updated as follows: rampThreshold=11, rampHWM=1, rampLWM=0.

Now, assuming that the number of calls reaches (at least) rampHWM and then decreases to rampLWM in every step, the values are updated as follows:

- rampThreshold=16, rampHWM=11, rampLWM=0
- rampThreshold=21, rampHWM=16, rampLWM=4

- rampThreshold=26, rampHWM=21, rampLWM=9
- rampThreshold=31, rampHWM=26, rampLWM=14
- rampThreshold=36, rampHWM=31, rampLWM=19
- rampThreshold=41, rampHWM=36, rampLWM=24
- rampThreshold=46, rampHWM=41, rampLWM=29
- rampThreshold=51, rampHWM=46, rampLWM=34
- rampThreshold=56, rampHWM=51, rampLWM=39
- rampThreshold=61, rampHWM=56, rampLWM=44
- rampThreshold=65, rampHWM=60, rampLWM=48

As can be observed, the difference between rampHWM and rampLWM in every step is 12 (HWM - LWM), and the values are generally increased by the ramp rate (channelstoincrease / numberofincrements = 5). In the last step however, the values have reached the target values of threshold, LWM and HWM.

### **The restrictedoutboundheaders group**

This group tells the MAS which SIP headers that should be excluded in an outbound call. The default value is to exclude no headers. This is normally adequate but may be modified for higher security. Multiple SIP headers may be excluded. The only supported values in this version are "remote-party-id" and "p-asserted-identity".

```
<restrictedoutboundheaders
 <header name="remote-party-id"/>
</restrictedoutboundheaders>
```

In the above example, the remote-party-id header field should be excluded in an outbound call. The default value is to exclude no headers and thus normally this group is left empty.

#### **7.5.1.5**

### **The content type mapper group**

The contenttypemapper group holds mappings between different content types, codecs and file extensions. Normally, the default values in this group are adequate and should not be modified.

```
<configuration>
 <contenttypemapper>
 <contenttype mimetype="video/quicktime">
 <codec mimetype="video/h263"/>
 <codec mimetype="audio/pcm"/>
 <fileext name="mov"/>
 </contenttype>
 </contenttypemapper>
</configuration>
```

```

 <contenttype mimetype="audio/wav">
 <codec mimetype="audio/pcmu"/>
 <fileext name="wav"/>
 </contenttype>
 <contenttype mimetype="video/3gpp">
 <codec mimetype="video/h263"/>
 <codec mimetype="audio/amr"/>
 <fileext name="3gp"/>
 </contenttype>
 <contenttype mimetype="audio/3gpp">
 <codec mimetype="audio/amr"/>
 <fileext name="3gp"/>
 </contenttype>
 <contenttype mimetype="text/plain">
 <codec mimetype="text/plain"/>
 <fileext name="txt"/>
 </contenttype>
 </contenttypemapper>
</configuration>

```

### The contenttype group

Each content type is configured in a contenttype group identified by its mimetype attribute. Each content type has:

- One or several codec groups
- One fileext group

### The codec group

The codec group holds a mimetype for a codec in a content type e.g. "audio/pcmu" or "audio/pcma"

### The fileext group

The fileext group holds the file extension used in a content type.

#### 7.5.1.6

### The executionengine group

The executionengine group holds the configuration for the handling and execution of applications.

It has the following syntax:

```

<executionengine enginestacksize="100"
 callmanagerwaittime="60000" accepttimeout="30000"
 createcalladditionaltimeout="30000" traceenabled="false"
 alwayscompile="false" generateops="false" opspath="."

```

```
hostname="10.11.0.38" watchdogtimeout="600000"/>
```

The executionengine group consists of a set of attributes. The default settings are normally adequate.

- The enginestacksize parameter is the size of an internal data structure. If the application contains very nested control flows, it may be necessary to increase the value. A sign of this is that you get log entries mentioning engine stack exhaustions.
- The callmanagerwaittime parameter is how long the Execution Engine shall wait for events from the Call Manager component before considering it an error. Unit: milliseconds.
- The accepttimeout parameter is how long the Execution Engine shall wait for the application to take appropriate action (like answering the call) after Execution Engine has notified the application of a new incoming call. Unit: milliseconds
- The createcalladditionaltimeout parameter is how long the Execution Engine shall wait for an event from the Call Manager following a "createcall" invocation. The createcalladditionaltimeout is used only if the application has defined a timeout for maximum time to connect the outgoing call. Execution Engine will actually wait for a maximum of (createcalladditionaltimeout + the maximum time defined by the application). Unit: milliseconds.
- The hostname parameter defines the hostname of the MAS. The value is automatically set during installation, and to defined the host name for SIP.

These parameters also exists, but are only for use by developers.

- The traceenabled parameter defines if low level tracing of executing engine shall be enabled.
- The alwayscompile parameter defines if Voice XML/CCXML documents shall be compiled every time a new MAS session (e.g. incoming call) is started. This is useful for debugging applications, but not recommended in a production environment.
- The generateops parameter defines if the execution engine shall generate ops-files when compiling Voice XML/CCXML documents.
- The opspath parameter defines where execution engine stores ops-files.
- The watchdogtimeout parameter defines how long to allow nothing to happen in a call before a warning log is issued. Unit: milliseconds.

### 7.5.1.7

#### The externalcomponentregister group

This XML tag groups together configuration for how the MAS handles services stored in the MCR or absence of services in the MCR.

It has the following syntax.

```
<externalcomponentregister>
 <mcr/>
 <propertymap>
 </propertymap>
 <services>
 </services>
 <algorithms>
 <localrandomchooser/>
 </algorithms>
</externalcomponentregister>
```

### The mcr group

The mcr group defines where the MAS locates services. For every individual service, the MAS will either use the configuration file or find the services in MCR. Changes after a reloadconfig operation will only take effect next time MCR is contacted, e.g. after the time defined by the periodicity attribute has elapsed.

```
<mcr hostname="mcrhost" port="389" timeout="5000"
 periodicity="300000" lookupdelay="500" userid="icomponent"
 password="abc123"
 suffix="emregistername=messagingcomponentregister,o=config"
 trylimit="3" trytimelimit="500"/>
```

- The hostname is the name of the MCR host.
- The port is the port number of the MCR host where the LDAP directory in MCR is located.
- The timeout is the timeout in milliseconds to wait for a response from the MCR before a retry is done.
- The periodicity is the time in milliseconds between the periodic MCR lookups when the cache is updated.
- The lookupdelay is the time in milliseconds between when all service instances has been error reported until the MCR is searched for service instances again.
- The userid is the userid to be used when accessing the MCR.

The authentication is made for the following distinguished name:

```
"uid=<userid>,<suffix>"
```

- The password is the password used for accessing the MCR.
- The suffix is the root suffix of the LDAP directory where the MCR:s services are located.

Typically this is `emregistername=messagingcomponentregister,o=config` and should never change.

- The `trylimit` is the maximum number of times to try contacting MCR in case of problems.
- The `trytimelimit` is the maximum time period in milliseconds to try contacting MCR in case of problems.

### The `propertymap` group

The `propertymap` maps the LDAP directory names to MAS names.

This map should never have to be edited and will have adequate default values after installation.

```
<propertymap>
 <port mcrname="ipserviceport"/>
 <protocol mcrname="ipserviceprotocol"/>
 <version mcrname="emcomponentversion"/>
 <hostname mcrname="emhostname"/>
 <logicalzone mcrname="emlogicalzone"/>
 <component mcrname="emcomponentname"/>
 <type mcrname="emcomponenttype"/>
 <rootoid mcrname="emrootoid"/>
 <language mcrname=""/>
 <replicationid mcrname="emcomponentmur-replicationid"/>
</propertymap>
```

### The `services` group

The `services` group contains a list of all services configured to retrieve from the MCR. It has the following syntax:

```
<services>
 <storage overridemcr="false">
 </storage>
 <smtpstorage overridemcr="false">
 </smtpstorage>
 <userregister overridemcr="false">
 </userregister>
 <userregisterwrite overridemcr="false"
 servicename="userregister">
 </userregisterwrite>
 <eventreporting overridemcr="false">
 </eventreporting>
 <speechrecognition overridemcr="true">
 </speechrecognition>
 <texttospeech overridemcr="true">
 </texttospeech>
 <provisioning overridemcr="false">
```

```

 </provisioning>
 <Accounting overridesmcr="false">
 </Accounting>
</services>

```

All services in this list has an attribute called overridesmcr, if this is set to "true", the MAS never asks the MCR for where the service is located and it uses the information given in the group instead.

*Added services to this section will not take effect after a reloadconfig operation, instead the MAS has to be restarted when adding services. Configuration changes in existing services will be updated at reloadconfig. **NR***

### The storage section

This service contains information for the Storage service usually provided by the MS component. It has the following syntax:

```

<storage overridesmcr="false">
 <default port="143"/>
 <!--<instance hostname=""/>-->
</storage>

```

- The port number is the IMAP server port in the MS component..
- The hostname is the name of the configured fallback MS host.

### The smtpstorage service

This service contains information for the SMTPStorage service usually provided by the MS component. It has the following syntax:

```

<smtpstorage overridesmcr="false">
 <default port="25"/>
 <!--<instance hostname=""/>-->
</smtpstorage>

```

- The port number is the SMTP server port in the MS component.
- The hostname is the name of the configured fallback MS host.

### The userregister service

This service contains information for the UserRegister service used for reading, usually provided by the MUR component. It has the following syntax:

```

<userregister overridesmcr="false">
 <default port="389"/>
 <instance hostname="murhost"/>
</userregister>

```



- The port number is the LDAP server port in the MUR component.
- The hostname is the name of the configured fallback MUR host.

### The userregisterwrite service

This service contains information for the UserRegister service used for writing, usually provided by the MUR component. It has the following syntax:

```
<userregisterwrite overridemcr="false" servicename="userregister">
 <default port="389"/>
 <instance hostname="murhost"/>
</userregisterwrite>
```

- The servicename is the service name to use when searching MCR for service instances providing the userregisterwrite service.
- The port number is the LDAP server port in the MUR component.
- The hostname is the name of the configured fallback MUR host.

### The eventreporting service

This service contains information for the EventReporting service usually provided by the MER component. It has the following syntax:

```
<eventreporting overridemcr="false">
 <default port="1813"/>
 <!--<instance hostname=""/>-->
</eventreporting>
```

- The port number is the RADIUS-MA server port in the MER component.
- The hostname is the name of the configured fallback MER host.

### The speechrecognition service

This service contains information for the SpeechRecognition service. It has the following syntax:

```
<speechrecognition overridemcr="true">
 <instance hostname="10.16.2.98"
 port="4900"
 protocol="mrtp"/>
</speechrecognition>
```

- The hostname is the name/address of the speech recognition host.
- The port is the speech recognition server port number.

- Protocol is the protocol (currently mrp only) used by the MAS when communicating with the speech recognition server.

### The texttospeech service

This service contains information for the TextToSpeech service. It has the following syntax:

```
<texttospeech overrides="true">
 <instance hostname="10.16.2.98"
 port="4900"
 protocol="mrp"/>
</texttospeech>
```

- The hostname is the name/address of the text to speech host.
- The port is the speech recognition server port number.
- The protocol is the protocol (currently mrp or xmp) used by MAS when communicating with the text to speech server.

### The provisioning service

This service contains information for the provisioning service. It has the following syntax:

```
<provisioning overrides="false">
 <instance hostname="" port="2400"/>
</provisioning>
```

- The hostname is the name/address of the provisioning server.
- The port is the provisioning server port number.

### The accounting service

This service contains information for the accounting service. It has the following syntax:

```
<Accounting overrides="false">
 <instance hostname="" port="8080"/>
</Accounting>
```

- The hostname is the name/address of the accounting (XMP) server
- The port is the accounting (XMP) server port number.

### The algorithms group

This group contains configuration for the different service instance choosing algorithms. Currently only the localrandomchooser algorithm is defined.

```
<algorithms>
 <localrandomchooser logicalzone=""
 netmask="255.255.255.0"/>
</algorithms>
```

- The logicalzone is the logical zone the MAS belongs to.
- The netmask is the netmask of the host of the MAS. This is used to find "close" hosts, when no logical zone is defined.

### The media content manager group

The mediacontentmanager group holds the configuration for the component in the MAS that handles the media content packages.

```
<configuration>
 <mediacontentmanager
 resourcepath="applications/mediacontentpackages">
 <mediaobjectcache
 policy="lfu"
 maxsize="100"
 elementtimeout="600"
 memorysensitive="false"/>
 </mediacontentmanager>
 </configuration>
```

The mediacontentmanager group has one attribute, the default value set is usually adequate.

- The resourcepath tells the MAS where to look for the media content packages. The path is relative to the MAS installation directory.

#### 7.5.1.8

### The mediaobjectcache group

When a Media Object is requested, the Media Content Manager can keep the object in a cache to speed up access to frequently used Media Objects. The mediaobject group holds the configuration for this cache. If the group is not configured, no cache will be used by the Media Content Manager. The default values of the attributes in the mediaobjectcache group are usually adequate.

```
<mediaobjectcache policy="lfu" maxsize="100"
 elementtimeout="600" memorysensitive="false"/>
```

- The policy attribute determines which caching policy that should be used. Possible values are "lfu" (default), "lru" and "fifo". "lfu" is Least Frequently Used, which means that the cache will remove objects starting with those least frequently used. "lru" is Least Recently Used. The cache will remove objects starting with those least recently used. "fifo" is First In First Out. The cache will remove objects starting with the oldest in the cache. **NR**

- The maxsize attribute determines the maximum number of objects that will be kept in the cache. When the cache is full and a new object is added, an old object will be removed according to the policy set. **NR**
- The elementtimeout attribute determines the life time in seconds of an object in the cache. When the timeout expires for an object, it is removed from the cache. **NR**
- The memorysensitive attribute determines if the objects added to the cache should be available for garbage collection. If "true", the objects are available for the garbage collector. Default is "false". **NR**

#### 7.5.1.9 The mediatranslationmanager group

The mediatranslationmanager group holds the configuration for the component in the MAS that handles TTS and ASR.

```
<configuration>
 <mediatranslationmanager
 texttospeechlanguages="en-GB,se-SV" />
</configuration>
```

The mediatranslationmanager group has one attribute which contains a list of the languages available for text to speech translations. The attribute value string is a comma separated list. The languages list must be consistent with the languages which actually are installed in the TTS engine.

#### 7.5.1.10 The mailbox group

The mailbox group holds the configuration for the component in the MAS that handles the subscriber mailboxes in the MS.

```
<mailbox>
 <message>
 <additionalproperty/>
 </message>
 <imap/>
</mailbox>
```

#### The message group

The message group holds parameters for configuring mailbox message handling.

```
<message>
 <additionalproperty name="<propertyname>"
 field="<fieldname>"/>
</message>
```

### The additionalproperty group

The additionalproperty group is used for extending the range of message properties without modifying the Mailbox implementation. A additional property name is mapped to a field in the underlaying storage implementation. Values are read once at first usage after system boot.

- The name parameter defines the value for mailbox client to use when accessing a additional property.
- The field parameter defines the name of the field in the underlaying storage implementation (i.e. If MIME messages are used in the storage implementation field is the name of an MIME message header). If field is left blank the value of the name parmeter will be used.

### The imap group

The imap group holds parameters for tuning the IMAP communication with the MS. Values are re-read when opening a mailbox.

```
<imap
 connectiontimeout="5000"
 commandtimeout="5000"
/>
```

- The connectiontimeout is the socket connection timeout value in milliseconds. Set to 5000 at installation.
- The commandtimeout is socket I/O timeout value in milliseconds. Set to 5000 at installation.

#### 7.5.1.11

### The message sender group

The messagesender group holds parameters for tuning the SMTP communication with the MS. Values are re-read when configuration is changed.

```
<messagesender
 smtpservicename="smtpstorage"
 smtpretries="3"
 smtpconnectiontimeout="3000"
 smtpcommandtimeout="3000"
/>
```

- The smtpservicename parameter defines which value Message Sender is using when locating an SMTP service from External Component Register.
- The smtpretries is how many times Message Sender tries to send a message to the MS. If Message Sender not is successful after the configured number of tries the failure is reported back to the client.
- The connectiontimeout is the SMTP socket connection timeout value in milliseconds. Set to 3000 at installation.

- The `commandtimeout` is SMTP socket I/O timeout value in milliseconds. Set to 3000 at installation.

#### 7.5.1.12

#### The operate and maintain manager group

The `operateandmaintainmanager` group holds the configuration for the component in the MAS that monitors the MAS activities and functions.

It has the following syntax.

```
<operateandmaintainmanager>
<omm/>
</operateandmaintainmanager>
```

#### The omm group

```
<omm
 hostname="localhost"
 rpcmaxnoofretries="10"
 rpcretrydelaysec="5"
 countersavetimeout="5"
 countersavefilenamesuffix="CounterData.dat"
 countersavefilepath="data"
 masnoresponsetimeoutlimituntildown="20"
 logicalzone="logicalzone"
 port="8081"
/>
```

- The `hostname` that the MAS will listen on when receiving commands from SNMP-Agent/User and Monitor.
- The `rpcmaxnoofretries` tells the Monitor how often it try to connect to the MAS before reporting error.
- The `rpcretrydelaysec` tells the Monitor the delay time between retries.
- The `countersavetimeout` tells the MAS how often it will save counter values to disk.
- The `countersavefilenamesuffix` tells the MAS what suffix name the file will have when storing counter data on disk.
- The `countersavefilepath` specifies the store path relative to install directory.
- The `masnoresponsetimeoutlimituntildown` specifies the max response time for the MAS before setting `adminstate` to DOWN. (This is currently not used.)
- The `logicalzone` specifies the logical zone that this MAS instance is installed on. **NR**

- The port specifies the port the MAS will listen on when receiving commands from SNMP-Agent/User and Monitor. **NR**

The service enabler group

```
<serviceenablers>
 <serviceenabler protocol="sip" initthreshold="60"
 lowwatermark="10" highwatermark="60"/>
 <serviceenabler protocol="xmp" initthreshold="20"
 lowwatermark="0" highwatermark="0"/>
</serviceenablers>
```

This group contains parameters for different service enabler protocols

- The protocol specifies the protocol this service enabler is using. **NR**
- The initthreshold specifies the initial threshold for this service enabler. **NR**

When the MAS reaches this value, the MAS starts to redirect calls back to the SSP (if an SSP is used in the deployment).

This value must always be larger than the highwatermark.

**Note:** Only initthreshold is used when protocol="xmp", hence the attributes "lowwatermark" and "highwatermark" are ignored.

- The lowwatermark specifies the default low watermark for this service enabler. **NR**

When the MAS reaches this value, the MAS registers in the SSP (if an SSP is used in the deployment) If previously unregistered due to have reached the highwatermark.

- The highwatermark default highwatermark for this service enabler. **NR**

The highwatermark specifies the maximum number of channels this host is accepting.

When number of simultaneously call reaches this value, the MAS unregisters in the SSP (if an SSP is used in the deployment).

**Note:** changing this value is not supported. If another value is desired, reinstallation of the MAS is necessary. The reason is that many other settings are dependent of the highwatermark, and they need to be recalculated by the installation script.

- **If this value is changed, check that the initthreshold parameter is larger than the highwatermark.**

**Note:** Only initthreshold is used when protocol="xmp", hence the attributes "lowwatermark" and "highwatermark" are ignored.

### 7.5.1.13 The profilemanager group

The profilemanager group contains configuration for the component in the MAS that handles communication with the user register. It has the following syntax:

```
<profilemanager limitscope="false" coscachetimeout="300000">
 <userregister/>
 <connectionpool/>
 <provisioning/>
 <attributemap>
 </attributemap>
</profilemanager>
```

- The limitscope controls if subscriber searches with limited scope is supported or not. If set to false, no search for a subscriber will use a limit scope for searching. Must be set to true only if IMUX is used.
- The coscachetimeout is the time in milliseconds when a cached CoS (Class of Service) will be considered old and reread from the user register.

Only CoS:es cached after a reloadconfig operation will have the new coscachetimeout, i.e. old CoS:es will be reread after the old coscachetimeout time period has passed. After rereading the CoS will be cached the new coscachetimeout period.

### The userregister group

The userregister group defines settings on how to access the user register.

```
<userregister readtimeout="5000" writetimeout="5000"
 admin="cn=directory manager" password="emmanager"
 defaultsearchbase="o=domain"
 trylimit="3" trytimelimit="500"/>
```

- The readtimeout is the timeout in milliseconds used for read operations, e.g. retrieving subscriber data or greetings.
- The writetimeout is the timeout in milliseconds used for write operations, e.g. modifying subscriber data.
- The admin is the userid to be used when accessing the user register.
- The password is the password used for accessing the user register.
- The defaultsearchbase is the searchbase to use when locating a subscriber when no other searchbase has been provided.
- The trylimit is the maximum number of times to try contacting MUR in case of problems.
- The trytimelimit is the maximum time period to try contacting MUR in case of problems.



### The connectionpool group

The connectionpool group defines settings for the LDAP connection pool.

```
<connectionpool maxsize="25" connectionlifetime="300000"/>
```

- The maxsize is the maximum pool size. There exists a pool for each userregister host.
- The connectionlifetime is the time after which a connection is closed, allowing for renewal of connections.

### The provisioning group

The provisioning group defines settings for provisioning administrators.

```
<provisioning password=""/>
```

- The password is the password used for authentication of provisioning administrators. The same password is used for all administrators.

### The attributemap group

The attributemap group defines which attributes that are available for a subscriber and how these attributes are used to create a subscriber object.

```
<attributemap searchorder="community,cos,user,billing">
```

- The searchorder is the default priority order used when retrieving subscriber data.

The other groups under the attributemap group each defines a subscriber attribute: name used in the user register, data type, which level to write data, and optional default values. If the type is boolean the strings used for true and false are also defined. It is also possible to define a searchorder which will override the default searchorder.

#### 7.5.1.14

### The service request manager group

The service request manager group holds the configuration for the component in the MAS that handles XMP communication. The service request manager acts as a server receiving XMP requests from other system components and a client sending XMP requests to other system components.

```
<servicerequestmanager requesttimeout="30000"
 requestretries="3" clientid="mas1@host.domain"/>
```

The service request manager has the following set of attributes:

- requesttimeout, the maximum time the XMP client should wait for an answer for an XMP request when using synchronous mode.

- requestretries, the number of retries that the XMP client should make if an XMP request fails.
- clientid, the unique id that the XMP client should use in the XMP requests. This is set during the installation of the MAS. **NR**

### The diagnose service service group

This clientid is used to identify the diagnose service client when diagnosing service status.

This value must not be used by any other service client.

```
<diagnoseservice clientid="diagnoseservice@localhost"/>
```

- clientid, the clientid the diagnoseservice will use to diagnose the service. **NR**

## 7.5.1.15

### The stream group

The stream group holds the configuration for the component in the MAS that handles RTP streaming.

```
<stream
 portpoolbase="23000"
 portpoolsize="900"
 movfileversion="1"
 synccallmaxwaittimesec="120"
 sendpacketsaheadms="40"
 expiretimeoutms="40000"
 streamabandonedms="32000"
 senderscontrolfraction="0.4"
 audioreplacewithsilencems="10"
 dispatchdtmfonkeydown="true"
 maximumtransmissionunit="2000"
 audioskipms="0"
 maxwaitforiframems="2000"
 skew="0"
 skewmethod="LOCAL"
 defaultinboundptime="40"
 defaultinboundmaxptime="40"
 outputprocessors="1"
 inputprocessors="4"
 silencedetectionmode="0"
 silencethreshold="0"
 initialsilenceframes="40"
 signaldeadband="10"
 silencedeadband="150"
 detectionframes="10"
 silencedetectiondebuglevel="0">
```

```

<supportedcontenttype name="audio/wav"/>
<supportedcontenttype name="video/quicktime"/>
<supportedcontenttype name="audio/3gpp"/>
<supportedcontenttype name="video/3gpp"/>
<rtppayloaddef primarytype="audio" subtype="pcmu"
 rtppayload="0" encoding="PCMU"
 clockrate="8000" bitrate="64000"/>
<rtppayloaddef primarytype="audio" subtype="amr"
 rtppayload="96" encoding="AMR"
 clockrate="8000" bitrate="12200" mediaformatparameters=
 "mode-set=7; octet-align=1"/>
<rtppayloaddef primarytype="audio" subtype="telephone-event"
 rtppayload="101" encoding="telephone-event"
 clockrate="8000" bitrate="0"/>
<rtppayloaddef primarytype="audio" subtype="cn"
 rtppayload="13" encoding="CN"
 clockrate="8000" bitrate="0"/>
<rtppayloaddef primarytype="video" subtype="h263"
 rtppayload="34" encoding="H263"
 clockrate="90000" bitrate="52000"/>
</stream>

```

The default values of the stream attributes are usually adequate but in some situations defaultinboundptime and defaultinboundmaxptime may have to be modified, see the descriptions for those attributes below. Attributes should normally not be edited unless there is a specific reason to do so.

- The portpoolbase defines the starting port number in the RTP port pool.
- The portpoolsize defines the range of the port pool (the number of RTP/RTCP port pairs; a voice call will allocate two port pairs and a video call will allocate four).
- The movfileversion ("0" or "1") defines the MOV-file layout of recorded messages. The "0" layout must be used if the MAS shall coexist with MVAS. In a MAS only system the "1" layout is recommended. The "1" layout gives better performance when playing MOV-files, but the messages recorded by the MAS can not be played by MVAS. **NR.**
- The synccallmaxwaittimesec defines max timeout in seconds for a synchronous call to return before an exception is thrown.
- The maximum time in seconds a function call waits for the function to return before throwing an exception.
- The sendpacketsaheadms defines when the RTP packets should be sent (ahead: amount of time before packet timestamp).
- The expiretimeoutms defines the amount of time in microseconds before a packet expires in the send queue (a packet expires when exceeding its timestamp).

- The `streamabandonedms` defines the abandoned stream detection timeout in milliseconds.
- The `senderscontrolfraction` defines fraction (a value between 0 and 1) of the total control bandwidth to be dedicated to senders reports.
- The `audioreplacewithsilencems` defines the amount of audio, from the beginning (in milliseconds), that will be replaced by silence during a record.
- The `dispatchdtmfonkeydown` defines if DTMF events (over RTP) are handled/dispatched or not (if false the application will not receive DTMF events).
- The `maximumtransmissionunit` defines the maximum payload size of an RTP packet.
- The `audioskipms` defines the amount of audio that will be skipped (from the beginning) when a record is started.
- The `maxwaitforframems` defines maximum amount of time that the video stream will wait for an I-frame.
- The `skew` defines the skew between audio and video (the number of milliseconds that the audio is ahead of the video). Please note when applying a negative number the video will be ahead of audio.
- The `skewmethod`. Valid values are: "LOCAL", "RTCP" and "LOCAL\_AND\_RTCP". Default value is: "LOCAL". The `skewmethod` defines how the skew is handled.
  - "LOCAL" means that the local skew configuration (see above) is utilized to adjust the offset between audio and video both when recording and when playing.
  - "RTCP" means that information from RTCP is utilized to adjust the offset between audio and video when recording. With this setting the skew is not adjusted when playing.
  - "LOCAL\_AND\_RTCP" means that both the local skew configuration and information from RTCP is utilized to adjust the offset between audio and video when recording. When playing only the local skew configuration is used.
- The `defaultinboundptime` is the preferred ptime of inbound audio RTP streams. The ptime defines the recommended size (in milliseconds) of the RTP audio packages that MAS wants to receive. Even though this is the preferred RTP packet size, other sizes are accepted as well. Valid values are: "20" and "40". Default value is "40". This parameter should be the same as configured in the gateway. Refer to the table below.
- The `defaultinboundmaxptime` is the maximum amount of media (in milliseconds) that can be encapsulated in each RTP packet for inbound audio RTP streams. The value should be an integer multiple

of the defaultinboundptime attribute. The recommendation is to set defaultinboundmaxptime to the same value as defaultinboundptime. Refer to the table below.

*Table 6 Possible configurations for defaultinboundptime and defaultinboundmaxptime.*

<b>Audio media format</b>	<b>defaultinboundptime (ms)</b>	<b>defaultinboundmaxptime (ms)</b>	<b>Comment</b>
PCMU/PCMA	40	40	Default values. Recommended configuration for PCMU.
PCMU/PCMA	20	20	Configuration if gateway does not support 40ms RTP packet size.
AMR	20	20	Recommended configuration for AMR.
AMR	20	40	

- The outputprocessors defines the number of native threads used for outbound streaming.
- The inputprocessors defines the number of native threads used for inbound streaming.

The following parameters are explained in Section 5.5 on page 29:

- silencedetectionmode
- silencethreshold
- initialsilenceframes
- signaldeadband
- silencedeadband
- detectionframes
- silencedetectiondebuglevel

### **The supportedcontenttype group**

This group defines which content types that are supported by the RTP streams in the MAS. The default values are normally adequate.

- The name defines the supported media content mime type.

There can be multiple supportedcontenttype currently only audio/wav, audio/quicktime, audio/3gpp and video/3gpp (.WAV, .MOV and .3GP) are supported.

### The rtppayloaddef group

This group defines which RTP payload types that are supported by the MAS (and their corresponding mime type). The default values are normally adequate and should not be modified unless there is a specific reason.

- The primarytype defines the mime primary type (audio or video).
- The subtype defines mime sub type (amr, cn, h263, pcmu, pcma and telephone-event).
- The rtppayload defines the RTP payload type ID number.
- The encoding defines the encoding name.
- The clockrate defines the RTP clockrate.
- The bitrate defines the amount of bandwidth (bit/s) consumed by the payload type/codecs.
- The mediaformatparameters defines format specific parameters for the payload, corresponding to the fmp attribute in the SDP media descriptor.

**Note:** This parameter is currently only used for the AMR where two parameters must be set; “mode-set” and “octet-align”. Mode-set specifies the AMR mode-set to use according to the table below (only one mode at a time may be specified), octet-align must always be set to “1” and specifies that octet-aligned mode is used.

For AMR there is a relation between the bitrate and the mode-set defined in the mediaformatparameters. The following table defines this relation:

Table 7

mode-set	bitrate (bit/s)	Comment
0	4750	
1	5150	
2	5900	
3	6700	
4	7400	
5	7950	
6	10200	
7	12200	Recommended default

Normally the mode-set and bitrate are left unchanged. Before modifying mode-set and bitrate consider the following:

- The bitrate and the mode-set must be set according to the table above.

- The bitrate and mode-set must be set equal for the whole system including other MAS'es and entities handling stored messages, greetings and other media where AMR audio might be included.
- The bitrate and mode-set must match the settings in the installed media content packages.

There can be multiple payload types.

For each rtppayload group there are six optional parameters related to bandwidth modifiers that can be specified:

*Table 8 Optional parameters*

Name	Description
rs	RTCP bandwidth allocated to active data senders. This value is usually set by the client in the SDP offer. If no such value is provided, we use the value specified in this parameter in the SDP answer.
minrs	Minimum value accepted on rs
maxrs	Maximum value accepted on rs.
rr	RTCP bandwidth allocated to other participants. This value is usually set by the client in the SDP offer. If no such value is provided, we use the value specified in this parameter in the SDP answer.
minrr	Maximum value accepted on rr.
maxrr	Maximum value accepted on rr.

Here is an example of how an rtppayload group may look like with the optional parameters specified:

```
<rtppayloaddef primarytype="audio" subtype="pcmu"
 rtppayload="0" encoding="PCMU" clockrate="8000" bitrate="64000"
 rs="3000" minrs="0" maxrs="10000"
 rr="800" minrr="0" maxrr="10000"/>
```

### 7.5.1.16 The provisionmanager group

The provisionmanager group holds the configuration for the component in the MAS that handles provisioning of subscribers in the user directory. The group can have these attributes:

```
<provisionmanager connectionpoolsize="5"
 connectiontimeout="10000"
```

```

connectionidletimeout="60000"
commandsendretries="1"
defaultmailhost="" />

```

- The connectionpoolsize is max size on the connection pool with connections to a CAI server.
- The connectiontimeout is the socket timeout value in milliseconds to the provision host and also the time limit to wait for a free connection if all available connections in the pool are used.
- The connectionidletimeout is the millisecond time limit in milliseconds for a connection to be idle before it is physically closed.
- The commandsendretries is number of retries to do when sending a command that gets an error response.
- The defaultmailhost is the default mailhost to set for created subscribers. If this is empty, the MAS host will be used instead.

#### 7.5.1.17 The chargingaccountmanager group

The chargingaccountmanager group holds the configuration for the component in MAS that handles the access to an Account Information and Refill server (AIR server). These parameters are located in the chargingaccountmanager.xml file.

Example:

```

<chargingaccountmanager>
 <airnodes>
 <node host="localhost" port="4444" username="test" password="xxx" />
 <node host="brage.mobeon.com" port="1337" />
 </airnodes>

 <element name="accountActivationFlag" type="boolean"/>
 <element name="accountGroupID" type="integer"/>
 <element name="accountHomeRegion" type="integer"/>

 <elementgroup parent="messageCapabilityFlag" structtype="struct">
 <member name="promotionNotificationFlag"/>
 <member name="firstIVRCallSetFlag"/>
 <member name="accountActivationFlag"/>
 </elementgroup>
</chargingaccountmanager>

```

#### airnodes group

This group contains a list of node groups. MAS will access them in an “round robin per call” fashion.



**node group**

This group contains information about an AIR server. It can have the following attributes:

- host is hostname for the AIR server.
- port for the AIR server.
- username for the AIR server.
- pwd for the AIR server.

**element group**

This group is used to configure each parameter in the UCIP specification. UCIP is the protocol used to send data to an AIR server. The group can have the following attributes:

- name on the parameter.
- type is datatype on the parameter. It can be string, integer, boolean or date.

**elementgroup group**

Some parameters in the UCIP specification are grouped together when a request is made. This group is used to configure which parameters that belongs to a specific group. It can have the following attributes:

- parent is the name on the parameter that contains the other (grouped) parameters.
- structtype type on the group called struct in the protocol. It can be struct or array.

This group also contains a list of member groups.

**member group**

This group describes which parameters that are members of the specific elementgroup group. It can have the following attributes:

- name on the parameter.

**7.5.2****Application dependent configuration**

These configuration files are provided by the installed application and only the definition is provided by the MAS.

### 7.5.2.1 The trafficevents group

The trafficevents group is used to configure each traffic event that can be sent in the MAS. The purpose is to be able to filter which events that shall be sent. When the application tries to send an event, the MAS will only send the event if its name is listed in the configuration, and the value of the "enabled" attribute is true.

The events that can be sent is defined in the XML file trafficevents.xml.

```
<trafficevents componentname="vva">
 <event name="callended" enabled="true" type="radius"/>
 <event name="emailreread" enabled="true" type="radius"/>
 <event name="cliinformationmessage" enabled="true"
 type="email">
 <emailqueue sizelimit="100" timelimit="100"/>
 </event>
 <event name="mwioff" enabled="true" type="email">
 <emailqueue sizelimit="100" timelimit="100"/>
 </event>
 <event name="slamdowninformation" enabled="true"
 type="email">
 <emailqueue sizelimit="100" timelimit="100"
 multievent="true"/>
 </event>
 <event name="accountingdebit" enabled="true"
 type="radius"/>
 <event name="accountingrefund" enabled="true"
 type="radius"/>
</trafficevents>
```

- The "componentname" parameter is the name identifying the component in the traffic events. When sending the event MAS adds @<hostname> to the value. Example: **vva**@<hostname>.

### The event group

This group contains event specific parameters. The group can have these attributes:

```
<event name="callended" enabled="true" type="radius"/>
```

- The "name" attribute identifies an event sent by the application. The value of the "name" attribute will not be sent in the event itself. It is used only for filtering purposes. The MAS will only send events, which have their names defined in the configuration, and only if the "enabled" attribute is true. Must be lowercase.
- The enabled is a boolean value that indicates if this event should be enabled or disabled.

- The type can be one of radius or email. Indicates if this event should be sent via Radius-MA or email.

The event group can also have a emailqueue group. Only relevant if the event group has the attribute type set to email.

### **emailqueue group**

Configures the queue that is used for this event. The group can have these attributes:

```
<emailqueue sizelimit="100" timelimit="100" multievent="true"/>
```

- The sizelimit is max size of the queue before it is emptied.
- The timelimit is max time since last time it was emptied before the queue is emptied again.
- The multievent is a boolean value that indicates if all the traffic events in the queue should be sent in one email only. Default is false which means that there will be one email for each traffic event.

## **7.5.2.2**

### **The numberanalyzer group**

The numberanalyzer group holds the configuration for the component in the MAS that handles number analysis.

This definition is specified in XML schema file numberanalyzer.xsd.

Here is an example of the numberanalyzer group:

```
<numberanalyzer>
<rule name="INBOUND_CALL">
<subrule name="NumberLength">
<input expr=""/>
<return expr="4,12"/>
</subrule>
<subrule name="All">
<input expr="^(1[09])([0-9]{2})$"/>
<return expr="466010$i1$i2"/>
</subrule>
</rule>
<rule name="CALLEROUTDIAL">
<subrule name="NumberLength">
<input expr=""/>
<return expr="4,6"/>
</subrule>
<subrule name="Office">
<input expr="^(1[09])([0-9]{2})$"/>
<return expr="$i0$i1"/>
</subrule>
```

```

<subrule name="Local" regioncoderule="RegionCodes">
<input expr="^([1-9])([0-9]{4,5})$"/>
<return expr="$i0$i1"/>
</subrule>
</rule>
<rule name="RegionCodes">
<input expr="060,061,062"/>
</rule>
</numberanalyzer>

```

The numberanalyzer group contains a list of rule groups.

### rule group

A rule is a set of subrules (rule group) for a certain type of scenario, for example inbound call, subscriber outbound call or fax retrieval. Here is an example of rule named INBOUNDCALL:

```

<rule name="INBOUNDCALL">
<subrule name="NumberLength">
<input expr=""/>
<return expr="4,12"/>
</subrule>
<subrule name="All">
<input expr="^(1[09])([0-9]{2})$"/>
<return expr="466010$i1$i2"/>
</subrule>
</rule>

```

A rule group can have this attribute:

- The name defines the name on the rule

The rule group can have one or many subrule groups:

### subrule group

A subrule is a branch of a rule and is defined to match a specific number type. The order in which the subrules are listed determines which rule is checked first. The first rule that matches the requirement is the one that is used.

A subrule group can have these attributes:

- The name defines the name on the subrule
- The useregioncode is used to reference a name on a special rule that contains regioncodes. (Optional)

Each subrule can have one input group and one return group.

### input group

The attribute `expr` is a regular expression string which is matched against the number to analyze. If the expression matches, this rule applies for the number analysis.

### return group

The attribute `expr` is an expression string that specifies what should be returned as result after the number analysis. There are three different types of return expressions:

- Group return expression. It can contain identifiers to the regular expression pattern from the input group. The identifier has the following format `$i<group_id>`. The `group_id` relates to the number of the regular expression group. For example in the following pattern `^(555)(\d*)$` where `$i1` relates to the (555) part and `$i2` relates to the `(\d*)` part. In case there is no groups `$i0` references the whole result.

The return expression may also contain numeric strings that should be applied to the resulting number.

- Length return expression. It is used to check correct length on a number. The expression contains of a min value and a max value separated by comma. Note that the input expression is ignored when this return expression is used.
- Block return expression. It is used to say that the number is blocked if the rule matches the number. The value is "Block".

### 7.5.2.3

#### Traffic Event template files

Some traffic events in the MAS are sent as email. These E-mails can be configurable with templates that specifies the structure of the email. The different attributes in a traffic event should be set into different mail headers (and body) of an email

Each event that is going to be sent as an email has its own template file. The name on the file is the same as the name on the event and ".vm" as file type. For example the `mwioff` event has a template file called `mwioff.vm` and similarly the `slamdowninformation` event has a file called `slamdowninformation.vm`.

**Note:** These template files are not XML files.

Here is an example of the `mwioff.vm` template file.

```
From: sink@domain
Subject: ipms/message
To: notification.off@$event.mailhost
Ipms-Message-Type: notification
Ipms-Notification-Version: 1.0
Ipms-Component-From: emComponent=$event.component
Ipms-Notification-Type: mvas.subscriber.logout
Ipms-Notification-Content: $event.username
```

Junk

The attributes from the traffic event can be set by using references to the event (the \$event in the template). The names on the attribute are defined in Table 9 on page 82:

Here is an example how the mail looks like when it has been generated:

```
Message-ID: <193985.0114491944.JavaMail.user1@mas1.domain>
From: sink@domain
To: notification.off@mail1.domain
Subject: ipms/message
MIME-Version: 1.0
Content-Type: text/plain; charset=us-ascii
Content-Transfer-Encoding: 7bit
Ipms-Message-Type: notification
Ipms-Notification-Version: 1.0
Ipms-Component-From: emComponent=mas1.domain
Ipms-Notification-Type: mvas.subscriber.logout
Ipms-Notification-Content: user1
```

Junk

This table lists which attributes that can be present in a traffic event and a small description of what they are used for:

*Table 9*

Attribute name	Description
username	Identifies a user, e.g. a phone number or an email address.
callednumber	The called phone number involved in the traffic event.
callingnumber	The calling phone number involved in the traffic event.
accesstype	The type of access into the system.
mailhost	Subscriber's mail host.
sender	The phone number of the sender of the message.
destination	The phone number of a recipient.
language	The language of the receiver of the message.
date	A date and time regarding the traffic event.
action	An action regarding the traffic event.
forwardingnumber	The number that should be used to forward calls to the voice mail system.

Attribute name	Description
unsetforwards	The forwards that were turned off.
emailaddress	The recipient's email address.
component	The name on the MAS that sent the event. (Automatically set if not present in the event.)

**Note:** The attribute names must be lowercase.

## 7.5.3 MAS Log Configuration

### 7.5.3.1 General

The log configuration files for the MAS are XML based and is easily edited with an XML aware editor. Either one that can handle XML schemes or one that can check start and ending tags.

The log configuration is specified in logmanager.xml and logmanageragent.xml.

logmanageragent.xml manages the loggings made by the mas agent and the logmanager.xml manages the rest of the logging.

The logmanager.xml and logmanageragent.xml are automatically reloaded whenever a change is made to the configuration.

The configuration files consists of a set of root tags.

- appender

This tag defines what log appender the logger will use to write the log to.

The following appenders exist:

- STDOUT

Prints log information to the standard output.

- PLUGIN

Prints log information to the plugin.

- ROLLFILE

Prints log information to the file specified.

- category

Defines what should be logged, to which appender and loglevel.

- priority

Defines which loglevel this category should use.

- appender-ref

Defines which appender this category should use.

If appender is omitted, the appender is inherited from the category above or root.

In this case, the root category.

- root

Defines the root log level. The root log level applies to all to all categories which has not overridden the root log level. Specifies an appender and loglevel for logger.

- priority

Defines which loglevel this category should use.

- appender-ref

Defines which appender this category should use.

This is the starting point for the logger. If no categories is specified, the logger will use the loglevel and appender specified here.

The following loglevels exist:

- fatal

Logs fatal information

- error

Logs error information.

- warn

Logs error and warn information.

- info

Logs level prints error, warn and info information.

- debug

Logs error, warn, info and debug information.

To change the root log level, edit the logmanager.xml or logmanageragent.xml file and change the loglevel for the root tag.

Ex.

```
<root>
```



```

 <priority value ="warn" />
 <appender-ref ref="ROLLFILE" />
 </root>

```

to

```

 <root>
 <priority value ="debug" />
 <appender-ref ref="ROLLFILE" />
 </root>

```

### 7.5.3.2 Call Trace Configuration

In order to configure the call trace functionality you must enable the session filter, specify which session to filter and the log level of the log entries of interest. When choosing the proper log level you should consider that error and fatal log entries are not filtered and that the filter will only operate on log entries matching the configured log level. Hence, in order to benefit from the call trace functionality you should use log level debug or info in order to receive meaningful information.

The log entries are filtered according to the session log data (the phone number of caller party, called party or redirecting party). The means that every log entry where any entry in the session data matches the filter session data is forwarded to the log file. Example: we have a session filter with session data caller:1234 which only will forward log entries related to sessions where the caller party has phonenumber 1234.

There are exceptions though, log events which are not session related (log events with no session data), these log events are always forwarded by the session filter. The filter will only operate on log events that relates to the current log level.

In the log configuration file the session data is a list of parameters where each value is a key-value-pair. The value-key-pair is a string in following format: <key>:<value>.

Where key can be one of (case sensitive, must be lower case):

- **calling**; which is the calling party of a call session.
- **called**; which is the called party of a call session.
- **redirecting**; which is the redirecting party of a call session.

And value is (when the key is one of above) a phone number. Example: if the list here below is the session data of a session filter then any log entry matching any of the entries in the list will be forwarded to the log file.

```

<param name="TraceItem" value="calling:1507432">
<param name="TraceItem" value="calling:1507434">
<param name="TraceItem" value="called:5510242">

```

```
<param name="TraceItem" value="redirecting:123400">
```

To start trace calls:

1. Uncomment the XML clause [ **<filter class="com.mobeon.masp.logging.SessionFilter">**] in the <MAS\_HOME>/cfg/logmanager.xml file for the ROLLFILE appender.

```
<appender name="ROLLFILE"
 class="org.apache.log4j.RollingFileAppender">
 <param name="File" value="log/mas.log"/>
 <param name="Append" value="true" />
 <param name="MaxFileSize" value="10MB" />
 <param name="MaxBackupIndex" value="10" />
 <layout class="org.apache.log4j.PatternLayout">
 <param name="ConversionPattern"
 value="%d{ISO8601} MET %c %5p [TID:%t]
 [SID:%X{session}] - %m%n"/>
 </layout>
 <!-- if session filtering should be used,
 uncomment the following: -->
 <!-- from here -->
 filter class="com.mobeon.masp.logging.SessionFilter">
 <param name="TraceItem" value="calling:1234">
 </filter>
 <!-- to here -->
</appender>
```

2. Set the appropriate log level.
3. Update the Traceltem list with appropriate session data by modifying the line [<param name="Traceltem" value=" *calling:1234*">].
4. Save and quit configuration.

**Note:** Do not forget undo the changes when finishing the call trace by disabling/commenting the session filter and reset the log level to what ever it was before call trace was used.

### 7.5.3.3 Stream Logging

The Stream component is partly implemented in C++ and uses a log manager which is different from the rest of the MAS. The Stream logging is configured in the file stream.log.properties. By default the appender is a rolling file appender named R and the log output is stored in stream.log. Here follows a configuration example:

```
log4j.rootCategory=WARN, R
log4j.appender.R=org.apache.log4j.RollingFileAppender
log4j.appender.R.File=log/stream.log
log4j.appender.R.MaxFileSize=10MB
log4j.appender.R.MaxBackupIndex=2
```

```
log4j.appender.R.layout=org.apache.log4j.PatternLayout
log4j.appender.R.layout.ConversionPattern=%d{ISO8601} %c %5p %m%n
```

As you can see on the first line the log level is warn. In order to set the log level to info you have to update the first line in stream.log.properties to:

```
log4j.rootCategory=INFO, R
```

The log levels are as mentioned above, but in upper case: FATAL, ERROR, WARN, INFO and DEBUG.

Please note that call trace is not handled by this log manager. Session ID is used though, when available, hence you can use a particular session ID found when call tracing in mas.log and search/filter for that session ID in the file stream.log.