GEMINI
Mobile Technologies

Gemini Webmail System (GWS)

Administrator's Guide

Document Release 1.9, Febuary 25, 2010

Copyright Notices

Contents

Preface

1 GWS Overview

2 M2G Administration

3 Dot M2G Properties File

4 M2G Properties File

5 M2G Map Files

6 M2G Logging

7 A2S/M2H Administration

8 M2H Configuration Files

12 M2H, A2S and GDSS Logging

A Index of Settings in .properties and .conf Files

Preface

This preface provides information that may be helpful as you start using the *Gemini Webmail System Administrator's Guide*. The preface covers these topics:

Intended Audience

This manual is intended for administrators of the Gemini Webmail System (GWS). The manual assumes that the reader is familiar with Linux system administration.

Typographic Conventions

This manual uses the following conventions to help you quickly recognize certain types of information:

GWSGuide Typographic Conventions

Convention	Purpose	Examples
italics	Denote cross references or points of emphasis.	◆ *Contacting Gemini Mobile Technologies, on page 15* ◆ ...applies the *lesser* of these two settings...
`monospaced (Courier)`	Denotes file names, directory paths, process names, configuration setting names, and computer input or output.	◆ `server.properties` ◆ the `m2g` process ◆ the `CLI.allowedhosts` setting ◆ the `shutdown` command
`<bracketed monospaced>`	Denotes variables in command syntax or directory paths .	◆ `show config <param>` ◆ `set <param>=<value>` ◆ `<M2G_HOME>/etc`
`[bracketed monospaced]`	Denotes *optional* parameters in command syntax.	◆ `[-r <daterange>]` ◆ `[-i <interval>]`
|	Vertical bar separates options of which you can choose one and only one.	◆ Specify the timeout in format `<n>h|<n>m|<n>s` ◆ `show stat <stat>|all`

Glossary

Below is the standard glossary of terms used for the Gemini Webmail System:

Term	*Description*
2 in 1 handset	The "2 in 1 handset" in a handset means it has two e-mail addresses, one the *A Address*, the other, the *B Address*. The A Address is mainly used.
A2S	Authentication and Authorization Storage Server, an Erlang Mnesia distributed DBMS for highly scalable applications.
BE	Back End
BOXID	M2G's M2G data box (mail, …) Identifier
Ciphers	Small application to keep encryption secrets
EBF	Erlang Binary Format
ETQ	Erlang Term Queue. Implements M2H's Job Queue
External Mailbox	Mailbox registered in a player's mail profile. Mailbox is typically accessed by POP3 and hosted by an ISP.
FE	Front End
GDSS	Gemini Distributed Storage Server, an Erlang-based, distributed store for highly scalable applications.
GTRID	Global Transaction Log (Item) ID. This is an atom and a unique 32 byte string pair. For example: {s1b,7d7760c8d1d44576af088298a5cdd30b}. s1b is the inferface of the original transaction came in over (the example given here is the Docomo SMTP listener).
ISP	Internet Service Provider
I/F	Interface
mail domain	The second half of the email address, similar to `xxx@mail.goo.ne.jp` or `xxx@goo.jp`.
MIMESUBPART ID	M2G's MIME sub-part Identifier. The sub-part index calculated by a breadth-first enumeration of a MIME message's sub-parts.
M2BE	Mail 2.0 Back-end. Implements M2H's "back-end" component.
M2CI	Mail 2.0 Client-interface. Implements M2H's "client-interface" component.
M2FE	Mail 2.0 Front-end. Implements M2H's "front-end" component.
M2G	Messaging 2.0 Gateway (hre)
M2H	Messaging 2.0 Helper (erlang, mnesia)
M2SI	Mail 2.0 Search Index. Implements M2H's Search Index component.
MTA	A mail transfer agent (MTA) (also called a mail transport agent, message transfer agent, or smtpd (short for SMTP daemon), is a computer program or software agent that transfers electronic mail messages from one computer to another.

Term	*Description*
OPCOID	M2G's Operator Identifier
RESERVEDID	M2G's reserved YAGUID sub-id
SERVICEID	M2G's Service Identifier
SMTPSVR	SMTP plugin listener
SMTPSVRX	SMTP Active X control plugin listener
TRID	Transaction Log (Item) ID
UBF	Erlang's Universal Binary Format. UBF is designed to be efficient and easy to implement. No grammar checking or parsing is involved and are compact. Since parsing an import stream involves looking at every character on the input, UBF parsing is intrinsically efficient. UBF can also use caching optimization for further efficiency improvement.
UID	Unique IDentifier
UIDL	Unique IDentifier List(ing). POP3 command and/or a list of UIDs.
VERSIONID	M2G's Version Identifier
VCARD	Virtual business card
YAGUID	Yet Another Global Unique ID is used for exchange between M2G and external clients and systems. YAGUID is constructed from the following sub-ids: ◆ VERSIONID (1 bytes) ◆ OPCOID (2 bytes) ◆ SERVICEID (2 bytes) ◆ PLAYERID (4 bytes) ◆ BOXID (2 bytes) ◆ UID (8 bytes) ◆ MIMESUBPARTID (2 bytes) ◆ RESERVEDID (4 bytes) Sub-ids are represented as non or zero padded hexadecimal 1 byte characters. Sub-ids are joined together by the '-' 1-byte character separator. The maximum length of a YAGUID is 32 bytes (and does not include the '\0' 1-byte character that terminates c-style strings). The "undefined" YAGUID is "0-00-00-00-0000-00-00000000-0000". The "maximum" YAGUID is "f-ff-ff-ff-ffff-ff-ffffffff-ffff". External clients and systems MUST treat YAGUID as an opaque identifier.

Below is the custom glossary of terms used for the NTTR version of the Gemini Webmail System:

Term	*Description*
CiRCUS	NTTR's i-mode system.
Cuenote SR-S	Mail "notification" service for CiRCUS.
goo Search Engine(AKA "EVA")	This is the search engine provided by NTTR. Our M2G component talks to this system via an HTTP API interface when receiving/sending/deleting/searching messages etc.

Term	*Description*
IDC	It is goo-ID Center where redIDs and gooIDs are managed.
i-mode	NTT-R DoCoMo's i-mode is a wireless internet service popular in Japan.
New goo Mail System	NTTR(http://www.goo.ne.jp/) has provided two e-mail services, one a "free mail service", the other one is an"advanced mail service" (200 YEN/ month). NTTR is replacing these mail services with the new system.
RED Mail System	DoCoMo is providing i-mode (internet mode) mobile phone subscribers with service to be able to see i-mode mails on their personal computer, similar service to one that au-one provides with KDDI (http:// www.auone.jp/index.html). If an i-mode subscriber registers for this service, CiRCUS system will forward a message to the system. Then the user can see i-mode mails using their PC browser/i-mode browser and PC mail client.
SHIVA	This gateway responds to the number of new arrival messages for red/ goo users to the client then accesses M2G's admin interface to get the number. Please see NTTRHTTPListenerM2GForAdmin in detail.
URA-ID	NTTR's Unique ID for Subscriber.

Contacting Gemini Mobile Technologies

To reach Gemini technical support, use the customer-specific email address that Gemini provides you.

To comment on this manual, use this email address:

```
pubs@geminimobile.com
```

1 GWS Overview

This chapter provides a high level overview of the basic design of the Gemini Webmail System, the GWS, and includes these overview topics:

Overview

Gemini Webmail System provides NTTR Red Mail mobile phone subscribers with the ability to check e-mails from their PC i-mode browser or their PC mail client.

In addition, the Gemini Webmail System replaces NTTR's goo mail system, a web-based email service provided by NTT Resolant Inc., Japan with two levels of service, one a "free mail service", the other a paid "advanced mail service".

GWS Components

There are three main componets of the GWS: the M2G+M2H, the A2S, and the GDSS.

The **M2G+M2H** is the Messaging 2.0 Gateway, Gemini's HyperScale® Runtime Environment (HRE) plus the Messaging 2.0 Helper and Erlang-based, Mnesia Database.

The **A2S** is the Authentication and Authorization Storage Server, an Erlang-based, Mnesia distributed DBMS.

The **GDSS** is the Gemini Distributed Storage Server, an Erlang-based, distributed DBMS for highly scalable applications.

The following diagram shows the components and the protocols used to communicate internally and externally with the system.

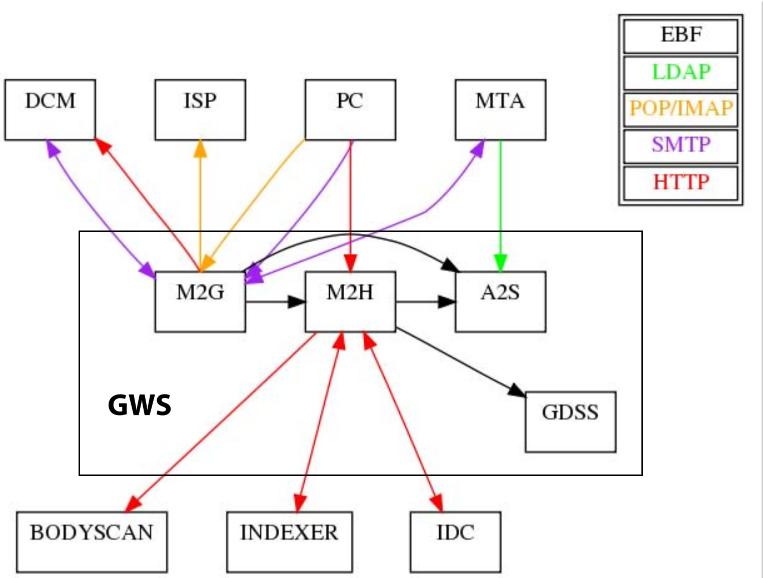


Figure 1 NTTR Components and Protocols

The five mail flows for the system are described below:

1 **internet ===> local**

The M2G gets a message from internet via VIRUS MTA on the SMTP Listener Interfaces below:

- NTTRMessageMIMEStructures
- NTTRSMTPListenerForInternet

2 **local ===> local**

Regardless of the destination, the M2G sends a message to the VIRUS MTA. The VIRUS MTA returns the message back to the SMTP Listener Interface on the M2G.

- NTTRSMTPClientForInternet
- NTTRSMTPListenerForInternet

3 **local ===> internet**

Regardless of the destination, M2G will send a message to VIRUS MTA and VIRUS MTA will deliver it to Internet.

- NTTRSMTPClientForInternet

4 **CiRCUS(i-mode) ===> local**

The M2G gets an i-mode forwarded message from CiRCUS on the SMTP Listener Interface below:

NTTRSMTPListenerForCiRCUS

5 **local ===> CiRCUS(i-mode) [SMTP Notification to CiRCUS]**

M2G will send a notification message to CiRCUS MTA if it's necessary.

- NTTRSMTPClientForCiRCUS

A Cuenote SR-S server is responsible for delivering notification messages to CiRCUS. The Gemini Webmail System communicates with this server via HTTP-based I/F when notification to CiRCUS is necessary.

Incoming Message Processing

The following diagram shows three inflows: the SMTP Listener for the Internet, POP Client for external POP and the SMTP Listener for CiRCUS.

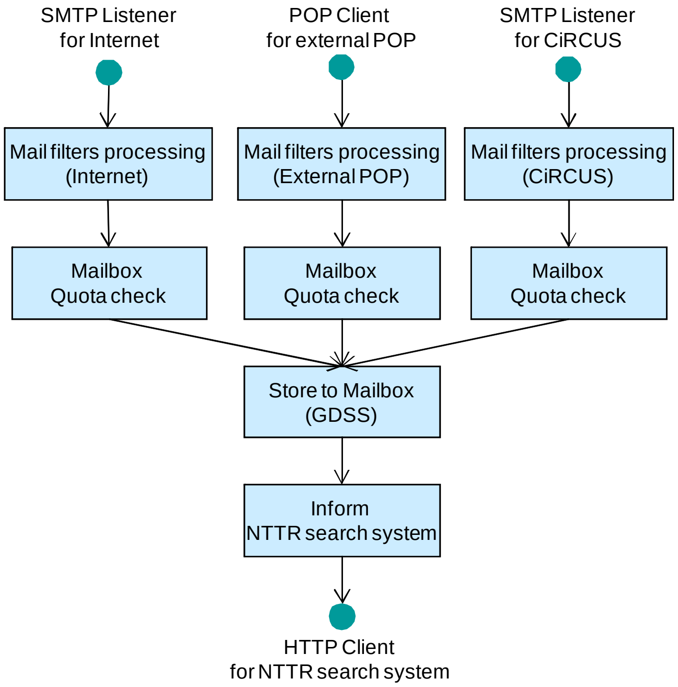

SMTP Listener for Internet
POP Client for external POP
SMTP Listener for CiRCUS
Mail filters processing (Internet)
Mail filters processing (External POP)
Mail filters processing (CiRCUS)
Mailbox Quota check
Mailbox Quota check
Mailbox Quota check
Store to Mailbox (GDSS)
Inform NTTR search system
HTTP Client for NTTR search system

Figure 2   Incoming Message Processing

There are four steps to process and incoming message. Messages are filtered, mailbox quotas are checked, messages are stored into the GDSS mailbox, then the message information is forwarded to the NTTR search system.

The filter processor examines each filter rule in order. If the trigger list matches the mail header, the filter's actions are appended to the pending actions list for the mail. There are three types of filters:, user, system and a custom filter.

When both black/white list filters are set, if the envelope-from matches the black list but doesn't match the white list, the mail is dropped. Otherwise, the mail is accepted and processing continues to the next filter. This allows you to configure `"drop all mail from the domain ‘example.com’ except mails from ‘tom@example.com’"`.

Quotas are set for individual mailboxes. If the total size (number of messages or bytes) is reached to the mailbox limit, then error processing is performed.

Outgoing Message Processing

The diagram below shows the steps in processing an outgoing message.

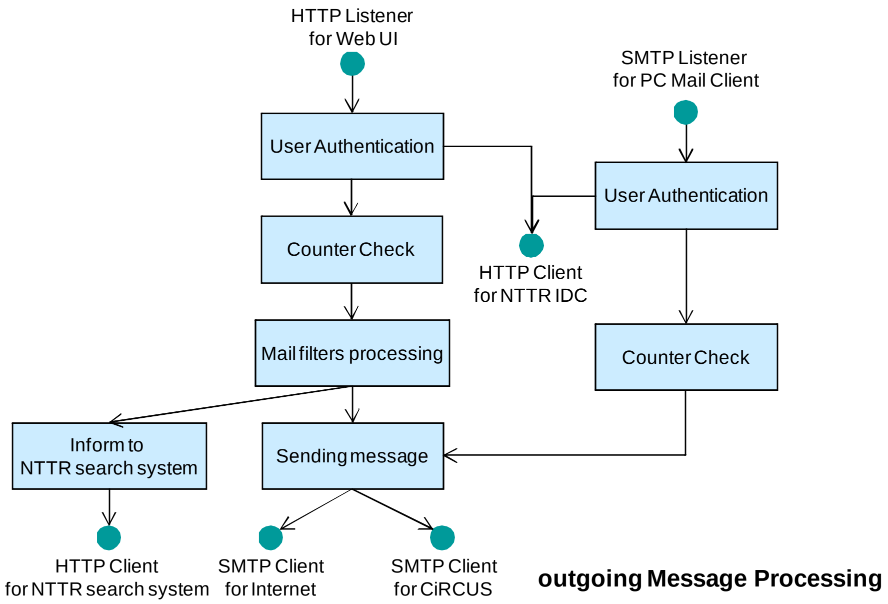


Figure 3

First the NTTR IDC (goo-ID Center where redIDs and gooIDs are managed) executes user authentication.

A counter check is then performed. There are two counters, one for the user id and the second for the source IP address. The counters count the number of sending messages in a specific period. If either counter reaches to the limit, the system rejects sending the message.

After mail filter processing is performed the sending message information goes to NTTR search system.

Client and Server Interfaces

The following diagram shows the server and client interfaces for the M2G, M2H and A2S.

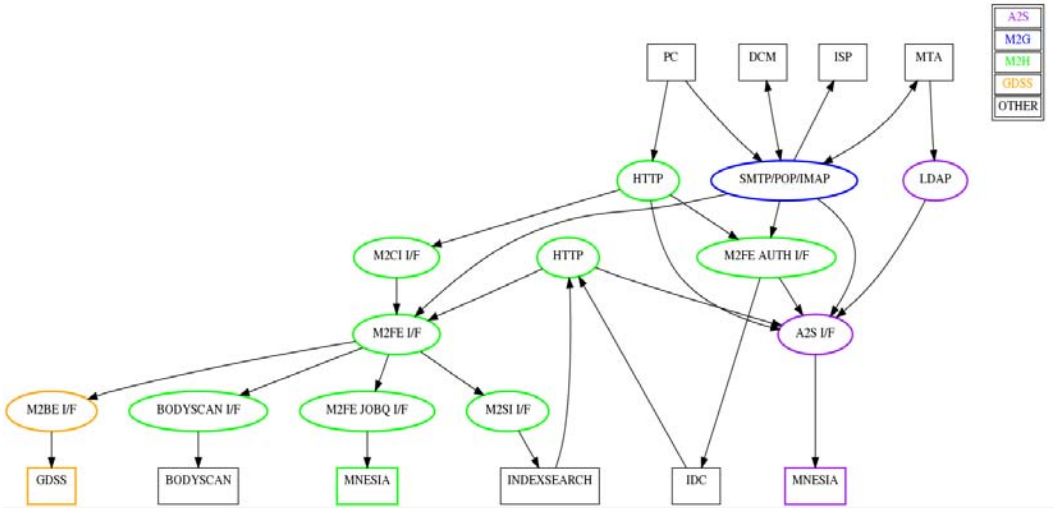


Figure 4 Client and Server Interfaces

The M2G server and client interfaces are summarized in the following table.

M2G Interfaces

Interface	Purpose
SMTP Listener Interface for Internet	This is used for receiving incoming messages from Internet. There are spam filter and virus check MTA servers between the M2G and the Internet. The client is the spam filter/virus check server.
SMTP Client Interface for Internet	This is used for sending outgoing messages from the GWS to the Internet.
SMTP Listener Interface for PC Mail Client	This is used for receiving messages from the PC Mail Client.
SMTP Listener Interface for CiRCUS(i-mode)	This is used for receiving forwarded messages from CiRCUS(i-mode).
POP Listener Interface for PC Mail Client Listener	This is used for retrieving messages from PC Mail Client.
POP Client Interface for external ISPs	This is used for retrieving messages from other ISPs POP servers.

The M2H server and client interfaces are summarized in the table below.

M2H Interfaces

Interface	Purpose
HTTP Listener Interface for FLASH/AJAX/CHTML	This is the core HTTP API Interface for FLASH/AJAX/CHTML. It provides all operation HTTP APIs to send/retrieve/move/delete mail, list, sort, search and so on.
HTTP Client Interface for Goo Search Engine(aka "EVA")	This is used for communicating with "EVA" system such as store message info, delete message info, search messages and so on.
HTTP Listener Interface for Goo Search Engine(aka "EVA")	This is used when EVA wants to have all messages for the user to re-create message indexes on EVA side, responding with all message information for the user.
HTTP Client Interface for Goo Body Parse Engine	This is used for communicating with goo body parse engine system when a user views a message body from the UI.
HTTP Client/ Listener Interface for goo ID Center System	This is used for subscriber's provisioning/deprovisioning from goo ID Center System
M2G HTTP Listener Interface for admin	This is the Administrative Interface for the M2G for user account/ mailbox operations such as view/add/modify/delete etc...

A2S Interface

There is one listener interface for the A2S, the LDAP Listener Interface for Spam Filter Server. This interface is used when the Spam Filter MTA receives messages from the Internet to identify a user’s existance and user attributres for spam action.

NTTR Node Topology

The following node topology diagram shows the GWS services, interfaces, load balancers, protocols and message flows used by the Gemini Webmail System:

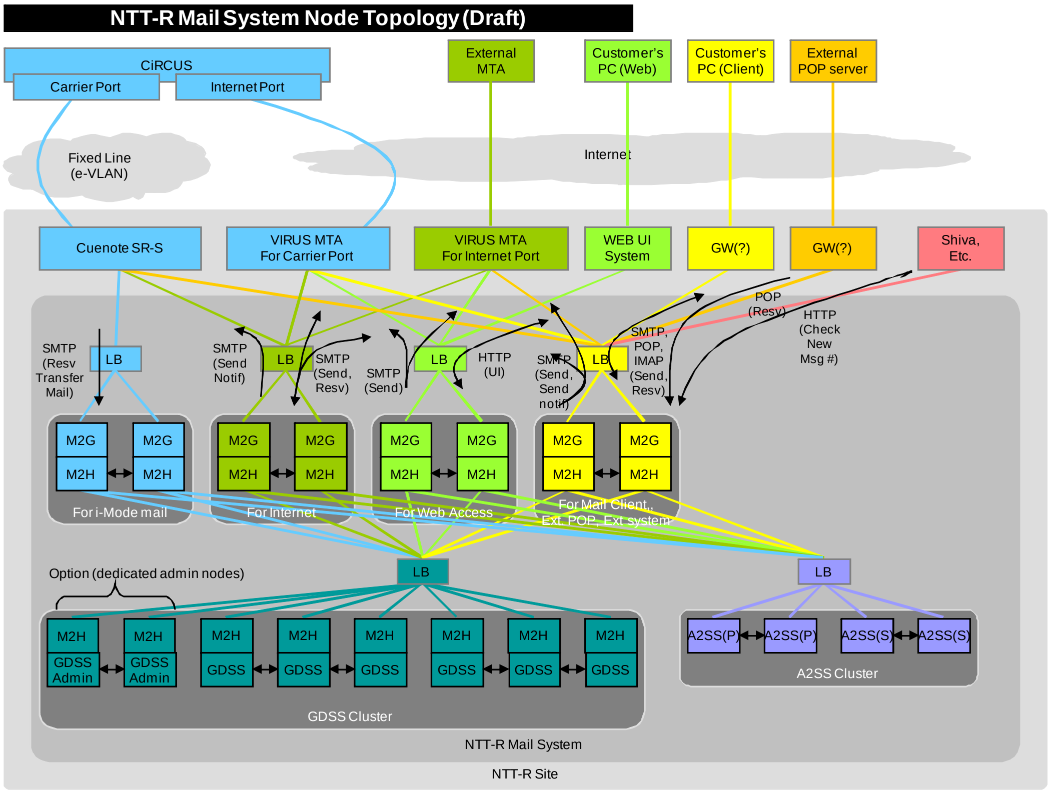


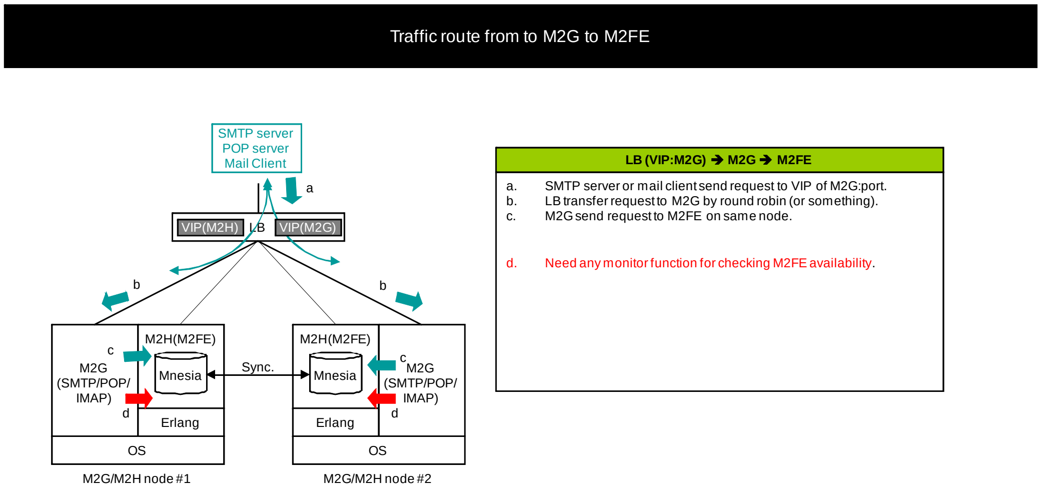
Figure 5   NTTR Mail System Topology

Traffic Routing

The four diagrams that follow show various traffic routes with load balancers. A fifth diagram shows traffic to the user profile and message storage.

- *M2G to M2FE, on page 25*
- *M2CI to M2FE, on page 26*
- *Anti-spam Servers, M2G/M2H to A2S, on page 26*
- *M2BE to GDSS, on page 27*
- *A2S and GDSS Profile and Message Store, on page 28*

M2G to M2FE


Traffic route from to M2G to M2FE

LB (VIP:M2G) → M2G → M2FE

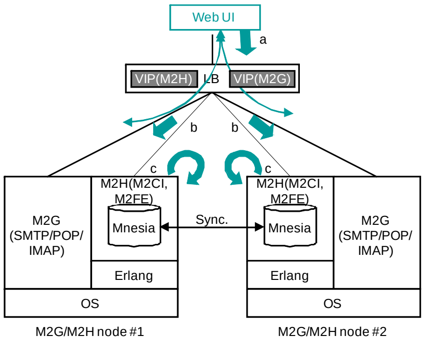
a. SMTP server or mail client send request to VIP of M2G:port.
b. LB transfer request to M2G by round robin (or something).
c. M2G send request to M2FE on same node.

d. Need any monitor function for checking M2FE availability.

Figure 6 M2G to M2E Traffic

M2CI to M2FE

Traffic route from to M2CI to M2FE



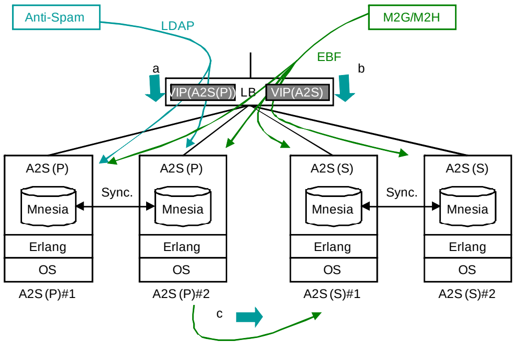
LB (VIP:M2CI) ➔ M2CI ➔ M2FE

a. Web UI system send request to VIP of M2CI:port.
b. LB transfer request to M2CI by round robin (or something).
c. M2CI send request to M2FE on same node.

Figure 7 M2CI to M2FE Traffic

Anti-spam Servers, M2G/M2H to A2S

Traffic route from Anti-Spam to A2S (P) and from M2G/M2H to A2S



If the request is write request, request re-route from A2S (P) to A2S (S) automatically.

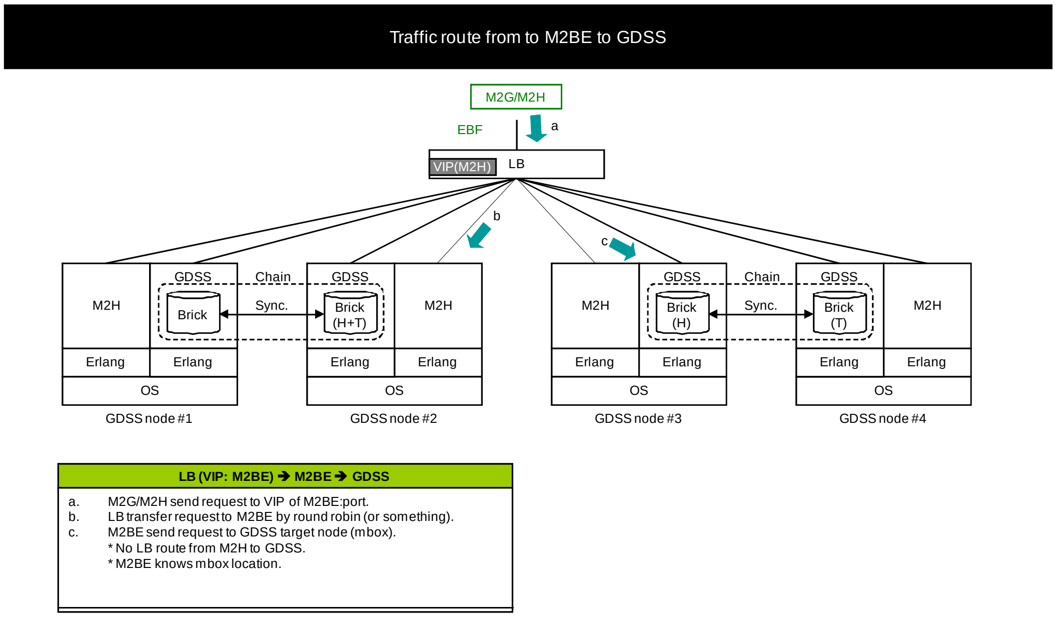
LB (VIP:A2S(P) ➔ A2S (P), LB (VIP:A2S) ➔ A2S (P &S)

a. Anti-Spam send request to VIP of A2S (P)

b. M2G/M2H send request to VIP of A2S (P&S)
c. If request is write request, A2S (P) re-route to A2S (S) automatically via erlang connectivity.

Figure 8 Anti-spam and M2G/M2H Traffic

M2BE to GDSS



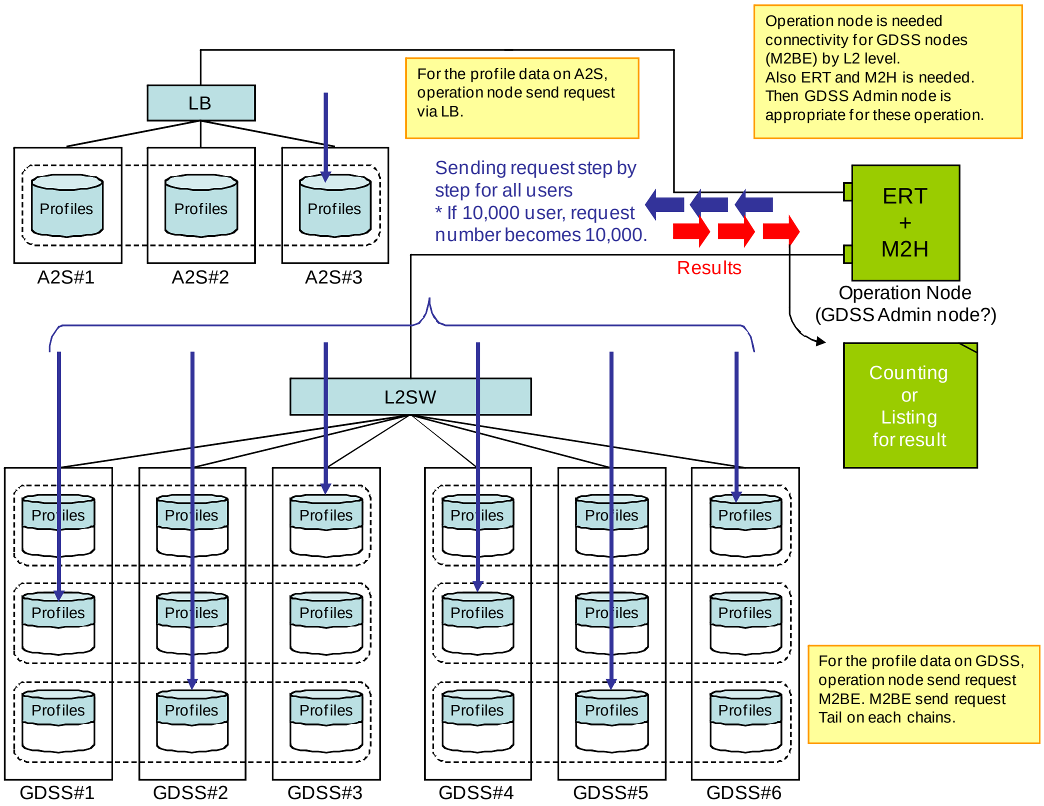
Traffic route from to M2BE to GDSS

LB (VIP: M2BE) → M2BE → GDSS

a. M2G/M2H send request to VIP of M2BE:port.
b. LB transfer request to M2BE by round robin (or something).
c. M2BE send request to GDSS target node (mbox).
* No LB route from M2H to GDSS.
* M2BE knows mbox location.

Figure 9 M2BE to GDSS Traffic

A2S and GDSS Profile and Message Store



For the profile data on A2S, operation node send request via LB.
Operation node is needed connectivity for GDSS nodes (M2BE) by L2 level.
Also ERT and M2H is needed.
Then GDSS Admin node is appropriate for these operation.
LB
Profiles
A2S#1
A2S#2
A2S#3
Sending request step by step for all users
* If 10,000 user, request number becomes 10,000.
Results
ERT
+
M2H
Operation Node
(GDSS Admin node?)
Counting or Listing for result
L2SW
GDSS#1
GDSS#2
GDSS#3
GDSS#4
GDSS#5
GDSS#6
For the profile data on GDSS, operation node send request M2BE. M2BE send request Tail on each chains.

Figure 10   A2S and GDSS Profile and Message Store

2 M2G Administration

This chapter describes administration tools for the M2G going into detail concerning the use of the M2G Server command line interface (CLI).

The chapter covers these topics:

M2G Administration Overview

The M2G provides a full set of administration tools to help you launch, operate, and monitor the messaging center

Automated Installer

The M2G's automated installer allows for easy and flexible product installation. The installer lets you make a set of initial product configuration choices, and to replicate settings across multiple installations. An uninstall option is also provided.

Initialization Script

The M2G includes an initialization script that makes it simple to:

- Start M2G server
- Restart M2G server
- Shut down M2G server
- Check the status of M2G processes
- Display M2G version information

The script implements appropriate environment settings for the M2G, and may be easily integrated into your Linux boot routine.

For further information on using the initialization script, see

Log Files

The M2G generates an application log and a transaction log.

Application Log

The M2G application logs record application-related alerts, warnings, informational messages, and debug messages. Each time-stamped log entry indicates the process ID and module with which the event is associated, as well as an event severity level and a brief descriptive message. For all events of level "INFO" or higher, the entry includes a numerical message code that aids in identifying and responding to the event.

- For M2G application logging configuration, see *page 81*.
- For general information about M2G application log content, see*page 207* .

- For a list of M2G application log messages of level INFO or higher, see the *MGS Error Guide*. For each message, this appendix briefly indicates the condition that caused the message, the condition's effect, and if appropriate, the suggested corrective action.

Transaction Log

The M2G transaction log records transactions in which the M2G acts as a server to incoming requests as well as transactions in which the M2G initiates requests as a client. A wide and configurable range of transaction data can be included in each time-stamped transaction record.

- For M2G transaction log configuration, see *page 85*.
- For transaction log content, see *page 210*.

Statistics

The M2G provides extensive statistical reporting to help you monitor the gateway's operations and performance. Statistics are available to you in several different forms.

- A limited set of *real-time* statistics is available through the M2G's command line interface. Through the `show stat` command you can instantly check on important state indicators such as the number of connections currently held by M2G listeners and connection pools or the number of messages in M2G delivery queues.
- The same set of statistics that is available through the CLI can also be recorded to the M2G's application log at a configurable interval, providing a series of snapshots of M2G status throughout the day.

Real-Time Statistics Viewable Through the CLI

Through M2G administrative commands, you can retrieve these real-time statistics:

- For each listener: connections currently open, total connections handled, and total connections refused
- For each connection pool: connections currently open, connections currently in use, number of spare connections, and number of queued requests waiting for a free connection

For further information, see *Viewing Real-Time Statistics (show stat), on page 49*.

Periodic Statistics Writes to the Application Log

The same set of statistics that is available through the CLI can also be recorded to the M2G's application log at a configurable interval, providing a series of snapshots of M2G status throughout the day.

Configuration Files

The M2G gives you configuration control over a wide variety of operation and service settings. Though you can update certain settings dynamically through CLI commands, the primary means of modifying your M2G's configuration is by editing configuration files. More information about configuring the M2G can be found in two chapters, *Chapter 2, M2G Configuration Overview* and *Chapter 4, M2G Properties File*.

About the M2G CLI

The M2G supports a command line interface through which you can perform a variety of administrative tasks.

Two versions of the command line interface listener are available for configuration. The `CLI` listener is a standard node-based listener for which incoming requests are load-balanced across `hgs` processes running on the host. By contrast, the `CLIX` listener is a process-based listener that allows for requests to be directed to a particular `hgs` process. Under normal circumstances you should use only the node-based `CLI` listener. The `CLIX` listener is provided primarily as a simple means of testing whether a particular server process is running and able to accept a connection. The `CLI` listener and the `CLIX` listener are separately configurable.

CLI Commands

The M2G includes a command line interface (CLI) server that supports a pre-defined set of administrative commands. With these commands you can perform a range of important operational tasks.

M2G Commands

For the M2G, you can perform the following tasks by using CLI commands:

- Check the status of M2G listeners and connection pools to see whether they are functioning normally
- Test connections to M2G listeners and from the M2G to peer servers
- View settings from M2G configuration properties files

- Dynamically change certain configuration properties, without having to restart the M2G
- View settings from M2G configuration map files
- Dynamically reload configuration map files without having to restart the M2G
- View real-time interface statistics
- Initialize real-time interface statistics
- Bind, unbind, or rebind connections to peer servers
- Shut down the M2G

About Shutdown

You can shut down an M2G server by using its initialization script or its command line interface. You can also use the initialization script to perform a restart whereby the server stops momentarily and then starts up again. In any of these operations, there is the question of how the server should handle open incoming and outgoing connections at the time that the server receives your shutdown command. You can configure whether the server simply drops the connections immediately, or rather exits in a more “graceful” manner that allows some time to complete in-progress transactions.

Command Line Interface (CLI)

The M2G supports a command line interface (CLI) through which you can perform a variety of administrative tasks. This section covers these introductory CLI topics:

- *Configuring the CLI, on page 34*
- *Connecting to the CLI, on page 34*
- *CLI Restrictions and Errors, on page 35*
- *CLI Logging, on page 35*
- *M2G CLI Commands at a Glance, on page 36*

Configuring the CLI

The M2G command line interface is configured by the `CLI` and `CLIX` parameters in the `<M2G_HOME>/1.0.0/etc/m2g.properties` file.

Connecting to the CLI

To use the M2G command line interface, telnet to its listening port, which by default is 21023. When you do so, your terminal should display the CLI banner and prompt. By default, the prompt is:

```
M2G>
```

Note If you want to connect to a process-based version of the CLI listener rather than to the standard node-based CLI listener, the default port number is 21024 plus the offset number of the particular `M2G` process in which the listener is based. For the process-based CLI listener, the prompt is `M2GX>`.

IMPORTANT If your CLI listener configuration includes a non-null setting for `.allowedhosts` and you try to connect to the CLI from an IP address other than one of the allowed ones, your connection attempt will be rejected.

CLI Restrictions and Errors

The CLI server is case-sensitive in its acceptance or denial of commands. This restriction comes into play particularly with commands that require component names as arguments. Some component names are partially or entirely in caps, and must be specified as such when used as command arguments.

By default, the CLI server accepts only ASCII printable characters as input. However, you can override this restriction by configuration.

You may encounter two common error types while using the CLI:

- Errors indicating an unrecognized command. If you receive this error, refer to this chapter to make sure that you are using a valid command.
- Errors indicating that the item could not be found or that the action could not be performed. This most often indicates that you have used an invalid argument with the command. Refer to this chapter to make sure that you are specifying a valid argument.

CLI Logging

CLI sessions are recorded in the M2G transaction log. Log entries are written for the session initialization and for each command-response transaction.

Example The sample below shows the transaction log record of a CLI session in which a user connects to the CLI listener port, enters the command `show stat all`, and then enters the session-ending command `quit`.

```
8063   04/08/2007:09:32:42   CLI                          CLI   OK
CLI.44DB5FAA.1F7F.0   127.0.0.1   pcx.ext.geminimobile.com   CLI
command. Session START.
8063   04/08/2007:09:32:42   CLI                          CLI   OK
CLI.44DB5FAA.1F7F.1   127.0.0.1   pcx.ext.geminimobile.com   CLI
command. c:show stat all s:OK
8063   04/08/2007:09:32:42   CLI                          CLI   OK
CLI.44DB5FAA.1F7F.2   127.0.0.1   pcx.ext.geminimobile.com   CLI
command. Session QUIT.
```

M2G CLI Commands at a Glance

The table below briefly describes each M2G CLI command. For details including command/response samples, see the referenced pages.

M2G CLI Commands

Command	Description	Details
`show status <COMPONENT>|all`	Checks whether M2G interface components are functioning normally.	*page 37*
`test component <COMPONENT>|all`	Confirms that M2G listeners are accepting commands, and that commands can be sent from M2G connection pools to peer servers.	*page 39*
`command <COMPONENT> bind|unbind|rebind`	Binds, unbinds, or rebinds the M2G's outgoing connection pools.	*page 41*
`show config <property>|all`	Shows configuration property settings.	*page 43*
`set <property>=<value>`	Changes configuration property settings.	*page 45*
`command <COMPONENT> dump <dumptofilename>`	Outputs map file data that the M2G currently has in its memory.	*page 47*
`reload <COMPONENT>.mapfile`	Dynamically reloads map files.	*page 48*
`show stat <statistic>|all`	Shows real-time statistics.	*page 49*
`reset stat <statistic>|all`	Initializes real-time statistics.	*page 52*
`command CHARSETCONV [show_|set_]replacementchar <char>`	Shows or updates replacement for invalid characters.	
`shutdown`	Shuts down the M2G.	*page 54*
`noop`	Confirms that the CLI listener is available.	*page 55*
`help`	Shows a list of supported commands.	*page 56*
`quit`	Terminates your session with the CLI.	*page 57*

Checking the Status of Interfaces (show status)

Use the `show status` command to check whether M2G interface components are functioning normally or are in an error condition. The command syntax is as follows:

```
show status <COMPONENT>|all
```

Parameter	Description
`<COMPONENT>`	The M2G component to be tested. For listeners, you can confirm that the listener is accepting connections and commands. For connection pools or outgoing interface managers, you can confirm that the M2G can connect and submit a command to the peer server.
	Listeners
	◆ `CLI` (CLI Listener) ◆ `CLIX` (CLIX Listener) ◆ `XCONV` (Transcoding Conversion EBF Listener) ◆ `mta/SMTPSVR` (Internet SMTP lLstener) ◆ `dcm/SMTPSVR` (DoCoMo SMTP Listener) ◆ `pcc/SMTPSVR` (PC CLIENT (Relay) SMTP Listener) ◆ `pcc/SMTPSSLSVR` (PC CLIENT (Relay) SMTP SSL Listener) ◆ `pcc/POPSVR` (PC CLIENT POP3 Listener) ◆ `pcc/POPSSLSVR` (PC CLIENT POP3 SSL Listener)
	Connection Pools and Outgoing Interface Managers
	◆ `M3CONMGR` (M3 Connection Manager) ◆ `LMTPPOOL0` (LMTP mail server connection pool) ◆ `POP3POOL0` (POP3 mail server connection pool) ◆ `POP3POOLMGR` (POP3 mail server connection pool manager) ◆ `SMTPPOOL0`(SMTP mail server connection pool) ◆ `a2s/EBFMGR` (A2S EBF connection pool manager) ◆ `a2s/EBFPOOL` (A2S EBF connection pool manager) ◆ `fe/EBFMGR` (Mail Front End EBF connection pool manager) ◆ `fe/EBFPOOL` (Mail Front EndEBF connection pool) ◆ `fe_auth/EBFPOOL` (Mail Front EndAuthorization EBF connection pool) ◆ `fe_jobq/EBFPOOL` (Mail Front JOBQ EBF connection pool) ◆ `HTTPNOTIFYQMGR` (HTTP Notify Queue manager) ◆ `HTTPNOTIFYPOOL` (HTTP Notify Queue connection pool) ◆ `SMTPNOTIFYMGR` (SMTP notification manager) ◆ `MTACONNPOOL` (Internet mail MTA connection pool) ◆ `DCMMTACONNPOOL` (DoCoMo mail MTA connection pool)
`all`	Use `test component all` to test all the above components at once.

In response to the `show status` command, the CLI will return for each requested component one of several possible statuses:

- `OK`
- `NA` (component not available)
- `WARNING <message>`

Possible `WARNING` messages depend on the component type:

- Listeners
  - `WARNING interface is unbound`

    The listener has been closed due to the configurable congestion limit `controlhigh` being exceeded.
- Connection pools
  - `WARNING Connection pool suspended.`

    The connection pool has been suspended due to manual operations such as the `unbind` command, or due to issues with the target peer server.
  - `WARNING Auto-reconnect disabled.`

    The connection pool's configurable `autoreconnect` setting is set to `false`, preventing the creation of new connections.
  - `WARNING Maximum connections reached.`

    The connection pool's configurable `maxconnections` limit has been reached.
  - `WARNING Maximum in-use connections reached.`

    The connection pool's configurable `maxinuse` limit has been reached.
  - `WARNING Maximum waiting connections in wait queue.`

    The connection pool's configurable `maxwaitqueue` limit has been reached.
  - `WARNING Unknown status.`

    A residual category indicating a warning status other than those listed above.

Example The sample below shows a successful check of the M2G's CLI listener. The listener's status is "OK".

```
M2G>show status CLI
OK
```

Testing Interfaces (test component)

Use `test component` to confirm that M2G listeners accept commands, and that commands can be sent from M2G connection pools to peer servers. The `test component` command sends a keep-alive check to the appropriate listening ports and then reports a status message.

The command syntax is as follows:

```
test component <COMPONENT>|all
```

Parameter	Description
`<COMPONENT>`	The M2G component to be tested. For listeners, you can confirm that the listener is accepting connections and commands. For connection pools or outgoing interface managers, you can confirm that the M2G can connect and submit a command to the peer server.
	Listeners
	◆ `CLI` (CLI Listener) ◆ `CLIX` (CLIX Listener) ◆ `XCONV` (Transcoding Conversion EBF Listener) ◆ `mta/SMTPSVR` (Internet SMTP ILstener) ◆ `dcm/SMTPSVR` (DoCoMo SMTP Listener) ◆ `pcc/SMTPSVR` (PC CLIENT (Relay) SMTP Listener) ◆ `pcc/SMTPSSLSVR` (PC CLIENT (Relay) SMTP SSL Listener) ◆ `pcc/POPSVR` (PC CLIENT POP3 Listener) ◆ `pcc/POPSSLSVR` (PC CLIENT POP3 SSL Listener)
	Connection Pools and Outgoing Interface Managers
	◆ `LMTPPOOL0` (LMTP mail server connection pool) ◆ `POP3POOL0` (POP3 mail server connection pool) ◆ `POP3POOLMGR` (POP3 connection pool to the POP3 mail server; requires `<id>`) ◆ `SMTPPOOL0`(SMTP mail server connection pool) ◆ `a2s/EBFPOOL` (A2S EBF connection pool) ◆ `fe/EBFPOOL` (Mail Front EndEBF connection pool) ◆ `fe_auth/EBFPOOL` (Mail Front EndAuthorization EBF connection pool) ◆ `fe_jobq/EBFPOOL` (Mail Front JOBQ EBF connection pool) ◆ `MTACONNPOOL` (Internet mail MTA connection pool) ◆ `DCMMTACONNPOOL` (DoCoMo mail MTA connection pool)
`all`	Use `test component all` to test all the above components at once.

In response to the `test component` command, the CLI will return for each requested component one of two possible test results:

- `OK`
- `ERR <message>`

Possible `ERR` messages depend on the component type. The most common potential error responses are:

- Listeners
  - `ERR Could not establish testing connection.`
  - `ERR Could not self test because self-hostname is not on allowedhosts list.`
- Connection pools
  - `ERR Could not open new connection.`
  - `ERR Keepalive test failed.`

Example The sample below shows a successful test of the M2G's CLI listener.

```
M2G>test component CLI
OK
```

Example

Binding or Unbinding Outgoing Connections (bind)

Use the `bind|unbind|rebind` command to bind, unbind, or rebind the M2G's outgoing connection pool connections. You can use this command with each of the M2G connection pools. The command syntax is as follows :

```
command <COMPONENT> bind|unbind|rebind
```

Parameter	Description
`<COMPONENT>`	Connection Pools and Outgoing Interface Managers
	◆ `LMTPPOOL0` (LMTP mail server connection pool) ◆ `POP3POOL0` (POP3 mail server connection pool) ◆ `POP3POOLMGR` (POP3 connection pool to the POP3 mail server; requires `<id>`) ◆ `SMTPPOOL0`(SMTP mail server connection pool) ◆ `a2s/EBFPOOL` (A2S EBF connection pool) ◆ `fe/EBFPOOL` (Mail Front EndEBF connection pool) ◆ `fe_auth/EBFPOOL` (Mail Front EndAuthorization EBF connection pool) ◆ `fe_jobq/EBFPOOL` (Mail Front JOBQ EBF connection pool) ◆ `MTACONNPOOL` (Internet mail MTA connection pool) ◆ `DCMMTACONNPOOL` (DoCoMo mail MTA connection pool)
`all`	Use `test component all` to test all the above components at once.

Option/Parameter	Description
`bind`	Enables connections to be made between the M2G and the target peer server. Upon M2G start-up, "bind" is by default enabled for each connection pool and thus the M2G can create connections as needed. However, if you use the `unbind` command for a connection pool, you will subsequently need to use the `bind` command in order to re-enable the creation of new connections. (Alternatively, you can restart the M2G and binding will again be enabled by default.)
`unbind`	Disables the creation of new connections to the target peer server and prevents the returning of existing connections to the connection pool after their use. In-use connections will be closed after they complete their in-progress transactions. Idle connections will be closed at the next keep-alive check.

Option/Parameter	Description
`rebind`	Disables the creation of new connections, closes existing connections after they complete in-progress transactions, and closes idle connections at the next keep-alive check; and then re-enables the creation of connections. Pending requests for connection are held over during this interval, not failed. The delay between the unbind and bind actions is set by the `rebinddelay` parameter for each connection pool.

Example The sample below shows a successful rebinding of the main connection pool to the POP3 server.

```
M2G>command POP3POOL0 rebind
OK
```

Viewing Configuration Properties (show config)

Use the `show config` command to view the configuration property settings that the M2G currently has loaded into its memory. You can view settings either individually or as a complete group.

The command syntax is as follows:

```
show config <property>|all
```

'show config' Parameters

Parameter	Description
`<property>`	A single property from any M2G properties file.
`all`	Use `show config all` to display all M2G configuration property settings at once. The following should be noted about the response to the `show config all` command: ◆ The `show config all` command displays not only customer-configurable property settings, but also internal property settings. Most of these internal settings are not meant for customer use and are not documented in this manual. The exception are properties that are documented in this manual as "setting not in file but may be added". To see whether an unfamiliar property in the `show config all` response is documented, refer to the *Index of Settings in .properties and .conf Files, on page 415*. ◆ The response to `show config all` is not divided by properties file. Rather, the response displays all property settings currently loaded into M2G memory, without reference to any `*.properties` files.

CLI responses to the `show config` command are in the form `<property>=<value>`. For the `show config all` command, the return will be in multiple lines with one configuration value per line. A successful response concludes with `'OK'`.

Example The sample below shows a successful retrieval of the application log format setting.

```
M2G>show config ALOG.format
ALOG.format=<PID>|<THREADID>|<DATE-CLF>|<MODULE:%-
24s>|<LEVEL>|<MESSAGECODE>|<MESSAGE>|<GTRID>
OK
```

Note The `show config` command does not retrieve map file data. To retrieve map file data that the server currently has in its memory, use the `dump` command as described on *page 47*.

Changing Configuration Properties (set)

Use the `set` command to dynamically modify M2G property settings without having to restart the server. The command syntax is as follows:

```
set <property>=<value>
```

Not all M2G properties can be dynamically reset with this command. Each property description in that chapter indicates whether or not the property can be dynamically reset through the CLI `set` command.

IMPORTANT When you change settings through the `set` command, your changes will persist only through the current running session of the M2G. Your dynamic setting changes are not registered in the properties file, and on the next restart the M2G will reload the properties file into memory and use whatever settings are in the properties file. (Note: If you use the `set` command to change the value of a property that is not listed in a properties file—a property marked in this manual as "setting not in file but can be added"—then on the next restart the property will revert to its internal default value.) If you want to make configuration changes that will persist across restarts, manually edit the properties file and then restart the server.

In the event that an error is returned in response to your `set` command, the configuration change is not applied, and the previously existing setting continues to be used.

'set' parameters

Parameter	Description
`<property>`	An M2G configuration property that supports dynamic resetting using the CLI.
`<value>`	The value that you wish to assign to the property. NOTE: If you attempt to assign a property a value that is outside the valid range for that property, the M2G rejects the value and returns an error response. See *Chapter 4, M2G Properties File* for valid ranges for each M2G configuration property.

Example The sample below successfully sets the application logging level to DEBUG.

```
M2G>set ALOG.loglevel=DEBUG
OK set ALOG.loglevel=DEBUG
```

Note The `set` command is not applicable to values in map files. For dynamic reloading of map files, see *Reloading Map Files (reload), on page 48*.

Viewing Map File Contents (dump)

Use the `dump` command to output map file data that the M2G currently has in its memory. For most map files, you can use this command to see how the M2G has loaded map file data into memory. The command syntax is as follows:

```
   reload <COMPONENT> dump <dumptofilename>
```

Map File Arguments

Valid Component Argument	Reloadable Map File
`SIDLIST.mapfile`	`sidlist.cfg`
`DOMAIN2HANDLERMAP.mapfile`	`map_domain_to_charhandler.cfg`
`DOMAIN2OPERATOR.mapfile`	`map_domain_to_operator.cfg`
`ERRORTEXTMAP.mapfile`	`map_error_text.cfg`

The `<dumptofilename>` argument is the name of the file into which you want the data written, including path, if appropriate.

Example The example below shows a successful request that the internal map data managed by the `ERRORTEXTMAP.mapfile` for the `map_error_text.cfg` map file be written to the file `output_of_dump.txt`.

```
M2G> command ERRORTEXTMAP dump output_of_dump.txt
OK
M2G> quit

EXACT|no_folder_exists|"Mail box does not exist"
EXACT|no_key_exists|"System error"
EXACT|quota_limit|"Message quota limit exceeded"
EXACT|retrylater|"System busy"
EXACT|system_down|"Mail system down"
EXACT|system_limit|"Mail system limit reached"
EXACT|timeout|"Time out error occurred"
EXACT|unauthorized|"Recipient unauthorized on system"
# END
```

Reloading Map Files (reload)

Use the `reload` command to dynamically reload M2G map configuration files. This action allows you to implement changes that you have made to a map file without having to restart the M2G. The command syntax is as follows:

```
reload <COMPONENT>
```

Note In contrast to using the `set` command for properties changes, when you edit a map file and then reload it using the `reload` command, your changes will persist even if the M2G is later shut down and restarted (since in this case you have actually modified the file, which will then be automatically reloaded into memory on each M2G restart).

In the event that an error is returned in response to your `reload` command, the configuration change is not applied, and the previous map file settings—stored in the server's memory—continue to be used

Map File Arguments

Valid Component Argument	Reloadable Map File
ALOG.codemapfile	errorcode.cfg
ALOG.eventmapfile	eventname.cfg
SIDLIST.mapfile	sidlist.cfg
DOMAIN2HANDLERMAP.mapfile	map_domain_to_charhandler.cfg
DOMAIN2OPERATOR.mapfile	map_domain_to_operator.cfg
IMGXMMAP.mapfile	magetransformmap.cfg
ERRORTEXTMAP.mapfile	map_error_text.cfg
TEXTXMMAP.mapfile	mail_transcoding.cfg

Example The sample below successfully reloads the map file `map_uri_allow.cfg`.

```
M2G>reload SIDLIST.mapfile
OK reload SIDLIST.mapfile
```

Viewing Real-Time Statistics (show stat)

Use the `show stat` command to view real-time statistics tracked by M2G interface components. The command syntax is as follows:

```
show stat <statistic>|all
```

'show stat' Parameters (Part 1 of 3)

Parameter	Description	
`<statistic>`	A single real-time statistic associated with a particular M2G listener component or connection pool component.	
	Listener Statistics ◆ Listener statistics are supported for each of the listeners listed below. ◆ `CLI` (CLI Listener) ◆ `CLIX` (CLIX Listener) ◆ `XCONV` (Transcoding Conversion EBF Listener) ◆ `mta/SMTPSVR` (Internet SMTP ILstener) ◆ `dcm/SMTPSVR` (DoCoMo SMTP Listener) ◆ `pcc/SMTPSVR` (PC CLIENT (Relay) SMTP Listener) ◆ `pcc/SMTPSSLSVR` (PC CLIENT (Relay) SMTP SSL Listener) ◆ `pcc/POPSVR` (PC CLIENT POP3 Listener) `pcc/POPSSLSVR` (PC CLIENT POP3 SSL Listener)	
	`<LISTENER>.conn.open`	Number of incoming connections currently open to the specified listener.
	`<LISTENER>.conn.handled`	Number of connections accepted by the specified listener since the statistics were last reset. This cumulative statistic is reset by either the `reset` command (*page 52*) or a restart of the M2G server or its host machine.

'show stat' Parameters (Part 2 of 3)

Parameter	Description	
`<statistic>` (continued)	`<LISTENER>.conn.refused`	Number of connection requests rejected by the specified listener since the statistics were last reset. This cumulative statistic is reset by either the CLI `reset` command (*page 52*) or a restart of the server or its host. This statistic is available only if the listener has an `allowedhosts` setting specified in a properties file.
	Connection Pool Statistics Connection pool statistics are supported for each of the connection pools listed below. For pool descriptions, see the referenced pages. ◆ `LMTPPOOL0` (LMTP mail server connection pool) ◆ `POP3POOL0` (POP3 mail server connection pool) ◆ `POP3POOLMGR` (POP3 connection pool to the POP3 mail server; requires `<id>`) ◆ `SMTPPOOL0`(SMTP mail server connection pool) ◆ `a2s/EBFPOOL` (A2S EBF connection pool) ◆ `fe/EBFPOOL` (Mail Front EndEBF connection pool) ◆ `fe_auth/EBFPOOL` (Mail Front EndAuthorization EBF connection pool) ◆ `fe_jobq/EBFPOOL` (Mail Front JOBQ EBF connection pool) ◆ `MTACONNPOOL` (Internet mail MTA connection pool) ◆ `DCMMTACONNPOOL` (DoCoMo mail MTA connection pool)	
	`<POOL>.conn.open`	Number of outgoing connections currently open from the identified connection pool. This includes spare connections as well as in-use connections.
	`<POOL>.conn.inuse`	Number of in-use connections currently in the specified connection pool. In-use connections are those that are currently servicing requests.

'show stat' Parameters (Part 3 of 3)

Parameter	Description	
`<statistic>` (continued)	`<POOL>.conn.spare`	Number of spare connections currently in the specified connection pool. Spare connections are those that are open but not currently in use. A connection pool's `conn.spare` value will typically equal its `conn.open` value minus its `conn.inuse` value—but not always, as some connections may be in a transition state at the moment of statistics retrieval.
	`<POOL>.conn.waiting`	Number of queued requests currently waiting for a free connection from the specified pool. Applicable only when all connections are currently in use, and only if you have the configuration properties `maxwaitqueue` and `maxwaittime` both set to values greater than zero for this connection pool.
`all`	Use `show stat all` to retrieve all available M2G real-time statistics.	

Example

The `show stat` sample below successfully retrieves the open connection statistic for the M2G-E's push notification request listener.

```
M2G>show stat CLI.conn.open
CLI.conn.conn.open=1
OK
```

Initializing Real-Time Statistics (reset stat)

Use the `reset stat` command to initialize cumulative real-time statistics (those that add up across time). Initializing a statistic resets the statistic's counter to zero, whereupon a new count for the statistic begins.

The command syntax is as follows:

```
reset stat <statistic>|all
```

'reset stat' Parameters

Parameter	Description
`<statistic>`	A cumulative real-time statistic that can be reset to zero. The following statistics can be reset: ◆ `<LISTENER>.conn.handled` ◆ `<LISTENER>.conn.refused` For information on these statistics, see *page 49*.
`all`	Use `reset stat all` to reset all cumulative real-time statistics.

Note These statistics are also reset whenever you restart the M2G or reboot the M2G's host machine.

Example The sample below successfully resets the CLI connections handled statistic.

```
M2G>reset stat CLI.conn.handled
OK CLI.conn.handled reset
```

Example The sample below successfully resets all cumulative real-time statistics.

```
M2G>reset stat all
OK all statistics reset
```

Setting a Replacement for Invalid Characters

Use the CHARSETCONV commands to see what exchange character the M2G is substituting for invalid characters that it encounters while performing text conversions, or to change the exchange character. The command syntax is as follows:

- `command CHARSETCONV show_replacementchar`
- `command CHARSETCONV set_replacementchar <char>`

Note The invalid character replacement can also be configured in m2g.properties as described on *page 53*.

Option/Parameter	Description
show_replacementchar	This option displays the current exchange character.
set_replacementchar	This option dynamically changes the exchange character, without having to restart the M2G. IMPORTANT: Your changes will persist only for the current running session. The M2G will revert to the invalid character replacement in hmg.properties upon the next restart.
<char>	When using the set_replacementchar option, the new exchange character. You can use a single character or a string. Characters must be from the UTF-8 encoding set.

Example The sample below shows a viewing of the current invalid character replacement.

```
M2G> command CHARSETCONV show_replacementchar
OK: [?] (in UTF-8)
```

Shutting Down the Server (shutdown)

Use the `shutdown` command to shut down the M2G. The command syntax is as follows:

```
shutdown
```

There are no arguments to the `shutdown` command.

Example

The sample below shows a successful shut-down of the M2G through the CLI.

```
M2G>shutdown
OK attempting shutdown using SIGTERM.
```

For confirmation that the shutdown succeeded, you can check the application log for INFO level messages indicating a "caught SIGTERM" and a "Handle Event SHUTDOWN" for each child `M2G` process, and a "Handle Event SHUTDOWN" for the parent `M2G` process. The number of child processes shut down should correspond to your `hre.numprocesses` setting from the `m2g.properties` file.

Pinging the CLI Listener (noop)

Use the `noop` command to 'ping' the CLI listener and confirm that it is available and receiving your commands. The command syntax is as follows:

```
noop
```

There are no arguments to the `noop` command.

Example This sample command pings the CLI server . The response indicates that the CLI server is receiving and responding to commands.

```
M2G>noop
OK
```

Viewing the CLI Command List (help)

Use the `help` option to view a list of supported commands. The command syntax is as follows:

```
help
```

There are no arguments to the `help` command.

Example This sample shows the list of supported commands.

```
M2G>help
Available commands:
    help
    quit
    command <component> <command> [<arg>]
    reload <component>.<paramname>
    reset stat [all|<stat>]
    set <param>=<value>
    show config [all|<param>]
    show stat [all|<stat>]
    show status [all|<component>]
    test component [all|<component>]
     shutdown
```

Terminating the CLI Session (quit)

Use the `quit` command to terminate your session with the command line interface. The command syntax is as follows:

```
quit
```

There are no arguments to the `quit` command.

Example This sample command terminates the CLI session.

```
M2G>quit
OK Bye Bye
```

M2G Configuration Directories and Files

By default, the M2G main configuration directory is:

```
<M2G_HOME>/etc
```

where `<M2G_HOME>` is the server's home directory as established during product installation. If during installation you accept the default location for `<M2G_HOME>`, then the M2G main configuration directory will be:

```
/usr/local/gemini/m2g/1.0.0/etc
```

Several configuration files reside directly under the main configuration directory. Other configuration files are stored in sub-directories under the main configuration directory.

Working with Properties Files

M2G components such as listeners, connection pools, and log managers are configured in a series of "properties" files, identified by the extension `.properties`. Though individual properties files differ in the components that they configure, all properties files have the same formatting and syntax..

Component-Based Configuration Scheme

Properties files are organized around configurable M2G components, with each component having a unique name and its own set of configuration properties. Common component types include:

Listeners Listeners listen for requests submitted to the M2G over a particular protocol, through a particular port.

Connection Pools Connection pools enable the M2G to efficiently submit requests to peer servers with which it frequently interacts by maintaining a set of persistent connections that remain open and ready to use.

Log Managers Log managers implement different types of M2G logging. Sample component names are `ALOG` (application logging) and `TLOG` (transaction logging).

Each properties file configures a set of components. Within the file, each component's configuration properties are grouped together as a series of lines in the form:

```
<COMPONENT>.<property> = <value>
```

Example For the M2G application log manager, named `ALOG`, the following block of parameters appears in the properties file `m2g.properties`:

```
ALOG.loglevel=INFO
ALOG.format=<DATE:%Y/%m/%d %H:%M:%S:000> <PID> <MODULE:%-24s>
<LEVEL> <MESSAGECODE> <MESSAGE>
ALOG.avoidrepeats=false
ALOG.maxheaderwidth=2000
ALOG.truncated_ind="..."
```

Default Values and Valid Range Enforcement

For each setting in each M2G configuration properties file, this manual indicates two types of "default" values:

Internal Default The "internal default" for a configuration setting is the default value that is written into the code of the M2G component to which the setting applies. Most settings have internal default values. Internal default values serve two purposes:

- If a setting is not explicitly assigned a value in a properties file, the internal default value for the setting is used.
- If you assign a setting an out-of-range value in a properties file, the server ignores the out-of-range value and instead uses the setting's internal default value.

If a setting is assigned a valid value in a properties file, *the value in the properties file overrides the internal default*.

File Default The "file default" for a configuration setting is the value assigned to the setting in the original version of the properties file included in your M2G release package.

A setting's file default will often match its internal default. However, file defaults sometimes differ from internal defaults, for one of these reasons:

- M2G components of a similar type—such as HTTP-based listeners—may share a common code base. The internal default configuration settings in the common code base may not be optimal for each variant of the component. In the properties files, Gemini Mobile Technologies sets file defaults appropriate to each component instance.
- Gemini Mobile Technologies' judgment as to the most appropriate default value for a particular setting may evolve over time. Gemini's periodic updates to file defaults may result in divergence from internal defaults.

In addition to documenting internal defaults, this manual documents the file default for each setting so that if you customize or experiment , you can always refer back to this manual to see what each setting's value was in the original properties files.

For most component properties, your settings must be within a valid range. If you assign an out-of-range value to a parameter in a properties file, the M2G ignores your setting in the properties file and instead uses the parameter's internal default value. When this occurs, the M2G writes a warning message to the application log.

There are two exceptions to this manner of enforcing valid ranges:

- For some parameters, the upper end of the valid range is INT_MAX, which is equal to 2147483647, a 32 bit signed integer. If you assign such a parameter a value greater than INT_MAX, the server will not start up.

- For a small number of parameters, the valid range is a set of M2G components. For example, interface congestion control settings allow you to specify one or more M2G listeners to shut down if the interface becomes too congested. For such settings, if you include an invalid component name in your setting, the server will fail to start because it will attempt unsuccessfully to instantiate the invalid component.

For each M2G configuration property, this manual indicates a valid range specified as a data type along with lower and upper limits (if applicable). The data types are:

INT Integer

BOOL Boolean

TIME Time value (see *page 62*)

TOKEN String with no whitespace

VECTOR Set of string values separated by white space or commas

TEXT Unrestricted free text

LOGLEVEL Set of log filtering levels—applicable only to logging configuration

ATOM One of a pre-defined set of option names, such as “on” or “off”

IMPORTANT The maximum allowed length of a string setting (TOKEN, VECTOR, or TEXT) is 1022 characters. If you assign a configuration property a value longer than 1022 characters the M2G will exit on start-up.

The table that follows—an excerpt of setting descriptions for the M2G command line interface listener—illustrates how valid ranges, file defaults, and internal defaults are documented in this manual. The significance of the “CLI Set” column is described on *page 67*.

Sample Property Documentation

Property Description	Valid Range	File Default	Internal Default	CLI Set
`CLI.timeout`				
If this much time passes without any communication from a connected client, the CLI listener closes the connection.	TIME (0s to 1h)	600s	600s	yes

Settings That You Can Add to Properties Files

For some components, the property description tables in this manual include a small number of parameters that do not appear in the properties files that you receive with your M2G server package. You can add these parameters to the component's configuration in the properties file if you wish to assign them values different than their internal defaults. If you do not explicitly add such parameters to the properties file, the server uses the internal default value.

In the property description tables, the "File Default" column for such parameters will indicate "Setting not in file but can be added."

Example For the application log manager component `ALOG`, the property description table on *page 84* lists an `ALOG.usegmt` property that does not appear in the `m2g.properties` file that Gemini Mobile Technologies provides you. This property has an internal default value of "false", as indicated in the property description table. If you want to assign this property a value of "true" (so that application log timestamps use Greenwich Mean Time), you can add the property as a new line at the end of the `ALOG` block, as in this example:

```
ALOG.loglevel=INFO
ALOG.format=<DATE:%Y/%m/%d %H:%M:%S:000> <PID> <MODULE:%-24s>
<LEVEL> <MESSAGECODE> <MESSAGE>
ALOG.avoidrepeats=false
ALOG.maxheaderwidth=2000
ALOG.truncated_ind="..."
ALOG.usegmt=true
```

Specifying Time Interval Values

Some M2G configuration parameters are set as a period of time—for example, timeout values for listeners or connection pools. These parameters are of data type TIME. All TIME parameters are set in one of these formats, in which `<n>` is an integer:

- `<n>s`, where `s` indicates seconds
- `<n>m`, where `m` indicates minutes
- `<n>h`, where `h` indicates hours
- `<n>d`, where `d` indicates days
- `<n>w`, where `w` indicates weeks

Example

Note If you simply enter a number for a time value, the server reads this as a number of seconds.

General Formatting Notes

When working with M2G properties files, note these formatting items:

- Near the top of most properties files is an "include" statement of this form:

  ```
  include = <filename>
  ```

  For example:

  ```
  include = ./.m2g.properties
  ```

  Do not modify these "include" settings.

- Lines that begin with a pound sign (#) are comment lines. Most typically comment lines are used to label sections of a properties file. For example:

  ```
  #
  # Client Connection Pool
  #
  ```

- A pair of forward slashes (//) within a line indicates that the text to the right of the forward slashes is comment text. For example:

  ```
  DEFAULT.in_timeout_max_network=185s // network i/o only
  ```

- It does not matter whether or not spaces surround the equal sign that links a property name to its assigned value. For example, these two lines would be equivalent:

  ```
  CLI.timeout=600s
  ```

  ```
  CLI.timeout = 600s
  ```

- For properties to which the assigned value is a string, it does not matter whether or not the string is enclosed in quotation marks. For example, these two lines would be equivalent:

  ```
  DEFAULT.rcptnomailboxerr="X-DCMUID"
  ```

  ```
  DEFAULT.rcptnomailboxerr=X-DCMUID
  ```

Activating Properties File Changes

When you make edits to a properties file, you must restart the M2G for your changes to take effect.

Note Certain settings that appear in properties files can also be dynamically modified through the M2G command line interface, without restarting the server. Such changes will last only for the current running session of the server.

Working with Map Files

Map files, typically named in form `map_*.cfg`, allow you to configure various types of lists against which the M2G will check for a match.

Map files are formatted as series of lines each in this form:

```
<match_type>|<match_key>|[<value>]
```

The `<match_type>` indicates the type of matching to execute. The options are:

- `EXACT`

  This option instructs the server to look for exact matches against your `<match_key>` value. If in a particular line in a map file you do not specify a `<match_type>`, then `EXACT` matching is used by default.

- `PREFIX`

  This option instructs the server to look for matches in which the matched item starts with your `<match_key>` value.

- `REGEX`

  This option instructs the server to look for matches against your `<match_key>` regular expression. POSIX 1003.2 Extended Regular Expressions are supported.

- `DEFAULT`

  This option dictates handling of cases that do not match against any other of your entries in the map file. The `DEFAULT` option is appropriate only for map files that use a `<value>` parameter.

The `<value>` is an optional parameter that is used if the server should handle matches differently depending on which entry line they match. Most map files do not use a `<value>`.

Note When checking an object against a map file, the system looks first for an `EXACT` match. If no `EXACT` match is found, the system next checks for a `PREFIX` match. If no `PREFIX` match is found, the system checks for a `REGEX` match. Since there is a slight performance cost to using the `PREFIX` or `REGEX` matching types, you should use the `EXACT` match type when possible.

Note Separation between lines in map files must be <LF> only, not <CR><LF>.

Example

Map File Entry Limits

Each map file has a corresponding map file manager component that loads data from the map file into memory for run-time use. A map file manager also configures a maximum number of entry lines for its map file, as well as the delimiter to be used between the `<match_type>`, the `<match_key>`, and the `<value>` in each line. In the detailed descriptions for each map file in this manual, the field delimiter and the maximum allowed entries are indicated.

Configuring the M2G Through the CLI

You can implement two types of configuration changes through the M2G command line interface:

- For certain properties file parameters, you can dynamically change settings through the CLI `set` command—without editing the file or restarting the server. In the configuration property description tables in this manual, a "CLI Set" column indicates "yes" or "no" for each parameter. For parameters marked as "yes", you can change the setting dynamically using this CLI command:

  ```
  set <property>=<value>
  ```

  For particular settings, you would specify the property name and desired value, as in this example:

  ```
  set STATSMGR.loginterval=60m
  ```

IMPORTANT When you change settings through the `set` command, your changes will persist only through the current running session of the M2G. Your dynamic setting changes are not registered in the properties file, and on the next restart the server will reload the properties file into memory and use whatever settings are in the properties file. (Note: If you use the `set` command to change the value of a property that is not listed in a properties file—a property marked in this manual as "setting not in file but can be added"—then on the next restart the property will revert to its internal default value.) If you want to make configuration changes that will persist across restarts, manually edit the properties file and then restart the server.

- For map files, you can dynamically reload an edited file through the CLI `reload` command, rather than restarting the M2G. In this manual, descriptions of map files include the specific `reload` argument for each file.

  When you reload a modified map file through the `reload` command, your changes will persist even if the server is later shut down and restarted (since in this case you have actually modified the file, which will then be automatically reloaded into memory on each server restart).

IMPORTANT When using either the `set` command or the `reload` command, you are making configuration changes only to the single node to which you are transmitting the command. To make changes to multiple nodes, you must work with each node individually.

You can also use the M2G command line interface to view the configuration property settings and map file data that the server currently has loaded in its memory. For viewing loaded configuration properties, use the `show config` command. For viewing loaded map file data, use the `command <COMPONENT> dump` command.

3 Dot M2G Properties File

This chapter provides information about the settings in the `.m2g.properties` configuration file.

Note This the “dot” M2G properties file. Note the dot in front of the file name.

Note If you want to quickly locate the description of a particular setting that you have seen in the M2G properties file, you can use *Index of Settings in .properties and .conf Files* starting on *page 631*.

.m2g.properties

Path `<M2G_HOME>/etc/.m2g.properties`

Purpose Configures basic (mostly internal) administrative settings for the M2G.

Type Properties file. For background see *Working with Properties Files, on page 59*.

Dynamic Reload You cannot dynamically reload this file. To activate changes that you make to the file, you must restart the M2G. For properties for which the "CLI Set" column is marked "yes" in the table that follows, you have the option of dynamically changing the setting using the CLI `set` command (*page 45*) rather than editing the properties file.

Most parameters in `.m2g.properties` are not configurable by the user. However, the `SSLCONFIG` component in this file is configurable and sets parameters for SSL protection for M2G listeners and clients. See

The table below describes each configuration property.

.m2g.properties Settings

`SSLCONFIG.clientcertverification`				
For the M2G's SSL-enabled clients, whether to validate certificates provided by target servers against the CA certificate stores. Options are: ◆ **`true`** Validate certificates provided by target servers against the CA certificate stores. ◆ **`false`** Accept target server cerficates without validation.	BOOL	true	false	yes
`SSLCONFIG.allownotyetvalidcert`				
When requiring a valid certificate from a client or a server, whether or not to to accept a certificate that is not yet in its valid date range. Options are: ◆ `true` Accept a certificate that is not yet in its valid date range. ◆ `false` Do not accept a certificate that is not yet in its valid date range. Instead, return an SSL handshake error and terminate the connection.	BOOL	true	true	yes

`SSLCONFIG.allowexpiredcert`				
When requiring a valid certificate from a client or a server, whether or not to to accept a certificate that has expired. Options are: ◆ `true` Accept a certificate that has expired. ◆ `false` Do not accept a certificate that has expired. Instead, return an SSL handshake error and terminate the connection.	BOOL	true	true	yes
`SSLCONFIG.supportsslv2`				
Whether to accept SSLv2 connections from clients. This version of the protocol is known to have security vulnerabilities. Options are: ◆ `true` Accept SSLv2 connections from clients. ◆ `false` Do not accept SSLv2 connections from clients. Instead, return an SSL handshake error and terminate the connection.	BOOL	false	false	yes
`SSLCONFIG.sslverifydepth`				
When requiring a valid certificate from a client or a server, the maximum allowable depth of the certifying authority chain. If a certificate has a deeper chain of intermediary authorities than this, the M2G returns an SSL handshake error and terminates the connection. Set to -1 for no maximum.	-1 to INT_MAX	-1	-1	yes
`SSLCONFIG.ciphers`				
List of OpenSSL supported ciphers to allow for SSL sessions. For formatting of this string, see: `http://www.openssl.org/docs/apps/ciphers.html` Specify “ALL” to allow all OpenSSL supported ciphers.	see description	ALL	all	no

SSLCONFIG.privatekeyfiletype				
Format of the private key file. Options are: ◆ PEM Privacy Enhanced Mail format ◆ asn1 Abstract Syntax Notation 1 format	PEM, asn1	PEM	pem	no
SSLCONFIG.certfiletype				
Format of the certificate file. Options are: ◆ PEM Privacy Enhanced Mail format ◆ asn1 Abstract Syntax Notation 1 format	PEM, asn1	PEM	pem	no
SSLCONFIG.certfile				
Path to the certificate file used by the M2G's SSL-protected servers and clients. If the file holds more than one certificate, the M2G uses the first certificate listed. Default = ssl/ssl_certificate.pem	none enforced	none	none	no
SSLCONFIG.privatekeyfile				
Path to the private key file used by the M2G's SSL-protected servers. The private key must correspond to the certificate pointed to by the certfile parameter. The private key and certificate may be stored in the same file. Default = ssl/server.key	none enforced	none	none	no

Example The parameter `SSLCONFIG.clientcertverification` below, set as `true`, allows certificates provided by target servers to be validated against the CA certificate store.

```
SSLCONFIG.clientcertverification=true
```

4 M2G Properties File

This chapter provides information about the settings in the M2G properties configuration file.

Note If you want to quickly locate the description of a particular setting that you have seen in the M2G properties file, you can use *Index of Settings in .properties and .conf Files* starting on *page 631*.

m2g.properties

Path `<M2H_HOME>/etc/m2g.properties`

Purpose Configures basic administrative settings for the M2G.

Type Properties file. For background see *Working with Properties Files, on page 59*.

Dynamic Reload You cannot dynamically reload this file. To activate changes that you make to the file, you must restart the M2H. For properties for which the "CLI Set" column is marked "yes" in the table that follows, you have the option of dynamically changing the setting using the CLI `set` command (*page 45*) rather than editing the properties file.

The table that starts on the next page describes each configuration property in the `m2g.properties` file. The properties are listed in the same order in which they appear in the file. The table also describes additional properties that you can add to `m2g.properties` if you want to assign them values different than the internal defaults.

Note The properties set in the DEFAULT block (for instance, `DEFAULT.timeout)` are used only when there are no definitions in specific components. Properties set in the component overrides default setting.

m2g.properties Settings (Part 1 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
`hre.numprocesses`				
The number of `M2G` child processes that the `M2G` parent process launches when you start the M2G server.	INT (1 to 16)	4	2	no
`hre.listeners`				
The HyperScale runtime environment listeners for the M2G. Do not change. Default = `CLI,CLIX,XCONV,mta/SMTPSVR,dcm/SMTPSVR,pcc/SMTPSVR,pcc/SMTPSSLSVR,pcc/POPSVR,pcc/POPSSLSVR`	See description	See description	See description	no

m2g.properties Settings (Part 2 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
`hre.clibaseport`				
Set the process to the CLI base port. Each HRE process may have a CLI listener dedicate to that process. These process CLI listeners are loaded at startup time. The port starts at the base port and increments by 1 for each HRE process started. It can be an integer or text. If text is used, it will be converted to an integer value by looking up in `services` file (*page 199*) File default = `server-i%n-%p+cli`	see description	see description	0	no
`hre.processclitimeout`				
Sets timeout value (in seconds) when connecting to process's CLI port by monitoring process. If the watcher process cannot obtain a response from the CLI process for the specified amount of time, the watcher may decided that this HRE process is non-responsive.	TIME	180	0	no
`hre.processcliinterval`				
Sets interval time in seconds when connecting to process's CLI port by monitoring process. Two successive pings to CLI port are separated this time interval.	TIME	60	0	no
`hre.climsgqueuepath`				
Sets message queue path name to enable non-responsive process monitoring.	TEXT	“”	“”	no
`hre.climsgqueuetype`				
Set message queue message type to communicate with non-responsive process monitor.	INT	72	72	no

m2g.properties Settings (Part 3 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
`DEFAULT.shutdowntimeout`				
When the server is responding to a shutdown or restart command, the interval in seconds for its listeners and connection pools to continue processing in-progress transactions, before shutting down. NOTE: Individual listeners also support the `shutdowntimeout` parameter, allowing you to set listener-specific overrides of the system-wide default value that you establish with `DEFAULT.shutdowntimeout`. For further information, see the configuration setting descriptions for individual server interfaces.	TIME (0 to 20s)	10s	10s	no
`DEFAULT.timeout`				
After sending a command to the target server over a particular connection from the pool, the maximum time to wait for a response from the target server. If the response is not received within the timeout period, the processing server terminates the connection. To disable timeout limit, set to 0s.	TIME 0s to 1h	70	0	yes
`DEFAULT.in_timeout_max_idle`				
For persistent HTTP connections to the SMTP listener, the maximum allowed idle time between request/response transactions. Within a persistent HTTP connection, this timer is restarted each time a request/response transaction is completed. If no new request is received from the client within the idle timeout interval, the listener closes the connection with the client. To disable this timeout, set this parameter to 0.	TIME 0 to 1h	0	0	yes

m2g.properties Settings (Part 4 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
`DEFAULT.disconnectforinvalidrecipient`				
The current receipt address checking rejects invalid addresses by assuming all address length are greater than three characters, the simplest term is "a@b". If the RCPT gets a "REJECT" status, the connection is closed, otherwise if it gets a "REJECTCOMMAND" or "ACCEPT" status, the connection will be kept open. If this configuration variable for the SMTP listener is set to "false", the connection will be kept open even for a shorter address such as "1" or "2@".	BOOL	false	true	no
`DEFAULT.rcptnomailboxerr`				
The SMTP response error message for an unknown mailbox. Default="550 Requested action not taken: mailbox unavailable" Internal default = "553 Syntax error"	TEXT	see description	see description	no
`DEFAULT.enable_pb4smtp`				
This feature is enabled when the client needs to contact the server via POP before sending a SMTP command.	BOOL	true	false	yes
`DEFAULT.smtpconnectiontimeout`				
For SMTP-based listeners: After sending a 220 greeting response to a connecting client, the maximum time that the listener will wait for a HELO or EHLO command from the client. If the command is not received within the timeout period, the listener terminates the connection.	TIME	5m	5m	no
`DEFAULT.smtpcommandtimeout`				
After sending a response to a client command other than those listed in the next several rows of this table (for instance, a NOOP), the maximum time that theSMTP listener will wait for the next command from the client. If the next command is not received within the timeout period, the listener terminates the connection.	TIME (0s to 60s)	5m	5m	no

m2g.properties Settings (Part 5 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
`DEFAULT.smtprcpttimeout`				
After sending a response to a RCPT TO command from a client, the maximum time that the SMTP listener will wait for either another RCPT TO command or a DATA command. If the command is not received within the timeout period, the listener terminates the connection.	TIME (0s to 1d)	5m	5m	no
`DEFAULT.smtpdatatimeout`				
After sending a 354 response to a DATA command from a client, the maximum time that the SMTP listener will wait for complete message data from the client. If the complete data—including the “.” that indicates the end of data transmission—is not received within the timeout period, the listener terminates the connection.	TIME (0s to 1d)	2m	2m	no
`DEFAULT.smtpmailtimeout`				
After sending a response to a HELO or EHLO command from a client, the maximum time that the MM1 SMTP listener will wait for a MAIL FROM command from the client. If the command is not received within the timeout period, the listener terminates the connection.	TIME (0s to 1d)	5m	5m	no
`DEFAULT.smtpdataendtimeout`				
After sending a response to an end of mail data indicator from a client (“.”), the maximum time that the SMTP listener will wait for a QUIT, RSET, or MAIL FROM command from the client. In addition, this parameter sets the timeout value between responding to "BDAT mm-size", "BDAT mm-size LAST", and DATA commands until the next SMTP command is returned from the target server. If the command is not received within the timeout period, the listener terminates the connection.	TIME (0s to 1d)	5m	5m	no
`DEFAULT.connectiontimeout`				
After accepting an incoming connection from a client, the maximum time that the listener will wait for a command from the client before dropping the connection.	TIME (0s to 1d)	10s	5m	no

m2g.properties Settings (Part 6 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
`DEFAULT.commandhelotimeout`				
After sending a HELO command to the target server, the maximum time for the client to wait for a response code from the server. If the response is not received within the timeout period, the client terminates the connection.	TIME (0s to 1d)	5m	5m	no
`DEFAULT.commandmailtimeout`				
After sending a MAIL FROM command to the target server, the maximum time for the client to wait for a response code from the server. If the response is not received within the timeout period, the client terminates the connection.	TIME (0s to 1d)	5m	5m	no
`DEFAULT.commandrcpttimeout`				
After sending a RCPT TO command to the target server, the maximum time for the client to wait for a response code from the server. If the response is not received within the timeout period, the client terminates the connection.	TIME (0s to 1d)	5m	5m	no
`DEFAULT.commanddatatimeout`				
After sending a DATA command to the target server, the maximum time for the client to wait for a response code from the server. If the response is not received within the timeout period, the client terminates the connection.	TIME (0s to 1d)	2m	2m	no
`DEFAULT.commanddataendtimeout`				
`commanddataendtimeout` is used by SMTP client. When the M2G sends amessage to the MTA, it waits `commanddataendtimeout` amount of time after sending "CRLF.CRLF" termination to the remote server. If server does not respond thenthe M2G assumes a timeout has occurred and the message will be retried.	TIME (0s to 1d)	10m	10m	no

m2g.properties Settings (Part 7 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
`DEFAULT.commandquittimeout`				
After sending a QUIT command to the target server, the maximum time for the client to wait for a response code from the server. If the response is not received within the timeout period, the client terminates the connection.	TIME (0s to 1d)	5m	5m	no
`DEFAULT.commandtimeout`				
After sending a command other than those previously specified in this table to the target server, the maximum time for the client to wait for a response code from the server. If the response is not received within the timeout period, the client terminates the connection.	TIME (0s to 1d)	5m	5m	no
`CHARSETCONV.replacementchar`				
This replacement character is used when the M2G fails to convert a character from source encoding to target encoding. Since we have special character conversion components that handle EMOJI character conversions, this replacementchar is really not "global". It is global when the EMOJI conversion component is not used.	TEXT	“?”	“?”	yes

m2g.properties Settings (Part 8 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
`ALOG.loglevel`				
The lowest severity level of messages to include in the application log. Each message that the M2G server can generate has an assigned severity level appropriate to the message. You can use the `loglevel` setting to filter the server's application logging so that only messages of your specified level and higher will be logged. Options are, from highest to lowest level: ◆ `ALERT` Messages indicating a condition requiring immediate correction. ◆ `WARNG` Warning messages indicating a potential problem. ◆ `INFO` Informational messages indicating normal activity. ◆ `DEBUG` Low level detail messages potentially of use when debugging the application. Setting `loglevel` to `DEBUG` will result in a very large number of messages being logged. ◆ `OFF` You can set `loglevel` to `OFF` if you do not want anything written to the application log. An application log file will still be generated, but it will be empty. For example, with `loglevel` set to `INFO`, the server will log messages of all levels except `DEBUG`.	LOG-LEVEL (ALERT, WARNG, INFO, DEBUG, OFF)	INFO	INFO	yes

m2g.properties Settings (Part 9 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
`ALOG.format`				
Format of entries written to the application log. Specify the fields that you want to include in log entries, in the order that you want, using the field delimiter that you want. Field options are: ◆ `<PID>` ◆ `<THREADID>` ◆ `<DATE-CLF>` ◆ `<MODULE:%-24s>` ◆ `<LEVEL>` ◆ `<MESSAGECODE>` ◆ `<MESSAGE>` ◆ `<GTRID>` For descriptions of these fields, see *page 421*. You can also include text strings in your log entry format. To set a field-specific *maximum* length, precede the field name with "`<n>:`", with `<n>` being the maximum allowed number of characters (bytes) for values in that field. For example, a field format of `<100:MESSAGE>` limits messages to 100 characters. However, note that the total maximum length of the field will be 100 plus the length of your configured `ALOG.truncated_ind` (*page 84*). By default this truncation indicator is 3 characters long. For fields to which you do not assign a field-specific maximum length, the default maximum length `ALOG.maxheaderwidth` (*page 83*) is applied. For fields that you have assigned a field-specific maximum length, the *lesser* of the field-specific maximum and the default maximum is applied. You should use `ALOG.maxheaderwidth` to set a maximum that you want no field value to exceed, and then use field-specific maximums for fields that you want to limit to lengths shorter than `ALOG.maxheaderwidth`. To set a field-specific *minimum* length, append "`:%-<n>s`" to the field name. If the field value is shorter than `<n>` characters, blank spaces are appended to the value until the `<n>` space minimum is reached. For example, `<MODULE:%-24s>`.	TEXT (max 1022 chars)	see Note at end of table	null	no

m2g.properties Settings (Part 10 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
`ALOG.avoidrepeats`				
This parameter is not in the configuration file but can be added. Whether to avoid consecutive repetition of messages. Options are: ◆ `true` In cases where the server would log multiple consecutive messages with the same `<MODULE>` and `<MESSAGECODE>`, instead replace the duplicate messages with one entry indicating "`Message Repeated <N> Times`". NOTE: This feature applies only to messages of level INFO or above. It does not apply to DEBUG messages. ◆ `false` Allow consecutive repetition of messages.	BOOL (true, false)	false	false	yes
`ALOG.maxheaderwidth`				
This parameter is not in the configuration file but can be added. Default maximum number of characters to allow in a single log entry field's value. Field values longer than this are truncated down to your specified maximum width. Note that the total maximum length of the field will be `ALOG.maxheaderwidth` plus the length of your configured `ALOG.truncated_ind` (*page 84*). By default this truncation indicator is 3 characters long. For example, if you have `ALOG.maxheaderwidth` set to 2000 and you use the default `ALOG.truncated_ind`, then the maximum possible field length will be 2003 characters. This default maximum width for application log field values can be superseded on a per-field basis as described in the `ALOG.format` definition on *page 82*. If you assign a particular field a field-specific maximum width, then the maximum allowed width for that field is the *lesser* of your field-specific setting and your `ALOG.maxheaderwidth` setting. For all fields to which you do not assign a field-specific maximum, the `ALOG.maxheaderwidth` setting is applied. You should use `ALOG.maxheaderwidth` to set a maximum that you want no field value to exceed, and then use field-specific maximums for fields that you want to limit to lengths shorter than `ALOG.maxheaderwidth`.	INT (1 to INT_MAX)	2000	null	no

m2g.properties Settings (Part 11 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
`ALOG.truncated_ind`				
This parameter is not in the configuration file but can be added. When an application log entry field has been truncated down to your `ALOG.maxheaderwidth` or to your field-specific maximum width (see `ALOG.format` on *page 82*), the text indicator to append to the truncated field value to indicate that the field value has been truncated.	TEXT (max 1022 chars)	". . ."	null	no
`ALOG.usegmt`				
This parameter is not in the configuration file but can be added. Whether to use Greenwich Mean Time (GMT) for application log entry timestamps. Options are: ◆ `true` Use GMT. ◆ `false` Use local time.	BOOL (true, false)	setting not in file but can be added	false	no
`SLOG.loglevel`				
Whether to write statistics to a dedicated statistics log. Options are: ◆ `ON` Statistics are written to the statistics log (*page 428*). Your `sidlist.cfg` configuration file (*page 202*) determines which statistics will be logged. ◆ `OFF` Statistics are not written to the statistics log. NOTE: If you set `SLOG.loglevel` to OFF, a statistics log file will still be generated, but it will be empty.	LOG-LEVEL (ON, OFF)	ON	ON	yes

m2g.properties Settings (Part 12 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
`SLOG.format`				
Format of entries written to the statistics log, specifying the fields within each entry, the order of those fields, and the field separator. Field options are: ◆ `<DATETIME>` ◆ `<STAT_ID>` ◆ `<STAT_VALUE>` ◆ `<DOMAIN>` ◆ `<STATUS>` File default = `<DATETIME> <STAT_ID> <STAT_VALUE> <DOMAIN> <STATUS>`	TEXT (max 1022 chars)	see description	null	no
`SLOG.usegmt`				
This parameter is not in the configuration file but can be added. Whether to use Greenwich Mean Time (GMT) for statistics log entry timestamps. Options are: ◆ `true` Use GMT. ◆ `false` Use local time.	BOOL (true, false)	setting not in file but can be added	false	no
`TLOG.loglevel`				
Whether transaction logging is turned on or off. Options are: ◆ `ON` Transactions are recorded to the transaction log. ◆ `OFF` Transactions are not recorded to the transaction log. NOTE: If you set `TLOG.loglevel` to `OFF`, a transaction log file will still be generated, but it will be empty.	LOG-LEVEL (ON, OFF)	ON	ON	yes

m2g.properties Settings (Part 13 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
`TLOG.format`				
Format of entries written to the transaction log. Specify the fields that you want to include in log entries, in the order that you want, using the field delimiter that you want. Field options are: ◆ `<PID>` ◆ `<THREADID>` ◆ `<DATE-CLF>` ◆ `<PROTO>` ◆ `<TYPE>` ◆ `<STATUS>` ◆ `<TRID>` ◆ `<CLIENT>` ◆ `<HOST>` ◆ `<DURATION>` ◆ `<GTRID>` ◆ `{YAGUID}` ◆ `<FROM>` ◆ `<RCPTS>` ◆ `{From}{To}{Cc}{Bcc}` ◆ `{Message-ID}` ◆ `<MESSAGE>` ◆ `<MAIL_FILTER_RESULT>` ◆ `<HTTP_USERNAME>` ◆ `<HTTP_REQ_BODY_SIZE>` ◆ `<HTTP_REQ_HOST>` ◆ `<HTTP_REQ_LINE>` ◆ `<HTTP_REQ_PORT>` ◆ `<HTTP_REQ_SIZE>` ◆ `<HTTP_RES_BODY_SIZE>` ◆ `<HTTP_RES_CODE>` ◆ `<HTTP_RES_SIZE>` For description of fields, see *page 424*. You can also include text strings in your log entry format. You can set transaction log field length limits in the same manner as for the application log (see *page 82*).	TEXT (max 1022 chars)	see the Note that follows this table	null	no

m2g.properties Settings (Part 14 of 86)

<table>
<tr><th>Property
Description</th><th>Valid
Range</th><th>File
Default</th><th>Internal
Default</th><th>CLI
Set</th></tr>
<tr><td colspan="5"><code>TLOG.escapechars</code></td></tr>
<tr><td>List of characters to escape when writing log entries.
When writing log entries, if any of the characters specified in this setting occur within the text of a log field, the characters will be escaped by having a backslash (“\”) inserted in front of them. In specifying your list, do not use comma-separation or any other separation—just list the characters.
Example: <code>TLOG.escapechars = “|\”</code>
In this example, the two characters that will be escaped are the vertical bar (which unless escaped is the field delimiter) and the back-slash (which unless escaped is the escape indicator). So if for instance the invalid phone number “312|123\4353” needs to be written to the {To} field, it will be written as:
<code>312\|123\\4353</code></td><td>none enforced</td><td>"|\""</td><td>none</td><td>no</td></tr>
<tr><td colspan="5"><code>TLOG.truncated_ind</code></td></tr>
<tr><td>Not in the file. This a parameter that can be added.
When a transaction log entry field value has been truncated down to your <code>TLOG.maxheaderwidth</code> or to your field-specific maximum width, the text indicator to append to the truncated field value to indicate that the field value has been truncated.</td><td>TEXT (max 1022 chars)</td><td>". . ."</td><td>null</td><td>no</td></tr>
</table>

m2g.properties Settings (Part 15 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
`TLOG.translatefromchars`				

Keeps the POP TLOG strings from wrapping to next line whenever it reads a `"\n\r"` (say, `^M`). There is one line for each TLOG entry.
`TLOG.translatefromchars` describes which group of characters needs to be replaced with `TLOG.translatetochars`.
Take the following case as an example:
POP needs to dump the following info to TLOG as a single entry:
`A\n\rBB\n\rCCC\n\r`
Without the configuration, it will appear as:
`A`
`BB`
`CCC`
With the configuration, it will become
`A BB CCC`
Note that the value for `TLOG.translatetochars` (`"  "`) has taken the place of `"\n\r"`.
File default = "`\n\t\r`"
	TEXT (any character combinations)	see description	null	no

`TLOG.translatetochars`				

The characters that replace the group characters set by `TLOG.translatefromchars.`
Take the following case as an example:
POP needs to dump the following info to TLOG as a single entry:
`A\n\rBB\n\rCCC\n\r`
Without the configuration, it will appear as:
`A`
`BB`
`CCC`
With the configuration, it will become
`A BB CCC`
Note that the value for `TLOG.translatetochars` (`"  "`) has taken the place of `"\n\r"`.
	TEXT (any char)	" "	" "	no

m2g.properties Settings (Part 16 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
`TLOG.maxheaderwidth`				
This parameter is not in the configuration file but can be added. Default maximum number of characters to allow in a single transaction log entry field value. Field values longer than this are truncated down to your specified maximum width. Note that the total maximum length of the field will be `TLOG.maxheaderwidth` plus the length of your configured `TLOG.truncated_ind`. By default this truncation indicator is 3 characters long. For example, if you have `TLOG.maxheaderwidth` set to 256 and you use the default `TLOG.truncated_ind`, then the maximum possible field length will be 259 characters. This default maximum width for transaction log field values can be superseded on a per-field basis using the `TLOG.format` setting (for a description, see the analogous `ALOG.format` setting definition on *page 82*). If you assign a particular field a field-specific maximum width, then the maximum allowed width for that field is the *lesser* of your field-specific setting and your `TLOG.maxheaderwidth` setting. For all fields to which you do not assign a field-specific maximum, the `TLOG.maxheaderwidth` setting is applied. You should use `TLOG.maxheaderwidth` to set a maximum that you want no field value to exceed, and then use field-specific maximums for fields that you want to limit to lengths shorter than `TLOG.maxheaderwidth`.	INT (1 to INT_MAX)	256	null	no

m2g.properties Settings (Part 17 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
`TLOG.echolog`				
This parameter is not in the configuration file but can be added. If you want your transaction log entries to be replicated as DEBUG level messages in the application log, set this parameter to `ALOG`. Replicating your transaction log entries to the application log in this way may be helpful when you are troubleshooting, since this allows you to see application messages and transaction records all in the same file. If you leave the `echolog` setting empty (null), then transaction log entries will be written only to the transaction log. When the server replicates a transaction log entry to the application log, the application log entry's `<MODULE>` field will indicate `"TLOG"`, and the entire transaction log entry will appear as the `<MESSAGE>` of the application log entry.	TOKEN (ALOG, null)	setting not in file but can be added	null	no
`TLOG.usegmt`				
This parameter is not in the configuration file but can be added. Whether to use Greenwich Mean Time (GMT) for transaction log entry timestamps. Options are: ◆ `true` Use GMT. ◆ `false` Use local time.	BOOL (true, false)	setting not in file but can be added	false	no
`STATSMGR.logstats`				
Whether or not to periodically write snapshots of the CLI-viewable statistics (*page 375*) to the *application log*. Options are: ◆ `true` Write the group of CLI-viewable statistics to the application log, at the interval specified by `STATSMGR.loginterval`. The values of these statistics will be their values at the time of writing. ◆ `false` Do not write statistics to the application log. IMPORTANT: This setting impacts only the writing of the CLI-viewable statistics to the application log—*not* the logging of statistics to the statistics log.	BOOL (true, false)	false	true	no

m2g.properties Settings (Part 18 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
`STATSMGR.loginterval`				
Interval at which to write snapshots of the CLI-viewable statistics (*page 375*) to the *application log*. Applicable only if `STATSMGR.logstats` is set to true.	TIME (1m to 1d)	60m	1m	yes
`STATSMGR.sampleinterval`				
This parameter is not in the configuration file but can be added. Interval at which to write statistics to the *statistics log*. Your `sidlist.cfg` file (*page 202*) determines which statistics will be logged.	TIME (5s to 1h)	5s	5s	yes
`SIDLIST.maxentries`				
Not in the file. This a parameter that can be added. Maximum number of entry lines allowed in the `sidlist.cfg` map file (*page 202*). If you place more than this number of entry lines in the file, the M2G will process the first `maxentries` number of lines in the file and ignore the additional lines. For example, if a map file’s `maxentries` setting is 40, and you enter 50 lines into the file, the first 40 listed lines are implemented and the last 10 lines are ignored.	INT	1000	100	no
`SIDLIST.delimiters`				
Not in the file. This a parameter that can be added. Delimiter to use between the fields in each entry line of the `sidlist.cfg` file.	TOKEN	“|”	space	no
`CLI.prompt`				
The command prompt that displays on a client console connected to the CLI listener.	TEXT (max 1022 chars)	"M2G> "	"M2G> "	no

m2g.properties Settings (Part 19 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
`CLI.banner`				
The banner that displays at the top of a client console connected to the CLI listener. File and internal default = `Hyperscale (r) MESSAGING 2.0 GATEWAY COMMAND LINE INTERFACE`	TEXT (max 1022 chars)	see description	see description	no
`CLI.allowedhosts`				
Comma-separated list of IP addresses that are allowed to connect to the CLI listener. Other hosts attempting to connect to the listener will be declined. This setting must be specified with IP addresses, not host names. Leave this setting empty (null) to accept connections from all hosts.	TEXT (max 1022 chars)	null	null	no
`CLI.timeout`				
If this much time passes without any communication from a connected client, the CLI listener closes the connection.	TIME (0s to 1h)	600s	10m	yes
`CLI.controllow`				
Congestion control low water mark for the CLI listener, in number of open connections. The low water mark serves two purposes: ◆ Triggers congestion warning message. When the number of open client connections to the listener rises above `controllow,` a message is written to the application log. ◆ Triggers reopening of closed interface. In the event that the number of open client connections to the listener rises past `controlhigh`, resulting in the closing of the interface to new connections, the listener is reopened when the number of connections falls back below `controllow`.	INT (0 to INT_MAX)	0	0	yes

m2g.properties Settings (Part 20 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
`CLI.controlhigh`				
Congestion control high water mark for the CLI listener, in number of open connections. When the number of open connections to the listener rises above `controlhigh`, the listener will temporarily stop accepting new connections. The listener will resume accepting new connections when the number of open connections falls back below `controllow`. To disable congestion control for the listener, set `controllow` and `controlhigh` to 0. NOTE: This setting can be changed through the CLI only if it is currently set to a non-zero value.	INT (0 to INT_MAX)	0	0	yes
`CLI.shutdowntimeout`				
In response to a shutdown command, the interval in seconds for the CLI listener to wait for connected clients to close their connections, before the listener terminates the connections for shutdown. During this interval no new connections are accepted. This setting (in seconds) is a listener-specific override of the `DEFAULT.shutdowntimeout` setting (*page 76*).	TIME (0 to 20s)	3	10s	no
`CLI.allow_non_printable`				
This parameter is not in the configuration file but can be added. Whether or not the CLI should allow non-ASCII input. Options are: ◆ `true` Allow UTF8 input with CLI commands. This can be useful when employing the CLI `set` command (*page 369*) to dynamically change configuration properties to which you want to assign non-ASCII string values. ◆ `false` Allow only ASCII input.	BOOL (true, false)	setting not in file but can be added	false	yes

m2g.properties Settings (Part 21 of 86)

Property	Valid	File	Internal	CLI
Description	Range	Default	Default	Set
`CLIX.prompt`				
The command prompt that displays on a client console connected to the process-based CLIX listener. File and internal default = `"M2GX> "`	TEXT (max 1022 chars)	see description	see description	no
`CLIX.banner`				
The banner that displays at the top of a client console connected to the process-based CLIX listener. File and internal default = `Hyperscale (r) MESSAGING 2.0 GATEWAY COMMAND LINE INTERFACE`	TEXT (max 1022 chars)	see description	see description	no
`CLIX.allowedhosts`				
Comma-separated list of IP addresses that are allowed to connect to the process-based CLIX listener. Other hosts attempting to connect to the listener will be declined. This setting must be specified with IP addresses, not host names. Leave this setting empty (null) to accept connections from all hosts.	TEXT (max 1022 chars)	null	null	no
`CLIX.timeout`				
If this much time passes without any communication from a connected client, the process-based CLI listener closes the connection.	TIME (0s to 1h)	600s	10m	yes

m2g.properties Settings (Part 22 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
`CLIX.controllow`				
Congestion control low water mark for the process-based CLIX listener, in number of open connections. The low water mark serves two purposes: ◆ Triggers congestion warning message. When the number of open client connections to the listener rises above `controllow`, a message is written to the application log. ◆ Triggers reopening of closed interface. In the event that the number of open client connections to the listener rises past `controlhigh`, resulting in the closing of the interface to new connections, the listener is reopened when the number of connections falls back below `controllow`.	INT (0 to INT_MAX)	0	0	yes
`CLIX.controlhigh`				
Congestion control high water mark for the process-based CLIX listener, in number of open connections. When the number of open connections to the listener rises above `controlhigh`, the listener will temporarily stop accepting new connections. The listener will resume accepting new connections when the number of open connections falls back below `controllow`. To disable congestion control for the listener, set `controllow` and `controlhigh` to 0. NOTE: This setting can be changed through the CLI only if it is currently set to a non-zero value.	INT (0 to INT_MAX)	0	0	yes
`CLIX.shutdowntimeout`				
In response to a shutdown command, the interval in seconds for the process-based CLIX listener to wait for connected clients to close their connections, before the listener terminates the connections for shutdown. During this interval no new connections are accepted. This setting (in seconds) is a listener-specific override of the `DEFAULT.shutdowntimeout` setting (*page 76*).	TIME (0 to 20s)	3	10s	no

m2g.properties Settings (Part 23 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
`CLIX.allow_non_printable`				
This parameter is not in the configuration file but can be added. Whether or not the CLI should allow non-ASCII input. Options are: ◆ `true` Allow UTF8 input with CLI commands. This can be useful when employing the CLI `set` command (*page 369*) to dynamically change configuration properties to which you want to assign non-ASCII string values. ◆ `false` Allow only ASCII input.	BOOL (true, false)	setting not in file but can be added	false	yes
`XCON.allowedhosts`				
EBF listener to enable M2G character set conversion. Do not change.	Do not change.	" "	null	no
`mta/SMTPSVR.allowedhosts`				
Comma-separated list of IP addresses that are allowed to connect to the SMTP Listener Interface for receiving incoming messages from Internet. There are spamfilter/virus check MTA servers between M2G and Internet. The client is a spamfilter/virus check server. Other hosts attempting to connect to the listener will be declined. This setting must be specified with IP addresses, not hostnames. Leave this setting empty to accept connections from all hosts. This setting can be separately configured for each SMTP listener.	TEXT	“”	null	no

m2g.properties Settings (Part 24 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
`mta/SMTPSVR.controlhigh`				
High water mark for congestion control for the SMTP Listener Interface for receiving incoming messages from Internet. Note, there are spamfilter/virus check MTA servers between M2G and Internet. The client is a spamfilter/virus check server. When the number of messages in queue rises above `controlhigh`, a message is recorded to the application log, and interfaces that you have specified with the `controlinterfaces` setting are shut down. Messages will not be processed out of the queue until you initiate batch processing as described on *page 728*. The closed interfaces reopen when batch processing reduces the number of queued messages below `controllow`. This setting must be at least as high as `controlwarn`. To disable congestion control for the queue processor, set `controllow` and `controlhigh` to 0. NOTE: This setting can be changed through the CLI only if it is currently set to a non-zero value.	INT (0 to INT_MAX)	0	0	see description
`mta/SMTPSVR.maxrecipients`				
Maximum number of recipients per MAIL FROM session, for service messages being transmitted to the MTA SMTP interface from the Internet. Note, there are spamfilter/virus check MTA servers between M2G and Internet. The client is a spamfilter/virus check server. Once the number of RCPT TO commands within one MAIL FROM session reaches your `maxrecipients` setting, no further RCPT TO commands are accepted. The listener rejects excess RCPT TO commands with a 452 SMTP error. Delivery processing of the message continues for the recipients that were accepted before the limit was exceeded. additional recipients rejected if over maxrecipients (unique)	0 to 200	64	64	yes

m2g.properties Settings (Part 25 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
`mta/SMTPSVR.errmaxsize`				
When the message data that a client transmits after the DATA command exceeds this number of bytes, the MTA SMTP listener will respond with error code 550 and close the connection with the client. The listener will measure the message size by the actual data received, not by the client's declared SIZE parameter.	INT (0 to INT_MAX)	10000000	10000000	no
`mta/SMTPSVR.sysmaxsize`				
Default SMTP maximum message size that if exceeded, the system then checks `mta/SMTPSVR.errmaxsize`. If that is exceeded then shuts down the client.	INT (0 to INT_MAX)	10000000	-1	yes
`mta/SMTPSVR.maxinputlinelen`				
Specifies the maximum line length permitted in an incoming message. This limit includes the carriage return (CR) and line feed (LF) characters at the end of a line.	INT (0 to INT_MAX)	1000	1000	no
`mta/SMTPSVR.linelenexceedresp`				
Specifies the SMTP response text to use when the incoming message contains a line whose length including the CR-LF characters exceeds the value specified in `maxinputlinelen` property. File default = "552 Line length exceeds 1000\r\n"	TEXT	see description	see description	no
`mta/SMTPSVR.smtpdomain`				
Used as part of connection response to the client or in other SMTP command responses. It is the domain name of installed SMTP service. Default = `"localhost"`	TEXT	see description	see description	no

m2g.properties Settings (Part 26 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
`mta/SMTPSVR.smtpgreeting`				
Used as part of connection response to client. Default =`"SMTP Geminimobile Technologies"`	TEXT	see description	see description	no
`mta/SMTPSVR.echopipelining`				
If true, the SMTP listener indicates to client that pipe-lining is allowed.	BOOL	true	true	no
`mta/SMTPSVR.ehlosyntaxerr`				
Used in response to client when HELO/EHLO command syntax is incorrect. Default =`"501 EHLO requires domain name as argument.\r\n"`	TEXT	see description	see description	no
`mta/SMTPSVR.mailsyntaxerr`				
Used in response to client when HELO/EHLO command syntax is incorrect. Default =`"553 MAIL command syntax error\r\n"`	TEXT	see description	see description	no
`mta/SMTPSVR.senderokmsg`				
Response to MAIL command when it is accepted.	TEXT	“”	“”	no
`mta/SMTPSVR.rcpttookmsg`				
Response to RCPT command when it is accepted.	TEXT	“”	“”	no
`mta/SMTPSVR.rcpttoomanyerr`				
Error response when there are too many RCPT commands received.	TEXT	“”	“”	no

m2g.properties Settings (Part 27 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
`mta/SMTPSVR.rcptsyntaxerr`				
Error response when RCPT command syntax error is detected. Default =`"553 RCPT command syntax error\r\n"`	TEXT	see description	see description	no
`mta/SMTPSVR.invalidcmdseq`				
Error response to client SMTP commands that are received out-of-sequence. Default =`"503 Bad sequence of commands\r\n"`	TEXT	see description	see description	no
`mta/SMTPSVR.datainitialresp`				
Prompt to client to send mail message. Default =`"354 Ok Send data ending with <CRLF>.<CRLF>\r\n"`	TEXT	see description	see description	no
`mta/SMTPSVR.sizeexceedresp`				
Error response when mail message is larger than configured max allowed size. Default = `"552 Message size exceeds maximum value.\r\n"`	TEXT	see description	see description	no
`mta/SMTPSVR.linelenexceedresp`				
Specifies the SMTP response text to use when the incoming message contains a line whose length including the CR-LF characters exceeds the value specified in `maxinputlinelen` property. Default = `"552 Line length exceeds 1000\r\n"`	TEXT	see description	see description	no

m2g.properties Settings (Part 28 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
`mta/SMTPSVR.smtptimeoutresp`				
Error response when client fails to communicate with server for a certain amount of time. Default =`"421 <local-domain> Service closing transmission channel due to timeouts\r\n"` Internal Default = `"221 <local-domain> Service closing transmission channel due to timeouts\r\n"`	TEXT	see description	see description	no
`mta/SMTPSVR.smtpcongestionresp`				
SMTP response to use when this listener is configured to handle congestion control. When 'handlecongestion' is set to true listener is responsible for rejecting client connection. If value of this property is empty, connection is closed without any response. Default = `"421 <local-domain> Service not available, closing transmission channel\r\n221 <local-domain> Service closing transmission channel\r\n"`	TEXT	see description	see description	no
`mta/SMTPSVR.mfsizeexceedcode`				
SMTP response code used when the size parameter value of MAIL command exceeds some set limit.	TEXT	542	542	no
`mta/SMTPSVR.heartbeatclients`				
List of client IP addresses from which heart beat pings may be received. This is used to eliminate excessive log entries in application and transaction log files for heart beat monitoring activities. Caution: the IP addresses must not be used by clients that send other requests.	TEXT	“”	“”	no

m2g.properties Settings (Part 29 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
dcm/SMTPSVR.allowedhosts				
Comma-separated list of IP addresses that are allowed to connect to the DoCoMo SMTP listener. Other hosts attempting to connect to the listener will be declined. This setting must be specified with IP addresses, not hostnames. Leave this setting empty to accept connections from all hosts. This setting can be separately configured for each SMTP listener.	TEXT	" "	null	no
dcm/SMTPSVR.controlhigh				
High water mark for congestion control for the Docomo SMTP Listener Interface used for receiving messages from CiRCUS (i-mode). When the number of messages in queue rises above controlhigh, a message is recorded to the application log, and interfaces that you have specified with the controlinterfaces setting are shut down. Messages will not be processed out of the queue until you initiate batch processing as described on *page 728*. The closed interfaces reopen when batch processing reduces the number of queued messages below controllow. This setting must be at least as high as controlwarn. To disable congestion control for the queue processor, set controllow and controlhigh to 0. NOTE: This setting can be changed through the CLI only if it is currently set to a non-zero value.	INT (0 to INT_MAX)	0	0	yes

m2g.properties Settings (Part 30 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
`dcm/SMTPSVR.maxrecipients`				
Maximum number of unique recipients per MAIL FROM session, for service messages being transmitted to the Docomo SMTP Listener Interface used for receiving messages from CiRCUS (i-mode). Once the number of RCPT TO commands within one MAIL FROM session reaches your `maxrecipients` setting, no further RCPT TO commands are accepted. The listener rejects excess RCPT TO commands with a 452 SMTP error. Delivery processing of the message continues for the recipients that were accepted before the limit was exceeded.	INT (0 to INT_MAX)	1	64	yes
`dcm/SMTPSVR.errmaxsize`				
When the message data that a client transmits after the DATA command exceeds this number of bytes, the Docomo SMTP Listener Interface used for receiving messages from CiRCUS (i-mode) will respond with error code 550 and close the connection with the client. The listener will measure the message size by the actual data received, not by the client's declared SIZE parameter.	INT (0 to INT_MAX)	10000000	10000000	no
`dcm/SMTPSVR.sysmaxsize`				
Default SMTP maximum message size for the Docomo SMTP Listener Interface used for receiving messages from CiRCUS (i-mode) that if exceeded, the system then checks `mta/SMTPSVR.errmaxsize`. If that is exceeded then shuts down the client.	INT (0 to INT_MAX)	-1	-1	yes
`dcm/SMTPSVR.maxinputlinelen`				
Specifies the maximum line length permitted in an incoming message. This limit includes the carriage return (CR) and line feed (LF) characters at the end of a line.	INT (0 to INT_MAX)	1000	1000	no

m2g.properties Settings (Part 31 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
`dcm/SMTPSVR.linelenexceedresp`				
Specifies the SMTP response text to use when the incoming message contains a line whose length including the CR-LF characters exceeds the value specified in `maxinputlinelen` property. File default = "552 Line length exceeds 1000\r\n"	TEXT	see description	see description	no
`dcm/SMTPSVR.smtpdomain`				
Used as part of connection response to client or in other SMTP command responses. It is the domain name of installed SMTP service. Default = `"localhost"`	TEXT	see description	see description	no
`dcm/SMTPSVR.smtpgreeting`				
Used as part of connection response to the Docomo SMTP server. Default =`"SMTP Geminimobile Technologies"`	TEXT	see description	see description	no
`dcm/SMTPSVR.echopipelining`				
If true, the SMTP listener indicates to client that pipe-lining is allowed.	BOOL	true	true	no
`dcm/SMTPSVR.ehlosyntaxerr`				
Used in response to client when HELO/EHLO command syntax is incorrect. Default =`"501 EHLO requires domain name as argument.\r\n"`	TEXT	see description	see description	no
`dcm/SMTPSVR.mailsyntaxerr`				
Error response to MAIL command error. Default =`"553 MAIL command syntax error\r\n"`	TEXT	see description	see description	no

m2g.properties Settings (Part 32 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
dcm/SMTPSVR.senderokmsg				
Response to MAIL command when it is accepted.	TEXT	“”	“”	no
dcm/SMTPSVR.rcpttookmsg				
Response to RCPT command when it is accepted.	TEXT	“”	“”	no
dcm/SMTPSVR.rcpttoomanyerr				
Error response when there are too many RCPT commands received.	TEXT	“”	“”	no
dcm/SMTPSVR.rcptsyntaxerr				
Error response when RCPT command syntax error is detected. Default ="553 RCPT command syntax error\r\n"	TEXT	see description	see description	no
dcm/SMTPSVR.invalidcmdseq				
Error response to client SMTP commands that are received out-of-sequence. Default ="503 Bad sequence of commands\r\n"	TEXT	see description	see description	no
dcm/SMTPSVR.datainitialresp				
Prompt to client to send mail message. Default ="354 Ok Send data ending with <CRLF>.<CRLF>\r\n"	TEXT	see description	see description	no
dcm/SMTPSVR.sizeexceedresp				
Error response when mail message is larger than configured max allowed size. Default = "552 Message size exceeds maximum value.\r\n"	TEXT	see description	see description	no

m2g.properties Settings (Part 33 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
`dcm/SMTPSVR.linelenexceedresp`				
Specifies the SMTP response text to use when the incoming message contains a line whose length including the CR-LF characters exceeds the value specified in `maxinputlinelen` property. Default = `"552 Line length exceeds 1000\r\n"`	TEXT	see description	see description	no
`dcm/SMTPSVR.smtptimeoutresp`				
Error response when client fails to communicate with server for a certain amount of time. Default =`"221 <local-domain> Service closing transmission channel due to timeouts\r\n"`	TEXT	see description	see description	no
`dcm/SMTPSVR.smtpcongestionresp`				
SMTP response to use when this listener is configured to handle congestion control. When 'handlecongestion' is set to true listener is responsible for rejecting client connection. If value of this property is empty, connection is closed without any response. Default = `"421 <local-domain> Service not available, closing transmission channel\r\n221 <local-domain> Service closing transmission channel\r\n"`	TEXT	see description	see description	no
`dcm/SMTPSVR.mfsizeexceedcode`				
MTP response code used when the size parameter value of MAIL command exceeds some set limit.	INT	542	542	no
`dcm/SMTPSVR.heartbeatclients`				
List of client IP addresses from which heart beat pings may be received. This is used to eliminate excessive log entries in application and transaction log files for heart beat monitoring activities. Caution: the IP addresses must not be used by clients that send other requests.		“”	“”	no

m2g.properties Settings (Part 34 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
`pcc/SMTPSVR.allowedhosts`				
Comma-separated list of IP addresses that are allowed to connect to receive SMTP messages from the PC Client (Relay) listener. Other hosts attempting to connect to the listener will be declined. This setting must be specified with IP addresses, not hostnames. Leave this setting empty to accept connections from all hosts. This setting can be separately configured for each SMTPX "plug in" listener.	TEXT	" "	null	no
`pcc/SMTPSVR.controlhigh`				
High water mark for congestion control for receiving messages from the SMTP PC Client (Relay) Listener interface. When the number of messages in queue rises above `controlhigh`, a message is recorded to the application log, and interfaces that you have specified with the `controlinterfaces` setting are shut down. Messages will not be processed out of the queue until you initiate batch processing as described on *page 728*. The closed interfaces reopen when batch processing reduces the number of queued messages below `controllow`. This setting must be at least as high as `controlwarn`. To disable congestion control for the queue processor, set `controllow` and `controlhigh` to 0. NOTE: This setting can be changed through the CLI only if it is currently set to a non-zero value.	INT (0 to INT_MAX)	0	0	yes
`pcc/SMTPSVR.maxrecipients`				
Maximum number of recipients per MAIL FROM session, for service messages received from the PC Client (Relay) SMTP interface. Once the number of RCPT TO commands within one MAIL FROM session reaches your `maxrecipients` setting, no further RCPT TO commands are accepted. The listener rejects excess RCPT TO commands with a 452 SMTP error. Delivery processing of the message continues for the recipients that were accepted before the limit was exceeded.	INT (0 to INT_MAX)	1	64	yes

m2g.properties Settings (Part 35 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
`pcc/SMTPSVR.errmaxsize`				
When the message data that a client transmits after the DATA command exceeds this number of bytes, the PC Client (Relay)SMTP listener will respond with error code 550 and close the connection with the client. The listener will measure the message size by the actual data received, not by the client's declared SIZE parameter.	INT (0 to INT_MAX)	10000000	10000000	no
`pcc/SMTPSVR.sysmaxsize`				
When the message data that a client transmits after the DATA command exceeds this number of bytes but is less than `pcc/SMTPSVR.errmaxsize` , the SMTP listener connection will respond with error code 550 and drop the message, while leaving the connection with the client open. The listener will measure the message size by the actual data received, not by the client's declared SIZE parameter.	INT (0 to INT_MAX)	10000000	-1	no
`pcc/SMTPSVR.maxinputlinelen`				
Specifies the maximum line length permitted in an incoming message. This limit includes the carriage return (CR) and line feed (LF) characters at the end of a line.	INT (0 to INT_MAX)	1000	1000	no
`pcc/SMTPSVR.linelenexceedresp`				
Specifies the SMTP response text to use when the incoming message contains a line whose length including the CR-LF characters exceeds the value specified in `maxinputlinelen` property. File default = "552 Line length exceeds 1000\r\n"	TEXT	see description	see description	no
`pcc/SMTPSVR.smtpdomain`				
Used as part of connection response to client or in other SMTP command responses from the PC client. It is the domain name of installed SMTP service. Default = `"localhost"`	TEXT	see description	see description	no

m2g.properties Settings (Part 36 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
`pcc/SMTPSVR.smtpgreeting`				
Used as part of connection response to PC client. Default = `"SMTP Geminimobile Technologies"`	TEXT	see description	see description	no
`pcc/SMTPSVR.echopipelining`				
If true, the SMTP listener indicates to PC client that pipe-lining is allowed.	BOOL	true	true	no
`pcc/SMTPSVR.ehlosyntaxerr`				
Used in response to client when HELO/EHLO command syntax is incorrect. Default = `"501 EHLO requires domain name as argument.\r\n"`	TEXT	see description	see description	no
`pcc/SMTPSVR.mailsyntaxerr`				
Error response to MAIL command error. Default = `"553 MAIL command syntax error\r\n"`	TEXT	see description	see description	no
`pcc/SMTPSVR.senderokmsg`				
Response to MAIL command when it is accepted.	TEXT	“”	“”	no
`pcc/SMTPSVR.rcpttookmsg`				
Response to RCPT command when it is accepted.	TEXT	“”	“”	no
`pcc/SMTPSVR.rcpttoomanyerr`				
Error response when there are too many RCPT commands received.	TEXT	“”	“”	no

m2g.properties Settings (Part 37 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
`pcc/SMTPSVR.rcptsyntaxerr`				
Error response when RCPT command syntax error is detected. Default =`"553 RCPT command syntax error\r\n"`	TEXT	see description	see description	no
`pcc/SMTPSVR.invalidcmdseq`				
Error response to client SMTP commands that are received out-of-sequence. Default =`"503 Bad sequence of commands\r\n"`	TEXT	see description	see description	no
`pcc/SMTPSVR.datainitialresp`				
Prompt to client to send mail message. Default =`"354 Ok Send data ending with <CRLF>.<CRLF>\r\n"`	TEXT	see description	see description	no
`pcc/SMTPSVR.sizeexceedresp`				
Error response when mail message is larger than configured max allowed size. Default = `"552 Message size exceeds maximum value.\r\n"`	TEXT	see description	see description	no
`pcc/SMTPSVR.linelenexceedresp`				
Specifies the SMTP response text to use when the incoming message contains a line whose length including the CR-LF characters exceeds the value specified in `maxinputlinelen` property. Default = `"552 Line length exceeds 1000\r\n"`	TEXT	see description	see description	no

m2g.properties Settings (Part 38 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
`pcc/SMTPSVR.smtptimeoutresp`				
Error response when client fails to communicate with server for a certain amount of time. Default =`"421 <local-domain> Service closing transmission channel due to timeouts\r\n"` Internal Default =`"221 <local-domain> Service closing transmission channel due to timeouts\r\n"`	TEXT	see description	see description	no
`pcc/SMTPSVR.smtpcongestionresp`				
SMTP response to use when this listener is configured to handle congestion control. When 'handlecongestion' is set to true listener is responsible for rejecting client connection. If value of this property is empty, connection is closed without any response. Default = `"421 <local-domain> Service not available, closing transmission channel\r\n221 <local-domain> Service closing transmission channel\r\n"`	TEXT	see description	see description	no
`pcc/SMTPSVR.mfsizeexceedcode`				
SMTP response code used when the size parameter value of MAIL command exceeds some set limit.	INT	542	542	no
`pcc/SMTPSVR.heartbeatclients`				
List of client IP addresses from which heart beat pings may be received. This is used to eliminate excessive log entries in application and transaction log files for heart beat monitoring activities. Caution: the IP addresses must not be used by clients that send other requests.	TEXT	“”	“”	no

m2g.properties Settings (Part 39 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
`pcc/SMTPSSLSVR.allowedhosts`				
Comma-separated list of IP addresses that are allowed to connect to receive SMTP messages from the PC Client (Relay) SSL SMTP listener connection. Other hosts attempting to connect to the listener will be declined. This setting must be specified with IP addresses, not hostnames. Leave this setting empty to accept connections from all hosts. This setting can be separately configured for each SMTPX "plug in" listener.	TEXT	" "	null	no
`pcc/SMTPSSLSVR.controlhigh`				
High water mark for congestion control for receiving messages from the SMTP PC Client (Relay) SSL SMTP listener connection. When the number of messages in queue rises above `controlhigh`, a message is recorded to the application log, and interfaces that you have specified with the `controlinterfaces` setting are shut down. Messages will not be processed out of the queue until you initiate batch processing as described on *page 728*. The closed interfaces reopen when batch processing reduces the number of queued messages below `controllow`. This setting must be at least as high as `controlwarn`. To disable congestion control for the queue processor, set `controllow` and `controlhigh` to 0. NOTE: This setting can be changed through the CLI only if it is currently set to a non-zero value.	INT (0 to INT_MAX)	0	0	yes
`pcc/SMTPSSLSVR.maxrecipients`				
Maximum number of recipients per MAIL FROM session, for service messages received from the PC Client (Relay) SSL SMTP listener connection. Once the number of RCPT TO commands within one MAIL FROM session reaches your `maxrecipients` setting, no further RCPT TO commands are accepted. The listener rejects excess RCPT TO commands with a 452 SMTP error. Delivery processing of the message continues for the recipients that were accepted before the limit was exceeded.	INT (0 to INT_MAX)	1	64	yes

m2g.properties Settings (Part 40 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
`pcc/SMTPSSLSVR.errmaxsize`				
When the message data that a client transmits after the DATA command exceeds this number of bytes, the PC Client (Relay)SSL SMTP listener connection will respond with error code 550 and close the connection with the client. The listener will measure the message size by the actual data received, not by the client's declared SIZE parameter.	INT (0 to INT_MAX)	10000000	10000000	no
`pcc/SMTPSSLSVR.sysmaxsize`				
When the message data that a client transmits after the DATA command exceeds this number of bytes but is less than `pcc/SMTPSSLSVR.errmaxsize` , the SSL listener connection will respond with error code 550 and drop the message, while leaving the connection with the client open. The listener will measure the message size by the actual data received, not by the client's declared SIZE parameter.	INT (0 to INT_MAX)	10000000	-1	no
`pcc/SMTPSSLSVR.maxinputlinelen`				
Specifies the maximum line length permitted in an incoming message. This limit includes the carriage return (CR) and line feed (LF) characters at the end of a line.	INT (0 to INT_MAX)	1000	1000	no
`pcc/SMTPSSLSVR.linelenexceedresp`				
Specifies the SMTP response text to use when the incoming message contains a line whose length including the CR-LF characters exceeds the value specified in `maxinputlinelen` property. File default = "552 Line length exceeds 1000\r\n"	TEXT	see description	see description	no
`pcc/SMTPSSLSVR.smtpdomain`				
Used as part of connection response to client or in other SMTP command responses. It is the domain name of installed SMTP service connectionto the SSL PC server. Default = `"localhost"`	TEXT	see description	see description	no

m2g.properties Settings (Part 41 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
`pcc/SMTPSSLSVR.smtpgreeting`				
Used as part of connection response to the SSL PC client. Default = `"SMTP Geminimobile Technologies"`	TEXT	see description	see description	no
`pcc/SMTPSSLSVR.echopipelining`				
If true, the SMTP listener indicates to client that pipe-lining is allowed.	BOOL	true	true	no
`pcc/SMTPSSLSVR.ehlosyntaxerr`				
Used in response to client when HELO/EHLO command syntax is incorrect. Default = `"501 EHLO requires domain name as argument.\r\n"`	TEXT	see description	see description	no
`pcc/SMTPSSLSVR.mailsyntaxerr`				
Error response to MAIL command error. Default = `"553 MAIL command syntax error\r\n"`	TEXT	see description	see description	no
`pcc/SMTPSSLSVR.senderokmsg`				
Response to MAIL command when it is accepted.	TEXT	“”	“”	no
`pcc/SMTPSSLSVR.rcpttookmsg`				
Response to RCPT command when it is accepted.	TEXT	“”	“”	no
`pcc/SMTPSSLSVR.rcpttoomanyerr`				
Error response when there are too many RCPT commands received.	TEXT	“”	“”	no

m2g.properties Settings (Part 42 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
`pcc/SMTPSSLSVR.rcptsyntaxerr`				
Error response when RCPT command syntax error is detected. Default =`"553 RCPT command syntax error\r\n"`	TEXT	see description	see description	no
`pcc/SMTPSSLSVR.invalidcmdseq`				
Error response to client SMTP commands that are received out-of-sequence. Default =`"503 Bad sequence of commands\r\n"`	TEXT	see description	see description	no
`pcc/SMTPSSLSVR.datainitialresp`				
Prompt to client to send mail message. Default =`"354 Ok Send data ending with <CRLF>.<CRLF>\r\n"`	TEXT	see description	see description	no
`pcc/SMTPSSLSVR.sizeexceedresp`				
Error response when mail message is larger than configured max allowed size. Default = `"552 Message size exceeds maximum value.\r\n"`	TEXT	see description	see description	no
`pcc/SMTPSSLSVR.linelenexceedresp`				
Specifies the SMTP response text to use when the incoming message contains a line whose length including the CR-LF characters exceeds the value specified in `maxinputlinelen` property. Default = `"552 Line length exceeds 1000\r\n"`	TEXT	see description	see description	no

m2g.properties Settings (Part 43 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
pcc/SMTPSSLSVR.smtptimeoutresp				
Error response when client fails to communicate with server for a certain amount of time. Default ="421 <local-domain> Service closing transmission channel due to timeouts\r\n" Internal Default ="221 <local-domain> Service closing transmission channel due to timeouts\r\n"	TEXT	see description	see description	no
pcc/SMTPSSLSVR.smtpcongestionresp				
SMTP response to use when this listener is configured to handle congestion control. When 'handlecongestion' is set to true listener is responsible for rejecting client connection. If value of this property is empty, connection is closed without any response. Default = "421 <local-domain> Service not available, closing transmission channel\r\n221 <local-domain> Service closing transmission channel\r\n"	TEXT	see description	see description	no
pcc/SMTPSSLSVR.mfsizeexceedcode				
SMTP response code used when the size parameter value of MAIL command exceeds some set limit.	IN	542	542	no
pcc/SMTPSSLSVR.heartbeatclients				
List of client IP addresses from which heart beat pings may be received. This is used to eliminate excessive log entries in application and transaction log files for heart beat monitoring activities. Caution: the IP addresses must not be used by clients that send other requests.	TEXT	“”	“”	no

m2g.properties Settings (Part 44 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
`mta/SMTPCMDRCPT.timeout`				
After sending a command to the target A2S server over a particular connection from the pool, the maximum time to wait for a response receipt of authorization from the target server. If the response is not received within the timeout period, the MTA processing server terminates the connection. To disable timeout limit, set to 0s.	TIME (0s to 1h)	30	0	yes
`mta/SMTPCMDRCPT.txntimeout`				
The maximum time to wait for the total time to get a receipt of authorization from the target A2S server and to notify the client. If the response and notification does not happen within the timeout period, the MTA processing server terminates the connection. To disable timeout limit, set to 0s.	0s to 1h	20	0	yes
`mta/SMTPCMDRCPT.perm_error_code`				
Permanent error code for the "receipt to" command for the MTA A2S validation. Requested action not taken: mailbox unavailable, for example, mailbox not found, no access.	INT	550	553	no
`mta/SMTPCMDRCPT.perm_error_text`				
Permanent error code text for the "receipt to" command for the MTA A2S validation. This could be because the recipient doesn't exist in the A2S. Default = `"Requested action not taken:mailbox unavailable"`	TEXT	see description	see description	no
`mta/SMTPCMDRCPT.temp_error_code`				
Temporary error code for the "receipt to" commandfor the MTA A2S validation. Requested action aborted: local error in processing. This could be because the A2S is temporarily down.	INT	451	451	no

m2g.properties Settings (Part 45 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
`mta/SMTPCMDRCPT.temp_error_text`				
Temporary error text for the "receipt to" command for the MTA A2S validation. Default = `"Requested action aborted:local error in processing."`	TEXT	see description	see description	no
`dcm/MIMEHEADERFILTER.remove_headers`				
Comma separated list of headers to remove that implements DoCoMo header filtering for non-DoCoMo interfaces. Default = `"X-RED-mailtype,X-RED-addresstype,X-RED-DELETE,X-REDMAIL-HEAD"`	TEXT	see description	see description	no
`dcm/SMTPFROMVALIDATOR.address_regex`				
Validates the DoCoMo SMTP MTA A2S Mail From against this regular expression. Default = `"^sender@docomo.ne.jp$"` Internal default = `"^[_a-z0-9-]+(.[_a-z0-9-]+)*@[a-z0-9-]+(.[a-z0-9-]+)*(.[a-z]{2,3})$";`	TEXT	see description	see description	no
`dcm/SMTPCMDRCPT.timeout`				
After sending a command to the target server over a particular connection from the pool, the maximum time to wait for a response from the target server. If the response is not received within the timeout period, the DoCoMo processing server terminates the connection. To disable timeout limit, set to 0s. DoCoMo MTA A2S validation	TIME (0s to 1h)	30	0	yes

m2g.properties Settings (Part 46 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
`dcm/SMTPCMDRCPT.txntimeout`				
The maximum time to wait for the total time to get a receipt of authorization from the target DoCoMo server and to notify the client. If the response and notification does not happen within the timeout period, the MTA processing server terminates the connection. To disable timeout limit, set to 0s.	0s to 1h	20	0	yes
`dcm/SMTPCMDRCPT.strip_domain`				
If true, the domain is stripped before authentication is done.	BOOL	true	false	no
`dcm/SMTPCMDRCPT.host_list`				
If set, only addresses that end with one of these will be authenticated. Default = `"docomo.ne.jp"`	VECTOR of TEXTS	see description	see description	no
`dcm/SMTPCMDRCPT.opcoid`				
The operator ID against which to validate addresses.	0 to INT_MAX	1	1	no
`dcm/SMTPCMDRCPT.perm_error_code`				
SMTP permanent error code. For example, mailbox not found, no access.	INT	550	553	no
`dcm/SMTPCMDRCPT.perm_error_text`				
SMTP permanent error text. For example, mailbox not found, no access. Default = `"Requested action not taken: mailbox unavailable"` Internal default = "RCPT Syntax error.	TEXT	see description	see description	no

m2g.properties Settings (Part 47 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
dcm/SMTPCMDRCPT.temp_error_code				
SMTP temporary error code. Requested action aborted: local error in processing.	INT	451	451	no
dcm/SMTPCMDRCPT.temp_error_text				
SMTP temporary error text. Requested action aborted: local error in processing. Default = "RCPT Processing Error" Internal default = Default = "RCPT Processing Error"	TEXT	see description	see description	no
dcm/HEADERCHECK.error_code				
This is a setting for the DoCoMo interface. If the correct DoCoMo message headers are not set, this SMTP error code is sent	INT	554	554	no
dcm/HEADERCHECK.error_text				
This is a setting for the DoCoMo interface. If the correct DoCoMo message headers are not set, this is the SMTP error text response. Default = "Invalid Header Error" Internal default = "Invalid Header Error"	TEXT	see description	see description	no
dcm/HEADERCHECK.hdr_rename_from				
This is a setting for the DoCoMo interface. New mails are examined for this value. If it doesn't exist, any existing headers with the HEADERCHECK.hdr_rename_to are removed. If the HEADERCHECK.hdr_rename_from header exists, it is removed and changed to HEADERCHECK.hdr_rename_to and given the prefix of the value for HEADERCHECK.hdr_rename_to_prefix. Default = "X-RED-DELETE" Internal default = "X-RED-DELETE"	TEXT	see description	see description	no

m2g.properties Settings (Part 48 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
`dcm/HEADERCHECK.hdr_rename_to`				
This is a setting for the DoCoMo interface. If the `HEADERCHECK.hdr_rename_from` header exists, it is removed and changed to `HEADERCHECK.hdr_rename_to` and given the prefix of the value for `HEADERCHECK.hdr_rename_to_prefix`. The system won't start if both `HEADERCHECK.hdr_rename_to` or `HEADERCHECK.hdr_rename_to_prefix` are not set. Default = `"X-RED-HEAD"`	TEXT	see description	“”	no
`dcm/HEADERCHECK.hdr_rename_to_prefix`				
This is a setting for the DoCoMo interface. If the `HEADERCHECK.hdr_rename_from` header exists, it is removed and changed to `HEADERCHECK.hdr_rename_to` and given the prefix of the value for `HEADERCHECK.hdr_rename_to_prefix`. The system won't start if both `HEADERCHECK.hdr_rename_to` or `HEADERCHECK.hdr_rename_to_prefix` are not set. Default = `"prefix"`	TEXT	see description	“”	no
`MAXHOPCHECK.error_code`				
SMTP response code if too many hops. This component checks number of Received headers in a message against configured maximum allowed value. If the maximum value is exceeded, the message is determined to be in a mail relay loop and won't beaccepted for delivery.	INT	554	554	no
`MAXHOPCHECK.error_text`				
SMTP command response text. Default = "Too many mail hops" Internal default = "Too many mail hops"	TEXT	see description	see description	no

m2g.properties Settings (Part 49 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
`MAXHOPCHECK.maxmailhops`				
The maximum number of MTA mail hops that are allowed. If the maximum value is exceeded, the message is determined to be in a mail relay loop and won't be accepted for delivery.	0 to INT_MAX	10	10	no
`MAXHOPCHECK.bounceiftoomanyhops`				
If true, sends a bounce message to sender. If `bounceiftoomanyhops` or `logiftoomanyhops` is true and the error_code is not 0 or less and the error_text is not empty, then the message is not delivered or relayed further. Otherwise, the decision of whether to deliver the message is passed to the next component.	BOOL	false	false	no
`MAXHOPCHECK.logiftoomanyhops`				
If true, sends a bounce message to sender. If `bounceiftoomanyhops` or `logiftoomanyhops` is true and the error_code is not 0 or less and the error_text is not empty, then the message is not delivered or relayed further. Otherwise, the decision of whether to deliver the message is passed to the next component.	BOOL	true	false	no
`MAXHOPCHECK.logdirectory`				
If set, messages determined to be in the replay loop will be stored under this directory. Default =`/var/maillop/store/`	TEXT	see description	empty	no
`MAXHOPCHECK.apploglevel`				
If a loop is detected, an entry is written in application log file. Log level is specified by this configuration.	TEXT	INFO	INFO	no
`SMTPACCEPTOR.timeout`				
Network polling timeout in microseconds.	TIME (0s to 1h)	310	30	no

m2g.properties Settings (Part 50 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
`SMTPACCEPTOR.txntimeout`				
Transaction timeout in milliseconds.	TIME (0s to 1h)	300	10	no
`SMTPACCEPTOR.perm_error_code`				
Permanent SMTP error code sent to the MTA.	INT	552	552	no
`SMTPACCEPTOR.perm_error_text`				
Permanent error text message sent to the sender. Default = `"Message size exceeds maximum value."` Internal default = `"Message size exceeds maximum value."`	TEXT	see description	see description	no
`SMTPACCEPTOR.temp_error_code`				
Temporary SMTP error code sent to the MTA to try again.	INT	451	451	no
`SMTPACCEPTOR.temp_error_text`				
Error text sent to the sender. Default = `"Processing Error"` Internal default = `"Processing Error"`	TEXT	see description	see description	no
`SMTPACCEPTOR.maxsize`				
Maximum size for the EBF request/response. The maxsize parameter effects the maximum size of a message that can be received. It is not the max message size though and should always be set to a higer value than the maximum mail size allowed. Its the max size of the internal protocol request that will contain the incoming mails so it needs to be bigger. Default = 26214400 Internal default = 26214400	0 to INT_MAX	see description	see description	no

m2g.properties Settings (Part 51 of 86)

Property	Valid	File	Internal	CLI
Description	Range	Default	Default	Set
`pcc/SMTPACCEPTOR.timeout`				
Network polling timeout in microseconds.	TIME (0s to 1h)	310	30	no
`pcc/SMTPACCEPTOR.txntimeout`				
Transaction timeout in milliseconds.	TIME (0s to 1h)	300	10	no
`SMTPCLIENT.checkmailfromdomain`				
When `checkmailfromdomain` is true, `smtphelodomain` may be appended to the reverse path address found in SMTP MAIL command if the reverse path (the sender) address does not already have a domain address.	BOOL	true	true	yes
`SMTPCLIENT.smtphelodomain`				
When `checkmailfromdomain` is true, `smtphelodomain` may be appended to the reverse path address found in SMTP MAIL command if the reverse path (the sender) address does not already have a domain address. `Default=virus.mta.nttr.com`	TEXT	see description	“”	no
`DCMSMTPCLIENT.checkmailfromdomain`				
When `checkmailfromdomain` is true, `smtphelodomain` may be appended to the reverse path address found in SMTP MAIL command if the reverse path (the sender) address does not already have a domain address.	BOOL	true	true	yes
`DCMSMTPCLIENT.smtphelodomain`				
For internal use only. Do not change. `Default=dcm.nttr.com`	TEXT	see description	see description	no

m2g.properties Settings (Part 52 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
`M3CONMGR.lmtpconnpool`				
For internal use only. Do not change. Default = `LMTPPOOL0`	TEXT	see description	see description	no
`M3CONMGR.popconnpool`				
For internal use only. Do not change. Default = `POP3POOLMGR`	TEXT	see description	see description	no
`M3CONMGR.esmtpconnpool`				
For internal use only. Do not change. Default = `SMTPPOOL0`	TEXT	see description	see description	no
`pcc/M2H4POP.max_list_messages`				
This is the maximum number of messages that the POP server will retrieve from GDSS and display in the listing to the POP client. Smaller numbers are better for performance/memory.	INT > 1 to 500000	200000	1000	yes
`pcc/M2H4POP.txntimeout`				
This is the transaction timeout for the erlang operation.	INT > 1 to 500000	60	60	no
`pcc/M2H4POP.timeout`				
This is the network read/write timeout for erlang operation.	INT > 1 to 500000	65	70	no
`pcc/POPSVR.connectiontimeout`				
After accepting an incoming connection from the customer's mail client, the maximum time that the POP listener will wait for a command from the client before dropping the connection.	TIME	70s	5m	no

m2g.properties Settings (Part 53 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
`pcc/POPSVR.commandtimeout`				
After sending a command other than those previously specified in this table to the target server, the maximum time for the client to wait for a response code from the server. If the response is not received within the timeout period, the client terminates the connection.	TIME	70s	60s	no
`pcc/POPSVR.controlhigh`				
High water mark for congestion control for receiving messages from the SMTP PC Client (Relay) SSL SMTP listener connection. When the number of messages in queue rises above `controlhigh`, a message is recorded to the application log, and interfaces that you have specified with the `controlinterfaces` setting are shut down. Messages will not be processed out of the queue until you initiate batch processing as described on *page 728*. The closed interfaces reopen when batch processing reduces the number of queued messages below `controllow`. This setting must be at least as high as `controlwarn`. To disable congestion control for the queue processor, set `controllow` and `controlhigh` to 0. NOTE: This setting can be changed through the CLI only if it is currently set to a non-zero value.	INT (0 to INT_MAX)	0	0	yes
`pcc/POPSVR.controllow`				
Congestion control low water mark for the PC Client listener, in number of open connections. The low water mark serves two purposes: ◆ Triggers congestion warning message. When the number of open client connections to the listener rises above `controllow`, a message is written to the application log. ◆ Triggers reopening of closed interface. In the event that the number of open client connections to the listener rises past `controlhigh`, resulting in the closing of the interface to new connections, the listener is reopened when the number of connections falls back below `controllow`.	INT (0 to INT_MAX)	0	0	yes

m2g.properties Settings (Part 54 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
`pcc/POPSVR.controlinterfaces`				
For internal use only. Do not change. Default = `pcc/POPSV`	TEXT	see description	null	no
`pcc/POPSVR.popgreeting`				
Greeting message sent out by our POP Listener to the POP client programs on the PC client side. Default = `"POP3 service ready"`	TEXT	see description	see description	no
`pcc/POPSVR.poplogout`				
Log out message sent out by our POP Listener to the POP client programs on the PC client side. Default = `"POP3 Server signing off"`	TEXT	see description	see description	no
`pcc/POPSVR.validuser`				
Valid user message sent out by our POP Listener to the POP client programs on the PC client side. Default = "is a valid user."	TEXT	see description	see description	no
`pcc/POPSVR.userloggedin`				
Log in message sent out by our POP Listener to the POP client programs on the PC client side. Default = `"logged in. Maildrop locked and ready."`	TEXT	see description	see description	no
`pcc/POPSVR.invaliduser`				
Invalid account message sent out by our POP Listener to the POP client programs on the PC client side. Default = `"invalid account"`	TEXT	see description	see description	no

m2g.properties Settings (Part 55 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
`pcc/POPSVR.invalidpass`				
Invalid password message sent out by our POP Listener to the POP client programs on the PC client side. Default = `"invalid password"`	TEXT	see description	see description	no
`pcc/POPSVR.illegalcmd`				
Illegal command message sent out by our POP Listener to the POP client programs on the PC client side. Default = `"illegal command:"`	TEXT	see description	see description	no
`pcc/POPSVR.unknowncmd`				
Unrecognizd commandmessage sent out by our POP Listener to the POP client programs on the PC client side. Default = `"unrecognized command:"`	TEXT	see description	see description	no
`pcc/POPSVR.missingarg`				
Missing argument message sent out by our POP Listener to the POP client programs on the PC client side. Default = `"missing command argument"`	TEXT	see description	see description	no
`pcc/POPSVR.nosuchmsg`				
No such message message sent out by our POP Listener to the POP client programs on the PC client side. Default = `"no such message"`	TEXT	see description	see description	no
`pcc/POPSVR.deleteisok`				
Message deleted message sent out by our POP Listener to the POP client programs on the PC client side. Default = `"message deleted"`	TEXT	see description	see description	no

m2g.properties Settings (Part 56 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
`pcc/POPSVR.wrongarg`				
Wrong argument message sent out by our POP Listener to the POP client programs on the PC client side. Default = `"incorrect argument for"`	TEXT	see description	see description	no
`pcc/POPSSLSVR.connectiontimeout`				
After accepting an incoming connection from the customer's mail client, the maximum time that the POP SSL listener will wait for a command from the client before dropping the connection.	TIME	60s	5m	no
`pcc/POPSSLSVR.commandtimeout`				
After sending a response to a client command other than those listed in the next several rows of this table (for instance, a NOOP), the maximum time that the PC Client SSL listener will wait for the next command from the client. If the next command is not received within the timeout period, the listener terminates the connection.	TIME	60s	60s	no
`pcc/POPSSLSVR.controlhigh`				
High water mark for congestion control for receiving messages from the PC Client SSL POP listener connection. When the number of messages in queue rises above `controlhigh`, a message is recorded to the application log, and interfaces that you have specified with the `controlinterfaces` setting are shut down. Messages will not be processed out of the queue until you initiate batch processing as described on *page 728*. The closed interfaces reopen when batch processing reduces the number of queued messages below `controllow`. This setting must be at least as high as `controlwarn`. To disable congestion control for the queue processor, set `controllow` and `controlhigh` to 0. NOTE: This setting can be changed through the CLI only if it is currently set to a non-zero value.	INT (0 to INT_MAX)	0	0	yes

m2g.properties Settings (Part 57 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
`pcc/POPSSLSVR.controlinterfaces`				
For internal use only. Do not change. Default = `pcc/POPSSLSVR`	TEXT	see description	null	no
`pcc/POPSSLSVR.popgreeting`				
Greeting message sent out by our POP SSL Listener to the POP client programs on the PC client side. Default = `"POP3 service ready"`	TEXT	see description	see description	no
`pcc/POPSSLSVR.poplogout`				
Log out message sent out by our POP SSL Listener to the POP client programs on the PC client side. Default = `"POP3 Server signing off"`	TEXT	see description	see description	no
`pcc/POPSSLSVR.validuser`				
Valid user message sent out by our POP SSL Listener to the POP client programs on the PC client side. Default="is a valid user.”	TEXT	see description	see description	no
`pcc/POPSSLSVR.userloggedin`				
Login message sent out by our POP SSL Listener to the POP client programs on the PC client side. Default= `"logged in. Maildrop locked and ready."`	TEXT	see description	see description	no

m2g.properties Settings (Part 58 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
`pcc/POPSSLSVR.invaliduser`				
Invalid account message sent out by our POP SSL Listener to the POP client programs on the PC client side. Default = `"invalid account"`	TEXT	see description	see description	no
`pcc/POPSSLSVR.invalidpass`				
Invalid password message sent out by our POP SSL Listener to the POP client programs on the PC client side. Default = `"invalid password"`	TEXT	see description	see description	no
`pcc/POPSSLSVR.illegalcmd`				
Illegal command message sent out by our POP SSL Listener to the POP client programs on the PC client side. Default = `"illegal command:"`	TEXT	see description	see description v	no
`pcc/POPSSLSVR.unknowncmd`				
Unknown command message sent out by our POP SSL Listener to the POP client programs on the PC client side. Default = `"unrecognized command:"`	TEXT	see description	see description	no
`pcc/POPSSLSVR.missingarg`				
Missaging command message sent out by our POP SSL Listener to the POP client programs on the PC client side. Default = `"missing command argument"`	TEXT	see description	see description	no
`pcc/POPSSLSVR.nosuchmsg`				
No such message message sent out by our POP SSL Listener to the POP client programs on the PC client side. Default = `"no such message"`	TEXT	see description	see description	no

m2g.properties Settings (Part 59 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
`pcc/POPSSLSVR.deleteisok`				
Message deleted message sent out by our POP SSL Listener to the POP client programs on the PC client side. Default = `"message deleted"`	TEXT	see description	see description	no
`pcc/POPSSLSVR.wrongarg`				
Incorrect argument message sent out by our POP SSL Listener to the POP client programs on the PC client side. Default = `"incorrect argument for"`	TEXT	see description	see description	no
`a2s/EBFPOOL.timeout`				
After sending a command to the target server over a particular connection from the pool, the maximum time to wait for a response from the target server. If the response is not received within the timeout period, the A2S processing server terminates the connection. To disable timeout limit, set to 0s.	TIME (0s to 1h)	15s	0	yes
`a2s/EBFPOOL.maxinuse`				
Maximum number of connections the A2S connection pool allows to be in use simultaneously. Unless specified differently, this setting defaults to equal your `maxconnections` setting.	INT (0 to INT_MAX)	1600	0	yes
`a2s/EBFPOOL.maxspare`				
Tool for managing the number of idle connections in the pool. After using a connection to complete a session with the target server, the A2S processing server either: ◆ Closes the connection if the current number of idle connections in the pool is greater than or equal to `maxspare` ◆ Puts the connection back into the connection pool if the current number of idle connections in the pool is less than `maxspare`. To disable the maximum spare connections limit, set this parameter to 0.	`minspare` to `maxconnections`	1500	0	yes

m2g.properties Settings (Part 60 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
`a2s/EBFPOOL.minspare`				
Tool for managing the number of idle connections in the A2S EBF pool. At your specified keep-alive period for idle connections, the server either: ◆ Closes the connection, if the current number of idle connections in the pool is greater than `minspare`; or ◆ Sends a keep-alive signal through the connection, if the current number of idle connections in the pool is less than or equal to `minspare`. If the keep-alive test succeeds, the connection remains open and in the pool; if the test fails, the connection is closed. To disable the minimum spare connections limit, set this parameter to 0.	0 to `max-spare`	5	0	yes
`a2s/EBFPOOL.maxconnections`				
Maximum number of connections allowed for the A2S EBF connection pool	INT (0 to INT_MAX)	1600	null	yes
`a2s/EBFPOOL.maxuses`				
Maximum number of times to reuse a connection from the A2S EBF connection pool before closing it. To disable the maximum reuse limit, set this parameter to 0.	INT (0 to 1000)	1000	0	yes
`a2s/EBFPOOL.keepalive`				
For idle connections in the FE connection pool, the periodic interval at which to either: ◆ Close the connection, if the current number of idle connections in the pool is greater than `minspare`; or Send a keep-alive signal (for example, a NOOP) through the connection, if the current number of idle connections in the pool is equal to or less than `minspare`. If the keep-alive test succeeds, the connection remains open and in the pool; if the test fails, the connection is closed.	TIME (0 to 1 hr)	30	1m	yes

m2g.properties Settings (Part 61 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
`a2s/EBFPOOL.rebinddelay`				
In the event that a rebind command is issued for this connection pool through theCLI, the interval between the execution of the unbind action and the subsequent bind action. IMPORTANT: This value should be larger than your setting for `keepalive` to ensure that all connections are properly unbound.	TIME (0 to 1hr)	31	5m	yes
`a2s/EBFPOOL.maxbadconnections`				
If this number of connections are rejected or die is reached, the incoming interface is shutdown. Zero (0) disables this health monitor.	INT	800	0	yes
`a2s/EBFPOOL.controlinterfaces`				
Comma separated list of compnent names that can be monitored. Default = `mta/SMTPSVR,dcm/SMTPSVR,pcc/SMTPSVR,pcc/SMTPSSLSVR,pcc/POPSVR,pcc/POPSSLSVR`	TEXT	see description	null	TEXT
`a2s/EBFPOOL.controlhigh`				
High water mark for congestion control for receiving messages from the SMTP PC Client (Relay) SSL SMTP listener connection. When the number of messages in queue rises above `controlhigh`, a message is recorded to the application log, and interfaces that you have specified with the `controlinterfaces` setting are shut down. Messages will not be processed out of the queue until you initiate batch processing as described on *page 728*. The closed interfaces reopen when batch processing reduces the number of queued messages below `controllow`. This setting must be at least as high as `controlwarn`. To disable congestion control for the queue processor, set `controllow` and `controlhigh` to 0. NOTE: This setting can be changed through the CLI only if it is currently set to a non-zero value.	INT (0 to INT_MAX)	1600	45	yes

m2g.properties Settings (Part 62 of 86)

Property **Description**	**Valid Range**	**File Default**	**Internal Default**	**CLI Set**
`fe/EBFPOOL.timeout`				
After sending a command to the target server over a particular connection from the pool, the maximum time to wait for a response from the target server. If the response is not received within the timeout period, the EBF processing server terminates the connection. To disable timeout limit, set to 0s.	TIME (0s to 1 hour)	15s	0	yes
`fe/EBFPOOL.maxinuse`				
Maximum number of connections the connection pool allows to be in use simultaneously. Unless specified differently, this setting defaults to equal your `maxconnections` setting.	INT (0 to INT_MAX)	1600	0	yes
`fe/EBFPOOL.maxspare`				
Tool for managing the number of idle connections in the pool. After using a connection to complete a session with the target server, the Front End EBF processing server either: ◆ Closes the connection if the current number of idle connections in the pool is greater than or equal to `maxspare` ◆ Puts the connection back into the connection pool if the current number of idle connections in the pool is less than `maxspare`. To disable the maximum spare connections limit, set this parameter to 0.	`minspare` to `maxconnections`	1500	0	yes

m2g.properties Settings (Part 63 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
`fe/EBFPOOL.minspare`				
Tool for managing the number of idle connections in the Front End EBF pool. At your specified keep-alive period for idle connections, the server either: ◆ Closes the connection, if the current number of idle connections in the pool is greater than `minspare`; or ◆ Sends a keep-alive signal through the connection, if the current number of idle connections in the pool is less than or equal to `minspare`. If the keep-alive test succeeds, the connection remains open and in the pool; if the test fails, the connection is closed. To disable the minimum spare connections limit, set this parameter to 0.	0 to `max-spare`	5	0	yes
`fe/EBFPOOL.maxconnections`				
Maximum number of connections allowed for the front end EBF connection pool.	INT (0 to INT_MAX)	1600	0	yes
`fe/EBFPOOL.maxuses`				
Maximum number of times to reuse a connection from the front end EBF connection pool before closing it. To disable the maximum reuse limit, set this parameter to 0.	INT (0 to 1000)	1000	0	yes
`fe/EBFPOOL.keepalive`				
For idle connections in the FE connection pool, the periodic interval at which to either: ◆ Close the connection, if the current number of idle connections in the pool is greater than `minspare`; or ◆ Send a keep-alive signal (for example, a NOOP) through the connection, if the current number of idle connections in the pool is equal to or less than `minspare`. If the keep-alive test succeeds, the connection remains open and in the pool; if the test fails, the connection is closed.	TIME (0 to 1 hr)	30	1m	yes

m2g.properties Settings (Part 64 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
`fe/EBFPOOL.rebinddelay`				
In the event that a rebind command is issued for this connection pool through theCLI, the interval between the execution of the unbind action and the subsequent bind action. IMPORTANT: This value should be larger than your setting for `keepalive` to ensure that all connections are properly unbound.	TIME 90 to 1hr)	31	5m	yes
`fe/EBFPOOL.maxbadconnections`				
If this number of connections are rejected or die is reached, the incoming interface is shutdown. Zero (0) disables this health monitor.	INT	800	0	yes
`fe/EBFPOOL.controlinterfaces`				
Comma separated list of compnent names that can be monitored. Default = `mta/SMTPSVR,dcm/SMTPSVR,pcc/SMTPSVR,pcc/SMTPSSLSVR,pcc/POPSVR,pcc/POPSSLSVR`	TEXT	see description	null	no
`fe/EBFPOOL.controlhigh`				
High water mark for congestion control for receiving messages from the SMTP PC Client (Relay) SSL SMTP listener connection. When the number of messages in queue rises above `controlhigh`, a message is recorded to the application log, and interfaces that you have specified with the `controlinterfaces` setting are shut down. Messages will not be processed out of the queue until you initiate batch processing as described on *page 728*. The closed interfaces reopen when batch processing reduces the number of queued messages below `controllow`. This setting must be at least as high as `controlwarn`. To disable congestion control for the queue processor, set `controllow` and `controlhigh` to 0. NOTE: This setting can be changed through the CLI only if it is currently set to a non-zero value.	INT (0 to INT_MAX)	1600	0	yes

m2g.properties Settings (Part 65 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
`fe_auth/EBFPOOL.timeout`				
After sending a command to the target server over a particular connection from the pool, the maximum time to wait for a response from the target server. If the response is not received within the timeout period, the EBF authorization processing server terminates the connection. To disable timeout limit, set to 0s.	TIME (0s to 1 hour)	15s	0	yes
`fe_auth/EBFPOOL.maxinuse`				
Maximum number of connections the front end authorization connection pool allows to be in use simultaneously. Unless specified differently, this setting defaults to equal your `maxconnections` setting.	INT (0 to INT_MAX)	1600	0	yes
`fe_auth/EBFPOOL.maxspare`				
Tool for managing the number of idle connections in the pool. After using a connection to complete a session with the target server, the Front End EBF authorization processing server either: ◆ Closes the connection if the current number of idle connections in the pool is greater than or equal to `maxspare` ◆ Puts the connection back into the connection pool if the current number of idle connections in the pool is less than `maxspare`. To disable the maximum spare connections limit, set this parameter to 0.	`min-spare` to `max-connections`	1500	0	yes

m2g.properties Settings (Part 66 of 86)

Property	Valid	File	Internal	CLI
Description	Range	Default	Default	Set
`fe_auth/EBFPOOL.minspare`				
Tool for managing the number of idle connections in the Front End EBF authorization pool. At your specified keep-alive period for idle connections, the server either: ◆ Closes the connection, if the current number of idle connections in the pool is greater than `minspare`; or ◆ Sends a keep-alive signal through the connection, if the current number of idle connections in the pool is less than or equal to `minspare`. If the keep-alive test succeeds, the connection remains open and in the pool; if the test fails, the connection is closed. To disable the minimum spare connections limit, set this parameter to 0.	0 to `max-spare`	5	0	yes
`fe_auth/EBFPOOL.maxconnections`				
Maximum number of connections allowed for the front end authorization EBF connection pool	INT (0 to INT_MAX)	1600	null	yes
`fe_auth/EBFPOOL.maxuses`				
Maximum number of times to reuse a connection from the front end authorization EBF connection pool before closing it. To disable the maximum reuse limit, set this parameter to 0.	INT (0 to 1000)	1000	0	yes
`fe_auth/EBFPOOL.keepalive`				
For idle connections in the FE connection pool, the periodic interval at which to either: ◆ Close the connection, if the current number of idle connections in the pool is greater than `minspare`; or Send a keep-alive signal (for example, a NOOP) through the connection, if the current number of idle connections in the pool is equal to or less than `minspare`. If the keep-alive test succeeds, the connection remains open and in the pool; if the test fails, the connection is closed.	TIME (0 to 1 hr)	30	1m	yes

m2g.properties Settings (Part 67 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
`fe_auth/EBFPOOL.rebinddelay`				
In the event that a rebind command is issued for this connection pool through theCLI, the interval between the execution of the unbind action and the subsequent bind action. IMPORTANT: This value should be larger than your setting for `keepalive` to ensure that all connections are properly unbound.	TIME (0 to 1hr)	31	5m	yes
`fe_auth/EBFPOOL.maxbadconnections`				
If this number of connections are rejected or die is reached, the incoming interface is shutdown. Zero (0) disables this health monitor.	INT	800	0	yes
`fe_auth/EBFPOOL.controlinterfaces`				
For internal use only. Do not change. Default = `mta/SMTPSVR,dcm/SMTPSVR,pcc/SMTPSVR,pcc/SMTPSSLSVR,pcc/POPSVR,pcc/POPSSLSVR`	TEXT	See description	null	no
`fe_auth/EBFPOOL.controlhigh`				
High water mark for congestion control for receiving messages from the SMTP PC Client (Relay) SSL SMTP listener connection. When the number of messages in queue rises above `controlhigh`, a message is recorded to the application log, and interfaces that you have specified with the `controlinterfaces` setting are shut down. Messages will not be processed out of the queue until you initiate batch processing as described on *page 728*. The closed interfaces reopen when batch processing reduces the number of queued messages below `controllow`. This setting must be at least as high as `controlwarn`. To disable congestion control for the queue processor, set `controllow` and `controlhigh` to 0. NOTE: This setting can be changed through the CLI only if it is currently set to a non-zero value.	INT (0 to INT_MAX)	1600	45	yes

m2g.properties Settings (Part 68 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
`fe_jobq/EBFPOOL.timeout`				
After sending a command to the target server over a particular connection from the pool, the maximum time to wait for a response from the target server. If the response is not received within the timeout period, the EBF job queue processing server terminates the connection. To disable timeout limit, set to 0s.	TIME (0s to 1 hour)	15s	0	yes
`fe_jobq/EBFPOOL.maxinuse`				
Maximum number of connections from the front end job queue EBF connection pool to allow to be in use simultaneously. Unless specified differently, this setting defaults to equal your `maxconnections` setting.	INT (0 to INT_MAX)	1600	0	yes
`fe_jobq/EBFPOOL.maxspare`				
Tool for managing the number of idle connections in the pool. After using a connection to complete a session with the target server, the Front End EBF job queue processing server either: ◆ Closes the connection if the current number of idle connections in the pool is greater than or equal to `maxspare` ◆ Puts the connection back into the connection pool if the current number of idle connections in the pool is less than `maxspare`. To disable the maximum spare connections limit, set this parameter to 0.	min-spare to max-connect-ions	1500	0	yes

m2g.properties Settings (Part 69 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
`fe_jobq/EBFPOOL.minspare`				
Tool for managing the number of idle connections in the Front End EBF job queue pool. At your specified keep-alive period for idle connections, the server either: ◆ Closes the connection, if the current number of idle connections in the pool is greater than `minspare`; or ◆ Sends a keep-alive signal through the connection, if the current number of idle connections in the pool is less than or equal to `minspare`. If the keep-alive test succeeds, the connection remains open and in the pool; if the test fails, the connection is closed. To disable the minimum spare connections limit, set this parameter to 0.	0 to `max-spare`	5	0	yes
`fe_jobq/EBFPOOL.maxconnections`				
Maximum number of connections allowed for the front end job queue EBF connection pool	INT (0 to INT_MAX)	1600	null	yes
`fe_jobq/EBFPOOL.maxuses`				
Maximum number of times to reuse a connection from thefront end job queue EBF connection pool before closing it. To disable the maximum reuse limit, set this parameter to 0.	INT (0 to 1000)	1000	0	yes
`fe_jobq/EBFPOOL.keepalive`				
For idle connections in the FE connection pool, the periodic interval at which to either: ◆ Close the connection, if the current number of idle connections in the pool is greater than `minspare`; or Send a keep-alive signal (for example, a NOOP) through the connection, if the current number of idle connections in the pool is equal to or less than `minspare`. If the keep-alive test succeeds, the connection remains open and in the pool; if the test fails, the connection is closed.	TIME (0 to 1hr)	30	1m	yes

m2g.properties Settings (Part 70 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
`fe_jobq/EBFPOOL.rebinddelay`				
In the event that a rebind command is issued for this connection pool through theCLI, the interval between the execution of the unbind action and the subsequent bind action. IMPORTANT: This value should be larger than your setting for `keepalive` to ensure that all connections are properly unbound.	TIME 90 to 1hr)	31	5m	yes
`fe_jobq/EBFPOOL.maxbadconnections`				
If this number of connections are rejected or die is reached, the incoming interface is shutdown. Zero (0) disables this health monitor.	INT	800	0	yes
`fe_jobq/EBFPOOL.controlinterfaces`				
For internal use only. Do not change. Default = `mta/SMTPSVR,dcm/SMTPSVR,pcc/SMTPSVR,pcc/SMTPSSLSVR,pcc/POPSVR,pcc/POPSSLSVR`	TEXT	see description	null	no
`fe_jobq/EBFPOOL.controlhigh`				
High water mark for congestion control for receiving messages from the SMTP PC Client (Relay) SSL SMTP listener connection. When the number of messages in queue rises above `controlhigh`, a message is recorded to the application log, and interfaces that you have specified with the `controlinterfaces` setting are shut down. Messages will not be processed out of the queue until you initiate batch processing as described on *page 728*. The closed interfaces reopen when batch processing reduces the number of queued messages below `controllow`. This setting must be at least as high as `controlwarn`. To disable congestion control for the queue processor, set `controllow` and `controlhigh` to 0. NOTE: This setting can be changed through the CLI only if it is currently set to a non-zero value.	INT (0 to INT_MAX)	1600	45	yes

m2g.properties Settings (Part 71 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
`SENDQTHREADMGR.expiredmessage`				
Text of the message to sender used when unable to deliver the message. Default = `"Unable to deliver message"`	TEXT	see description	see description	no
`SENDQTHREADMGR.processerrormessage`				
Message to sender when there is a processing error. Default = `"System error when attempt to deliver your message"`	TEXT	see description	see description	no
`SENDQTHREADMGR.quotaerrormessage`				
Message sent to sender when quota limit has been reached. Default = `"Message quota limit exceeded"`	TEXT	see description	see description	no
`SENDQTHREADMGR.maxretries`				
Maximum number of retries for sending outgoing message.	INT(0 to INT_MAX)	3	3	no
`SENDQTHREADMGR.waittime`				
Wait time between succeeding retries.	TIME	2000	100 millis	no
`SENDQTHREADMGR.rotatetime`				
Timing out the queue entries.	INT	100000s	100s	no
`SENDQTHREADMGR.numthreads`				
Number of threads for the send queue.	INT	50	2	yes

m2g.properties Settings (Part 72 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
`SENDQTHREADMGR.postmaster1`				
Configured domain name for goo. Default = postmaster1@localhost.localdomain	TEXT	see description	see description	no
`SENDQTHREADMGR.postmaster2`				
Configured domain name for red. Default = postmaster2@localhost.localdomain	TEXT	see description	see description	no
`SENDQTHREADMGR.removeheaders`				
Comma separated list of headers to be removed before sending message. Default = `Return-Path,X-GMT-Sender-Operator, BCC`	TEXT	see description	`Return-Path`	no
`SENDQTHREADMGR.bounceonconvfailure`				
For non-relay messages when message formatting fails, a bounce message is generated if `bounceonconvfailure` is set to true. The default is not to generate abounce message to sender when outgoing message formatting fails.	BOOL	false	false	no
`DCMSENDQTHREADMGR.expiredmessage`				
Expiry message sent to the DoCoMo interface. Default = “Unable to deliver message”	TEXT	see description	see description	no
`DCMSENDQTHREADMGR.processerrormessage`				
Processing error message sent to the DoCoMo interface. Default = `"System error when attempt to deliver your message"`	TEXT	see description	see description	no

m2g.properties Settings (Part 73 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
`DCMSENDQTHREADMGR.quotaerrormessage`				
Quota error message sent to the DoCoMo interface. Default = `"Message quota limit exceeded"`	TEXT	see description	see description	no
`DCMSENDQTHREADMGR.maxretries`				
Maximum number of retries for sending outgoing messages to DoCoMo.	INT(0 to INT_MAX)	3	3	no
`DCMSENDQTHREADMGR.waittime`				
Wait time between succeeding retries for this queue.	TIME	2000	100 millis	no
`DCMSENDQTHREADMGR.rotatetime`				
Maximum rotate time for sendingDoCoMo queue messages.	TIME	100000	100s	no
`DCMSENDQTHREADMGR.numthreads`				
Number of threads for the DoCoMo queue.	INT	50	2	yes
`DCMSENDQTHREADMGR.postmaster1`				
Configured domain name for goo. Default = postmaster1@localhost.localdomain	TEXT	see description	see description	no
`DCMSENDQTHREADMGR.postmaster2`				
Configured domain name for red. Default = postmaster1@localhost.localdomain	TEXT	see description	see description	no

m2g.properties Settings (Part 74 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
`DCMSENDQTHREADMGR.removeheaders`				
Remove these headers for messages sent from this queue to DoCoMo recipents. Default = `Return-Path,X-GMT-Sender-Operator, Bcc`	TEXT	see description	see description	no
`DCMSENDQTHREADMGR.bounceonconvfailure`				
For non-relay messages when message formatting fails, a bounce message is generated if `bounceonconvfailure` is set to true. The default is not to generate abounce message to sender when outgoing message formatting fails.	BOOL	false	false	no
`HTTPNOTIFYQMGR.maxretries`				
Maximum number of retries for sending outgoing messages to DCM notification server.	INT(0 to INT_MAX)	3	3	no
`HTTPNOTIFYQMGR.waittime`				
Wait time between succeeding retries for this queue.	TIME	2000	100 millis	no
`HTTPNOTIFYQMGR.rotatetime`				
Maximum rotate time for sending HTTP notification messages.	TIME	100000	100s	no
`HTTPNOTIFYQMGR.numthreads`				
Number of threads for the HTTP notification queue.	INT	50	2	yes
`HTTPNOTIFYQMGR.ab_flag`				
I-mode address flag used in composing the HTTP Request body part.	STRING	`"a"`	`"a"`	no
`HTTPNOTIFYQMGR.mail_no`				
Operation type used in composing the HTTP Request body part.	STRING	`"003"`	`"003"`	no

m2g.properties Settings (Part 75 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
`HTTPNOTIFYPOOL.timeout`				
After sending a command to the target server over a particular connection from the pool, the maximum time to wait for a response from the target server. If the response is not received within the timeout period, the processing server terminates the connection. To disable timeout limit, set to 0s.	TIME 0s to 1h	15s	0	yes
`HTTPNOTIFYPOOL.maxinuse`				
Maximum number of connections from the HTTP Notify connection pool to allow to be in use simultaneously. Unless specified differently, this setting defaults to equal your `maxconnections` setting.	INT (0 to INT_MAX)	1600	0	yes
`HTTPNOTIFYPOOL.maxspare`				
Tool for managing the number of idle connections in the pool. After using a connection to complete a session with the target server, the Front End EBF job queue processing server either: ◆ Closes the connection if the current number of idle connections in the pool is greater than or equal to `maxspare` ◆ Puts the connection back into the connection pool if the current number of idle connections in the pool is less than `maxspare`. To disable the maximum spare connections limit, set this parameter to 0.	min-spare to max-connections	1500	0	yes

m2g.properties Settings (Part 76 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
`HTTPNOTIFYPOOL.minspare`				
Tool for managing the number of idle connections in the HTTP notification pool. At your specified keep-alive period for idle connections, the server either: ◆ Closes the connection, if the current number of idle connections in the pool is greater than `minspare`; or ◆ Sends a keep-alive signal through the connection, if the current number of idle connections in the pool is less than or equal to `minspare`. If the keep-alive test succeeds, the connection remains open and in the pool; if the test fails, the connection is closed. To disable the minimum spare connections limit, set this parameter to 0.	0 to `max-spare`	5	0	yes
`HTTPNOTIFYPOOL.maxconnections`				
Maximum number of connections allowed for the HTTP Notify connection pool.	INT (0 to INT_MAX)	1600	null	yes
`HTTPNOTIFYPOOL.maxuses`				
Maximum number of times to reuse a connection from the HTTP Notify connection pool before closing it. To disable the maximum reuse limit, set this parameter to 0.	INT (0 to 1000)	1	0	yes
`BOUNCEQTHREADMGR.maxretries`				
Maximum number of retries for sending outgoing bounce messages.	INT(0 to INT_MAX)	3	3	no
`BOUNCEQTHREADMGR.waittime`				
Wait time between succeeding retries for this queue.	TIME	2000	100 millis	no
`BOUNCEQTHREADMGR.rotatetime`				
Maximum rotate time for sending bounce messages.	TIME	100000	100s	no

m2g.properties Settings (Part 77 of 86)

Property	Valid	File	Internal	CLI
Description	Range	Default	Default	Set
`BOUNCEQTHREADMGR.numthreads`				
Number of threads for the bounce queue.	INT	50	2	yes
`BOUNCEQTHREADMGR.removeheaders`				
Removethese headers for bounce messages sent from this queue. Default = `Return-Path,X-GMT-Sender-Operator, Bcc`	TEXT	see description	see description	no
`SMTPNOTIFYMGR.maxretries`				
Maximum number of retries for sending outgoing SMTP notify messages.	INT(0 to INT_MAX)	3	3	no
`SMTPNOTIFYMGR.waittime`				
Wait time between succeeding retries for SMTP notification messages.	TIME	2000	100 millis	no
`SMTPNOTIFYMGR.rotatetime`				
Maximum rotate time for sending of SMTP notification messages.	TIME	100000	100s	no
`SMTPNOTIFYMGR.numthreads`				
Number of threads for this queue.	INT	50	2	yes
`SMTPNOTIFYMGR.postmaster1`				
This value is used as the SMTP MAIL FROM command arguent when sending SMTP notification message to red subscribers. Default = postmaster1@localhost.localdomain	TEXT	see description	see description	no

m2g.properties Settings (Part 78 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
`SMTPNOTIFYMGR.postmaster2`				
This value is used as the SMTP MAIL FROM command arguent when sending SMTP notification message to goo subscribers. Default = postmaster1@localhost.localdomain	TEXT	see description	see description	no
`SMTPNOTIFYMGR.postmaster_from_header1`				
From header in the MIME notification message for red. If `postmaster_from_header1` is not specified, then "header from" is not set in the MIME message. The Sender MIME message header is set to the `SMTPNOTIFYMGR.postmaster1` property value, not the value of `SMTPNOTIFYMGR.postmaster_from_header1` property. Default = postmaster1@localhost.localdomain	TEXT	see description	see description	no
`SMTPNOTIFYMGR.postmaster_from_header2`				
From header in the MIME notification message for goo. Also, if `postmaster_from_header2` is not specified, then "header from" is not set in the MIME the message. The Sender MIME message header is set to the `SMTPNOTIFYMGR.postmaster2` property value, not the value of `SMTPNOTIFYMGR.postmaster_from_header2` property. Default = postmaster2@localhost.localdomain	TEXT	see description	see description	no
`SMTPNOTIFYMGR.removeheaders`				
Remove these headers for messages sent through the SMTP notification manager. Default = `Return-Path,X-GMT-Sender-Operator, Bcc`	TEXT	see description	see description	no

m2g.properties Settings (Part 79 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
`MTACONNPOOL.timeout`				
After sending a command to the target server over a particular connection from the pool, the maximum time to wait for a response from the target server. If the response is not received within the timeout period, the EBF job queue processing server terminates the connection. To disable timeout limit, set to 0s.	TIME (0s to 1 hour)	15s	0	yes
`MTACONNPOOL.maxinuse`				
Maximum number of connectionst the MTA connection pool allows to be in use simultaneously. Unless specified differently, this setting defaults to equal your `maxconnections` setting.	INT (0 to INT_MAX)	0	0	yes
`MTACONNPOOL.maxspare`				
Tool for managing the number of idle connections in the pool. After using a connection to complete a session with the target server, the Internet mail MTA pool either: ◆ Closes the connection if the current number of idle connections in the pool is greater than or equal to `maxspare` ◆ Puts the connection back into the connection pool if the current number of idle connections in the pool is less than `maxspare`. To disable the maximum spare connections limit, set this parameter to 0.	`minspare` to `maxconnections`	0	0	yes

m2g.properties Settings (Part 80 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
`MTACONNPOOL.minspare`				
Tool for managing the number of idle connections in the Internet Mail MTA pool. At your specified keep-alive period for idle connections, the server either: ◆ Closes the connection, if the current number of idle connections in the pool is greater than `minspare`; or ◆ Sends a keep-alive signal through the connection, if the current number of idle connections in the pool is less than or equal to `minspare`. If the keep-alive test succeeds, the connection remains open and in the pool; if the test fails, the connection is closed. To disable the minimum spare connections limit, set this parameter to 0.	0 to `max-spare`	0	0	yes
`MTACONNPOOL.maxconnections`				
Maximum number of connections allowed for the MTA connection pool.	INT (0 to INT_MAX)	1600	null	yes
`DCMMTACONNPOOL.timeout`				
After sending a command to the target server over a particular connection from the pool, the maximum time to wait for a response from the target server. If the response is not received within the timeout period, theDocomo MTA connection pool terminates the connection. To disable timeout limit, set to 0s.	0s to 1 hour	100s	0	yes
`DCMMTACONNPOOL.maxinuse`				
Maximum number of connections the Docomo MTA connection pool allows to be in use simultaneously. Unless specified differently, this setting defaults to equal your `maxconnections` setting.	INT (0 to INT_MAX)	0	0	yes

m2g.properties Settings (Part 81 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
DCMMTACONNPOOL.maxspare				
Tool for managing the number of idle connections in the pool. After using a connection to complete a session with the target server, the Docomo MTA connection pool either: ◆ Closes the connection if the current number of idle connections in the pool is greater than or equal to maxspare ◆ Puts the connection back into the connection pool if the current number of idle connections in the pool is less than maxspare. To disable the maximum spare connections limit, set this parameter to 0.	minspare to maxconnections	0	0	yes
DCMMTACONNPOOL.minspare				
Tool for managing the number of idle connections in the Docomo MTA connection pool. At your specified keep-alive period for idle connections, the server either: ◆ Closes the connection, if the current number of idle connections in the pool is greater than minspare; or ◆ Sends a keep-alive signal through the connection, if the current number of idle connections in the pool is less than or equal to minspare. If the keep-alive test succeeds, the connection remains open and in the pool; if the test fails, the connection is closed. To disable the minimum spare connections limit, set this parameter to 0.	0 to maxspare	5	0	yes
DCMMTACONNPOOL.maxconnections				
Maximum number of connections allowed for the Docmo MTA connection pool.	INT	1600	null	yes
REDBOUNCEFORMATTER.reportingmta				
Part of the bounce message, the reporting MTA domain name for red subscribers. Default = red.webmail.nttr.com	TEXT	see description	“”	no

m2g.properties Settings (Part 82 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
`REDBOUNCEFORMATTER.postmaster`				
Domain name used for bounce messages from red formatter. Default = `"<MAILER-DAEMON@red.webmail.nttr.com>"`	TEXT	see description	“”	no
`REDBOUNCEFORMATTER.addoriginalmsg`				
If true, the bounce message includes the original message.	BOOL	true	true	no
`REDBOUNCEFORMATTER.addheadersonly`				
If true the message only includes only the message header.	BOOL	true	true	no
`GOOBOUNCEFORMATTER.reportingmta`				
Part of the bounce message’s reporting MTA name for goo subscribers. Default = `goo.webmail.nttr.com`	TEXT	see description	“”	no
`GOOBOUNCEFORMATTER.postmaster`				
The sender of the bounce message used in the bounce message itself. Default = `"<MAILER-DAEMON@goo.webmail.nttr.com>"`	TEXT	see description	“”	no
`GOOBOUNCEFORMATTER.addoriginalmsg`				
If true, the bounce message includes the original message.	BOOL	true	true	no
`GOOBOUNCEFORMATTER.addheadersonly`				
If true, the bounce message only includes the header.	BOOL	true	true	no
`FETCHQTHREADMGR.timeout`				
Network polling timeout in microseconds.	TIME (0s to 1h)	310	30	no

m2g.properties Settings (Part 83 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
`FETCHQTHREADMGR.txntimeout`				
Transaction timeout in milliseconds.	TIME (0s to 1h)	300	10	no
`FETCHQTHREADMGR.maxrotate`				
The maximum number of rotation for POP fetch jobs in ETQ	TIME	0		
`FETCHQTHREADMGR.maxretrievemessagesize`				
Max number of POP messages which can be retrieve in a single POP session. Default = `2048000`	INT	see description	see description	no
`FETCHQTHREADMGR.maxsize`				
Maximum size of UBF/EBF rpc request. File default = `2050000` Internal default = 128*1024	INT	see description	see description	no
`FETCHQTHREADMGR.config_waittime`				
	TIME	2000	100	yes
`FETCHQTHREADMGR.config_rotatetime`				
Default = 100000	TIME	see description	see description	yes
`FETCHQTHREADMGR.numthreads`				
Number of threads for the fetch queue.	INT	50	2	yes
`FETCHQTHREADMGR.maxlistingmessages`				
Number of threads for the fetch queue.	INT (>0)	10000	10000	yes

m2g.properties Settings (Part 84 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
`FETCHSSLCONF.allownotyetvalidcert`				
Determines if a certificate will be considered valid if it’s validity date is later than the present date.	BOOL	true	true	yes
`FETCHSSLCONF.allowexpiredcert`				
Determines if a certificate will be considered valid if it’s expiry date is earlier than the present date.	BOOL	true	true	
`FETCHSSLCONF.supportsslv2`				
Determines if the connections utilizing sslv2 (a protocol with known security issues) will be allowed.	BOOL	false	false	yes
`FETCHSSLCONF.sslverifydepth`				
Deterimines the number of itermediate certificate authorities that will be allowed for a certificate to be considered valid.	INT	-1	-1	yes
`FETCHSSLCONF.ciphers`				
A list of ciphers that are supported by the installation.	TEXT	ALL	all	no
`FETCHSSLCONF.clientcertverification`				
Determines if a client verifies the server certificate or not.	INT	true	true	no
`FETCHSSLCONF.cacertfile`				
Path to a pem file containing valid certificate authority certificates. Default =`/etc/pki/tls/cert.pem`	TEXT	see description	pem	no
`VERIFYQTHREADMGR.maxretries`				
Maximum number of retries for address verification queue.	INT (0 to 1000)	5	3	no

m2g.properties Settings (Part 85 of 86)

Property Description	Valid Range	File Default	Internal Default	CLI Set
`VERIFYQTHREADMGR.waittime`				
Wait time between succeeding retries for this queue.	TIME	2000	100 millis	no
`VERIFYQTHREADMGR.rotatetime`				
Maximum rotate time for address verification queue.	TIME	`100000`	100s	no
`VERIFYQTHREADMGR.numthreads`				
Number of threads for the address verification queue.	INT	50	2	yes
`VERIFYQTHREADMGR.postmaster1`				
Password Verification Message (address_ext) for the SMTP MAIL From configured domain name for red. Default = `postmaster1@localhost.localdomain`	TEXT	see description	see description	no
`VERIFYQTHREADMGR.postmaster2`				
Password Verification Message (address_ext) for the SMTP MAIL From configured domain name for goo. Default= `postmaster2@localhost.localdomain`	TEXT	see description	see description	no
`VERIFYQTHREADMGR.postmaster_from_header1`				
Password Verification Message (address_ext) for "header from" configured domain name for red. Also, if `postmaster_from_header1` is not specified, then "header from" is not set in the MIME message. The Sender MIME message header is set to the `VERIFYQTHREADMGR.postmaster1` property value, not the value of `VERIFYQTHREADMGR.postmaster_from_header1` property. Default = postmaster1@localhost.localdomain	TEXT	see description	see description	no

m2g.properties Settings (Part 86 of 86)

Property	Valid	File	Internal	CLI
Description	Range	Default	Default	Set
`VERIFYQTHREADMGR.postmaster_from_header2`				
Password Verification Message (address_ext) for "header from" configured domain name for goo. Also, if the `postmaster_from_header2` is not specified, then "header from" is not set in MIME the message. The Sender MIME message header is set to the `VERIFYQTHREADMGR.postmaster2` property value, not the value of `VERIFYQTHREADMGR.postmaster_from_header2` property. Default = postmaster2@localhost.localdomain	TEXT	see description	see description	no
`VERIFYQTHREADMGR.removeheaders`				
Remove this header for messages in the address verification queue. Default = `Return-Path,X-GMT-Sender-Operator, Bcc`	TEXT	see description	see description	no

Note The file default for `alog.format` is:

```
ALOG.format=<PID>|<THREADID>|<DATE-CLF>|<MODULE:%-
24s>|<LEVEL>|<MESSAGECODE>|<MESSAGE>
```

Note The file default for `tlog.format` is:

```
TLOG.format=<PID>|<THREADID>|<DATE-
CLF>|<PROTO>|<TYPE>|<STATUS>|<TRID>|<CLIENT>|<HOST>|<DURATION>
|{Message-ID}|<MESSAGE>|<HTTP_REQ_LINE>|<HTTP_RES_CODE>
```

5 M2G Map Files

This chapter provides information about the settings in the M2G map files including the following:

Note If you want to quickly locate the description of a particular setting that you have seen in the M2G properties file, you can use *Index of Settings in .properties and .conf Files* starting on *page 631*.

au_dcm_jis_map.cfg

Path `<M2G_HOME>/1.0.0/etc/emojimaps/au_dcm_jis_map.cfg`

Purpose This map file provides conversion for AU/Tuka ISO-2022-JP to Docomo UTF-16BE.

Max Entry Lines .65536. Cannot be changed.

Field Delimiter Vertical bar, "|"

Dynamic Reload This file is not dynamically reloadable. To apply changes, you must restart the M2G.

The mapping files are text files. The first line has three fields: operator, replacement character and charset encoding name.

```
<operator>,<replacement character>,<charset_encoding_name>
```

Example

```
au|?|iso-2022-jp
```

The first line tells the char handler component that is map is for AU/TUKA ISO-2022-JP encoding to Docomo UTF-16BE encoding map.

The rest of lines are EMOJI maps; each line is one character.

Important Specify character encodings in hexadecimal format, using the 0x prefix

Example

```
0x7541|0xE63E
0x7546|0xE63F
```

The first column, above, is the operator's EMOJI in indicated encoding (in this example, it is an ISO-2022-JP character). The second column is Docomo's EMOJI character in UTF-16BE.

Example The sample below shows properly formatted entries in the `au_dcm_jis_map.cfg` file.

```
au|?|iso-2022-jp
0x7521|0x3013
0x7522|0xE643
0x7523|0xE66D
0x7524|0xE733
0x7525|0x3013
```

yahoo_dcm_jis_map.cfg

Path `<M2G_HOME>/1.0.0/etc/emojimaps/yahoo_dcm_jis_map.cfg`

Purpose Yahoo ISO-2022-JP to Docomo UTF-16BE.

Max Entry Lines .65536. Cannot be changed.

Field Delimiter Vertical bar, "|"

Dynamic Reload This file is not dynamically reloadable. To apply changes, you must restart the M2G.

The mapping files are text files. The first line has three fields: operator, replacement character and charset encoding name.

```
<operator>,<replacement character>,<charset_encoding_name>
```

Example

```
yahoo|?|iso-2022-jp
```

The first line tells the char handler component that is map is for YAHOO ISO-2022-JP encoding to Docomo UTF-16BE encoding map.

The rest of lines are EMOJI maps; each line is one character.

Important Specify character encodings in hexadecimal format, using the 0x prefix

Example

```
0x7541|0xE63E
0x7546|0xE63F
```

The first column, above, is the operator's EMOJI in indicated encoding (in this example, it is an ISO-2022-JP character). The second column is Docomo's EMOJI character in UTF-16BE.

Example The sample below shows properly formatted entries in the `yahoo_dcm_jis_map.cfg` file.

```
0x7541|0xE63E
0x7546|0xE63F
0x7545|0xE640
0x753E|0xE641
```

sb_dcm_sjis_map.cfg

Path `<M2G_HOME>/1.0.0/etc/emojimaps/sb_dcm_jis_map.cfg`

Purpose SoftBank SHIFT_JIS to Docomo UTF-16BE map.

Max Entry Lines .65536. Cannot be changed.

Field Delimiter Vertical bar, "|".

Dynamic Reload This file is not dynamically reloadable. To apply changes, you must restart the M2G.

The mapping files are text files. The first line has three fields: operator, replacement character and charset encoding name.

```
<operator>,<replacement character>,<charset_encoding_name>
```

Example

```
sbm|?|SHIFT_JIS
```

The first line tells the char handler component that is map is for SoftBank SHIFT_JIS to Docomo UTF-16BE mapping.

The rest of lines are EMOJI maps; each line is one character.

IMPORTANT Specify character encodings in hexadecimal format, using the 0x prefix

Example

```
0xF98B|0xE63E
0xF98A|0xE63F
```

The first column, above, is the operator's EMOJI in indicated encoding (in this example, it is a SoftBank SHIFT_JIS character). The second column is Docomo's EMOJI character in UTF-16BE.

Example The sample below shows properly formatted entries in the `sb_dcm_sjis_map.cfg` file.

```
0xF98B|0xE63E
0xF98A|0xE63F
0xF98C|0xE640
0xF989|0xE641
```

dcm_dcm_sjis_map.cfg

Path `<M2G_HOME>/1.0.0/etc/emojimaps/dcm_dcm_jis_map.cfg`

Purpose Docomo SHIFT_JIS to Docomo UTF-16BE map.

Max Entry Lines .65536. Cannot be changed.

Field Delimiter Vertical bar, “|”

Dynamic Reload This file is not dynamically reloadable. To apply changes, you must restart the M2G.

The mapping files are text files. The first line has three fields: operator, replacement character and charset encoding name.

```
<operator>,<replacement character>,<charset_encoding_name>
```

Example

```
docomo|?|SHIFT_JIS
```

The first line tells the char handler component that is map is for Docomo SHIFT_JIS to Docomo UTF-16BE mapping.

The rest of lines are EMOJI maps; each line is one character.

IMPORTANT Specify character encodings in hexadecimal format, using the 0x prefix

Example

```
0xF89F|0xE63E
0xF8A0|0xE63F
```

The first column, above, is the operator's EMOJI in indicated encoding (in this example, it is a Docmo SHIFT_JIS character). The second column is Docomo's EMOJI character in UTF-16BE.

Example The sample below shows properly formatted entries in the `dcm_dcm_sjis_map.cfg` file.

```
0xF89F|0xE63E
0xF8A0|0xE63F
0xF8A1|0xE640
0xF8A2|0xE641
```

au_dcm_sjis_map.cfg

Path `<M2G_HOME>/1.0.0/etc/emojimaps/au_dcm_sjis_map.cfg`

Purpose #AU/Tuka SHIFT_JIS to Docomo UTF-16BE map.

Max Entry Lines .65536. Cannot be changed.

Field Delimiter Vertical bar, “|”

Dynamic Reload This file is not dynamically reloadable. To apply changes, you must restart the M2G.

The mapping files are text files. The first line has three fields: operator, replacement character and charset encoding name.

```
<operator>,<replacement character>,<charset_encoding_name>
```

Example

```
au|?|SHIFT_JIS
```

The first line tells the char handler component that is map is for Docomo SHIFT_JIS to Docomo UTF-16BE mapping.

The rest of lines are EMOJI maps; each line is one character.

IMPORTANT Specify character encodings in hexadecimal format, using the 0x prefix

Example

```
0xF660|0xE63E
0xF665|0xE63F
```

The first column, above, is the operator's EMOJI in indicated encoding (in this example, it is a AU character). The second column is Docomo's EMOJI character in SHIFT_JIS.

Example The sample below shows properly formatted entries in the `au_dcm_sjis_map.cfg` file.

```
0xF660|0xE63E
0xF665|0xE63F
0xF664|0xE640
0xF65D|0xE641
```

dcm_au_jis_map.cfg

Path `<M2G_HOME>/1.0.0/etc/dcm_au_jis_map.cfg`

Purpose DoCoMo EMOJI Unicode to AU JIS EMOJI mapping.

Max Entry Lines 65536. Cannot be changed.

Field Delimiter Comma . Cannot be changed.

Dynamic Reload This file is not dynamically reloadable. To apply changes, you must restart the M2G.

These Emoji encoding mappings are configurable so that you may modify the files periodically to keep them up to date with the latest Emoji encodings.

For each Emoji character, the map file has a line composed of these comma-separated hexadecimal values:

```
<unicode>,<destination_encoding>,<shift_in>,<shift_out>
```

The table that follows describes the values to include in each line of the map file. The “Position” column indicates the position of each parameter within the entry line. A sample of properly formatted lines follows the table.

IMPORTANT Specify character encodings in hexadecimal format, using the 0x prefix.

Emoji Map Parameters (Part 1 of 2)

Position	Parameter	Description
1	`<unicode>`	Unicode encoding of the Emoji character. This is the encoding in which the message will arrive into the M2G. The maximum key length has no limit. IMPORTANT: The table should have one line for every Unicode Emoji that the M2G may encounter. Otherwise, the M2G will process unlisted Emoji as invalid characters.
2	`<destination_ encoding>`	Destination encoding of the Emoji character. The destination encoding set is ISO-2022-JP. If an Emoji character does not have a representation in the target encoding set, enter something other than a valid hexadecimal in this field (for example, a dash). The M2G will replace these unmappable Emoji characters with your default Emoji replacement character.

Emoji Map Parameters (Part 2 of 2)

Position	Parameter	Description
3	`<shift_in>`	Destination encoding for Shift In from Kanji mode back into ASCII mode.
4	`<shift_out>`	Destination encoding for Shift Out from ASCII mode to Kanji mode.

Example The sample below shows properly formatted entries in the `dcm_au_jis_map.cfg` file.

```
0xE63E,0x7541,0x1b 0x24 0x42,0x1b 0x28 0x42
0xE63F,0x7546,0x1b 0x24 0x42,0x1b 0x28 0x42
0xE640,0x7545,0x1b 0x24 0x42,0x1b 0x28 0x42
0xE641,0x753E,0x1b 0x24 0x42,0x1b 0x28 0x42
```

dcm_au_sjis_map.cfg

Path `<M2G_HOME>/1.0.0/etc/dcm_au_sjis_map.cfg`

Purpose Emoji conversion for outgoing messages to the Internet.

Max Entry Lines 65536. Cannot be changed.

Field Delimiter Comma . Cannot be changed.

Dynamic Reload This file is not dynamically reloadable. To apply changes, you must restart the M2G.

These Emoji encoding mappings are configurable so that you may modify the files periodically to keep them up to date with the latest Emoji encodings.

For each Emoji character, the map file has a line composed of these comma-separated hexadecimal values:

```
<unicode>,<destination_encoding>,<shift_in>,<shift_out>
```

The table that follows describes the values to include in each line of the map file. The “Position” column indicates the position of each parameter within the entry line. A sample of properly formatted lines follows the table.

IMPORTANT When an EMOJI character is mapped to an ASCII string, do not provide values in the `<shift_in>` and `<shift_out>` columns.

IMPORTANT Specify character encodings in hexadecimal format, using the 0x prefix.

Emoji Map Parameters (Part 1 of 2)

Position	Parameter	Description
1	`<unicode>`	Unicode encoding of the Emoji character. This is the encoding in which the message will arrive into the M2G. The maximum key length has no limit. IMPORTANT: The table should have one line for every Unicode Emoji that the M2G may encounter. Otherwise, the M2G will process unlisted Emoji as invalid characters.

Emoji Map Parameters (Part 2 of 2)

Position	Parameter	Description
2	`<destination_ encoding>`	Destination encoding of the Emoji character. The destination encoding set is ISO-2022-JP. If an Emoji character does not have a representation in the target encoding set, enter something other than a valid hexadecimal in this field (for example, a dash). The M2G will replace these unmappable Emoji characters with your default Emoji replacement character.
3	`<shift_in>`	Destination encoding for Shift In from Kanji mode back into ASCII mode.
4	`<shift_out>`	Destination encoding for Shift Out from ASCII mode to Kanji mode.

Example

```
0xE63E,0xF660,-,-
0xE63F,0xF665,-,-
0xE640,0xF664,-,-
0xE641,0xF65D,-,-
```

dcm_emb_jis_map.cfg

Path `<M2G_HOME>/1.0.0/etc/dcm_emb_jis_map.cfg`

Purpose DoCoMo EMOJI Unicode to Emobile JIS EMOJI mapping.

Max Entry Lines 65536. Cannot be changed.

Field Delimiter Comma . Cannot be changed.

Dynamic Reload This file is not dynamically reloadable. To apply changes, you must restart the M2G.

These Emoji encoding mappings are configurable so that you may modify the files periodically to keep them up to date with the latest Emoji encodings.

For each Emoji character, the map file has a line composed of these comma-separated hexadecimal values:

```
<unicode>,<destination_encoding>,<shift_out>,<shift_in>
```

The table that follows describes the values to include in each line of the map file. The “Position” column indicates the position of each parameter within the entry line. A sample of properly formatted lines follows the table.

IMPORTANT When an EMOJI character is mapped to an ASCII string, do not provide values in the `<shift_in>` and `<shift_out>` columns.

IMPORTANT Specify character encodings in hexadecimal format, using the 0x prefix.

Emoji Map Parameters (Part 1 of 2)

Position	Parameter	Description
1	`<unicode>`	Unicode encoding of the Emoji character. This is the encoding in which the message will arrive into the M2G. The maximum key length has no limit. IMPORTANT: The table should have one line for every Unicode Emoji that the M2G may encounter. Otherwise, the M2G will process unlisted Emoji as invalid characters.

Emoji Map Parameters (Part 2 of 2)

Position	Parameter	Description
2	`<destination_ encoding>`	Destination encoding of the Emoji character. The destination encoding set is ISO-2022-JP. If an Emoji character does not have a representation in the target encoding set, enter something other than a valid hexadecimal in this field (for example, a dash). The M2G will replace these unmappable Emoji characters with your default Emoji replacement character.
3	`<shift_out>`	Destination encoding for Shift Out from ASCII mode to Kanji mode.
4	`<shift_in>`	Destination encoding for Shift In from Kanji mode back into ASCII mode.

Example

```
0xE600,0x7761,0x1b 0x24 0x42,0x1b 0x28 0x42
0xE601,0x7762,0x1b 0x24 0x42,0x1b 0x28 0x42
0xE602,0x7763,0x1b 0x24 0x42,0x1b 0x28 0x42
```

dcm_emb_sjis_map.cfg

Path `<M2G_HOME>/1.0.0/etc/dcm_emb_sjis_map.cfg`

Purpose DoCoMo EMOJI Unicode to Emobile Shift_JIS EMOJI mapping.

Max Entry Lines 65536. Cannot be changed.

Field Delimiter Comma . Cannot be changed.

Dynamic Reload This file is not dynamically reloadable. To apply changes, you must restart the M2G.

These Emoji encoding mappings are configurable so that you may modify the files periodically to keep them up to date with the latest Emoji encodings.

For each Emoji character, the map file has a line composed of these comma-separated hexadecimal values:

```
<unicode>,<Shift_JIS>
```

The table that follows describes the values to include in each line of the map file. The “Position” column indicates the position of each parameter within the entry line. A sample of properly formatted lines follows the table.

IMPORTANT Specify character encodings in hexadecimal format, using the 0x prefix.

Emoji Map Parameters

Position	Parameter	Description
1	`<unicode>`	Unicode encoding of the Emoji character. This is the encoding in which the message will arrive into the M2G. The maximum key length has no limit. IMPORTANT: The table should have one line for every Unicode Emoji that the M2G may encounter. Otherwise, the M2G will process unlisted Emoji as invalid characters.
2	`<destination_ encoding>`	Destination encoding of the Emoji character. The destination encoding set is Shift_JIS. If an Emoji character does not have a representation in the target encoding set, enter something other than a valid hexadecimal in this field (for example, a dash). The M2G will replace these unmappable Emoji characters with your default Emoji replacement character.

Example

```
0xE600,0xF860,-,-
0xE601,0xF861,-,-
0xE602,0xF862,-,-
0xE603,0xF863,-,-
```

dcm_emojitable.cfg

Path `<M2G_HOME>/1.0.0/etc/dcm_emojitable.cfg`

Purpose This table contains a list of DoCoMo EMOJI characters in UTF-16BE encoding. This table is used to identify EMOJI characters and replace them with some replacement string.

Max Entry Lines .65536. Cannot be changed.

Field Delimiter Comma

IMPORTANT All keys in this mapfile must be lowercase only.

Dynamic Reload This file is not dynamically reloadable. To apply changes, you must restart the M2G.

This Emoji encoding map file is configurable so that you may modify the files periodically to keep them up to date with the latest Emoji encodings.

For each Emoji character, the map file has a line composed of these comma-separated hexadecimal values:

```
<unicode>,<invalid hexadecimal>
```

The table that follows describes the values to include in each line of the map file. The “Position” column indicates the position of each parameter within the entry line. A sample of properly formatted lines follows the table.

IMPORTANT Specify character encodings in hexadecimal format, using the 0x prefix.

_emojitable.cfg Parameters

Position	Parameter	Description
1	`<unicode>`	DoCoMo EMOJI characters in UTF-16BE encoding.
2	`<invalid hexadecimal>`	Any invalid hexadecimal character that cannot be converted to hex number.

Example

```
0xE63E,-,-,-
0xE63F,-,-,-
0xE640,-,-,-
```

dcm_emojiurltable.cfg

Path `<M2G_HOME>/1.0.0/etc/dcm_emojiurltable.cfg`

Purpose This table contains list of DoCoMo EMOJI characters and URLs.

Max Entry Lines 65536. Cannot be changed.

Field Delimiter Vertical bar, "|"

Dynamic Reload This file is not dynamically reloadable. To apply changes, you must restart the M2G.

These Emoji encoding mappings are configurable so that you may modify the files periodically to keep them up to date with the latest Emoji encodings.

For each Emoji character, the map file has a line composed of these comma-separated hexadecimal values:

```
<unicode>,<image_name>,<shift_out>,<shift_in>
```

The table that follows describes the values to include in each line of the map file. The "Position" column indicates the position of each parameter within the entry line. A sample of properly formatted lines follows the table.

IMPORTANT Specify character encodings in hexadecimal format, using the 0x prefix.

Emoji Map Parameters

Position	Parameter	Description
1	`<unicode>`	Unicode encoding of the Emoji character. This is the encoding in which the message will arrive into the M2G. The maximum key length has no limit. IMPORTANT: The table should have one line for every Unicode Emoji that the M2G may encounter. Otherwise, the M2G will process unlisted Emoji as invalid characters.
2	`<image_name>`	URL of the EMOJI character graphics image file.
3	`<shift_out>`	If destination charset is ISO-2022-JP, escape sequence should be prepended.
4	`<shift_in>`	If destination charset is ISO-2022-JP, escape sequence may need to be appended.

Example

```
0xE63E|<br /><img style="margin-left: 1px;" src="http://
i.yimg.jp/i/mesg/tsmileys2/001.gif" width="18"
height="18">|0x1b 0x28 0x42|
0xE63F|<br /><img style="margin-left: 1px;" src="http://
i.yimg.jp/i/mesg/tsmileys2/002.gif" width="18"
height="18">|0x1b 0x28 0x42|

0xE640|<br /><img style="margin-left: 1px;" src="http://
i.yimg.jp/i/mesg/tsmileys2/003.gif" width="18"
height="18">|0x1b 0x28 0x42|
```

dcm_sb_sjis_map.cfg

Path `<M2G_HOME>/1.0.0/etc/dcm_sb_sjis_map.cfg`

Purpose DoCoMo EMOJI Unicode to SoftBank PDC mapping.

Max Entry Lines 65536. Cannot be changed.

Field Delimiter Comma . Cannot be changed.

Dynamic Reload This file is not dynamically reloadable. To apply changes, you must restart the M2G.

These Emoji encoding mappings are configurable so that you may modify the files periodically to keep them up to date with the latest Emoji encodings.

For each Emoji character, the map file has a line composed of these comma-separated hexadecimal values:

```
<unicode>,<destination_encoding>,<shift_out>,<shift_in>
```

The table that follows describes the values to include in each line of the map file. The “Position” column indicates the position of each parameter within the entry line. A sample of properly formatted lines follows the table.

IMPORTANT Specify character encodings in hexadecimal format, using the 0x prefix.

Emoji Map Parameters

Position	Parameter	Description
1	`<unicode>`	UTF-16BE in hexidecimal.
2	`<destination_encoding>`	SBM JIS code in hexdecimal
3	`<shift_out>`	Shift-out in hexidecimal.
4	`<shift_in>`	Shift-in in hexidecimal.

Example

```
0xE63E,0x6A,0x1b 0x24 0x47,0x0F
0xE63F,0x69,0x1b 0x24 0x47,0x0F
0xE640,0x6B,0x1b 0x24 0x47,0x0F
0xE641,0x68,0x1b 0x24 0x47,0x0F
```

dcm_sjis_map.cfg

Path `<M2G_HOME>/1.0.0/etc/dcm_sjis_map.cfg`

Purpose Docomo Unicode to Docomo Shift_JIS mapping.

Max Entry Lines 65536. Cannot be changed.

Field Delimiter Comma . Cannot be changed.

Dynamic Reload This file is not dynamically reloadable. To apply changes, you must restart the M2G.

These Emoji encoding mappings are configurable so that you may modify the files periodically to keep them up to date with the latest Emoji encodings.

For each Emoji character, the map file has a line composed of these comma-separated hexadecimal values:

```
<unicode>,<destination_encoding>,<shift_out>,<shift_in>
```

The table that follows describes the values to include in each line of the map file. The “Position” column indicates the position of each parameter within the entry line. A sample of properly formatted lines follows the table.

IMPORTANT Specify character encodings in hexadecimal format, using the 0x prefix.

Emoji Map Parameters

Position	Parameter	Description
1	`<unicode>`	UTF-16BE in hexidecimal.
2	`<destination_ encoding>`	Docomo JIS code in hexdecimal
3	`<shift_out>`	Shift-out in hexidecimal.
4	`<shift_in>`	Shift-in in hexidecimal.

Example

```
0xE63E,0xF89F
0xE63F,0xF8A0
0xE640,0xF8A1
0xE641,0xF8A2
```

dcm_sb_jis_map.cfg

Path `<M2G_HOME>/1.0.0/etc/dcm_sb_jis_map.cfg`

Purpose DoCoMo EMOJI Unicode to SoftBank PDC mapping.

Max Entry Lines 65536. Cannot be changed.

Field Delimiter Comma . Cannot be changed.

Dynamic Reload This file is not dynamically reloadable. To apply changes, you must restart the M2G.

These Emoji encoding mappings are configurable so that you may modify the files periodically to keep them up to date with the latest Emoji encodings.

For each Emoji character, the map file has a line composed of these comma-separated hexadecimal values:

```
<unicode>,<destination_encoding>,<shift_out>,<shift_in>
```

The table that follows describes the values to include in each line of the map file. The “Position” column indicates the position of each parameter within the entry line. A sample of properly formatted lines follows the table.

IMPORTANT Specify character encodings in hexadecimal format, using the 0x prefix.

Emoji Map Parameters

Position	Parameter	Description
1	`<unicode>`	UTF-16BE in hexidecimal.
2	`<destination_ encoding>`	SBM JIS code in hexdecimal
3	`<shift_out>`	Shift-out in hexidecimal.
4	`<shift_in>`	Shift-in in hexidecimal.

Example

```
0xE63E,0x6A,0x1b 0x24 0x47,0x0F
0xE63F,0x69,0x1b 0x24 0x47,0x0F
0xE640,0x6B,0x1b 0x24 0x47,0x0F
```

dcm_yahoo_jis_map.cfg

Path `<M2G_HOME>/1.0.0/etc/dcm_yahoo_jis_map.cfg`

Purpose DoCoMo to Yahoo JIS mapping.

Max Entry Lines 65536. Cannot be changed.

Field Delimiter Comma . Cannot be changed.

Dynamic Reload This file is not dynamically reloadable. To apply changes, you must restart the M2G.

These Emoji encoding mappings are configurable so that you may modify the files periodically to keep them up to date with the latest Emoji encodings.

For each Emoji character, the map file has a line composed of these comma-separated hexadecimal values:

```
<unicode>,<destination_encoding>,<shift_out>,<shift_in>
```

The table that follows describes the values to include in each line of the map file. The “Position” column indicates the position of each parameter within the entry line. A sample of properly formatted lines follows the table.

IMPORTANT Specify character encodings in hexadecimal format, using the 0x prefix.

IMPORTANT When an EMOJI character is mapped to an ASCII string, do not provide values in the `<shift_in>` and `<shift_out>` columns.

Emoji Map Parameters

Position	Parameter	Description
1	`<unicode>`	UTF-16BE in hexidecimal.
2	`<destination_ encoding>`	Yahoo JIS code in hexdecimal
3	`<shift_out>`	Shift-out in hexidecimal.
4	`<shift_in>`	Shift-in in hexidecimal.

Example

```
0xE63E,0x7541,0x1b 0x24 0x42,0x1b 0x28 0x42
0xE63F,0x7546,0x1b 0x24 0x42,0x1b 0x28 0x42
0xE640,0x7545,0x1b 0x24 0x42,0x1b 0x28 0x42
```

errorcode.cfg

Path `<M2G_HOME>/etc/errorcode.cfg`

Purpose Configures customized log message severity levels. For overview see *Configuring Application Logging, on page 72*.

Type Map file.

Max Entry Lines 250. Limit cannot be changed.

Field Delimiter Space. Delimiter cannot be changed.

Dynamic Reload You can dynamically reload this file with this CLI command:

```
   reload ALOG.codemapfile
```

Use the map file `errorcode.cfg` if you want to customize the severity levels assigned to messages that the M2G server writes to its application log. Gemini Mobile Technologies assigns each application log message that the server can generate a default severity level of either ALERT, WARNG, INFO, or DEBUG. These per-message default severity levels are documented in *Appendix d, MGS Application Log Messages* starting on *page 513*. If for some messages you would prefer a severity level different than the default, you can use the `errorcode.cfg` file to make this customization.

For example, if you are repeatedly seeing in your logs a particular INFO level message that you would prefer to deprecate to DEBUG level, you can do so using the `errorcode.cfg` file. Likewise, if you would prefer that a WARNG level message that you have seen in your logs be elevated to ALERT level, you can use `errorcode.cfg` to make this change.

For each message that you want to assign a severity level different than its default, enter into the map file a line composed of these space-separated values:

```
<errorcode> <errorlevel>
```

The table that follows describes each of these parameters. A sample of properly formatted lines follows the table.

errorcode.cfg Parameters

Parameter	Description
`<errorcode>`	Message code of the message that you want to assign a non-default severity level.
`<errorlevel>`	Severity level that you want to assign to the message. The valid options are: ◆ `ALERT` ◆ `WARNG` ◆ `INFO` ◆ `DEBUG` ◆ `NONE` NOTE: Assign level `NONE` if you do not want the message to be logged at all.

Example The sample below shows properly formatted entries for the `errorcode.cfg` file.

```
0030001 WARNG
0880004 ALERT
```

eventname.cfg

Path `<M2G_HOME>/etc/eventname.cfg`

Purpose Configures customized log severity levels for HRE system events. For overview see *Configuring Application Logging, on page 72*.

Type Map file.

Max Entry Lines 250. Limit cannot be changed.

Field Delimiter Space. Delimiter cannot be changed.

Dynamic Reload You can dynamically reload this file with this CLI command:

```
reload ALOG.eventmapfile
```

The M2G server writes application log entries with message codes 0090003 or 0090004 to record the occurrence of certain HyperScale Runtime Environment (HRE) internal system events. There are multiple HRE events that may trigger the writing of one of these log entries. The specific event that triggered a particular log entry is identified in the `<MESSAGE>` field of the entry.

By default 0090003 and 0090004 log entries are assigned a severity level of "INFO". With the `eventname.cfg` file you can customize the severity level of the log entry according to the particular HRE system event being handled. For instance, if you want to deprecate the log entries associated with certain system events to DEBUG level, you can do so using the `eventname.cfg` file.

For each HRE event that you want to assign a customized severity level, enter into the map file a line composed of these space-separated values:

```
<event_name> <severity_level>
```

The table on the next page describes each of these parameters. A sample of a properly formatted line follows the table.

Note To assign *all* the HRE system events listed in the table the *same* customized logging level, it is easier to use the `errorcode.cfg` file (*page 183*). In that file you can assign a logging level to code 0090003 and to code 0090004. That level will be applied to all HRE system events listed in the table, with the exception of any events that you set in the `eventname.cfg` file. Settings in the `eventname.cfg` file supersede those in the `errorcode.cfg` file. For example, you might use the `errorcode.cfg` file to set codes 0090003 and 0090004 to DEBUG, and then use `eventname.cfg` to keep one or two particular events at INFO level.

eventname.cfg Parameters

Parameter	Description
`<event_name>`	Name of HRE event to which you want to assign a custom severity level. Options are: ◆ `ALERT_STATE_CHANGE` ◆ `ARRAY_RELOAD` ◆ `FORK` ◆ `MAP_RELOAD` ◆ `PID` ◆ `POSTFORK` ◆ `POSTPID` ◆ `PREFORK` ◆ `SHUTDOWN` ◆ `SHUTDOWN2` ◆ `SIGHUP` ◆ `START_QUEUE` ◆ `START_QUEUETHREADS` ◆ `STATMGR_SAMPLE`
`<severity_level>`	Severity level that you want to assign to the event. The valid options are: ◆ `ALERT` ◆ `WARNG` ◆ `INFO` ◆ `DEBUG` ◆ `NONE` NOTE: Assign level `NONE` if you do not want the event to be logged at all. NOTE: Since all HRE events are assigned the INFO level by default, the only reason to use the INFO level in the `eventname.cfg` file is if you have changed the level of the 0090003 and 0090004 message codes to a level other than INFO in the `errorcode.cfg` file. In that scenario you could use the `eventname.cfg` file to keep certain events at the INFO level.

Example

The sample below shows properly formatted entries for the `eventname.cfg` file. The event `STATMGR_SAMPLE` is a good candidate for deprecation to DEBUG level because it occurs frequently.

```
STATMGR_SAMPLE DEBUG
```

hosts

Path `<M2G_HOME>/etc/hosts`

Purpose Maps M2G and partner server host names to IP addresses.

Max Entry Lines. Unlimited.

Field Delimiter. Space. Delimiter cannot be changed.

Dynamic Reload. You cannot dynamically reload this file. To implement changes that you have made to this file, you must restart the M2G.

For each M2server, and for each of your other network servers with which the M2G will interact, enter into the `<M2G_HOME>/etc/hosts` file a line composed of these space-separated values:

```
<ip_address> <hostname>
```

The table on the next page describes each of these parameters. A sample of properly formatted lines follows the table.

hosts Parameters

Parameter	Description
`<ip_address>`	IP address.
`<hostname>`	Host name of an M2G server or another server with which the M2G interacts. In the default hosts file, each server type is listed in these two notations: ◆ `<base_hostname>` This host name represents a load balancer sitting in front of multiple nodes of the server type denoted by the `<base_hostname>`, if applicable. ◆ `<base_hostname>%n` This host name entry is listed to show you how to expand the host name if you have multiple nodes of the server type denoted by the `<base_hostname>`. If you have multiple nodes for this server type, you will need one entry for each node, with the `%n` in each entry replaced by a node number: for example, `<base_hostname>0`, `<base_hostname>1`, `<base_hostname>2`, and so on. The "`<base_hostname>%n`" entry itself can be commented out or removed when you are done. Default base host names are listed below.

Example The sample below shows properly formatted entries for the `hosts` file. The first entry denotes a CLI server. The second line lists an Internet SMTP server. The third line shows the Docomo SMTP server. The fourth line lists thePC Client POP3 server.

```
0.0.0.0 server-i0 #@
0.0.0.0 smtpsvr-u0 #@
0.0.0.0 dcmsmtpsvr-u0 #@
0.0.0.0 pccsmtpsvr-u0 #
```

imagetransformmap.cfg

Path `<M2G_HOME>/etc/imagetransformmap.cfg`

Purpose For each destination domain whether to apply image transformation, delete certain types of images or allow certain types of images to pass through.

Max Entry Lines Unlimited.

Field Delimiter Vertical bar, "|"

Dynamic Reload You cannot dynamically reload this file. To implement changes that you have made to this file, you must restart the M2G.

You can configure different image transformation for different domains.

For each expected destination domain or domain pattern for which you want to configure message conversions, enter into the map file a line composed of these space-separated values:

```
[<match_type>] <domain> <max_size> <delete_contents>;
<resize_contents> <lists>
```

The following table describes each of these parameters. A sample of properly formatted lines follows the table

Position	Parameter	Description
0	`<domain>`	Sender's domain name.
1	`<max_size>`	The maximum size to allow.
2	`<delete_ contents>`	List of content types to delete. The format is content_type2;content_type2...
3	`<resize_ contents>`	List of content types that will be resized. The format is content_type : resolution : depth : quality ;
4	`<lists>`	Lists are ';' separated content-types. Resize options follow a content-type with ':'

Example The sample below shows properly formatted entries for the `imagetransformmap.cfg` file.

```
EXACT test.nttr.moc|2048000|image/x-wbmp;image/x-gif|image/
jpeg:640x480:8:75
```

mail_transcoding.cfg

Path `<M2G_HOME>/1.0.0/etc/mail_transcoding.cfg`

Purpose Map file for outgoing SMTP transcoding.

Max Entry Lines 100.

Field Delimiter Space. Cannot be changed.

Dynamic Reload You can dynamically reload this file with this CLI command:

```
reload MAIL_TRANSCODING.mapfile
```

To configure message conversions for destination domains, use the map file `mail_transcoding.cfg`. You can configure different conversions for different domains. The configurable conversions include character set encoding, MIME encoding, and header filtering.

For each expected destination domain or domain pattern for which you want to configure message conversions, enter into the map file a line composed of these space-separated values:

```
[<match_type>] <domain_pattern> <displayname>
<usesmilconvert> <usetextconversion> <hdrcharset>
<hdrmimeenc> <textcharset> <textmimeenc>
<textattachmentcharset> <textattachmentmimeenc>
<maxlinelength> <nontext_mimeenc> <imageresolution>
```

The following table describes each of these parameters. A sample of properly formatted lines follows the table

Position	Parameter	Description
0	`<domain_ pattern>`	A FQDN or IP_address specified as an EXACT|SUBDOMAIN|REGEX type.
1	`<displayname>`	A true/false value specifying whether or not to keep display name in email address.
2	`<usesmilconvert>`	A true/false value specifying whether to perform SMIL conversion.
3	`<usetextconverion>`	A true/false value specifying whether to perform charset conversion.
4	`<hdrcharset>`	The target charset for RFC 2047 encoding of headers

Position	Parameter	Description
5	<hdrmimeenc>	The target MIME encoding for RFC 2047 encoding of headers
6	<textcharset>	The target charset for text body.
7	<textmimeenc>	The MIME transfer encoding to use. e.g., Base64 for text body.
8	<textattachmentcharset>	The target charset for attachment text body.
9	<textattachmentmimeenc>	The MIME transfer encoding to use. e.g., Base64 for attachment text body.
10	<maxlinelength>	Maximum line length in octets of text body.
11	<nontextmimeenc>	The MIME transfer encoding to use. e.g., Base64 for non-text body.
12	<image resolution>	Resolution of the image.

Example

The sample below shows properly formatted entries for the `mail_transcoding.cfg` file.

```
EXACT test.nttr.moc TRUE TRUE TRUE UTF-8 Base64 UTF-8 8Bit UTF-
8 Base64 1000 Base64 DOCOMOMAP GETA 384x256

DEFAULT TRUE TRUE TRUE UTF-8 Base64 UTF-8 8Bit UTF-8 Base64
1000 Base64 128x128
```

map_error_text.cfg

Path `<M2G_HOME>/1.0.0/etc/map_error_text.cfg`

Purpose Entries in this map file are mail_add error code to description mapping. The description may be used in bounce message system notice.

Max Entry Lines 100. Cannot be changed.

IMPORTANT All keys in this mapfile must be lowercase only.

Field Delimiter Vertical bar. Cannot be changed.

Dynamic Reload You can dynamically reload this file with this CLI command:

```
reload ERRORTEXTMAP.mapfile
```

For each domain, enter into the `map_error_text.cfg` file a line composed of these vertical bar-separated values:

```
[<match_type>]|<mail_add_code>|<description>
```

The table that follows describes the values to include in each line of the map file. The "Position" column indicates the position of each parameter within the entry line. A sample of properly formatted lines follows the table.

customcharhandlermap.cfg Parameters (Part 1 of 2)

Position	Parameter	Description
0	[<match_type>]	The type of matching to implement when the M2G checks domain names for matches against the `<domain_pattern>`. Options are: ◆ `EXACT` ◆ `SUBDOMAIN` ◆ `REGEX` ◆ `DEFAULT` If you specify a match type, separate it from the `<domain_pattern>` with a space. If you do not specify a match type for a particular line in the map file, then EXACT matching is implemented for that line. For further information on match types, see *page 39*.

***customcharhandlermap.cfg* Parameters (Part 2 of 2)**

Position	Parameter	Description
1	`<mail_add_code>`	Mail add code from the following list: ◆ unauthorized ◆ badarg ◆ system_down ◆ system_limit ◆ timeout ◆ retrylater ◆ quota_limit ◆ no_folder_exists ◆ no_key_exists
2	`<description>`	◆ Text description, in quotes, of the error code added. Quoted text can also be added for DEFAULT to dictate the handling of cases that do not match against any other of your entries in the map file.

Example Below is an example of a correctly formatted `map_error_text.cfg` file.

```
EXACT|quota_limit|"Message quota limit exceeded"
EXACT|no_folder_exists|"Mail box does not exist"
EXACT|no_key_exists|"System error"
EXACT|unauthorized|"Recipient unauthorized on system"
EXACT|system_down|"Mail system down"
EXACT|system_limit|"Mail system limit reached"
EXACT|timeout|"Time out error occurred"
EXACT|retrylater|"System busy"
DEFAULT "Unknown error"
```

mailhosts

Path `<M2G_HOME>/etc/mailhosts`

Purpose Specifies the MX record IP address/hostname, port number, and domain for each mail server which M2G sends mail to. Each entry in this file eliminates an MXDNS record lookup by M2G when it sends mail.

Max Entry Lines Unlimited.

Field Delimiter Space. Delimiter cannot be changed.

Dynamic Reload You cannot dynamically reload this file. To implement changes that you have made to this file, you must restart the M2G.

For each of the mail servers for which M2G will send mail to, enter into the `<M2G_HOME>/1.0.0/etc/mailhosts` file a line composed of these space-separated values:

```
<ip_address_or_hostname>:<port> <domain_name>
```

The following table describes each of these parameters. A sample of properly formatted lines follows the table.

mailhosts Parameters

Parameter	Description
`<ip_address_or_hostname>`	The IP address or hostname of a mail server M2G interacts with.
`<port>`	The port number of the mail server.
`<domain_name>`	The domain name associated with the MX record of the mail server whose IP address or hostname is specified on the same line.

Example The sample below shows properly formatted entries for the `mailhosts` file.

```
localhost:5026 docomo.ne.jp
localhost:5025 mmsc.test.ne.jp
```

map_domain_to_charhandler.cfg

Path `<M2G_HOME>/1.0.0/etc/map_domain_to_charhandler.cfg`

Purpose Maps the mail sender's domain to the EMOJI converter component name.

Max Entry Lines 100. Cannot be changed.

IMPORTANT All keys in this mapfile must be lowercase only.

Field Delimiter Vertical bar. Cannot be changed.

Dynamic Reload You can dynamically reload this file with this CLI command:

```
reload DOMAIN2HANDLERMAP.mapfile
```

For each domain, enter into the `map_domain_to_charhandler.cfg` file a line composed of these vertical bar-separated values:

```
[<match_type>] <sender_domain> <EMOJI_converter_component
_name>
```

The table that follows describes the values to include in each line of the map file. The "Position" column indicates the position of each parameter within the entry line. A sample of properly formatted lines follows the table.

customcharhandlermap.cfg Parameters (Part 1 of 2)

Position	Parameter	Description
0	[<match_type>]	The type of matching to implement when the M2G checks domain names for matches against the `<domain_pattern>`. Options are: ◆ EXACT ◆ SUBDOMAIN ◆ REGEX If you specify a match type, separate it from the `<domain_pattern>` with a space. If you do not specify a match type for a particular line in the map file, then EXACT matching is implemented for that line. For further information on match types, see *page 91*. IMPORTANT: Do not set a DEFAULT line in this file.

customcharhandlermap.cfg Parameters (Part 2 of 2)

Position	Parameter	Description
1	`<domain_pattern>`	Mail sender's domain or domain pattern. Maximum key length is 60 bytes.
2	`<EMOJI_converter_component_name>`	M2G converter component that will convert from the sender's Emoji character encoding to Unicode. Options are: ◆ `SMBSJIS2UTF8`—Softbank Mobile Shift JIS to UTF8 ◆ `DCMSJIS2UTF8`—Docomo Shift JIS to UTF8 NOTE: You must specify the component names in all capital letters, as indicated above.

Example

Below is an example of a correctly formatted `map_domain_to_charhandler.cfg` file.

```
EXACT|test.localhost.localhost|DCMSJIS2UTF8
EXACT|docomo|DCMSJIS2UTF8,SJISHANDLERS
EXACT|au|JISHANDLERS,SJISHANDLERS
EXACT|sbm|SJISHANDLERS
EXACT|yahoo|JISHANDLERS
```

map_domain_to_operator.cfg

Path `<M2G_HOME>/1.0.0/etc/map_domain_to_operator.cfg`

Purpose Maps the mail sender's domain to the operator name.

Max Entry Lines 100. Cannot be changed.

Field Delimiter Vertical bar.

IMPORTANT All keys in this mapfile must be lowercase only.

Dynamic Reload You can dynamically reload this file with this CLI command:

```
reload DOMAIN2OPERATOR.mapfile
```

For each domain, enter into the `map_domain_to_operator.cfg` file a line composed of these vertical bar-separated values:

```
[<match_type>] <sender_domain> <operator_name>
```

The table that follows describes the values to include in each line of the map file. The "Position" column indicates the position of each parameter within the entry line. A sample of properly formatted lines follows the table.

customcharhandlermap.cfg Parameters (Part 1 of 2)

Position	Parameter	Description
0	[<match_type>]	The type of matching to implement when the M2G checks domain names for matches against the `<domain_pattern>`. Options are: ◆ `EXACT` ◆ `SUBDOMAIN` ◆ `REGEX` If you specify a match type, separate it from the `<domain_pattern>` with a space. If you do not specify a match type for a particular line in the map file, then EXACT matching is implemented for that line. For further information on match types, see *page 91*. IMPORTANT: Do not set a DEFAULT line in this file.
1	`<sender_domain>`	Mail sender's domain or domain pattern. Maximum key length is 60 bytes.

***customcharhandlermap.cfg* Parameters (Part 2 of 2)**

Position	Parameter	Description
2	`<operator_name>`	Operator to map to the sender's domain name.

Example

Below is an example of a correctly formatted `map_domain_to_charhandler.cfg` file.

```
EXACT|sbm.ne.jp|sbm
EXACT|docomo.ne.jp|docomo
EXACT|ezweb.ne.jp|au
EXACT|disney.ne.jp|disney
EXACT|emobile.ne.jp|emb
```

services

Path `<M2G_HOME>/1.0.0/etc/services`

Purpose Maps M2G and partner server service names to TCP port numbers.

Max Entry Lines Unlimited.

Field Delimiter Space. Delimiter cannot be changed.

Dynamic Reload You cannot dynamically reload this file. To implement changes that you have made to this file, you must restart the M2G.

For each M2G service, and for each of your other network services with which the M2G will interact, enter into the `<M2G_HOME>/1.0.0/etc/services` file a line composed of these space-separated values:

```
<service_name> <port_number/protocol>
```

The table below describes each of these parameters. A sample of properly formatted lines follows the table.

IMPORTANT Do not change the service names in the `services` file. The M2G application has internal references to these names. You may change the port number associated with a service name, but not the service name itself.

Note The server in its internal service mapping table performs per-node (`%n`) and per-process (`%p`) expansions automatically, based on your settings for `DNSMGR.n_expansions` and `DNSMGR.p_expansions`. You do not need to perform these expansions manually.

For process-based listeners (which include a `%p` notation), each process is assigned its own port number by adding the process index offset (0, 1, 2, and so on) to the base port number assigned to the listener in the `services` file.

services Parameters

Parameter	Description
`<service_name>`	Symbolic service name.
`<port_number/protocol>`	Port number used by the service, followed by a “`/tcp`” or “`/udp`” suffix indicating the transport protocol.

Example The sample below shows properly formatted entries for the `services` file. It shows a the PC Client Relay POP3 Server.

```
pccpop3svr-u%n+svr 11025/tcp
```

Default Port Assignments, Service Names

The table below indicates default port assignments and service names established in the `services` file.

TCP port numbers

Note The service names can be expanded as follows:

%p is replaced with process ID

%n is replaced with node ID

CLI:

```
server-i+cli 7023/tcp
```

```
server-i%n+cli 7023/tcp
```

```
server-i%n-%p+cli 7024/tcp
```

XCONV:

```
server-i%n+xconv 7571/tcp
```

A2S:

```
a2s-i+ebf 7575/tcp
```

M2FE:

```
m2fe-i+ebf 7577/tcp
```

```
m2fe_auth-i+ebf 7577/tcp
```

```
m2fe_jobq-i+ebf 7577/tcp
```

INTERNET SMTP SERVER:

```
smtpsvr-u+svr 9025/tcp
```

```
smtpsvr-u%n+svr 9025/tcp
```

DOCOMO SMTP SERVER

```
dcmsmtpsvr-u+svr 8025/tcp
```

```
dcmsmtpsvr-u%n+svr 8025/tcp
```

PC CLIENT (RELAY) SMTP SERVER

```
pccsmtpsvr-u+svr 10025/tcp
```

```
pccsmtpsvr-u%n+svr 10025/tcp
```

```
pccsmtpsslsvr-u+svr 10465/tcp
```

```
pccsmtpsslsvr-u%n+svr 10465/tcp
```

PC CLIENT (RELAY) POP3 SERVER

```
pccpop3svr-u+svr 11025/tcp
```

```
pccpop3svr-u%n+svr 11025/tcp
```

```
pccpop3sslsvr-u+svr 11995/tcp
```

```
pccpop3sslsvr-u%n+svr 11995/tcp
```

NOTIFICATION CLIENT

```
notifyclt+http 7680/tcp
```

INTERNET MTA CLIENT

```
mtaclt+smtp 25/tcp
```

DOCOMO MTA CLIENT

```
dcmmtaclt+smtp 25/tcp
```

sidlist.cfg

Path `<M2G_HOME>/etc/sidlist.cfg`

Purpose Sets list of statistics to write to statistics log. For overview see *Configuring Statistics Logging, on page 78*.

Type Map file. For background see *Working with Map Files, on page 64*.

Max Entry Lines 1000. To change, see *page 91*.

Field Delimiter Vertical bar. To change, see *page 91*.

Dynamic Reload You can dynamically reload this file with this CLI command:

```
reload SIDLIST.mapfile
```

The map file `sidlist.cfg` lists the statistics that the M2G records to its statistics log. This map file lets you limit the server's statistics logging so that only a specified subset of statistics is logged, rather than all possible statistics. For each statistic or statistics pattern that you want to include in the statistics log, enter into the map file a line composed of these vertical bar-separated values:

```
<match_type>|<stat_id_key>
```

The table that follows describes each of these parameters. A sample of properly formatted lines follows the table.

sidlist.cfg Parameters (Part 1 of 2)

Parameter	Description
`<match_type>`	The type of matching to implement. Options are: ◆ `EXACT` This option logs statistics that exactly match your specified `<stat_id_key>` value. Exact matching is the option used in the default `sidlist.cfg` file. ◆ `PREFIX` This option logs statistics that start with your specified `<stat_id_key>` prefix. ◆ `REGEX` This option logs statistics that match against your specified `<stat_id_key>` regular expression.

sidlist.cfg Parameters (Part 2 of 2)

Parameter	Description
`<stat_id_key>`	A statistics ID, statistics ID prefix, or statistics ID regular expression. Gemini recommends that you use the default list of statistics that appear in the `sidlist.cfg` file, as shown on the next page: ◆ For each M2G listener: connections currently open. ◆ For each M2G connection pool: connections currently open, connections currently in use, number of spare connections, and number of queued requests waiting for a free connection. These statistics are written at your configurable statistics log write interval, set by the `STATSMGR.sampleinterval` property (*page 91*).

Example

The sample below shows an example of a configuration setting for the `sidlist.cfg` file.

```
# listeners
EXACT|CLI.conn.open
EXACT|CLIX.conn.open
```

6 M2G Logging

This chapter describes M2G logging. The information in this chapter applies to only to the M2G, not o Erlang-based servers. The chapter covers these topics:

M2G Logging Overview

The M2G generates several logs through which you can monitor the server's operations:

Application Log. The application log records application-related alerts, warnings, informational messages, and debug messages. Each log entry indicates the process and component with which the event is associated, as well as an event severity level and a brief descriptive message. For all events of level "INFO" or higher, the entry includes a numerical message code that aides in identifying and responding to the event. This log is described in detail in *Application Logging, on page 207*.

Transaction Log. The transaction log records transactions between different M2G services, between the M2G and the M2H, and between the M2G and external clients and servers. The transaction log is described in detail in *Transaction Logging, on page 210*.

Statistics Log. This log records statistical data for M2G conditions and transactions. The statistic log can serve as input into a statistical analysis tool. This log is described in detail in *Statistics Logging, on page 214*.

IMPORTANT We have not configured .m2g.properties and m2g.properties for statistics output.

For overviews of M2G logging configuration see:

- *Application Log, on page 30*
- *Transaction Log, on page 31*
- *Statistics, on page 31*

For information on M2G log rotation and removal, see *Log Rotation and Removal, on page 218*.

Application Logging

The M2G application log records application-related alerts, warnings, informational messages, and debug messages. By default this log is written to this file:

```
   <M2G_HOME>/1.0.0/var/log/m2g-app.log
```

Each application log entry is by default composed of these fields in this order, with | as the delimitation:

```
<PID>|<THREADID>|<DATE-CLF>|<MODULE:%-24s>|<LEVEL>|
<MESSAGECODE>|<MESSAGE>|<GTRID>
```

Each of the default application log entry fields is described in the table on the next page. The "Position" column indicates the default position of the field within an application log entry. The end of the table covers additional fields that may appear in your logs depending on your configuration choices.

For an overview of M2G application logging configuration, including log entry formatting, see *page 72*.

M2G Application Log Fields (Part 1 of 2)

Position	Field	Description
0	`<PID>`	System-assigned process identifier (PID) of the `M2G` process that generated the log message. Because multiple `M2G` processes may write to the application log, sometimes nearly simultaneously, it is important to pay attention to the PID associated with each message. A running M2G will be composed of two or more `M2G` processes—a parent process plus one or more child processes as configured in `m2g.properties`. Each process will have its own PID.
1	`<THREADID>`	Internal processing thread with which the message is associated.
2	`<DATE-CLF>`	Timestamp in format `DD/MM/YYYY:HH:MM:SS`
3	`<MODULE:%-24s>`	The internal component with which the message is associated. For example, the component may be a particular listener or connection pool. By default this field is configured to fill a minimum of 24 spaces. If the component name is less than 24 characters, it is appended with blank spaces until the 24 space minimum is reached for the field.
4	`<LEVEL>`	The severity level of the message. The level will be one of the following: ◆ `ALERT` A condition requiring immediate correction. ◆ `WARNG` A warning message, indicating a potential problem. ◆ `INFO` An informational message, indicating normal activity. ◆ `DEBUG` A low level detail message potentially of use when debugging the application.

M2G Application Log Fields (Part 2 of 2)

Position	Field	Description
5	`<MESSAGECODE>`	Integer code assigned to all messages of level INFO or higher. For a list of application log messages arrayed by message code, see *Gemini Guide to Error Code Messages*. NOTE: If you downgrade an INFO or higher level message to DEBUG level using the `errorcode.cfg` file the message will will retain its original `<MESSAGECODE>` when it is logged. Apart from this scenario, for all DEBUG messages the `<MESSAGECODE>` field will empty.
6	`<MESSAGE>`	A brief description of the event being logged.
7	`<GTRID>`	Global Transaction Log (Item) ID. This is an atom and a unique 32 byte string pair. For example: {s1b,45a571e4b3714440897e159fbb4ef662}.

Transaction Logging

The transaction log records the M2G's transactions with other clients and servers. By default this log is written to this file:

```
   <M2G_HOME>/1.0.0/var/log/m2g-tx.log
```

Each transaction log entry is by default composed of these fields in this order, with tab delimitation:

```
<PID>|<THREADID>|<DATE-CLF>|<PROTO>|<TYPE>|<STATUS>|<TRID>|
<CLIENT>|<HOST>|<DURATION>|<GTRID>|<YAGUID>|<FROM>|<RCPTS>|{Fr
om}|{To}|{Cc}|{Bcc}|{Message-ID}|<MESSAGE>|
<MAIL_FILTER_RESULT>|<HTTP_USERNAME>|<HTTP_REQ_BODY_SIZE>|
<HTTP_REQ_HOST>|<HTTP_REQ_LINE>|<HTTP_REQ_PORT>|
<HTTP_REQ_SIZE>|<HTTP_RES_BODY_SIZE>|<HTTP_RES_CODE>|
<HTTP_RES_SIZE>|
```

Fields that are not applicable to a particular transaction will be empty in the log entry for that transaction. The exception is when you have configured a transaction log field to include a string prefix or suffix in addition to the transaction-specific variable. In these cases, if the field variable is not applicable to a particular transaction, then only the string value will appear in the field for that transaction's log entry.

Example

```
The example below shows an M2G transaction log entry.

19259|0x92e4df8|16/11/2009:09:41:00|mta/SMTPSVR |INFO
|0082048|,status=I/O error,status=I/O error encountered
executing SMTP DATA
command|{s1c,f8ca8b161efe4c0dba2942b7ae9d7bfb}

19262|0x92e4df8|16/11/2009:09:41:02|mta/SMTPSVR |INFO
|0082048|,status=I/O error,status=I/O error encountered
executing SMTP EHLO
command|{s1c,a34ad21721ba49fe9c626e0de1e09da2}

19259|0x92e4df8|16/11/2009:09:41:03|mta/SMTPSVR |INFO
|0082048|,status=I/O error,status=I/O error encountered
executing SMTP EHLO
command|{s1c,99c1c733e99f484fb7d4f549d5003d0e}
```

Each of these transaction log entry fields is described in the table on the next page. The "Position" column indicates the default position of the field within a transaction log entry. Empty fields may also indicate that the values are in fact empty strings.

M2G Transaction Log Fields (Part 1 of 3)

Position	Field	Description
Default Fields		
1	`<PID>`	ID of the `M2G` process that executed the transaction.
2	`<THREADID>`	ID of the processing thread that executed the transaction.
3	`<DATE-CLF>`	Timestamp in format `DD/MM/YYYY:HH:MM:SS`
4	`<PROTO>`	Transaction protocol.
5	`<TYPE>`	The transaction type, for example, SMTP.
6	`<STATUS>`	Transaction status. The set of possible status values will depend on the transaction type, but generally the status field will indicate whether the transaction succeeded or failed.
7	`<TRID>`	Unique transaction ID. This will be a text string, with format dependent on the type of transaction.
8	`<CLIENT>`	The IP address or hostname of the client in the transaction.
9	`<HOST>`	The IP address or hostname of the host in the transaction.
10	`<DURATION>`	Duration of the transaction in milliseconds. Times are rounded off using the 'floor' function (highest integer equal to or less than the real number). A duration of "0" indicates that the transaction took less than 1 millisecond. NOTE: The manner in which a transaction duration is measured will be specific to the transaction type, but the duration will often run approximately from the initiation of a TCP connection to the closing of the TCP connection; or in the case of outgoing transactions that employ a connection pool, from the requesting of a connection from the pool to the the return of the connection to the pool.
11	`<GTRID>`	Global Transaction Log (Item) ID. This is an atom and a unique 32 byte string pair. For example: {s1b,45a571e4b3714440897e159fbb4ef662}.

M2G Transaction Log Fields (Part 2 of 3)

Position	Field	Description
12	`<YAGUID>`	Yet Another Global Unique ID is used for exchange between M2G and external clients and systems. YAGUID is constructed from the following sub-ids: ◆ VERSIONID (1 byte) ◆ OPCOID (2 bytes) ◆ SERVICEID (2 bytes) ◆ USERID (4 bytes) ◆ BOXID (2 bytes) ◆ UID (8 bytes) ◆ MIMEPARTID (2 bytes) ◆ RESERVEDID (4 bytes) Sub-ids are represented as non or zero padded hexadecimal 1 byte characters. Sub-ids are joined together by the '-' 1-byte character separator. The maximum length of a YAGUID is 35 bytes (and does not include the '\0' 1-byte character that terminates c-style strings). The "undefined" YAGUID is "0-00-00-00-0000-00-00000000-00-0000". The "maximum" YAGUID is "f-ff-ff-ff-ffff-ff-ffffffff-ff-ffff". External clients and systems MUST treat YAGUID as an opaque identifier.
13	`<FROM>`	Thesender of the message in the transaction.
14	`<RCPTS>`	The recipient of the message in the transaction.
15	`{From}`	"From" header value extracted from the message being transacted, if applicable.
16	`{To}`	"To" header value extracted from the message being transacted, if applicable.
17	`{Cc}`	"Cc" header value extracted from the message being transacted, if applicable.
18	`{Bcc}`	"Bcc" header value extracted from the message being transacted, if applicable.
19	`{Message-ID}`	"Message-ID" header value extracted from the message being transacted, if applicable.
20	`<MESSAGE>`	Custom message appropriate for certain transaction types. Field will be empty if no message is applicable to the transaction.
21	`<MAIL_FILTER_RESULT>`	
22	`<HTTP_USERNAME>`	From an HTTP request, the user's name.

M2G Transaction Log Fields (Part 3 of 3)

Position	Field	Description
23	`<HTTP_REQ_BODY_SIZE>`	From an HTTP request, the request body size in bytes.
4	`<HTTP_REQ_HOST>`	From an HTTP request, the target host.
25	`<HTTP_REQ_LINE>`	From an HTTP request, the request line.
26	`<HTTP_REQ_PORT>`	From an HTTP request, the target port.
27	`<HTTP_REQ_SIZE>`	From an HTTP request, the request size in bytes.
28	`<HTTP_RES_SIZE>`	From an HTTP response, the response size in bytes.
29	`<HTTP_RES_BODY_SIZE>`	From an HTTP response, the response body size in bytes.
30	`<HTTP_RES_CODE>`	From an HTTP request, the response code.
31	`<HTTP_RES_SIZE>`	From an HTTP request, the response size in bytes.

Statistics Logging

IMPORTANT We have not configured .m2g.properties and m2g.properties for statistics output.

The M2G makes statistics available to you in three different ways:

- A set of real-time statistics is available through the M2G command line interface. Through the `show stat` command you can instantly check on important state indicators such as the number of open connections currently held by the server's listeners and connection pools. For details, see *page 375*.
- The same set of statistics that is available through the CLI for real-time viewing can also be recorded to the application log at a configurable interval, providing a series of snapshots of system status throughout the day.
- A configurable set of statistics can be recorded to a dedicated statistics log.

This section describes how the M2G records statistics in the application log and in the statistics log. The material is divided into these sub-sections:

- *Statistics in the Application Log, on page 215*
- *Statistics in the Statistics Log, on page 216*

Statistics in the Application Log

If you wish, you can have the M2G write to the application log the same set of statistics that you can view in real time through the CLI. At a configurable interval, the M2G can log all of the CLI-viewable statistics, with one application log entry line for each statistic. The entries are formatted in the same way as other application log entries. The severity level is “INFO”, and the statistic name and value appear in the `<MESSAGE>` field, prefixed by the string “STATS DUMP”.

For an overview of statistics configuration, including enabling/disabling of statistics dumps to the application log, see *page 78*.

For description of CLI-viewable statistics, see *page 375*.

For further information on application log entry formatting, see *page 207*.

Statistics in the Statistics Log

The M2G can write statistics to a dedicated statistics log, which by default is:

```
   <M2G_HOME>/1.0.0/var/log/m2g-stats.log
```

Each statistics log entry is by default composed of these fields in this order, with space delimitation:

```
<DATE-MILLI:%Y/%m/%d %H:%M:%S:> <STAT_ID> <STAT_VALUE>
<DOMAIN> <STATUS>
```

For an overview of statistics configuration, including statistics log entry formatting, see *page 78*.

Each of the default statistics log entry fields is described in the table on the next page. The “Position” column indicates the position of the field within a statistics log entry.

M2G Statistics Log Fields

Position	Field	Description
1	`<DATE-MILLI:%Y/%m/%d %H:%M:%S:>`	Entry timestamp, where `%Y` = four digit year; `%m` = two digit month; `%d` = two digit date; `%H` = two digit hour; `%M` = two digit minute; `%S` = two digit seconds; followed by three digit milliseconds.
2	`<STAT_ID>`	The name of the statistic. The M2G will log the set of statistics configured by `sidlist.cfg` (*page 127*). By default the logged statistics are: ◆ `<LISTENER>.conn.open` Number of incoming connections currently open to the identified listener. ◆ `<POOL>.conn.open` Number of outgoing connections currently open from the identified connection pool. ◆ `<POOL.conn.inuse` Number of open connections currently in use in the identified connection pool. A connection is "in use" if a transaction is currently being conducted through the connection. ◆ `<POOL>.conn.spare` Number of open connections that currently are not in use in the identified connection pool. These connections are considered "spare". ◆ `<POOL>.conn.waiting` Number of queued requests currently waiting for a free connection from the identified pool. Applicable only when all connections are currently in use. For descriptions of individual listeners and connection pools by name, see *page 508*.
3	`<STAT_VALUE>`	The value of the statistic.
4	`<DOMAIN>`	This functionality is not used in the current release, so in each entry this field will be empty.
5	`<STATUS>`	This functionality is not used in the current release, so in each entry this field will be empty.

Log Rotation and Removal

Rotation of M2G application, transaction, and statistics logs is implemented by this Perl script that is installed with your application package:

```
<M2G_HOME>/1.0.0/bin/logrotate.pl
```

The script is invoked by cron tab entries established during product installation. The cron tab entries are configured in this file:

```
<M2G_HOME>/1.0.0/bin/cron.d/m2g-logrotate
```

A cron job that deletes old rotated logs is configured in this same file.

Default Log Rotation and Removal

The table below summarizes how application, transaction, and statistics log files are rotated on the M2G by default—if you install the product using the silent option, or if you install interactively and accept the default log rotation settings. If a “Size Threshold” is specified, then at each rotation interval the file will be rotated only if it is larger than the size threshold.

M2G Log Rotation

Log	Rotation Frequency	Size Threshold	GZip?	Location and Name Once Rotated
Application	30 min (1 past and 31 past the hour)	10MB	No	`<M2G_HOME>/1.0.0/var/log/archive/ m2g-app_<HOST>_<TIME>_<SEQNUM>.log`
Transaction	30 min (at the hour and 30 past the hour)	none	No	`<M2G_HOME>/1.0.0/var/log/archive/ m2g-tx_<HOST>_<TIME>_<SEQNUM>.log`
Statistics	10 min (at the hour and 10, 20, 30, 40, and 50 past the hour)	none	No	`<M2G_HOME>/1.0.0/var/log/archive/ m2g-stats_<HOST>_<TIME>_<SEQNUM>.log`

In the naming format for rotated logs, the `<TIME>` will be in format `YYYYMMDDHHMMSS`, and the `<SEQNUM>` is a locally generated sequence number unique for the node and file type.

Example The m2g-logrotate file also contains an entry that triggers a once-per-hour deletion of DFR interception files that have been stored for more than 30 days in the directory <M2G_HOME>/1.0.0/var/intercept/. DFR interception files are not rotated.

Changing Log Rotation and Removal

To change the timing of log rotation or log removal, edit the cron job timing specifications in the m2g-logrotate file.

To change the criteria for removal of archived log files, edit the options for the find command entry in the M2G-logrotate file. For example, if you want archived log files to be deleted after 10 days rather than after 30 days you would change the -mtime +30 option to -mtime +10.

To change the manner in which logs are rotated—for example, if you want to specify a size threshold for rotation or if you want log files to be gzipped as they are rotated—then you must revise the way in which the logrotate.pl script is invoked in the m2g-logrotate file.

The logrotate.pl script has the following syntax:

```
logrotate.pl -f <file> [-p <port>] [-s <size>]
[-t <dir>] [-g] [-d] <LOGNAME>
```

The table that follows describes each syntax element.

'logrotate.pl' parameters (Part 1 of 2)

Parameter	Description
-f <file>	Properties file in which the log types for the M2G are declared. This is the dot-file .m2g.properties. You must include the full path. For example, for an M2G-E installation, with the default path: -f /usr/local/gemini/m2g/1.0.0/etc/ .m2g.properties
[-p <port>]	Listening port number for the M2G command line interface. The logrotate.pl script sends a signal to the M2G CLI to execute the rotating of the logs. If you do not use the -p option, logrotate.pl uses a default CLI port number of 0.

'logrotate.pl' parameters (Part 2 of 2)

Parameter	Description
`[-s <size>]`	Rotate log file only if it is larger than this size. Specify as a number of bytes—for example 10M or 500k. If you do not use the `-s` option, `logrotate.pl` rotates the log file regardless of its size.
`[-t <dir>]`	Target directory in which to store rotated logs. If you do not use the `-t` option, `logrotate.pl` stores rotated logs into the `<M2G_HOME/var/log/archive` directory.
`[-g]`	Use the `-g` option if you want `logrotate.pl` to compress rotated files with GNU zip (gzip).
`[-d]`	Use the `-d` option if you want `logrotate.pl` to run in debug mode, with debug-level logging.
`<LOGNAME>`	Component name of the log that is to be rotated. Options are: ◆ `ALOG` Application log. ◆ `TLOG` Transaction log. ◆ `SLOG` Statistics log.

7 A2S/M2H Administration

This chapter describes how to use server commands and utilities for Gemini's Erlang-based A2S and M2H servers. The chapter covers these topics:

Installation and Startup

Installation

Use the following commands to install the A2S, M2H or GDSS:

```
% sudo ./installer-a2s.sh -o silent
```

```
% sudo ./installer-m2h.sh -o silent
```

Startup

Use the following commands to start the A2S or M2H, respectively:

```
% /etc/init.d/a2s start
```

```
% /etc/init.d/m2h start
```

Server Configuration Directory and Files

By default, the A2S | M2H configuration directory is:

```
<A2S | M2H_HOME>/1.0.0/etc
```

where `<A2S | M2H_HOME>` is the server's home directory as established during product installation. If during installation you accept the default for `<A2S | M2H_HOME>`, then the A2S | GDSS | M2H configuration directory is:

```
/usr/local/gemini/a2s/1.0.0/etc
```

or

```
/usr/local/gemini/m2h/1.0.0/etc
```

Working with the central.conf File

Each line of the `central.conf` file has the form

```
parameter: value
```

where `parameter` is the name of the configuration option being set and `value` is the value that the configuration option is being set to.

Valid data types for configuration settings are INT (integer), STRING (string), and ATOM (one of a pre-defined set of option names, such as “on” or “off”). Apart from data type restrictions, no further valid range restrictions are enforced for `central.conf` parameters.

All time values in `central.conf` (such as delivery retry intervals or transaction timeouts) must be set as a number of seconds.

Blank lines and lines beginning with the pound sign `(#)` are ignored.

To apply changes that you have made to the `central.conf` file, you must restart the server.

The variables in the configuration files (e.g. pGBSYSDIR) correspond to variables that are given values during the installation process. If the person doing the install does a "silent" install, the variable will default to the value indicated in the documentation.

If the person does an interactive install, he or she can specify a different value if desired. The installer file, shows what each variable defaults to.

CLI Overview

The Erlang-based server supports a command line interface (CLI) through which you can perform a variety of administrative tasks.

The Erlang-based server listening port number is configurable in your `m2h.properties` or `a2s.properties` file. By default the port number for the A2S is 7586, the GDSS, 7597 and the M2H is 7585.

No other aspects of the Erlang-based server command line interface are configurable.

To use the Erlang-based server command line interface, telnet to its listening port. When you do so, your terminal should display the A2S|M2H CLI prompt:

```
CLI>
```

On the next page, a summary of A2S |M2H CLI commands is provided. The remainder of this section provides detailed descriptions of each command, as well as command/response samples.

CLI Commands in Common for A2S and M2H Servers

This section describes how to use commands supported by Gemini's Erlang-based server command line interface.

Common A2S and M2H CLI Commands List

- `show nodes <NODEID>` — check if `<NODEID>` is active or not
- `show nodes` — list all active mnesia nodes
- `show tables <TABLEID>` — show information for `<TABLEID>`
- `show tables` — show information for all mnesia tables
- `show ubf` — show number of ubf connections and its maximum
- `set loglevel ALERT|WARNG|INFO|DEBUG` — sets application log level
- `reload central.conf` — reload `central.conf`
- `exit|logout|q|quit` — exit CLI

Note This A2S and M2H CLI command list is the same as the GDSS list of commands as described in the section *CLI Commands for the GDSS, on page 361*.

CLI Commands in Common

This section discusses the CLI Common Commands.

Viewing an Active Node List (show_nodes)

Use the `show_nodes` command to see which mnesia nodes in your Erlang-based server cluster are currently active.

The command syntax is as follows:

```
   show_nodes [<NODEID>]
```

'show_nodes' Parameters

Parameter	Description
`[<NODEID>]`	Optional node ID. If you specify a node ID, the command will determine whether that node is active or not. If you do not specify a node ID, the command returns a list of active nodes in the cluster. A sample node name is `a2s1@machine1`.

The following example shows that node `m2h1@hotate` is active.

```
CLI> show nodes m2h1@hotate
'm2h1@hotate' exists.
```

Example

The following example shows all active mnesia nodes.

```
CLI> show nodes
m2h1@hotate
```

Viewing a Database Table List (show_tables)

Use the `show_tables` command to view high level information about a particular database table, or about the full set of Erlang-based server database tables. The command displays the number of records and number of words in each table, as well as the average number of words per record.

The command syntax is as follows:

```
show_tables [<TABLEID>]
```

'show_tables' Parameters

Parameter	Description
`[<TABLEID>]`	Optional table ID. If you specify a table ID, the command returns information about that particular table. Valid table IDs are those in the example below.If you do not specify a table ID, the command returns information about all Erlang-based server tables.

Example The following shows information for all mnesia tables:

```
CLI> show tables

table name             |records   |words      |(average words)
 -------------------------------------------------------------
etq_pending_m2h1@hotate|         0|        124|              --
etq_ready_m2h1@hotate  |         0|        124|              --
etq_waiting_m2h1@hotate|         0|        124|              --
etq_data_m2h1@hotate   |         0|        332|              --
gcxKV                  |         1|        343|          343.00
gdict                 |         0|         124|              --
strmap                 |          0|        124|            --
schema                |          8|       1240|          155.00
```

Example The example below shows information for `<TABLEID> etq_pending_m2h1` showing `records` as the number of records in the table, `words` as the table size in erlang words, and `average words` as `words` divided by `records`:

```
CLI> show tables etq_pending_gdss@hotate

table name              |records   |words      |(average words)
 -------------------------------------------------------------
etq_pending_m2h1@hotate |         0|        124|              --
```

Viewing UBF Connections (show_ubf)

Use the `show_ubf` command to view the number of UBF connections for this server

The command syntax is as follows:

```
show_ubf
```

Example The following shows the number of UBF connections:

```
CLI> show ubf
m2fe_ebf has 0 connections(max=10000)
m2fe_ubf has 0 connections(max=10000)
m2be_ebf has 5 connections(max=10000)
m2ci_ebf has 0 connections(max=10000)
m2si_ebf has 0 connections(max=10000)
etq_ebf has 0 connections(max=10000)
m2h_ebf has 0 connections(max=10000)
m2h_ubf has 0 connections(max=10000)
```

Set Application Log Level (set_loglevel)

Use the `set_loglevel` command to set the application log level to the lowest severity level of messages to include in the application log.

Each message that the server can generate has an assigned severity level appropriate to the message. You can use the `set_loglevel` setting to filter the server's application logging so that only messages of your specified level and higher will be logged.

Options are, from highest to lowest level:

- `ALERT`
  Messages indicating a condition requiring immediate correction.
- `WARNG`
  Warning messages indicating a potential problem.
- `INFO`
  Informational messages indicating normal activity.
- `DEBUG`
  Low level detail messages potentially of use when debugging the application. Setting `loglevel` to `DEBUG` will result in a very large number of messages being logged.

For example, with the log level set to `INFO`, the server will log messages of all levels except `DEBUG`.

The command syntax is as follows:

```
set_loglevel [ALERT|WARNG|INFO|DEBUG]
```

Example The following sets the application log level to `DEBUG:`

```
CLI> set loglevel DEBUG
OK.
```

Reloading central.conf (reload central.conf)

Use the `reload central` command to reload a `central.conf` file that you have edited to dynamically change some of the file's settings. This command allows you to apply your new settings without restarting the M2H or A2S.

IMPORTANT This command applies only to the node to which you are submitting the command. It does not apply to other nodes in your M2H or A2S cluster.

The command syntax is as follows:

```
reload central
```

IMPORTANT Only some of the settings are able to be changed dynamically. Consult the parameter list below to see if the setting you wish to change is able to be changed dynamically.

The list of reloadable settings for the A2S include:

- `isp_fetch_interval`
- `restriction_period`
- `counter_max_1`
- `counter_max_2`
- `counter_max_3`
- `counter_reset_interval_1`
- `counter_reset_interval_2`
- `counter_reset_interval_3`
- `counter_same_rcpt_weight`
- `counter_resend_period`
- `counter_resend_weight`
- `popb4smtp_expiry`

The list of reloadable settings for the M2H include:

- `isp_fetch_interval`
- `m2ci_max_content_param`
- `m2ci_max_content_default`
- `m2ci_max_content_status`
- `m2ci_post_size_too_big`
- `m2fe_maillist_maxuids_perses`
- `m2fe_maillist_maxsess_peruser`
- `m2fe_etq_rate_control_ready`
- `m2fe_etq_rate_control_append`
- `m2fe_external_fetch_max`
- `m2be_mail_singleton_limit`
- `m2be_mail_shortcutdel_maxlen`
- `m2be_maillist_idletimeout`
- `m2be_maillist_stoptimeout`

- m2be_maillist_maxsess
- m2be_maillist_maxuids
- m2be_maillist_numuids
- m2be_maillist_danglinguids
- m2be_label_max_deltas
- address_ext_reg_max_tries
- nttr_max_mig_uid
- nttr.idc1_name: goo
- nttr.idc1_webui_auth_url
- nttr.idc1_webui_cookie
- nttr.idc1_webui_key_a
- nttr.idc1_webui_key_b
- nttr.idc1_webui_site
- nttr.idc1_webui_auth_enable
- nttr.idc1_pcmc_auth_url
- nttr.idc1_pcmc_prod_code_a
- nttr.idc1_pcmc_prod_code_b
- nttr.idc1_pcmc_src_ip_list
- nttr.idc2_name
- nttr.idc2_webui_auth_url
- nttr.idc2_webui_cookie
- nttr.idc2_webui_key_a
- nttr.idc2_webui_key_b
- nttr.idc2_webui_site
- nttr.idc2_webui_auth_enable
- nttr.idc2_pcmc_auth_url
- nttr.idc2_pcmc_prod_code_a
- nttr.idc2_pcmc_prod_code_b
- nttr.idc2_pcmc_src_ip_list
- nttr.idc_error_map_1
- nttr.idc_error_map_2
- nttr.idc_error_map_3
- nttr.idc_error_map_4
- nttr.idc_error_map_5
- nttr.idc_error_map_6

Terminating the CLI Session (quit)

Use any of the `exit|logout|q|quit` commands to terminate your session with the command line interface. The command syntax is as follows:

```
   exit|logout|q|quit
```

There are no arguments to the `exit|logout|q|quit` command.

Example The following command exits the CLI session:

```
CLI> exit
Goodbye!
```

A2S Only CLI Commands

This section includes the following A2S CLI commands:

- `show send_counter ip <IP>` — show information for send counter for `<IP>`
- `show send_counter id <URAID>` — show information for send counter for `<URAID>`
- `clear send_counter ip <IP>`—clear send counter for `<IP>`
- `clear send_counter id <URAID>`—clear send counter for `<URAID>`
- `show ldap` — show number of LDAP connections and its maximum
- `reload counter` — reload counter configuration files

Viewing Send Counter Information (show send_counter)

Counter information can be viewed by using either the IP address or the URAID.

Viewing Send Counter Information <IP>

The following command shows send counter information for an IP address.

```
show send_counter <IP>
```

The output will show the following information:

`state:` whether normal or restricted

`white:` true if it's listed in the white list

`black:` true if it's listed in the black list

`counter:` the actual value of the counter

`last_restricted`: the time stamp when restriction started(0 means not restricted now)

`last_reset`: the time stamp when the counter reset(or initialized) to 0

`last_incr`: the time stamp of the last increment

The following example shows information for send counter for IP address 127.0.0.1:

```
CLI> show send_counter ip 127.0.0.1
source_ip=<<"127.0.0.1">>, counter#=1, max=100, interval=10:
   state=normal
   white=false
   black=false
   counter=0
   last_restricted=0
   last_reset=0
   last_incr=0
source_ip=<<"127.0.0.1">>, counter#=2, max=200, interval=20:
   state=normal
   white=false
   black=false
   counter=0
   last_restricted=0
   last_reset=0
   last_incr=0
source_ip=<<"127.0.0.1">>, counter#=3, max=300, interval=30:
   state=normal
   white=false
   black=false
   counter=0
   last_restricted=0
   last_reset=0
   last_incr=0
```

Viewing Send Counter Information <URAID>

The following command shows the send counter information for <URAID>:

```
show send_counter <URAID>
```

The output will show the following information:

`state:` whether normal or restricted

`white:` true if it's listed in the white list

`black:` true if it's listed in the black list

`counter:` the actual value of the counter

`last_restricted`: the time stamp when restriction started(0 means not restricted now)

`last_reset`: the time stamp when the counter reset(or initialized) to 0

`last_incr`: the time stamp of the last increment

Example The following example shows information for send counter for IP address 127.0.0.1:

```
CLI> show send_counter ip 127.0.0.1
userid=<<"100">>, counter#=1, max=100, interval=10:
   state=normal
   white=false
   black=false
   counter=0
   last_restricted=0
   last_reset=0
   last_incr=0
userid=<<"100">>, counter#=2, max=200, interval=20:
   state=normal
   white=false
   black=false
   counter=0
   last_restricted=0
   last_reset=0
   last_incr=0
userid=<<"100">>, counter#=3, max=300, interval=30:
   state=normal
   white=false
   black=false
   counter=0
   last_restricted=0
   last_reset=0
   last_incr=0
```

Clearing Send Counter Information (clear send_counter)

Counter information can be cleared by using either the IP address or the URAID.

Clearing Send Counter Information <IP>

The following command clears send counter information for an IP address.

```
clear send_counter ip <IP>
```

Example The following shows the `clear send_counter` command for `<IP> 127.0.0.1` and the result.

```
CLI> clear send_counter ip 127.0.0.1
ok
```

Clearing Send Counter Information <URAID>

The following command clear send counter information for an IP address.

```
   clear send_counter id <URAID>
```

Example The following shows the `clear send_counter` command for `<URAID> 100` and the result.

```
CLI> clear send_counter id 100
ok
```

Viewing LDAP Connections (show ldap)

The following commands shows the number of LDAP connecctions and its maximum:

```
show ldap
```

Example

```
CLI> show ldap

ldap has 0 connections(max=10000)
```

Reloading the Counter Configuration File (reload counter.conf)

The following commands reloads the `counter.conf` configuration files after edits are made.

```
reload counter.conf
```

Example

```
CLI> reload counter
OK.
```

M2H Only CLI Commands

This section includes the M2H CLI commands.

- `show brokers` — show configuration of ticket brokers
- `show queues` — show job queues
- `apply brokers <FILENAME>` — apply configuration to ticket brokers
- `reload regex_euc` — reload `regex.euc`
- `show http` — show number of http connections and its maximum

Viewing Current Rate Restrictions (show_brokers)

Use the `show_brokers` command to view the ticket-based transaction restrictions currently being applied by the cluster. In addition to showing the rates currently in use, this command shows the currently available number of transaction tickets of each size.

The command syntax is as follows:

```
show brokers
```

Example

The sample below shows the configuration of ticket brokers on the M2H:

```
CLI> show brokers
m2fe_etq_rc_ready_send_docomo: 1000(tickets)/1000(ms), 1000
available
m2fe_etq_rc_ready_send_undefined: 1000(tickets)/1000(ms), 1000
available
m2fe_etq_rc_ready_bounce_undefined: 1000(tickets)/1000(ms),
1000 available
m2fe_etq_rc_ready_index_undefined: 1000(tickets)/1000(ms), 1000
available
m2fe_etq_rc_ready_notify_docomo: 1000(tickets)/1000(ms), 1000
available
m2fe_etq_rc_ready_notify_undefined: 1000(tickets)/1000(ms),
1000 available
m2fe_etq_rc_ready_fetch_undefined: 1000(tickets)/1000(ms), 1000
available
m2fe_etq_rc_ready_verify_ext_address_undefined: 1000(tickets)/
1000(ms), 1000 available
m2fe_etq_rc_ready_deprovision_undefined: 1000(tickets)/
1000(ms), 1000 available
m2fe_etq_rc_append_send_docomo: 1000(tickets)/1000(ms), 1000
available
m2fe_etq_rc_append_send_undefined: 1000(tickets)/1000(ms), 1000
available
m2fe_etq_rc_append_bounce_undefined: 1000(tickets)/1000(ms),
1000 available
m2fe_etq_rc_append_index_undefined: 1000(tickets)/1000(ms),
1000 available
m2fe_etq_rc_append_notify_docomo: 1000(tickets)/1000(ms), 1000
available
m2fe_etq_rc_append_notify_undefined: 1000(tickets)/1000(ms),
1000 available
m2fe_etq_rc_append_fetch_undefined: 1000(tickets)/1000(ms),
1000 available
```

```
m2fe_etq_rc_append_verify_ext_address_undefined:
1000(tickets)/1000(ms), 1000 available
m2fe_etq_rc_append_deprovision_undefined: 1000(tickets)/
1000(ms), 1000 available
```

Viewing Broker Queue Size (show_queues)

Use the `show_queues` command to view the current size of the Erlang-based server job broker queues.

The command syntax is as follows:

```
show queues [<brokername>]
```

'show_queues' Parameters

Parameter	Description
`[<brokername>]`	Optionally, you can specify a particular job-brokering entity. If you do not specify a `<brokername>`, then the command response will show the queue sizes for all job-brokering entities. The examples on the next page show how to interpret the command response.

Example

The following example shows `pqueue` as the size of the producer queue and `cqueue` as the size of the consumer queue:

```
CLI> show queues
queue name                                   |pqueue  |cqueue
 -------------------------------------------------------------
{deprovision,undefined}                      |       0|       1
{index,undefined}                            |       0|       1
```

Viewing Current Rate Restrictions (show_brokers)

Use the `show_brokers` command to view the ticket-based transaction restrictions currently being applied by the M2H cluster. In addition to showing the rates currently in use, this command shows the currently available number of transaction tickets of each size.

The command syntax is as follows:

```
show brokers
```

Example The sample below shows the output from the `show brokers` command:

```
CLI> show brokers
m2fe_etq_rc_ready_send_docomo: 1000(tickets)/1000(ms), 1000
available
m2fe_etq_rc_ready_send_undefined: 1000(tickets)/1000(ms), 1000
available
m2fe_etq_rc_ready_bounce_undefined: 1000(tickets)/1000(ms),
1000 available
m2fe_etq_rc_ready_index_undefined: 1000(tickets)/1000(ms), 1000
available
m2fe_etq_rc_ready_notify_docomo: 1000(tickets)/1000(ms), 1000
available
m2fe_etq_rc_ready_notify_undefined: 1000(tickets)/1000(ms),
1000 available
m2fe_etq_rc_ready_fetch_undefined: 1000(tickets)/1000(ms), 1000
available
m2fe_etq_rc_ready_verify_ext_address_undefined: 1000(tickets)/
1000(ms), 1000 available
m2fe_etq_rc_ready_deprovision_undefined: 1000(tickets)/
1000(ms), 1000 available
m2fe_etq_rc_append_send_docomo: 1000(tickets)/1000(ms), 1000
available
m2fe_etq_rc_append_send_undefined: 1000(tickets)/1000(ms), 1000
available
m2fe_etq_rc_append_bounce_undefined: 1000(tickets)/1000(ms),
1000 available
m2fe_etq_rc_append_index_undefined: 1000(tickets)/1000(ms),
1000 available
m2fe_etq_rc_append_notify_docomo: 1000(tickets)/1000(ms), 1000
available
m2fe_etq_rc_append_notify_undefined: 1000(tickets)/1000(ms),
1000 available
m2fe_etq_rc_append_fetch_undefined: 1000(tickets)/1000(ms),
1000 available
```

```
m2fe_etq_rc_append_verify_ext_address_undefined:
1000(tickets)/1000(ms), 1000 available
m2fe_etq_rc_append_deprovision_undefined: 1000(tickets)/
1000(ms), 1000 available
```

Applying Rate Restrictions (apply_brokers)

Use the `apply_brokers` command to have the Erlang-based server cluster implement the transaction restriction rates that you have configured in a particular ticket broker file.

The command syntax is as follows:

```
apply brokers <filename>
```

where `<filename>` is the name of the M2H broker file to be applied.

Example The following output applies configurations in the file to ticket brokers:

```
CLI> apply brokers broker.conf
OK.
```

Viewing HTTP Connections (show http)

Use the `show http` command to view the number of HTTP connections and it's maximum.

The command syntax is as follows:

```
show http
```

Example The following output shows the number of HTTP connections for each server and the maximum allowed:

```
CLI> show http
http port 7588 has 0 connections(max=10000)
http port 7587 has 0 connections(max=10000)
http port 7596 has 0 connections(max=10000)
http port 7595 has 0 connections(max=10000)
http port 7594 has 0 connections(max=10000)
```

Applying regex.euc (reload regex_euc)

Use the `apply_brokers` command to have the Erlang-based server cluster implement the transaction restriction rates that you have configured in a particular ticket broker file.

The command syntax is as follows:

```
   reload regex.euc
```

Example The following output applies the configurations in the `regex.euc` file to ticket brokers:

```
CLI> reload regex_euc
OK.
```

Using Mnesia Database Utilities

In addition to the CLI commands described previously in this chapter, the A2S and M2H also supports several utilities that let you perform administrative operations directly on the Mnesia distributed database that underlies these servers.

To run the Mnesia database utilities described in this section, you must be logged on as the A2S or M2H runtime user. By default, the runtime user is named `mmssys`.

For some Mnesia utilities, the A2S or M2H must be running when the utility is executed. For others, the server must be shut down when the utility is run. See the descriptions of individual utilities for details.

The basic syntax for the Mnesia database utilities is:

```
<A2S|M2H_HOME>/1.0.0/bin/<command> <arguments>
```

By default `<A2S|M2H_HOME>` is either `/usr/local/gemini/a2s`, or `/usr/local/gemini/m2h.`

The default Mnesia utility syntax is:

```
/usr/local/gemini/a2s|m2h/1.0.0/bin/<command> <arguments>
```

Each Mnesia utility after running exits with status code “0” if the operation was successful or with status code “1” if the operation encountered an error.

You can perform the following administrative tasks using Mnesia database utilities:

- *Generating an Mnesia DB Report (mnesia-info), on page 250*
- *Making an Mnesia DB Replica (make-mnesia-replica), on page 251*
- *Backing Up the Mnesia DB (mnesia-backup), on page 253*
- *Filtering a DB Backup File for a Single Node (filter-backup-nodelist), on page 254*
- *Restoring the Mnesia DB (mnesia-restore), on page 255*
- *Manually Setting an Mnesia Master Node (force-master-node), on page 258*

Note There is a “man” page for each Mnesia utility. The default location for the “man” pages is:

```
/usr/local/gemini/a2s|m2h/man
```

Generating an Mnesia DB Report (mnesia-info)

The mnesia-info utility creates an ASCII-formatted report that summarizes the current state of the Mnesia database.

The report includes summaries of:

- Local locking and transaction manager activity
- Table names and current memory utilization
- Mnesia version information
- Mnesia cluster membership (running nodes and down nodes)
- Storage locations of each table
- Aggregate transaction statistics

You must be logged in as the runtime user to run mnesia-info, and the must be running on the local host. The command syntax is as follows:

```
<A2S_HOME|M2H_HOME>/bin/mnesia-info
```

The mnesia-info utility exits with status code “0” if the operation was successful or with status code “1” if an error occurred.

Making an Mnesia DB Replica (make-mnesia-replica)

For the A2S only, the `make-mnesia-replica` utility creates on the local node an exact replica of all Mnesia tables residing on a source node. You can use this utility if, for example, you are expanding your cluster by adding a new node.

Note The M2H uses the utility script `m2h-make-mnesia-replica`. See *Replicating M2H Nodes (m2h-make-mnesia-replica), on page 262*.

There are several prerequisites to using `make-mnesia-replica` to replicate Mnesia tables from a source node to the local node:

- A2S must be installed on the local node.
- In the local node's `central.conf` file, the network monitoring application must have settings identical to the settings used by existing nodes in the A2S cluster.
- A2S must not be running on the local node.
- There must be no existing Mnesia tables on the local node.
- You must be logged in as the A2S runtime user (`mmssys`).
- Before using `make-mnesia-replica`, copy the file `.erlang.cookie` from the `~mmssys` directory on the source A2S | M2H node to the `~mmssys` directory on the local A2S | M2H node. After copying over the `.erlang.cookie` file, ensure that the file has `mmssys` as its owner and that it has permissions 0400.

The command syntax is as follows:

```
<A2S_HOME>/bin/make-mnesia-replica <source-node>
```

or

```
<M2H_HOME>/bin/make-mnesia-replica <source-node>
```

The `make-mnesia-replica` utility exits with status code “0” if the operation was successful or with status code “1” if an error occurred.

After a successful `make-mnesia-replica` operation, you can start the M2H | A2S application on the local node.

'make-mnesia-replica' Parameters

Parameter	Description
`<source-node>`	Existing M2H | A2S node from which to copy Mnesia tables. All Mnesia tables from this source M2H | A2S node will be exactly replicated to the local host, into the `<M2H_HOME>/var/data` directory or the `<A2S_HOME>/var/data` directory. A sample node name is `a2s1@machine1`.

Backing Up the Mnesia DB (mnesia-backup)

The `mnesia-backup` utility writes to a specified file a transaction-consistent backup of the Mnesia database. This backup file can then be used by the `mnesia-restore` utility (*page 255*) to restore the cluster in the event of a failure.

You must be logged in as the A2S | M2H runtime user to run `mnesia-backup`, and the A2S | M2H must be running on the local host. If you wish you may perform a "hot" backup, while the A2S | M2H is actively servicing user traffic.

The command syntax is as follows:

```
<A2S_HOME>/bin/mnesia-backup <output-file>
```

or

```
<M2H_HOME>/bin/mnesia-backup <output-file>
```

The `mnesia-backup` utility exits with status code "0" if the operation was successful or with status code "1" if an error occurred.

'mnesia-backup' Parameters

Parameter	Description
`<output-file>`	Path to the backup file to be created by `mnesia-backup`.

Filtering a DB Backup File for a Single Node (filter-backup-nodelist)

You can use the `filter-backup-nodelist` utility to create an Mnesia backup file suitable for restoring the Mnesia database on a single A2S | M2H node. By contrast, when the `mnesia-backup` utility (*page 253*) creates a standard backup file, the file contains a full description of *all* of the A2S | M2H nodes running at the time of backup. The `mnesia-restore` utility (*page 255*) then requires that all of the nodes described in the backup file be available when data is restored. If one or more of the nodes is unavailable, for example due to a hardware failure, the `mnesia-restore` utility cannot proceed.

The `filter-backup-nodelist` utility converts a standard backup file (that you have previously created using `mnesia-backup`) to a new backup file intended for restoration of only one A2S | M2H node. In the converted version of the backup file, the Mnesia cluster "schema" specification is edited to remove references to all nodes other than the desired node. You can then run the `mnesia-restore` utility on that single node, and the node is restored.

The command syntax for `filter-backup-nodelist` is as follows:

```
<A2S | M2H_HOME>/bin/filter-backup-nodelist <node-name>
<backup-file> <output-file>
```

The `filter-backup-nodelist` utility exits with status code "0" if the operation was successful or with status code "1" if an error occurred.

'filter-backup-nodelist' Parameters

Parameter	Description
`<node-name>`	Node name of the single node for which information will remain in the filtered backup file. A sample node name is `a2s1@machine1`.
`<backup-file>`	Path to the existing backup file produced by `mnesia-backup`.
`<output-file>`	Path to the new, filtered backup file to be created by `filter-backup-nodelist`.

Restoring the Mnesia DB (mnesia-restore)

The `mnesia-restore` utility restores Mnesia table definitions and table data, using a backup file that you previously created with the `mnesia-backup` utility (*page 253*).

The `mnesia-restore` utility performs the following steps:

- Deletes all existing Mnesia database tables.
- Uses the backup file to create new Mnesia database tables, including the schema table that specifies the Mnesia nodes in the cluster and their basic Mnesia configuration.
- Restores all data elements in each table.

IMPORTANT If one of your nodes has more up-to-date correct data than is contained in your backup file, do not use `mnesia-restore`. Instead, use the recovery procedure described in *Manually Setting an Mnesia Master Node (force-master-node), on page 258*.

You must be logged in as the A2S | M2H runtime user to run `mnesia-restore`, and the A2S | M2H must *not* be running on the local host.

The command syntax for `mnesia-restore` is as follows:

```
<A2S | M2H _HOME>/bin/mnesia-restore <backup-file>
```

The `mnesia-restore` utility exits with status code “0” if the operation was successful or with status code “1” if an error occurred.

‘mnesia-restore’ Parameters

Parameter	Description
`<backup-file>`	Path to the back-up file to be used by `mnesia-restore`.

The restore procedure is different depending on whether you are restoring all nodes in the cluster, or a subset of the nodes in the original cluster:

- *Restoring All Nodes in a Cluster, on page 256*
- *Restoring a Subset of Nodes, on page 257*

Restoring All Nodes in a Cluster

To restore all nodes in an A2S | M2H cluster, obtain the latest backup file for your cluster and then follow the steps below.

▼ To restore a node when all cluster nodes are available

1 Stop the A2S | M2H service on all nodes in the Mnesia cluster.

2 Select one of the data-full Mnesia nodes to be the data restoration node.

3 On each node in the cluster *except for the data restoration node*, start the Erlang runtime environment by executing:

```
cd /usr/local/gemini/a2s|m2h/var/data
```

and

```
su mmssys -c '/usr/local/gemini/erlang/bin/erl -sname a2s1'
```

or

```
su mmssys -c '/usr/local/gemini/erlang/bin/erl -sname m2h1'
```

each of the nodes. (The latter command presumes that the configurable `application_nodename` [*page 316*] is set to its default of `a2s1|m2h1`. Otherwise, on each node use the correct `application_nodename` value in place of `a2s1|m2h1`.)

4 On the data restoration node, run:

```
/usr/local/gemini/a2s|m2h/bin/mnesia-restore <backup-
file>
```

where `<backup-file>` is the path to the back-up file that you are using for the restoration.

5 On each node in the cluster except for the data restoration node, stop the Erlang runtime system using the command:

```
q().
```

at the Erlang command shell prompt. Note that the trailing period is required.

6 Start the A2S | M2H application on the data restoration node.

7 Start the A2S | M2H application on all other nodes in the cluster.

Restoring a Subset of Nodes

If one or more of the nodes in the original cluster is unavailable, you must first create a single node backup file using `filter-backup-nodelist` (*page 254*), then restore that single node, and then add the remaining nodes as they become available. Follow the steps described below.

▼ To restore a subset of nodes

1. Select one of the available data-full Mnesia nodes (i.e. a node with all A2S | M2H tables stored on local disk) to be the data restoration node.
2. Run the `filter-backup-nodelist` utility (*page 254*) on your back-up file, using the data restoration node name.
3. Stop the A2S | M2H service on the data restoration node.
4. On the data restoration node, run:

   ```
   /usr/local/gemini/a2s|m2h/bin/mnesia-restore <backup-file>
   ```

   where `<backup-file>` is the path to the filtered, single node back-up file that you are using for the restoration.
5. Start the A2S | M2H application on the data restoration node.
6. Now the cluster consists of one node, the data restoration node. To add the remaining nodes, use the `make-mnesia-replica` utility (*page 251*).

IMPORTANT When you add the remaining nodes, Gemini recommends that you give each added node a new A2S | M2H node name, for Mnesia clarity. You can create a new A2S | M2H node name by assigning a new `application_nodename` setting, a new machine name, or both.

Manually Setting an Mnesia Master Node (force-master-node)

The `force-master-node` utility allows you to manually specify the "master" node in a Mnesia cluster. The Mnesia master node is the node containing the most recent database data. In a healthy cluster Mnesia can determine which node is the master. However, in some failure scenarios such as a network partition, Mnesia cannot determine the correct master node. In these scenarios, an `inconsistent_database` event is triggered and the nodes shut down. In this case you have two options:

- If one of the nodes contains all of the latest database data, use the `force-master-node` utility to force the cluster to accept this node as the master node.
- If none of the nodes contains an uncorrupted version of the most current database, and if you have a backup file, you can restart the cluster from a backup file. See *Restoring the Mnesia DB (mnesia-restore), on page 255*.

After running the `force-master-node` utility on a node, the next A2S | M2H application start will force the local node to load all A2S | M2H tables from the specified node. Most Mnesia safeguards for table and transaction consistency are ignored after this command has been used, so `force-master-node` must be used only as a last resort when an A2S | M2H application cluster refuses to restart.

Note Application log messages generated by the `force-master-node` utility will always be written to the A2S | M2H application log file.

You must be logged in as the A2S | M2H runtime user to run `force-master-node`, and the A2S | M2H must *not* be running on the local host. For additional information, see *To restart the Mnesia cluster after a failure, using force-master-node, on page 259*.

The command syntax for `force-master-node` is as follows:

```
<A2S | M2H_HOME>/bin/force-master-node <master-node-name>
```

The `force-master-node` utility exits with status code "0" if the operation was successful or with status code "1" if an error occurred.

'force-master-node' Parameters

Parameter	Description
`<master-node-name>`	Name of the A2S | M2H node that you have determined to be the node with the most recent database data. A sample node name is `m2h1@machine1`.

▼ To restart the Mnesia cluster after a failure, using force-master-node

1. Determine which node is the master Mnesia node, by examining tables or log files to see which node has the most recent data.
2. Shut down A2S | M2H on all nodes.
3. Run `force-master-node` on all nodes to specify which node contains the most up-to-date copy of the Mnesia database.
4. Start A2S | M2H on the master Mnesia node.
5. Start A2S | M2H on all other nodes.

M2H Node Utility Scripts

The M2H package contains two scripts (m2h-user and m2h-node) for provisioning and deprovisioning and one script (m2h-quota) for quota policy management.

These scripts are installed in the `/usr/local/gemini/m2h/1.0.0/bin` directory.:

- *Viewing Contents and Sizes of User Profiles (m2h-admin), on page 260*.
- *Provisioning Users (m2h-user), on page 261*.
- *Replicating M2H Nodes (m2h-make-mnesia-replica), on page 262*.
- *Administering Mailbox Quota Policies (m2h-quota), on page 263*.
- *Bootstrapping the GDDS Cluster (m2h-node), on page 265*.

Viewing Contents and Sizes of User Profiles (m2h-admin)

Use the m2h-admin script to view provisioned user profile data. The script's usage is as follows:

```
m2h-admin -m2h <M2H@NODE> -a2s <A2S@NODE> -userid <USERID>
-uraid <URAID> [-user -mail -ui -filter -quota]
```

Both of the m2h node and a2s node names are required.

```
-m2h <M2H@NODE>
```

```
-a2s <A2S@NODE>
```

Either the UserID or URAID may be given. If both are given, they must refer to the same user in the A2S.

```
-userid <USERID>
```

```
-uraid <URAID>
```

If any of the following options are given, then only that profile data will be output. If none are given, then all user profile data will be output.

```
-user
```

```
-maild
```

```
-ui
```

```
-filter
```

```
-quota
```

Note The A2S "aa" record data is always shown.

Provisioning Users (m2h-user)

The m2h-user script is mainly used to provision (add) a new user to the system. The caller provides a userid for provisioning or lets the M2H automatically choose a new userid.

The m2h-user script can also delete, set, and get users. The m2h-user script permits only a small subset of a user's overall profile(s) to be set via this tool.

The m2h-user script supports the following fields:

- `userid (primary key)`
- `uraid (secondary key)`
- `serviceid`
- `nickname`
- `address_int list`
- `opcoinfo`
- `opcouid`
- `msisdn`
- `cos`
- `mail_max_migration_uid`

Example The following example adds a user with an `userid`=100 and `serviceid`=1:

```
/usr/local/gemini/m2h/1.0.0/bin/m2h-user add \
-userid 100 -uraid uraid100 -nickname nickname1 \
-address_int 1@domain1 undefined undefined \
-opcoinfo_undefined -cos free -serviceid 1 \
-strict \
-m2h m2h1@$sname
```

Replicating M2H Nodes (m2h-make-mnesia-replica)

The `make-mnesia-replica` utility creates on the local M2H node an exact replica of all Mnesia tables residing on a source M2H node. You can use this utility if, for example, you are expanding your M2H cluster by adding a new node.

This script is same as the *make-mnesia-replica* script except it has been customized for the M2H node's mnesia schema.

There are several prerequisites to using *make-mnesia-replica* to replicate Mnesia tables from a source M2H node to the local M2H node:

- M2H must be installed on the local node.
- In the local node's `central.conf` file, the network monitoring application must have settings identical to the settings used by existing nodes in the M2H cluster.
- M2H must not be running on the local node.
- There must be no existing Mnesia tables on the local node.
- You must be logged in as the M2H runtime user (`mmssys`).
- Before using *make-mnesia-replica*, copy the file `.erlang.cookie` from the `~mmssys` directory on the source M2H node to the `~mmssys` directory on the local M2H node. After copying over the `.erlang.cookie` file, ensure that the file has `mmssys` as its owner and that it has permissions 0400.

The command syntax is as follows:

```
<M2H_HOME>/bin/make-mnesia-replica <source-node>
```

The *make-mnesia-replica* utility exits with status code “0” if the operation was successful or with status code “1” if an error occurred.

After a successful *make-mnesia-replica* operation, you can start the M2H application on the local node.

‘make-mnesia-replica’ Parameters

Parameter	Description
`<source-node>`	Existing M2H node from which to copy Mnesia tables. All Mnesia tables from this source M2H node will be exactly replicated to the local host, into the `<M2H_HOME>/var/data` directory. A sample node name is `M2H1@machine1`.

Administering Mailbox Quota Policies (m2h-quota)

Quotas have been implemented at the m2fe and m2be levels ("front-end" and "back-end").

This script provides a simple API to administer the global mailbox quota policies. These policies are stored in GDSS per M2BE+GDSS cluster. The m2h-quota script has four main features:

- **set** - set a mail quota policy
- **del** - delete a mail quota policy
- **list** - list all mail quota policies
- **get** - get a mail quota policy
- **cache** - refresh mail quota cache (for a particular m2h "m2be" node)

Quotas can be assigned via two methods: a setting for an individual mailbox, or via a quota template. By default, there are no quota templates defined. Any developer or QA efforts should first define a quota template before assigning that template to a mailbox. If a mailbox's quota refers to an undefined quota template, quota enforcement will be disabled.

A quota template name has a binary() key. All quota templates are stored via the "label" mechanism, using an otherwise unprovisioned user: see the QUOTA_TEMPLATE_USER macro in api_m2be_mail.erl.

The quota template mechanism is similar to Gemini's GMS "Class of Service policy template" for controlling quotas. The main difference is that GMS allows individual limits in a template to be overridden on a per-mailbox basis. The m2be implementaion does not have such override capability.

The limits enforced are:

- Number of messages (i.e. items)
- Number of bytes (i.e. sum of all message sizes)

Like GMS's policy system, if a limit is equal to zero, that limit is not enforced. For example, if items = 0 and bytes = 0, quota enforcement will be completely disabled.

The definitions for quota templates are cached by the M2BE application for up to 10 seconds. Therefore, any change to a quota template definition may not be visible immediately. If this delay cannot be tolerated, the Erlang function `m2be_qcache:sync_refresh_cache()` can be called to flush the cache and reload it.

Example The following shows an example of using the `set` command for *m2h-quota*:

```
/usr/local/gemini/m2h/1.0.0/bin/m2h-quota set -quotatype mail -
policyname 4 \

-quotaitems 200000 -quotabytes 5368709120 -m2h m2h1@$sname
```

Bootstrapping the GDDS Cluster (m2h-node)

Use *m2h-node* to bootstrap and create tables or add M2H-BE nodes as clients to the GDSS cluster. This script should be run only once after a fresh installation.

The usage for this script is as follows:

```
m2h-node COMMAND [ARGS]
```

with these `COMMAND` options and `[ARGS]`:

```
start -gdssadmin <nodeid>

add -gdssadmin <nodeid> -m2h <nodeid>...

del -gdssadmin <nodeid> -m2h <nodeid>...

list -gdssadmin <nodeid>

http_connections -m2h <nodeid>

ldap_connections -a2h <nodeid>

dump_profiles -m2h <nodeid> -dump_dir <directory>

bootstrap -gdssadmin <nodeid>... -gdss <nodeid>...
[-bricksperchain <num>]

createtables -m2h <nodeid> -gdssadmin <nodeid>
-gdss <nodeid>...[-chainlength <num>]
[-num_nodes_per_block <num>]
[-block_mult_factor <num>]
```

Example The following example adds M2H-BE nodes as clients to the GDSS cluster:

```
/usr/local/gemini/m2h/1.0.0/bin/m2h-node add \
-gdssadmin gdss1@tkqa-464-8 \
-m2h m2h1@tkqa-464-8 \
-m2h m2h1@tkqa-464-9 \
-m2h m2h1@tkqa-464-10
```

Example The following example bootstraps the GDSS cluster:

```
/usr/local/gemini/m2h/1.0.0/bin/m2h-node bootstrap \
-gdssadmin gdss1@tkqa-464-8 \
-gdss gdss1@tkqa-464-9 \
-gdss gdss1@tkqa-464-10 \
-bricksperchain 2
```

Example The following example creates the tables for the GDSS cluster:

```
/usr/local/gemini/m2h/1.0.0/bin/m2h-node createtables \
-gdssadmin gdss1@tkqa-464-8 \
-gdss gdss1@tkqa-464-9 \
-gdss gdss1@tkqa-464-10 \
-chainlength 2 -num_nodes_per_block 0 -block_mult_factor 0
```

8 M2H Configuration Files

This chapter describes Messaging 2.0 Helper (M2H) configuration files and settings and includes these configuration files:

Note For overviews of how to configure M2H features, see *Chapter 6, M2H, A2S, GDSS Configuration*.

If you want to quickly locate the description of a particular setting that you have seen in the M2H `central.conf` file, you can use *Index of Settings in .properties and .conf Files, on page 415*.

broker.conf

IMPORTANT This file is used internally. Do not modify the settings in this file.

Path `<M2H_HOME>/1.0.0/etc/broker.conf`

Purpose Ticket-based transaction rate controls. For overview see *Configuring Transaction Rate Control, on page 237*.

For further information on the `apply_brokers` command, see *page 269*.

You can configure ticket-based transaction rate controls with settings in this file. For each ticket type that you want to subject to rate control, enter a line composed of these comma-separated values:

```
ticket type, ticket per time period, time period length
(milliseconds), constant 1, blank
```

After editing this file, you can dynamically apply the file through the M2H command line interface, using this command:

```
   apply_brokers broker.conf
```

IMPORTANT Applying the broker file with the `apply_brokers` command will result in the M2H immediately starting to implement the restriction levels in the reloaded file.

Note If you want to view the broker settings currently being used by the M2H, you can use the `show_brokers` command as described on *page 267*.

The table that follows describes each parameter in the `broker.conf` file. An example follows the table.

broker.conf Parameters

Parameter	Description
`<ticket_type>`	The ticket type. Currently, only one ticket type is supported. ◆ `message_new` Ticket for new messages queued for delivery. New messages are messages that M2G is inserting into the M2H delivery queue database or existing messages that are moving from one queue processor to another. This ticket implements rate control for such messages.
`<tickets_per_period>`	Total number of tickets of this type to generate per time period. When this is set to `0` (zero), manual restriction takes effect. IMPORTANT: Note that this is the *total* number of tickets of this type to generate per time period.
`<time period length>`	Length of the ticket generating period, in milliseconds.
`<constant 1>`	Do not use.
`<blank>`	Do not use.

The sample below shows a properly formatted entry for `broker.conf`.

```
m2fe_etq_rc_ready_send_docomo,1000,1000,1,
m2fe_etq_rc_ready_send_undefined,1000,1000,1,
m2fe_etq_rc_ready_bounce_undefined,1000,1000,1,
m2fe_etq_rc_ready_index_undefined,1000,1000,1,
m2fe_etq_rc_ready_notify_docomo,1000,1000,1,
m2fe_etq_rc_ready_notify_undefined,1000,1000,1,
m2fe_etq_rc_ready_fetch_undefined,1000,1000,1,
m2fe_etq_rc_ready_verify_ext_address_undefined,1000,1000,1,
m2fe_etq_rc_ready_deprovision_undefined,1000,1000,1,
m2fe_rc_mail_recv,1000,1000,1,
m2fe_rc_mail_relay,1000,1000,1,
m2fe_rc_mail_send,1000,1000,1,
m2fe_rc_mail_migrate,1000,1000,1,
m2fe_rc_mail_label,1000,1000,1,
m2fe_rc_mail_delete,1000,1000,1,
m2fe_rc_async_fetch,1000,1000,1,
m2fe_rc_addr_verify,1000,1000,1,
m2fe_rc_user_deprovision,1000,1000,1,
```

central.conf

Path `<M2H_HOME>/1.0.0/etc/central.conf`

Purpose M2H configuration file.

Dynamic Reload You cannot dynamically reload this file. To activate changes that you make to the file, you must restart the M2H.

The table that follows describes each parameter in the `central.conf` file. For background information about how to work with the `central.conf` file, see *page 295*.

central.conf Parameters (Part 1 of 47)

Parameter Description	Valid Range	File Default	Internal Default
`application_home`			
M2H top-level directory. File default = set during install (installer defaults to `/usr/local/gemini/m2h`)	STRING	see description	null
`application_bootname`			
M2H boot name. This parameter disables unnecessary applications, that is, unnecessary (but harmless) application logs to be generated every 60 seconds or so, not required for the respective role of FE or BE.	M2H, M2BE, or M2FE	M2H	null

central.conf Parameters (Part 2 of 47)

Parameter Description	Valid Range	File Default	Internal Default
`application_nodename`			
First part of the M2H application node name. Each M2H server is assigned a “node name”. A node name identifies a specific Linux process on a specific physical machine. Different node names can be used to run multiple M2H services on the same physical machine, if desired. An M2H node name has three parts: 1) An application name local to the physical machine. 2) The “@” symbol. 3) The hostname of the physical machine, as shown by the output of the system command `uname -n`. If the hostname as shown by `uname -n` has one or more dots in it (for example `machine1.company.com`) then only the left-most part is used for the M2H node name (from the example, `machine1`). A sample three-part M2H node name is `m2h1@machine1`. The first part of the M2H node name is determined by your `application_nodename` setting—in the sample, `m2h`.	STRING	m2h1	null
`application_data_dir`			
Data directory for the database. File default = set during install (installer defaults to `/usr/local/gemini/m2h/var/data`) IMPORTANT: Do not change the location of the data directory after installation.	STRING	see description	null

central.conf Parameters (Part 3 of 47)

Parameter Description	Valid Range	File Default	Internal Default
`application_txn_log_path`			
Path to the M2H transaction log file, including file name. File default = <set during install>`/m2h-tx.log` (installer defaults to `/usr/local/gemini/gdss/var/log` for the directory path portion)	STRING	see description	dev/null
`application_tx_log_flush`			
Limit for the transaction log file's number of log entries to buffer before storing to disk.	INT (0 to INT_MAX)	1	0
`application_app_log_path`			
Path to the M2H application log file, including file name. File default = <set during install>`/m2h-app.log` (installer defaults to `/usr/local/gemini/gdss/var/log` for the directory path portion)	STRING	see description	dev/null

central.conf Parameters (Part 4 of 47)

Parameter	Valid	File	Internal
Description	**Range**	**Default**	**Default**
`application_app_log_level`			
The lowest severity level of messages to include in the application log. Each message that the M2H can generate has an assigned severity level appropriate to the message. You can use the `application_app_log_level` setting to filter M2H logging so that only messages of your specified level and higher will be logged. Options are, from highest to lowest level: ◆ `ALERT` Messages indicating a condition requiring immediate correction. ◆ `WARNG` Warning messages indicating a potential problem. ◆ `INFO` Informational messages indicating normal activity. ◆ `DEBUG` Low level detail messages potentially of use when debugging the application. For example, with `application_app_log_level` set to `INFO`, the M2H will log messages of all levels except `DEBUG`.	ATOM	INFO	INFO
`application_app_log_log_fmt`			
Application log entry format type, indicating the log entry fields and their order. Options are: ◆ `default` The default M2H application log format, as follows: `<PID> <DATE> <MODULE> <LEVEL> <MESSAGECODE> <MESSAGE>` ◆ `cstm1` Custom format #1, as follows: `<DATE> <MESSAGECODE> <LEVEL> <PID> <THREADID> <MODULE> <MESSAGE>` NOTE: The format of the `<DATE>` field is specified by the `application_app_log_date_fmt` setting. The delimiter between the fields is specified by the `application_app_log_field_sep` setting.	ATOM	default	default

central.conf Parameters (Part 5 of 47)

Parameter Description	Valid Range	File Default	Internal Default
`application_app_log_date_fmt`			

Application log entry timestamp type. Options are:
◆ `default`
The default M2H application log timestamp format, as follows:
`YYYYMMDDHHMMSS`
◆ `cstm1`
Custom timestamp #1, as follows:
`YYYY/MM/DD HH:MM:SS:000`
NOTE: For the `cstm1` timestamp, the “000” in the millisecond part is a fixed constant.
	ATOM	default	default

`application_app_log_field_sep`			
ASCII decimal code indicating the desired delimiter between fields in an application log entry. Options are: ◆ `32` Single byte space. ◆ `124` Vertical bar (|).	ATOM	124	124
`vm_swappiness_value`			
Linux virtual memory "swappiness" correction. The default is for the type of Linux kernel being used. The typical value is 60%. The maximum is 100%. This value should be 0 (zero) for all production environments. The default for Red Hat EL4.4 default is 60 (percent). See the following for more information: `http://kerneltrap.org/node/3000 http://www.westnet.com/~gsmith/content/linux-pdflush.htm`	0 to 100	0	0
`cli_port`			
TCP port number for the command line interface.	INT	7585	7585

central.conf Parameters (Part 6 of 47)

Parameter Description	Valid Range	File Default	Internal Default
`cli_hello`			
Command line interface hello message. Default = M2H CLI Server	STRING	see description	see description
`cli_prompt`			
Prompt for the command line interface.	STRING	CLI>	CLI>
`cli_module`			
This parameter is used for internal module configuration. Do not change this.	ATOM	m2h_cli	m2h_cli
`cli_ubf_servers`			
This parameter is used for internal module configuration. Do not change this. Default = `m2fe_ebf m2fe_ubf m2be_ebf m2ci_ebf m2si_ebf etq_ebf m2h_ebf m2h_ubf`	ATOM	see description	see description
`ticket_server_tcp_port`			
Port on which the A2S listens for requests to its internal ticket broker. In the current release, this listener is not used.	INT	set during install (installer defaults to 2299)	2299

central.conf Parameters (Part 7 of 47)

Parameter Description	Valid Range	File Default	Internal Default
`ticket_server_distributed_nodes`			
List of all nodes running the distributed ticket server used by the main application startup Tcl script. This node list is required for M2H start-up. The node list must be identically configured on each of your M2H nodes. If you are using only one M2H node, specify just the one node's name for this setting. File default = `/etc/central.conf`	STRING	set during install	null
`ticket_maker_reset_timeout`			
This setting works together with the congestion monitoring controls that you establish in the `congestion_watcher.conf` file (*page 317*). When you start or restart the M2H, or when you dynamically reload the `congestion_watcher.conf` file, the ticket broker will wait for `ticket_maker_reset_timeout` seconds before issuing any messaging tickets. This pause allows time for the congestion monitor to send restriction requests to the ticket broker, if congestion has been detected.	INT	6	6

central.conf Parameters (Part 8 of 47)

Parameter Description	Valid Range	File Default	Internal Default
`network_monitor_enable`			
Enable network partition monitoring. Options are: ◆ `true` Enable network partition monitoring. You can enable network monitoring only if you have set up two networks, A and B, that connect your M2H nodes. Gemini recommends that A and B be physically separate networks. Network monitoring works by comparing heartbeats from network A and network B. For further information, see *page 298*. ◆ `false` Disable network partition monitoring. IMPORTANT: For network partition monitoring to function properly, these `central.conf` settings must be assigned identical values on each M2H node: ◆ `network_monitor_enable` ◆ `network_a_*` ◆ `network_b_*` ◆ `heartbeat_*`	ATOM	set during install (installer defaults to 'false')	false
`network_monitor_monitored_nodes`			
Enter the list of all M2H node names (without single quotes) for this cluster. The installer defaults to `m2h1@node-a, m2h1@node-b.`		set during install	2299
`network_a_address`			
IP address for the A network. This network *must* be the same network used by the Erlang network distribution protocol (i.e. the network used for Mnesia replication traffic). File default = set during install (installer defaults to 10.1.1.12)	STRING	see description	null

central.conf Parameters (Part 9 of 47)

Parameter Description	Valid Range	File Default	Internal Default
`network_a_broadcast_address`			
IP broadcast address for the A network. This network *must* be the same network used by the Erlang network distribution protocol (i.e. the network used for Mnesia replication traffic). File default = set during install (installer defaults to 10.1.1.255)	STRING	see description	null
`network_a_tiebreaker`			
IP address for the A network to act as a tiebreaker. If the network monitoring application determines that the A network is partitioned and the B network is not partitioned, then if `network_a_tiebreaker` responds to an ICMP echo (a ping), then the local M2H node is on the "correct" side of the partition. If the local M2H node is not on the correct side of the partition (if the attempt to ping the tiebreaker address fails), then it shuts down immediately. The `network_a_tiebreaker` address must be extremely reliable and must be as close to the local M2H node as possible (from a network Layer 1 and 2 point of view) as well as close to all other M2H nodes. Ideally the tiebreaker should be the address of the Layer 2 switch or Layer 3 router that all Mnesia communications flow through. File default = set during install (installer defaults to 10.1.1.254)	STRING	see description	null
`network_b_address`			
IP address for the B network. This network should be physically separate from the A network. File default = set during install (installer defaults to 10.10.10.12)	STRING	see description	null

central.conf Parameters (Part 10 of 47)

Parameter Description	Valid Range	File Default	Internal Default
`network_b_broadcast_address`			
IP broadcast address for the B network. This network should be physically separate from the A network. File default = set during install (installer defaults to 10.10.10.255)	STRING	see description	null
`heartbeat_beacon_interval`			
Heartbeat beacon interval in milliseconds. At this interval, UDP heartbeart signals are transmitted from the local M2H node to each other M2H node in the cluster. The heartbeats are sent out both through network A and through network B. Gemini recommends that this interval be between 250 and 1000 (milliseconds).	INT	1000	1000
`heartbeat_warning_interval`			
Heartbeat alarm interval in seconds. If this interval passes without the local M2H node receiving a heartbeat signal from a peer M2H node, an alert is written to the local application log.	INT	set during install (installer defaults to 5)	5

central.conf Parameters (Part 11 of 47)

Parameter Description	Valid Range	File Default	Internal Default
`heartbeat_failure_interval`			
Heartbeat failure interval in seconds. A serious error has occurred if during this interval a heartbeat from a peer M2H node has been detected on network B but no heartbeat from that node has been detected on network A. The `network_a_tiebreaker` (*page 278*) address will be pinged to determine whether or not the local M2H node should be shut down to avoid database damage. NOTE: The value of `heartbeat_failure_interval` should be larger than the value of `heartbeat_warning_interval` by a factor of at least 1.5x but preferably 2x or more. Cluster timeout interval. If there is a network partition (or other failure that will cause network traffic from a node to be dropped or delayed), PSS/LSS protocol operations will hang. Therefore, WARNING: The "cluster_timeout" value must be larger than the "heartbeat_failure_interval" value, preferably by five (5) seconds or more.	INT	set during install (installer defaults to 15)	15
`cluster_timeout`			
Enter the cluster timeout interval in seconds. Erlang nodes will force a disconnect from each other if this timeout value is exceeded. WARNING: The "cluster_timeout" value must be larger than the "heartbeat_failure_interval" value, preferably by five (5) seconds or more. File default = <set during install> (installer defaults to 20 seconds for the timeout).	INT	set during install (installer defaults to 20)	20
`heartbeat_status_udp_port`			
UDP port for heartbeat listener.	INT	63099	63099

central.conf Parameters (Part 12 of 47)

Parameter Description	Valid Range	File Default	Internal Default
`heartbeat_status_xmit_udp_port`			
This is the UDP port for the heartbeat transmitter, the base port. Note that the actual port may be higher.	INT	63100	63100
`mnesia_diskmon_dir_attrname`			
Name of the central.conf attribute, to locate the directory pathname to monitor. This may be a comma-separated list. If none of the listed attributes appear in central.conf, then the current directory, ".", will be monitored. Default = application_data_dir	STRING	see description	see description
`mnesia_diskmon_dir_minfree`			
Minimum disk space for alarm trigger, units in kilobytes.	INT	800000	800000
`mnesia_diskmon_latest_log_max`			
Maximum size for Mnesia LATEST.LOG file, units in kilobytes.	INT	500000	500000
`congestion_watcher_config`			
Sets the filename for `congestion_watcher.config`, the file used for congestion watcher settings. Should it be changed to another file that is readable, it must have the same format as the `congestion_watcher.conf` file. There is no internal default. Default = pM2HETCDIR/congestion_watcher.conf See *congestion_watcher.conf, on page 317* for documentation on the format of the file.	STRING	see description	see description
`isp_fetch_interval`			
ISP async fetch interval from webmail in seconds.	INT > than 0	10	10

central.conf Parameters (Part 13 of 47)

Parameter	Valid	File	Internal
Description	Range	Default	Default
`m2ci_ebf_port`			
Port number for the Mail 2.0 Client-interface. Implements M2H's "client-interface" component. Used for test purposes.	INT	7578	7578
`m2ci_ebf_maxconn`			
Mail 2.0 Client-interface. Implements M2H's "client-interface" component. Used for test purposes.	INT	10000	10000
`m2ci_ebf_timeout`			
Mail 2.0 Client-interface. Implements M2H's "client-interface" component. Used for test purposes.	INT	60	60
`m2ci_yaws_base_url_rpc`			
Mail 2.0 Client-interface. Implements M2H's "client-interface" component for IDC authorization.	STRING	/rpc	/rpc
`m2ci_yaws_base_url_attach`			
Base URL for HTTP "attach" requests. Default = `/attach`	STRING	see description	see description
`m2ci_yaws_base_url_vcard`			
Base URL for HTTP "vcard" requests. Default = `/vcard`	STRING	see description	see description
`m2ci_yaws_auth_source_ip`			
HTTP header field name for the source IP address. Default = `X-Auth-Source-IP-Address`	STRING	see description	see description

central.conf Parameters (Part 14 of 47)

Parameter	Valid	File	Internal
Description	**Range**	**Default**	**Default**
`m2ci_max_content_param`			
The property name for the maximum content length. Default = `maxclen`	STRING	see description	see description
`m2ci_max_content_default`			
The default value for the maximum content length, -1 means there is no limit.	INT	-1	-1
`m2ci_max_content_status`			
HTTP status code showing the maximum content length.	INT (must be a valid HTTP response code)	512	512
`yaws_maxconn`			
The maximum number of yaws HTTP connections. Default = `10000`	INT	10000	10000
`yaws_timeout`			
Yaws HTTP timeout in milleseconds. Default = `60000`	TIME	60000	60000
`m2ci_post_size_too_big`			
HTTP response code when the post body size too big.	INT (must be a valid HTTP response code)	400	400

central.conf Parameters (Part 15 of 47)

Parameter Description	Valid Range	File Default	Internal Default
m2ci_vcard_quota_http_err			
This HTTP status code is returned to the client when they try to import more vcards than is allowed by the quota. The body of the message will contain the number of vcards imported in both quota exceeded and successful cases.	TIME	413	413
m2ci_yaws_port1			
Port number for the yaws administration listener.	INT	7590	7590
m2ci_yaws_host1			
HTTP listener for the yaws administration listener. Default = localhost	STRING	see description	see description
m2ci_yaws_authmod1			
The internal module name for authentication checking the uraid though the A2S interface. Default = api_nttr_auth_idc Internal default = api_m2fe_auth	ATOM	see description	see description
m2ci_yaws_authproxy1			
If true, enables m2ci_yaws_auth_source_ip for source IP authentication.	BOOL	false	false
m2ci_yaws_appmod1			
The internal UBF module name. Default = ubf_m2ci_plugin	ATOM	see description	see description
m2ci_max_post_size1			
The maximum HTTP body size for POST requests.	INT	-1	-1

central.conf Parameters (Part 16 of 47)

Parameter Description	Valid Range	File Default	Internal Default
`m2ci_yaws_port2`			
Port number for the yaws restricted client listener, non-proxied.	INT	`7594`	`7594`
`m2ci_yaws_host2`			
HTTP listener for the yaws restricted client listener, non-proxied. Default = `localhost`	STRING	see description	see description
`m2ci_yaws_authmod2`			
The internal module name for authentication for the restricted client listener, non-proxied. Default = `api_nttr_auth_idc` Internal default = `api_m2fe_auth`	ATOM	see description	see description
`m2ci_yaws_authproxy2`			
If true, enables `m2ci_yaws_auth_source_ip` for source IP authentication for the restricted client listener, non-proxied.	BOOL	false	false
`m2ci_yaws_appmod2`			
The internal UBF module name for the restricted client listener, non-proxied. Default = `ubf_m2ci_ruser_plugin`	ATOM	see description	see description
`m2ci_max_post_size2`			
The maximum HTTP body size for POST requests for the restricted client listener, non-proxied.	INT	-1	-1
`m2ci_yaws_port3`			
Port number for the yaws restricted client listener, proxied.	INT	`7595`	`7595`

central.conf Parameters (Part 17 of 47)

Parameter Description	Valid Range	File Default	Internal Default
`m2ci_yaws_host3`			
HTTP listener for the yaws restricted client listener, proxied. Default = `localhost`	STRING	see description	see description
`m2ci_yaws_authmod3`			
The internal module name for authentication for the yaws restricted client listener, proxied. Default = `api_nttr_auth_idc` Internal default = `api_m2fe_auth`	ATOM	see description	see description
`m2ci_yaws_authproxy3`			
If true, enables `m2ci_yaws_auth_source_ip` for source IP authentication for the yaws restricted client listener, proxied.	BOOL	true	false
`m2ci_yaws_appmod3`			
The internal UBF module name for the yaws restricted client listener, proxied. Default = `ubf_m2ci_ruser_plugin`	ATOM	see description	see description
`m2ci_max_post_size3`			
The maximum HTTP body size for POST requests for the yaws restricted client listener, proxied.	INT	-1	-1
`m2ci_yaws_port4`			
Port number for the yaws restricted administration listener.	INT	`7596`	`7596`
`m2ci_yaws_host4`			
HTTP listener for the yaws restricted administration listener. Default = `localhost`	STRING	see description	see description

central.conf Parameters (Part 18 of 47)

Parameter Description	Valid Range	File Default	Internal Default
`m2ci_yaws_authmod4`			
The internal module name for authentication for the yaws restricted administration listener. Default = `api_nttr_auth_idc` Internal default = `api_m2fe_auth`	ATOM	see description	see description
`m2ci_yaws_authproxy4`			
If true, enables `m2ci_yaws_auth_source_ip` for source IP authentication for the yaws restricted administration listener.	BOOL	false	false
`m2ci_yaws_appmod4`			
The internal UBF module name for the yaws restricted administration listener. Default = `ubf_m2ci_radmin_plugin`	ATOM	see description	see description
`m2ci_max_post_size4`			
The maximum HTTP body size for POST requests for the yaws restricted administration listener.	INT	-1	-1
`m2ci_max_timeout`			
In seconds, this value limits the caller's timeout() to a configured maximum. This timeout is applied to all of the m2ci JSON HTTP interfaces, the admin, restricted admin, and restricted user, as well as the m2ci EBF and UBF listeners. There is no restriction on the limit.	TIME	300	60
`m2fe_ebf_port`			
Used internally. Migration interface port number for Erlang Binary Format.	INT	7577	7577

central.conf Parameters (Part 19 of 47)

Parameter Description	Valid Range	File Default	Internal Default
`m2fe_ebf_maxconn`			
Used internally. Migration interface for the number of maximum connections for Erlang Binary Format.	INT	10000	10000
`m2fe_ebf_timeout`			
Used internally. Migration interface timeout for Erlang Binary Format.	INT	300	300
`m2fe_ubf_port`			
Used internally. Migration interface port number for Universal Binary Format.	INT	7572	7572
`m2fe_ubf_maxconn`			
Used internally. Migration interface for the number of maximum connections for Universal Binary Format.	INT	10000	7572
`m2fe_ubf_timeout`			
Used internally. Migration interface timeout for Universal Binary Format.	INT	300	300
`m2fe_m2fe_xss_re`			
Used internally. Do not change. Default = `bin/init.d/m2h/regex.euc`	STRING	see description	see description
`m2fe_filter_skip`			
Location of the `skip.euc` file. Used internally. Do not change. Default = `bin/init.d/m2h/skip.euc`	STRING	see description	see description

***central.conf* Parameters (Part 20 of 47)**

Parameter Description	Valid Range	File Default	Internal Default
`m2fe_reply_supress`			
Location of the `reply_supress.euc` file. Used internally. Do not change. Default = `bin/init.d/m2h/reply_supress.euc`	STRING	see description	see description
`m2fe_default_filters`			
Location of the `default_filters.euc` file. Used internally. Do not change. Default = `bin/init.d/m2h/default_filters.euc`	STRING	see description	see description
`m2fe_default_mail_profile`			
Location of the `default_mail_profile.euc` file. Used internally. Do not change. Default = `bin/init.d/m2h/default_mail_profile.euc`	STRING	see description	see description
`m2fe_default_quota_policy`			
Location of the `default_quota_policy.euc` file. Used internally. Do not change. Default = `bin/init.d/m2h/default_quota_policy.euc`	STRING	see description	see description
`m2fe_default_ui_properties`			
Location of the `default_ui_properties.euc` file. Used internally. Do not change. Default = `bin/init.d/m2h/default_ui_properties.euc`	STRING	see description	see description

central.conf Parameters (Part 21 of 47)

Parameter Description	Valid Range	File Default	Internal Default
`m2fe_other_headers`			

A mail summary contains a list of certain well defined headers such as to/from/cc etc. The client application is sometimes interested in "other" headers that a mail might contain. These "other" headers can then also be kept with the normal mailheaders in the summary so that the client can easily reference their existence and their values.
 	
`m2fe_other_headers` controls what extra other headers to keep. The application default is to keep no extra headers. If set, the parameter must only use either an empty list or a list of erlang strings as its value. The erlang strings are complete header names. The case of the header does not matter.
 	
Examples:
 	
`m2fe_other_headers: []`
`m2fe_other_headers: ["X_REDMAIL-HEAD"]`
`m2fe_other_headers: ["X_REDMAIL-HEAD", "User-Agent"]`
 	
Default = `["X-REDMAIL-HEAD"]`
	STRING	see description	see description

`m2fe_access_file`			
Location of the default admin password file that stores the basic authentication passwords for all of m2fe HTTP API. The value of this file should be changed for security purposes. Default = `/usr/local/gemini/m2h/1.0.0/etc/.gmtpasswd`	STRING	see description	see description
`m2fe_mail_authmod`			
The module used for authentification. Default=`api_nttr_auth_idc`	ATOM	see description	see description

central.conf Parameters (Part 22 of 47)

Parameter	Valid	File	Internal
Description	Range	Default	Default
`m2fe_maillist_maxuids_perses`			
Configurable limit for the maximum number of uids permitted per mail list session. A value of negative one (-1) is unlimited.	INT	-1	-1
`m2fe_maillist_maxsess_peruser`			
Configurable limit for the maximum number of simultaneous mail list sessions per user. A value of negative one (-1) is unlimited.	INT	-1	-1
`m2fe_etq_rate_control_ready`			
Enables or disables the rate control for etq ready operations, the Front End Erlang Term Queue, i.e., the M2H job queue. Negative one (-1) is unlimited.	on or off	on	off
`m2fe_etq_rate_control_append`			
Enables or disables the rate control for etq append operations, the Front End Erlang Term Queue, i.e., the M2H job queue. Negative one (-1) is unlimited.	on or off	on	off
`m2fe_etq_rate_control_append_timeout`			
This is the default timeout to wait for tickets for appending operations to the job queues. When there is congestion this means that after this number of seconds, the operation will return "restricted". The configured default is 5000 which is 5 seconds. Value takes a timeout in milliseconds.	TIME	5000	5000
`m2fe_external_fetch_max`			
The maximum number of messages to fetch for a single fetch job from the external ISP.	INT > 0	1000	1000

central.conf Parameters (Part 23 of 47)

Parameter Description	Valid Range	File Default	Internal Default
`m2fe_xconv_ebf_connpool.host`			
Connection pool host name for the M2G that provides a charset and emoji conversion service for the M2H. The M2H uses this service for mail retrieval and mail listing purposes. M2G is the server and M2H is client.	STRING	localhost	localhost
`m2fe_xconv_ebf_connpool.port`			
Port for the connection pool that provides a charset and emoji conversion service for M2H. M2G is server and M2H is client.	INT > 0	7571	7571
`m2fe_xconv_ebf_connpool.options`			
Constant values. Do not change. Default = `[{proto,ebf}]`	TEXT	see description	see description
`m2fe_xconv_ebf_connpool.serverhello`			
Constant values. Do not change. Default = `meta_server`	TEXT	see description	see description
`m2fe_xconv_ebf_connpool.service`			
Constant values. Do not change. Default = `char`	TEXT	see description	see description
`m2fe_xconv_ebf_connpool.timeout`			
Client-side timeout interval for accessing the M2G, in seconds. If a M2H request to the M2G does not receive a response within this time, the M2H closes the connection to the M2G.	TIME	270	30

central.conf Parameters (Part 24 of 47)

Parameter Description	Valid Range	File Default	Internal Default
`m2fe_xconv_ebf_connpool.maxuses`			
In the connection pool to the M2G, the maximum number of times to use a single connection. After a connection has been used this many times, the M2H closes the connection rather than placing it back in the pool.	INT, infinity	1000	infinity
`m2fe_xconv_ebf_connpool.controlhigh`			
In the connection pool to the M2G, the maximum allowed number of simultaneously open connections. If the pool reaches a state where this many connections are open to the M2G, then the M2H temporarily stops creating new connections to the M2G. No new connections will be created until the number of open connections falls below the value of the `m2fe_xconv_ebf_connpool.controllow` setting. Once the number of connections falls below, the M2H resumes creating new connections as needed. If a request for a connection comes into the pool at a time when all open connections are in use and no new connections are being created due to the `m2fe_xconv_ebf_connpool.controlhigh` limit, then the request may time out, depending on the transaction timeout interval applicable to the request.	INT, infinity	750	infinity
`m2fe_xconv_ebf_connpool.controllow`			
See the previous description of `m2fe_xconv_ebf_connpool.controlhigh.`	INT, infinity	725	infinity
`m2fe_xconv_ebf_connpool.maxspare`			
In the connection pool to the M2G, the maximum number of idle connections to keep open simultaneously. Idle or “spare” open connections are those that are not currently being used for service requests. At any given moment, the connection pool will consist of a mix of connections that are busy servicing requests and connections that are idle and available for use.	INT	225	0

central.conf Parameters (Part 25 of 47)

Parameter Description	Valid Range	File Default	Internal Default
`m2fe_xconv_ebf_connpool.minspare`			
In the connection pool to the M2G, the minimum number of idle connections to keep open simultaneously.	INT	15	0
`m2fe_xconv_ebf_connpool.maxkeepalives`			
In the connection pool to the M2G, the maximum number of consecutive keep-alive intervals for which to keep an idle connection open. If an open connection remains idle for this many consecutive keep-alive intervals, the M2H closes the connection. For example, if `m2fe_xconv_ebf_connpool.maxkeepalives` is set to 5, then when an open connection has been idle for a fifth consecutive keep-alive interval, the M2H will close the connection rather than performing another keep-alive test on the connection.	INT, infinity	5	infinity
`m2fe_xconv_ebf_connpool.keepalive`			
In the connection pool to the UDB, the interval at which to perform keep-alive tests on an idle open connection, in seconds. The keep-alive tests ensure that a query can still be successfully sent to the UDB over the idle connection.	INT, infinity	60	infinity
`m2fe_deprov_retry_interval`			
Retry interval in millisecond for JOBQ jobs. This parameter is used when deprovisioning job has an error and will be retried. The file and internal defaults are both 30000 millisecond which equals 30 seconds. Zero (0) means immediate retry.	INT >-1	30000	30000

central.conf Parameters (Part 26 of 47)

Parameter	Valid	File	Internal
Description	**Range**	**Default**	**Default**
`m2fe_deprov_max_retries`			
The maximum number of retries for JOBQ jobs. This parameter is used when deprovisioning job has an error and will be retried. If the number of retries exceed this value then remove the job from the JOBQ and log error message. Zero (0) means no retry. Negative one (-1) means unlimited. The second try is the first retry if the maximum retires is five (5) then the indexer client will try six (6) times at most.	INT >-1 or -1	-1 (times)	-1 (times)
`m2fe_deprov_batch_size`			
The batch size for deprovisioning a user's items. Batches are repeated until all of a user's items have been deleted.	INT > 0	2500	2500
`m2fe_deprov_num_workers`			
The number of indexer workers each worker can do deprovisioning jobs independently.	INT	3	3
`m2si_ebf_port`			
Erlang binary format interface port to the NTTR indexer.	INT > 0	7584	7584
`m2si_ebf_maxconn`			
Maximum number of connections to the NTTR indexer .	INT	10000	10000
`m2si_ebf_timeout`			
Timeout for connection to the NTTR Indexer.	TIME	300	300
`m2be_ebf_host`			
The host for the m2be_ebf connection which is between m2fe and m2be. This communication is via TCP/IP using EBF (erlang binary format). The communication between M2BE and GDSS uses the native Erlang distribution method. Default = localhost	STRING	see description	see description

central.conf Parameters (Part 27 of 47)

Parameter Description	Valid Range	File Default	Internal Default
`m2be_ebf_port`			
Back end interface port to GDSS.	INT > 0	7576	7576
`m2be_ebf_maxconn`			
M2H's "back-end" max imum number of connections to the GDSS.	INT	10000	10000
`m2be_ebf_timeout`			
M2H's "back-end" timeout for connections to the GDSS.	TIME	300	300
`m2be_ebf_connpool.host`			
Host for the M2H Backend Erlang Binary Format connection. Default = localhost	STRING	see description	see description
`m2be_ebf_connpool.port`			
Port for the M2H Backend Erlang Binary Format connection.	INT	7576	7576
`m2be_ebf_connpool.options`			
Do not change. Default = `[{proto,ebf}]`	TEXT	see description	see description
`m2be_ebf_conn.serverhello`			
Do not change. Default = `m2be_meta_server`	TEXT	see description	see description

central.conf Parameters (Part 28 of 47)

Parameter Description	Valid Range	File Default	Internal Default
`m2be_ebf_conn.service`			
Constant values. Do not change.	TEXT	`m2be`	`m2be`
`m2be_ebf_connpool.timeout`			
Clinet side connection timeout from the M2 Backend Erlang Binary Format connection pool.	TIME	270	270
`m2be_ebf_connpool.maxuses`			
In the connection pool to the M2 Back End Erlang Binary Format server, the maximum number of times to use a single connection. After a connection has been used this many times, the M2H closes the connection rather than placing it back in the pool.	INT, infinity	1000	infinity
`m2be_ebf_connpool.controlhigh`			
High water mark for congestion control for receiving messages from the M2BE listener connection. When the number of messages in queue rises above `controlhigh`, a message is recorded to the application log, and interfaces that you have specified with the `controlinterfaces` setting are shut down. Messages will not be processed out of the queue until you initiate batch processing. The closed interfaces reopen when batch processing reduces the number of queued messages below `controllow`. This setting must be at least as high as `controlwarn`. To disable congestion control for the queue processor, set `controllow` and `controlhigh` to 0. NOTE: This setting can be changed through the CLI only if it is currently set to a non-zero value.	INT (0 to INT_MAX)	750	250

central.conf Parameters (Part 29 of 47)

Parameter Description	Valid Range	File Default	Internal Default
`m2be_ebf_connpool.controllow`			
Congestion control low water mark for the M2BE listener, in number of open connections. The low water mark serves two purposes: ◆ Triggers congestion warning message. When the number of open client connections to the listener rises above `controllow,` a message is written to the application log. ◆ Triggers reopening of closed interface. In the event that the number of open client connections to the listener rises past `controlhigh`, resulting in the closing of the interface to new connections, the listener is reopened when the number of connections falls back below `controllow`.	INT (0 to INT_MAX)	725	250
`m2be_ebf_connpool.maxspare`			
Tool for managing the number of idle connections in the pool. After using a connection to complete a session with the target server, the Back End EBF job queue processing server either: ◆ Closes the connection if the current number of idle connections in the pool is greater than or equal to `maxspare` ◆ Puts the connection back into the connection pool if the current number of idle connections in the pool is less than `maxspare`. To disable the maximum spare connections limit, set this parameter to 0.	min-spare to max-connect-ions	225	100

central.conf Parameters (Part 30 of 47)

Parameter Description	Valid Range	File Default	Internal Default
`m2be_ebf_connpool.minspare`			
Tool for managing the number of idle connections in the Back End EBF job queue pool. At your specified keep-alive period for idle connections, the server either: ◆ Closes the connection, if the current number of idle connections in the pool is greater than `minspare`; or ◆ Sends a keep-alive signal through the connection, if the current number of idle connections in the pool is less than or equal to `minspare`. If the keep-alive test succeeds, the connection remains open and in the pool; if the test fails, the connection is closed. To disable the minimum spare connections limit, set this parameter to 0.	0 to `max-spare`	15	5
`m2be_ebf_connpool.maxkeepalives`			
In the connection pool to the M2G, the maximum number of consecutive keep-alive intervals for which to keep an idle connection open. If an open connection remains idle for this many consecutive keep-alive intervals, the M2H closes the connection. For example, if `m2be_ebf_connpool.maxkeepalives` is set to 5, then when an open connection has been idle for a fifth consecutive keep-alive interval, the M2H will close the connection rather than performing another keep-alive test on the connection.	INT to infinity	5	infinity
`m2be_ebf_connpool.keepalive`			
In the connection pool to the m2BE, the interval at which to perform keep-alive tests on an idle open connection, in seconds. The keep-alive tests ensure that a query can still be successfully sent to the M2BE over the idle connection.	INT to infinity	60	infinity

central.conf Parameters (Part 31 of 47)

Parameter Description	Valid Range	File Default	Internal Default
`m2be_mail_singleton_limit`			
Temporarily set to a large number until the mail delete implementation is complete. The deletion of mail summaries is currently only implemented for singleton mails. Default = `100000000`	INT	see decrip-tion	see decrip-tion
`m2be_mail_shortcutdel_maxlen`			
Internal tuning parameter for message deletion operations.	INT > 0	20	20
`m2be_maillist_tracedump`			
Enables/disables dumping these trace files to the `/tmp` directory. This feature is intended only as a debugging facility and is highly NOT recommended for use in production service.	BOOL	false	false
`m2be_maillist_idletimeout`			
Configurable idle timer (in seconds) to automatically stop a mail list session. The idle timer is reset by a mail list *get*.	TIME	300	300
`m2be_maillist_stoptimeout`			
Configurable maximum session time (in seconds) to automatically stop a mail list session. The stop timer is not resetable.	TIME	1200	1200

central.conf Parameters (Part 32 of 47)

Parameter Description	Valid Range	File Default	Internal Default
`m2be_maillist_maxsess`			
Limits the number of simultaneous sessions per m2be node. Negative one (-1) means unlimited. If one of these limits: `m2be_maillist_maxsess`, `m2be_maillist_maxuids`, `m2be_maillist_numuids` is reached, new mail list sessions are rejected and the 'restricted' return value is returned until all values fall below the configurable limit. Existing mail list sessions are not affected if a limit is reached. The purpose of these restrictions is to indirectly protect the memory usage of a single m2be node by preventing too many mail list sessions from being created.	INT	-1	-1
`m2be_maillist_maxuids`			
Configurable limit for the maximum number of estimated uids permitted per mail list session. A value of negative one (-1) means unlimited. If one of these limits: `m2be_maillist_maxsess`, `m2be_maillist_maxuids`, `m2be_maillist_numuids` is reached, new mail list sessions are rejected and the 'restricted' return value is returned until all values fall below the configurable limit. Existing mail list sessions are not affected if a limit is reached. The purpose of these restrictions is to indirectly protect the memory usage of a single m2be node by preventing too many mail list sessions from being created.	INT	-1	-1

central.conf Parameters (Part 33 of 47)

Parameter Description	Valid Range	File Default	Internal Default
`m2be_maillist_numuids`			
Configurable limit of the actual number of uids per m2be node. A value of negative one (-1) means unlimited. If one of these limits: `m2be_maillist_maxsess`, `m2be_maillist_maxuids`, `m2be_maillist_numuids` is reached, new mail list sessions are rejected and the 'restricted' return value is returned until all values fall below the configurable limit. Existing mail list sessions are not affected if a limit is reached. The purpose of these restrictions is to indirectly protect the memory usage of a single m2be node by preventing too many mail list sessions from being created.	INT	-1	-1
`m2be_maillist_danglinguids`			
As a side-effect of mail listing, the m2be maillist driver asynchronously cleans the mail label store of dangling uids. A dangling uid is a uid that no longer exists in the mail store but is still present in the mail label store. This parameter enables asynchronous dangling uid cleanup for mail labels.	BOOL	true	false
`m2be_force_mail_folder_table_to_disk`			
If present and equal to "true", force this table to disk. The values in the "mail_folder" table will be stored on disk instead of in RAM. The value of this setting only applies at table creation time. Check the `http://{host}:23080/ summary "Brick Options"` to confirm the table's parameters after creation time.	BOOL	false	false

central.conf Parameters (Part 34 of 47)

Parameter Description	Valid Range	File Default	Internal Default
`m2be_force_mail_label_table_to_disk`			
If present and equal to "true", force this table to disk. The values in the "mail_folder" table will be stored on disk instead of in RAM. The value of this setting only applies at table creation time. Check the `http://{host}:23080/ summary "Brick Options"` to confirm the table's parameters after creation time.	BOOL	false	false
`m2be_force_profile_store_table_to_disk`			
If present and equal to "true", force this table to disk. The values in the "mail_folder" table will be stored on disk instead of in RAM. The value of this setting only applies at table creation time. Check the `http://{host}:23080/ summary "Brick Options"` to confirm the table's parameters after creation time.	BOOL	false	false
`m2be_force_vcard_label_table_to_disk`			
If present and equal to "true", force this table to disk. The values in the "mail_folder" table will be stored on disk instead of in RAM. The value of this setting only applies at table creation time. Check the `http://{host}:23080/ summary "Brick Options"` to confirm the table's parameters after creation time.	BOOL	false	false
`m2be_force_vcard_store_table_to_disk`			
If present and equal to "true", force this table to disk. The values in the "mail_folder" table will be stored on disk instead of in RAM. The value of this setting only applies at table creation time. Check the `http://{host}:23080/ summary "Brick Options"` to confirm the table's parameters after creation time.	BOOL	false	false

central.conf Parameters (Part 35 of 47)

Parameter Description	Valid Range	File Default	Internal Default
`m2be_label_max_deltas`			
The maximum length of the M2BE label deltas lists. This number must be non-negative.	INT >0	5	5
`a2s_ebf_connpool.host`			
Host name for the A2S Erlang Binary Format connection pool.	STRING	localhost	localhost
`a2s_ebf_connpool.port`			
Port for the A2S Erlang Binary Format connection pool.	INT > 0	7575	7575
`a2s_ebf_connpool.options`			
Constant values. Do not change. Default = `[{proto,ebf}]`	TEXT	see description	see description
`a2s_ebf_connpool.serverhello`			
Constant values. Do not change. Default = `a2s_meta_server`	TEXT	see description	see description
`a2s_ebf_connpool.service`			
Constant values. Do not change. Default = `a2s`	TEXT	see description	see description
`a2s_ebf_connpool.timeout`			
Client-side timeout interval for accessing the A2s, in seconds. If a M2H request to the A2s does not receive a response within this time, the M2H closes the connection to the A2s.	TIME	270	270

central.conf Parameters (Part 36 of 47)

Parameter Description	Valid Range	File Default	Internal Default
`a2s_ebf_connpool.maxuses`			
In the connection pool to the A2s, the maximum number of times to use a single connection. After a connection has been used this many times, the M2H closes the connection rather than placing it back in the pool.	INT, infinity	1000	infinity
`a2s_ebf_connpool.controlhigh`			
In the connection pool to the A2S, the maximum allowed number of simultaneously open connections. If the pool reaches a state where this many connections are open to the A2S, then the M2H temporarily stops creating new connections to the A2S. No new connections will be created until the number of open connections falls below the value of the `a2s_ebf_connpool.controllow` setting. Once the number of connections falls below, the M2H resumes creating new connections as needed. If a request for a connection comes into the pool at a time when all open connections are in use and no new connections are being created due to the `a2s_ebf_connpool.controlhigh` limit, then the request may time out, depending on the transaction timeout interval applicable to the request.	INT, infinity	750	infinity
`a2s_ebf_connpool.controllow`			
See the previous description of `a2s_ebf_connpool.controlhigh.`	INT, infinity	725	infinity
`a2s_ebf_connpool.maxspare`			
In the connection pool to the A2S, the maximum number of idle connections to keep open simultaneously. Idle or “spare” open connections are those that are not currently being used for service requests. At any given moment, the connection pool will consist of a mix of connections that are busy servicing requests and connections that are idle and available for use.	INT	225	0

central.conf Parameters (Part 37 of 47)

Parameter Description	Valid Range	File Default	Internal Default
`a2s_ebf_connpool.minspare`			
In the connection pool to the A2S, the minimum number of idle connections to keep open simultaneously.	INT	15	0
`a2s_ebf_connpool.maxkeepalives`			
In the connection pool to the A2S, the maximum number of consecutive keep-alive intervals for which to keep an idle connection open. If an open connection remains idle for this many consecutive keep-alive intervals, the M2H closes the connection. For example, if `a2s_ebf_connpool.maxkeepalives` is set to 5, then when an open connection has been idle for a fifth consecutive keep-alive interval, the M2H will close the connection rather than performing another keep-alive test on the connection.	INT, infinity	5	infinity
`a2s_ebf_connpool.keepalive`			
In the connection pool to the UDB, the interval at which to perform keep-alive tests on an idle open connection, in seconds. The keep-alive tests ensure that a query can still be successfully sent to the UDB over the idle connection.	INT, infinity	60	infinity
`m2h_ebf_port`			
For internal use only. Do not change without first consulting Gemini.	INT	7583	7583
`m2h_ebf_maxconn`			
For internal use only. Do not change without first consulting Gemini.	INT	10000	10000
`m2h_ebf_timeout`			
For internal use only. Do not change without first consulting Gemini.	INT	60	10000

central.conf Parameters (Part 38 of 47)

Parameter Description	Valid Range	File Default	Internal Default
`m2h_ubf_port`			
For internal use only. Do not change without first consulting Gemini.	INT	7589	7589
`m2h_ubf_maxconn`			
For internal use only. Do not change without first consulting Gemini.	INT	10000	10000
`m2h_ubf_timeout`			
For internal use only. Do not change without first consulting Gemini.	INT	60	60
`address_ext_reg_max_tries`			
When a user registers an email address, the system send an unique password for the login to that email address. This parameter is the number of retries allowed to type in the passowrd. Over this number and the user must re-register the email address.	INT	10	10
`gms_imapd_conf`			
IMAP is not used in this version of the product. Do not modify or delete this value.	STRING	/dev/null	/dev/null
`nttr_max_mig_uid`			
Custom NTTR for provisioning the maximum migration uid (user ID). This parameter is used at the time of user provisioning to enable/disable mail migration. Mail migration is a process used to relocate a user's mail from the old mail system to our mail system. The default value is 0 and the default of zero indirectly signals the end of migration processing.	INT	0	0

central.conf Parameters (Part 39 of 47)

Parameter Description	Valid Range	File Default	Internal Default
`custom_xconv_module`			
Custom character set conversion. M2G provides a charset and emoji conversion service for M2H. M2H uses this service for mail retrieval and mail listing purposes. Default = `api_m2fe_xconv`	ATOM	see description	see description
`nttr_yaws_port1`			
TCP port number for NTTR custom admin listener.	INT	7587	7587
`nttr_yaws_host1`			
Host name for NTTR custom Admin Listener. Default = `localhost` Internal default = `hostname of the system`	STRING	see description	see description
`nttr_max_post_size1`			
The maximum size of the post body for the NTTR custom administration listener. Negative one (-1) means no limit.	INT	-1	-1
`nttr_yaws_port2`			
TCP port number for the EVA NTTR custom restoration listener.	INT	7588	7588
`nttr_yaws_host2`			
Host name for the EVA NTTR custom restoration listener. Default = `localhost` Internal default = `hostname of the system`	STRING	see description	see description
`nttr_max_post_size2`			
The maximum size of the post body for the NTTR restoration listener. Negative one (-1) means no limit.	INT	-1	-1

central.conf Parameters (Part 40 of 47)

Parameter Description	Valid Range	File Default	Internal Default
`m2fe_xform_body_mod`			
For body parse of NTTR custom. Internal use only. Do not change. Default = `api_nttr_body_parse`	ATOM	see description	see description
`nttr.body_parse_engine_mobile_url`			
URL for body parse for mobile. Default = `http://localhost:7680/text_analyze_mobile`	STRING	see description	see description
`nttr.body_parse_engine_pc_url`			
URL for body parse for PC Web UI. Default = `http://localhost:7680/text_analyze_pc`	STRING	see description	see description
`nttr.idc1_name`			
Goo mail component for the authorization ID Center for NTTR customization.	STRING	goo	goo
`nttr.idc1_webui_auth_url`			
The URL for authentication for the goo Web UI. This is a mandatory parameter. Default = `http://localhost:7680/idc/goo/webui_auth`	STRING	see description	“”
`nttr.idc1_webui_cookie`			
Cookie name for the goo Web UI. This is a mandatory parameter.	STRING	NGID	“”

central.conf Parameters (Part 41 of 47)

Parameter Description	Valid Range	File Default	Internal Default
`nttr.idc1_webui_key_a`			
The type A encryption key for cookie for Web UI. This should be 32 hex characters. This will be 16 byte binary data. Default = `aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa` Internal default = `aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa`	STRING	see description	see description
`nttr.idc1_webui_key_b`			
The type B encryption key for cookie for the goo Web UI. This should be 32 hex characters. This will be 16 byte binary data. Default = `bbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbb` Internal default = `bbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbb`	STRING	see description	see description
`nttr.idc1_webui_site`			
The site parameter for the goo mail Web UI. Default = `goo_site`	STRING	see description	aa
`nttr.idc1_webui_auth_enable`			
Enables authentication for the goo mail Web UI. ◆ If true, the M2H requests authentication HTTP to IDC. ◆ If false the M2H doesn’t request authentication HTTP to IDC .	BOOL	true	true

central.conf Parameters (Part 42 of 47)

Parameter Description	Valid Range	File Default	Internal Default
`nttr.idc1_pcmc_auth_url`			
The URL for authentication for goo mail PC mail client. This is a mandatory parameter. Default = `http://localhost:7680/idc/goo/pcmail_auth`	STRING	see description	“”
`nttr.idc1_pcmc_prod_code_a`			
The product code A for the goo PC mail client. Default = `goo_prod_a`	STRING	see description	“proda”
`nttr.idc1_pcmc_prod_code_b`			
The product code B for the goo PC mail client. Default = `goo_prod_b`	STRING	see description	“prodb”
`nttr.idc1_pcmc_src_ip_list`			
The source IP list to determine to use product code B if the source IP is in this list then M2H will use product code B otherwise, it will use product code A. The format is comma separated IP address list without any space character. For example: 10.1.0.1,10.1.0.2 10.1.0.1, 10.1.0.2 —INVALID space should be removed Default = `127.0.1.1,127.0.1.2`	STRING	see description	“”
`nttr.idc2_name`			
Red mail component for the authorization ID Center for NTTR customization.	STRING	red	red

central.conf Parameters (Part 43 of 47)

Parameter Description	Valid Range	File Default	Internal Default
`nttr.idc2_webui_auth_url`			
The URL for authentication for the goo Web UI. This is a mandatory parameter. Default = `http://localhost:7680/idc/red/webui_auth`	STRING	see description	“”
`nttr.idc2_webui_cookie`			
Cookie name for the red mail Web UI. This is a mandatory parameter.	STRING	RID	“”
`nttr.idc2_webui_key_a`			
The type A encryption key for cookie for red mail Web UI. This should be 32 hex characters. This will be 16 byte binary data. Default = `aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa` Internal default = `aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa`	STRING	see description	see description
`nttr.idc2_webui_key_b`			
The type B encryption key for cookie for red mail Web UI. This should be 32 hex characters. This will be 16 byte binary data. Default = `bbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbb` Internal default = `bbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbb`	STRING	see description	see description
`nttr.idc2_webui_site`			
The site parameter for the red mail Web UI. Default = `red_site`	STRING	see description	aa

central.conf Parameters (Part 44 of 47)

Parameter Description	Valid Range	File Default	Internal Default
`nttr.idc2_webui_auth_enable`			
Enables authentication for the red mail Web UI. ◆ If true, the M2H requests authentication HTTP to IDC. ◆ If false the M2H doesn't request authentication HTTP to IDC .	BOOL	true	true
`nttr.idc2_pcmc_auth_url`			
Default = `http://localhost:7680/idc/red/pcmail_auth`	STRING	see description	""
`nttr.idc2_pcmc_prod_code_a`			
The product code A for the red PC mail client. Default = `red_prod_a`	STRING	see description	"proda"
`nttr.idc2_pcmc_prod_code_b`			
The product code B for the red PC mail client. Default = `red_prod_b`	STRING	see description	"prodb"
`nttr.idc2_pcmc_src_ip_list`			
The source IP list to determine to use product code B if the source IP is in this list then M2H will use product code B otherwise, it will use product code A. The format is comma separated IP address list without any space character. For example: 10.1.0.1,10.1.0.2 10.1.0.1, 10.1.0.2 —INVALID space should be removed Default = `127.0.2.1,127.0.2.2`	STRING	see description	""

central.conf Parameters (Part 45 of 47)

Parameter	Valid	File	Internal
Description	Range	Default	Default
`nttr.idc_error_map_1`			
This parameter defines the HTTP response to the IDC web UI when the error code for Authn is 1.	INT	492	503
`nttr.idc_error_map_2`			
This parameter defines the HTTP response to the IDC web UI when the error code for Authn is 2.	INT	493	503
`nttr.idc_error_map_3`			
This parameter defines the HTTP response to the IDC web UI when the error code for Authn is 3.	INT	494	503
`nttr.idc_error_map_4`			
This parameter defines the HTTP response to the IDC web UI when the error code for Authn is 4.	INT	491	503
`nttr.idc_error_map_5`			
This parameter defines error code when Authn is 5.	INT	495	503
`nttr.idc_error_map_6`			
This parameter defines error code when Authn is 6.	INT	495	503
`m2si_retry_interval`			
Retry interval in millisecond for JOBQ jobs. This parameter is used when transactions to the NTTR indexer get an error that needs to be retried. The default is 1000 (millisecond) (= 1 second). Zero (0) means immediate retry.	INT (non-negative)	1000	1000

central.conf Parameters (Part 46 of 47)

Parameter	Valid	File	Internal
Description	**Range**	**Default**	**Default**
`m2si_max_retries`			
The maximum number of retries for JOBQ jobs. This parameter is used when transactions to the NTTR indexer gets an error to be retried. If the number of retries exceed this value then remove the job from JOBQ and log error message. NOTE: The second try is the first retry. If the number of maximum retires is five (5) times, then the indexer client will try six (6) times at most. Zero (0) means no retry. The second try is the first retry. For example, if `m2si_max_retries` is five (5) then the indexer client will try six (6) times at most.	INT (non-negative)	5	3
`m2si_num_workers`			
The number of indexer workers. Each worker can do an independent indexer HTTP request.	INT (non-negative)	5	3
`m2si_indexer_mod`			
This parameter defines the indexer module that should be exported [store/3, label/3, delete/3]. If this parameter is undefined, it means that there is no actual effect. Default = `api_nttr_eva` Internal Default = `undefined` (meaning no actual effect)	ATOM	see description	see description
`m2fe_mail_list_search_mod`			
This parameter defines the mail list search module. The module should export [mail_list_search/3]. If this parameter is undefined, it means that there is no actual effect. Default = `api_nttr_eva` Internal Default = `undefined` (meaning no actual effect)	ATOM	see description	see description

central.conf Parameters (Part 47 of 47)

Parameter Description	Valid Range	File Default	Internal Default
nttr.eva_base_url			
This parameter is the base URL of EVA. If http://some.com/base/ (the last / is important) then: ◆ the actual URL for mail_recv is http://some.com/base/mail_recv ◆ the actual URL for label_chg is http://some.com/base/label_chg ◆ the actual URL for mail_del is http://some.com/base/mail_del ◆ the actual URL for proc_ctrl is http://some.com/base/proc_ctrl ◆ the actual URL for mail_search is http://some.com/base/mail_search Default = http://localhost:7680/ Internal default = http://localhost/	STRING	see description	see description

congestion_watcher.conf

Path `<M2H_HOME>/1.0.0/etc/congestion_watcher.conf`

Purpose Configures congestion monitoring for M2H messaging.

Use the `congestion_watcher.conf` file to configure monitoring of M2H messaging tables for congestion. You can specify congestion levels at which the M2H should either log warning messages or implement temporary messaging restrictions.

For each M2H messaging table that you want to monitor for congestion, enter in the `congestion_watcher.conf` file a line composed of these comma-separated values:

```
<what_to_watch>,<interval>,<enabled>,<low_mark>,
<warn_mark>, <high_mark>,<restrict_who>,<restrict_msec>,
<restrict_what>
```

Each item description to watch is specified on 1 line. An item description is comprised of 9 different comma separated fields. No spaces are allowed around the commas. The following table describes each field.

congestion_watcher.conf Parameters (Part 1 of 2)

Parameter	Description
`<what_to_watch>`	The options for the first field are either `mnesia`, `apply` or `gen_server`. ◆ For `mnesia`—the second field is either `items` or `bytes`. The third field is `table name`. ◆ For `apply` —the second is `module`. The third is `function` with `0 args` ◆ For `gen_server`—the second field is `items`. The third is the `gen_server_process_name`
`<interval>`	The interval in milliseconds at which `to` check this `<what_to_watch>` item for congestion.
`<enabled>`	Whether or not congestion monitoring is enabled for the messaging table specified by `<what_to_watch>`. Options are: ◆ `true` Congestion monitoring is enabled to check this `<what_to_watch>` item for congestion ◆ `false` Congestion monitoring is disabled for the item specified by `<what_to_watch>`.
`<low_water_ mark>`	If congested, the item will return to the uncongested state once the size falls below this mark.

congestion_watcher.conf Parameters (Part 2 of 2)

Parameter	Description
`<warn_water_ mark>`	A warning will be written to the log file once the size of the item reaches this size. No further warning will be issued unless the size falls below the low water mark or rises above the high water mark.
`<high_water_ mark>`	Once the high water mark is reached for this item to be considered congested. A restriction message will be sent to the `<restrict_who>` field.
`<restrict_who>`	The process which needs to receive the restriction messages or undefined for logging only if there is nobody to restrict.
`<restrict_msec>`	Number of milliseconds the `<restrict_who>` process should impose a restriction for before assuming it can continue again.
`<restrict_what>`	A colon separated list of items to restrict.

Example

The sample below shows properly formatted entries for the `congestion_watcher.conf` file. Note that in the actual configuration file, the lines would not wrap the way they do below.

The following line monitors the term queues and stop accepting new jobs if congested:

```
apply:etq_util:my_data_items,5000,true,10000,50000,500000,ticket_m
aker,10000,m2fe_rc_mail_recv:m2fe_rc_mail_relay:m2fe_rc_mail_send:
m2fe_rc_mail_migrate:m2fe_rc_mail_label:m2fe_rc_mail_delete:m2fe_r
c_async_fetch:m2fe_rc_addr_verify:m2fe_rc_user_deprovision
```

The following line monitors erlang processes. The action is pure logging:

```
apply:congestion_util:process_count,5000,true,5000,15000,25000,und
efined,0,undefined
```

The following monitors Erlang process mailbox sizes. The action is pure logging:

```
apply:congestion_util:mbox_count,300000,true,20,50,100,undefined,0
,undefined
```

The following monitors the memory used by erlang processes looking for large sizes:

```
apply:congestion_util:process_maxmem,300000,true,20000000,8000
0000,100000000,undefined,0,undefined
```

default_filters.euc

Path <M2H_HOME>/1.0.0/etc/default_filters.euc

Purpose Determines incoming, outgoing and notification filters for M2H messaging.

Take care before modifiying this file or the system will not start up properly. it is in Erlang syntax and is read in by the M2H at start up time with the file:consult(FileName) function.

Reference the Erlang manual at:

http://www.erlang.org/doc/reference_manual/part_frame.html

For each of the possible {ServiceId,Cos} pairs, give a properties list with {out_filters, [FILTERLIST]}, the filters to be applied to outgoing mail,{in_filters, [FILTERLIST]}, the filters to be applied to incoming mail, {notify_filters, [FILTERLIST]}, the filters to be appended to the in_filters depending on a provisioning parameter "NewArrival" sent in by the IDC.

Example Below is an example of a properly formatted sequence:

```
{{1,free},[ {out_filters, [{{system, human}, true, <<"human">>,
'or',[true],[{x_action, {human_label, [xHumanLabel]}}]}]},
        {in_filters,       [{{system, spam}, true, <<"Spam">>,
'and',[{{header, <<"X-goo-spam">>}, {does, contain},
<<"black">>}],[{set, spam}]},
                         {{system,bw}, true,
<<"BlackWhite_Black">>, 'or', [], [{set, delete}]},
```

default_mail_profile.euc

Path `<M2H_HOME>/1.0.0/etc/default_mail_profile.euc`

Purpose Maps the ServiceID with the terms of service.

Take care before modifiying this file or the system will not start up properly. it is in Erlang syntax and is read in by the M2H at start up time with the file:consult(FileName) function.

Reference the Erlang manual at:

```
http://www.erlang.org/doc/reference_manual/
part_frame.html
```

For each of the possible `{ServiceId,Cos}` pairs, give a properties list of values for the user's mail profile.

`{spam_level,0|1|2|3}` is the value returned to spam filter that determines what to do with mail identified as spam.

`{auto_fetch,Boolean}` says whether or not to automatically fetch messages.

`{max_msg_attach,Int}` determines how many attachments may be sent with a mail.

`{max_msg_bytes,Int}` gives the maximum size in bytes of message including attachments.

Example Below is a properly formatted sequence:

```
{{1,free},[{spam_level,1},{auto_fetch,true},{max_msg_attach,10
},{max_msg_bytes,20971520}]}.
{{1,paying},[{spam_level,1},{auto_fetch,true},{max_msg_attach,
10},{max_msg_bytes,20971520}]}.
{{2,free},[{spam_level,1},{auto_fetch,false},{max_msg_attach,1
0},{max_msg_bytes,20971520}]}.
{{2,paying},[{spam_level,1},{auto_fetch,false},{max_msg_attach
,10},{max_msg_bytes,20971520}]}.
```

default_quota_policy.euc

Path `<M2H_HOME>/1.0.0/etc/default_quota_policy.euc`

Purpose Sets up mapping for each of the possible {ServiceId,Cos} pairs to each quota policy, the named quota policies must be set up separately using the quota policy tool.

Take care before modifiying this file or the system will not start up properly. it is in Erlang syntax and is read in by the M2H at start up time with the file:consult(FileName) function.

Reference the Erlang manual at:

```
http://www.erlang.org/doc/reference_manual/
part_frame.html
```

Example Below is a properly formated file.

```
{{1,free},[{quota_policy,"1"}]}.
{{1,paying},[{quota_policy,"2"}]}.
{{2,free},[{quota_policy,"3"}]}.
```

default_ui_properties.euc

Path `<M2H_HOME>/1.0.0/etc/default_ui_properties.euc`

Purpose For each of the possible {ServiceId,Cos} pairs, give a JSON style proplist of values to be sent to the UI as defaults.

Take care before modifiying this file or the system will not start up properly. it is in Erlang syntax and is read in by the M2H at start up time with the file:consult(FileName) function.

Reference the Erlang manual at:

```
http://www.erlang.org/doc/reference_manual/
part_frame.html
```

Example Below is part of a properly formated file.

```
{{1,free},[{<<"editPref">>,
   {'#P',
    [{<<"quotationFlag">>,true},
    {<<"quotationReply">>,<<">">>},
    {<<"quotationForward">>,<<>>},
    {<<"mailEditor">>,1}]}},
  {<<"signaturePref">>,
```

regex.euc

Path `<M2H_HOME>/1.0.0/etc/regex.euc`

Purpose Configures a list of regular expressions and the replacements for the matching portions of the message. This is used to manipulate HTML mails so, for example, imbedded scripts will not be run by the browser.

Take care before modifiying this file or the system will not start up properly. it is in Erlang syntax and is read in by the M2H at start up time with the file:consult(FileName) function.

Reference the Erlang manual at:

```
http://www.erlang.org/doc/reference_manual/
part_frame.html
```

Example

Below is part of a properly formated file.

```
[{"<script(.*?)>", "<_script\\1>"},
 {"<\/script(.*?)>", "</_script\\1>"},
 {"<img(.+?)src\s*=\s*(\"??)j(.+?)>", "<img\\1src=\\2_j\\3>"},
 {"<a(.+?)href\s*=\s*(\"??)j(.+?)>", "<a\\1href=\\2_j\\3>"},
 {"on(change|mouseover|load|error)", ""},
 {"<style(.*?) type=\"text\/javascript\"(.*?)>", "<style\\1
type=\"text/_javascript\"\\2>"},
 {":expression\\((.+?>)", ":_expression(\\1"},
 {"(\\(\s*)eval(\\(.+?>)", "\\1_eval(\\2"},
 {"@import", "_@import"},
 {"\\\\/", "\\\\ "},
 {"javascript:/","_javascript:"},
 {"xml:namespace", "xml:_namespace"},
 {"\\?import", "?_import"}].
```

reply_supress.euc

Path `<M2H_HOME>/1.0.0/etc/reply_supress.euc`

Purpose Configures do not reply to messages that have a sender matching one of the listed strings. This is to prevent infinite mail loops.

Take care before modifiying this file or the system will not start up properly. it is in Erlang syntax and is read in by the M2H at start up time with the file:consult(FileName) function.

Reference the Erlang manual at:

```
http://www.erlang.org/doc/reference_manual/
part_frame.html
```

Example Below is part of a properly formated file.

```
["autoanswer",
 "echo",
 "list.manager",
 "listproc",
 "listserv",
 "mailer",
 "mailerdaemon",
 "mailer-daemon",
 "majordomo",
 "mirror",
 "netserv",
 "server",
 "uucp"].
```

skip.euc

Path `<M2H_HOME>/1.0.0/etc/skip.euc`

Purpose Proplist of symbolic interface designations to {system, Type} in filters. Each Type refers to the system filters which will be skipped (not run) for messages coming in over the listed interface.

Take care before modifiying this file or the system will not start up properly. it is in Erlang syntax and is read in by the M2H at start up time with the file:consult(FileName) function.

Reference the Erlang manual at:

`http://www.erlang.`

Example This means do not run `{system,bw}, {system,stop}` etc.. filters for messages coming in from Docomo.

```
{s1b, [bw,stop,notify,notify_no_sf,forward,vacation]}
```

sys.conf

For internal user only. Do not change.

9 A2S Configuration Files

This chapter describes the Authentication and Authorization Storage Server (A2S) configuration files and settings.

The chapter covers the following configuration files:

- *central.conf, on page 328*
- *counter_white_black.conf, on page 342*
- *counter.conf, on page 343*

For overviews of how to configure M2H features, see *Chapter 6, M2H, A2S, GDSS Configuration*.

If you want to quickly locate the description of a particular setting that you have seen in the A2S `central.conf` file, you can use *Index of Settings in .properties and .conf Files, on page 415*.

---

central.conf

Path `<A2S_HOME>/1.0.0/etc/central.conf`

Purpose A2S configuration file.

Dynamic Reload You cannot dynamically reload this file. To activate changes that you make to the file, you must restart the A2S.

The table that follows describes each parameter in the `central.conf` file. For background information about how to work with the `central.conf` file, see *page 225*.

central.conf Parameters (Part 1 of 14)

Parameter Description	Valid Range	File Default	Internal Default
`application_home`			
A2S top-level directory. File default = set during install (installer defaults to `/usr/local/gemini/a2s`)	STRING	see description	null

central.conf Parameters (Part 2 of 14)

Parameter Description	Valid Range	File Default	Internal Default
`application_nodename`			
First part of the A2S application node name. Each A2S server is assigned a “node name”. A node name identifies a specific Linux process on a specific physical machine. Different node names can be used to run multiple A2S services on the same physical machine, if desired. An A2S node name has three parts: 1) An application name local to the physical machine. 2) The “@” symbol. 3) The hostname of the physical machine, as shown by the output of the system command `uname -n`. If the hostname as shown by `uname -n` has one or more dots in it (for example `machine1.company.com`) then only the left-most part is used for the A2S node name (from the example, `machine1`). A sample three-part A2S node name is `a2s1@machine1`. The first part of the A2S node name is determined by your `application_nodename` setting—in the sample, `a2s`.	STRING	a2s1	null
`a2s_ebf_host`			
Host name for the A2S EBF listener. Do not change.	TEXT	local	NA
`a2s_ebf_port`			
TCP port number for A2S EBF listener. Do not change.	INT	7575	NA
`a2s_ebf_maxconn`			
Maximum number of connections for A2S EBF listener.	INT	10000	TBD
`a2s_ebf_timeout`			
The timeout for A2S EBF listener in seconds.	TIME	300	TBD

central.conf Parameters (Part 3 of 14)

Parameter Description	Valid Range	File Default	Internal Default
`gldaps_port`			
Port number for the gldaps application.	INT	7574	TBD
`gldaps_timeout`			
Timeout for the gldaps application.	INT	6000	TBD
`gldaps_maxconn`			
Maximum number of connections for the gldaps listener.	INT	10000	TBD
`application_data_dir`			
Data directory for Mnesia database. File default = set during install (installer defaults to `/usr/local/gemini/a2s/var/data`) IMPORTANT: Do not change the location of the data directory after installation.	STRING	see description	null
`application_tx_log_path`			
Path to the A2S transaction log file, including file name. File default = <set during install>`/a2s-tx.log` (installer defaults to `/usr/local/gemini/a2s/var/log` for the directory path portion)	STRING	see description	dev/null
`application_tx_log_flush`			
Limit for the transaction log file's number of log entries to buffer before storing to disk.	INT (0 to INT_MAX)	1	0

central.conf Parameters (Part 4 of 14)

Parameter Description	Valid Range	File Default	Internal Default
`application_app_log_path`			
Path to the A2S application log file, including file name. File default = <set during install>`/a2s-app.log` (installer defaults to `/usr/local/gemini/a2s/var/log` for the directory path portion)	STRING	see description	dev/null
`application_app_log_level`			
The lowest severity level of messages to include in the application log. Each message that the A2S can generate has an assigned severity level appropriate to the message. You can use the `application_app_log_level` setting to filter A2S logging so that only messages of your specified level and higher will be logged. Options are, from highest to lowest level: ◆ `WARNG` Warning messages indicating a potential problem. ◆ `INFO` Informational messages indicating normal activity. ◆ `DEBUG` Low level detail messages potentially of use when debugging the application. For example, with `application_app_log_level` set to `INFO`, the A2S will log messages of all levels except `DEBUG`.	ATOM	INFO	INFO

***central.conf* Parameters (Part 5 of 14)**

Parameter Description	Valid Range	File Default	Internal Default
`application_app_log_log_fmt`			

Application log entry format type, indicating the log entry fields and their order. Options are:
◆ `default`
The default A2S application log format, as follows:
`<PID> <DATE> <MODULE> <LEVEL> <MESSAGECODE> <MESSAGE>`
◆ `cstm1`
Custom format #1, as follows:
`<DATE> <MESSAGECODE> <LEVEL> <PID> <THREADID> <MODULE> <MESSAGE>`
 	
If this parameter is commented out, the default log entry format is used.
 	
NOTE: The format of the `<DATE>` field is specified by the `application_app_log_date_fmt` setting. The delimiter between the fields is specified by the `application_app_log_field_sep` setting.
	ATOM	cstm1	default

`application_app_log_date_fmt`			

Application log entry timestamp type. Options are:
◆ `default`
The default A2S application log timestamp format, as follows:
`YYYYMMDDHHMMSS`
◆ `cstm1`
Custom timestamp #1, as follows:
`YYYY/MM/DD HH:MM:SS:000`
 	
If this parameter is commented out, the default timestamp format is used.
 	
NOTE: For the `cstm1` timestamp, the "000" in the millisecond part is a fixed constant. This parameter is commented out so the default timestamp format is used.
	ATOM	cstm1	default

central.conf Parameters (Part 6 of 14)

Parameter Description	Valid Range	File Default	Internal Default
`application_app_log_field_sep`			
ASCII decimal code indicating the desired delimiter between fields in an application log entry. Options are: ◆ `32` Single byte space. ◆ `124` Vertical bar (|).	ATOM	124	124
`vm_swappiness_value`			
Linux virtual memory "swappiness" correction. Should be set to 0 for all production environments. Red Hat EL4.4 default is `http://kerneltrap.org/node/3000 http://www.westnet.com/~gsmith/content/linux-pdflush.htm`	0 to INT_MAX	0	0
`cli_port`			
TCP port number for the command line interface.	INT	7586	7586
`cli_hello`			
CLI hello message. Default = A2S CLI Server	TEXT	see description	see description
`cli_prompt`			
CLI prompt.	TEXT	CLI>	CLI>
`cli_module`			
This parameter is used for internal module configuration. Do not change this. Default = `eldap_cli`	TEXT	see description	see description

central.conf Parameters (Part 7 of 14)

Parameter Description	Valid Range	File Default	Internal Default
`cli_ubf_servers`			
This parameter is used for internal module configuration. Do not change this. Default = `a2s_ebf`	TEXT	see description	see description
`network_monitor_enable`			
Enable network partition monitoring. Options are: ◆ `true` Enable network partition monitoring. You can enable network monitoring only if you have set up two networks, A and B, that connect your A2S nodes. Gemini recommends that A and B be physically separate networks. Network monitoring works by comparing heartbeats from network A and network B. For further information, see *page 298*. ◆ `false` Disable network partition monitoring. IMPORTANT: For network partition monitoring to function properly, these `central.conf` settings must be assigned identical values on each A2S node: ◆ `network_monitor_enable` ◆ `network_a_*` ◆ `network_b_*` ◆ `heartbeat_*`	ATOM	set during install (installer defaults to ‘false’)	false
`network_monitor_enable`			
Flag to enable network monitoring for network partitions. Must equal "true" or monitoring will be disabled	BOOL	true	null
`network_monitor_monitored_nodes`			
Comma-separated list of nodes that will be monitored. Do not include single quotes. Default = `a2s1@node-a, a2s1@node-b`	TEXT	see description	see description

***central.conf* Parameters (Part 8 of 14)**

Parameter Description	Valid Range	File Default	Internal Default
`network_a_address`			
IP address for the A network. This network should be physically separate from the B network and must be the same network used by the Erlang network distribution protocol (i.e. the network used for Mnesia replication traffic). File default = set during install (installer defaults to 10.10.10.12)	STRING	see description	null
`network_a_broadcast_address`			
IP broadcast address for the A network. This network *must* be the same network used by the Erlang network distribution protocol (i.e. the network used for Mnesia replication traffic). File default = set during install (installer defaults to 10.1.1.255)	STRING	see description	null
`network_a_tiebreaker`			
IP address for the A network to act as a tiebreaker. If the network monitoring application determines that the A network is partitioned and the B network is not partitioned, then if `network_a_tiebreaker` responds to an ICMP echo (a ping), then the local A2S node is on the "correct" side of the partition. If the local A2S node is not on the correct side of the partition (if the attempt to ping the tiebreaker address fails), then it shuts down immediately. The `network_a_tiebreaker` address must be extremely reliable and must be as close to the local A2S node as possible (from a network Layer 1 and 2 point of view) as well as close to all other A2S nodes. Ideally the tiebreaker should be the address of the Layer 2 switch or Layer 3 router that all Mnesia communications flow through. File default = set during install (installer defaults to 10.1.1.254)	STRING	see description	null

central.conf Parameters (Part 9 of 14)

Parameter Description	Valid Range	File Default	Internal Default
`network_b_address`			
IP address for the B network. This network should be physically separate from the A network. File default = set during install (installer defaults to 10.10.10.12)	STRING	see description	null
`network_b_broadcast_address`			
IP broadcast address for the B network. This network should be physically separate from the A network. File default = set during install (installer defaults to 10.10.10.255)	STRING	see description	null
`heartbeat_beacon_interval`			
Heartbeat beacon interval in milliseconds. At this interval, UDP heartbeart signals are transmitted from the local A2S node to each other A2S node in the cluster. The heartbeats are sent out both through network A and through network B. Gemini recommends that this interval be between 250 and 1000 (milliseconds).	INT	1000	1000
`heartbeat_warning_interval`			
Heartbeat alarm interval in seconds. If this interval passes without the local A2S node receiving a heartbeat signal from a peer A2S node, an alert is written to the local application log.	INT	set during install (installer defaults to 5)	5

central.conf Parameters (Part 10 of 14)

Parameter Description	Valid Range	File Default	Internal Default
`heartbeat_failure_interval`			
Heartbeat failure interval in seconds. A serious error has occurred if during this interval a heartbeat from a peer A2S node has been detected on network B but no heartbeat from that node has been detected on network A. The `network_a_tiebreaker` (*page 335*) address will be pinged to determine whether or not the local A2S node should be shut down to avoid database damage. NOTE: The value of `heartbeat_failure_interval` should be larger than the value of `heartbeat_warning_interval` by a factor of at least 1.5x but preferably 2x or more. Cluster timeout interval. If there is a network partition (or other failure that will cause network traffic from a node to be dropped or delayed), PSS/LSS protocol operations will hang. Therefore, WARNING: The "cluster_timeout" value must be larger than the "heartbeat_failure_interval" value, preferably by five (5) seconds or more.	INT	set during install (installer defaults to 15)	15
`cluster_timeout`			
Enter the cluster timeout interval in seconds. Erlang nodes will force a disconnect from each other if this timeout value is exceeded. WARNING: The "cluster_timeout" value must be larger than the "heartbeat_failure_interval" value, preferably by five (5) seconds or more. File default = <set during install> (installer defaults to 20 seconds for the timeout).	INT	see description	see description
`heartbeat_status_udp_port`			
UDP port for heartbeat listener.	INT	63099	63099

central.conf Parameters (Part 11 of 14)

Parameter Description	Valid Range	File Default	Internal Default
`heartbeat_status_xmit_udp_port`			
UDP port for heartbeat transmitter. This is the base port, the actual port may be higher.	INT	`6310`	`6310`
`mnesia_diskmon_dir_attrname`			
Name of the central.conf attribute, to locate the directory pathname to monitor. This may be a comma-separated list. If none of the listed attributes appear in central.conf, then the current directory, ".", will be monitored. Default = `application_data_dir`	TEXT	see description	see description
`mnesia_diskmon_dir_minfree`			
Minimum disk space for alarm trigger, units in kilobytes.	INT	800000	800000
`mnesia_diskmon_latest_log_max`			
Maximum size for Mnesia LATEST.LOG file, units in kilobytes.	INT	500000	500000
`restriction_id_period`			
Restriction period based on user IDs in seconds.	INT	`90`	`90`
`restriction_ip_period`			
Restriction period based on IPs in seconds.	INT	`90`	`90`
`counter_id_max_1`			
The maximum value for this counter based on user ID. Negative one (-1) means nolimit.	INT	1000	-1
`counter_id_max_2`			
The maximum value for this counter based on user ID. Negative one (-1) means nolimit.	INT	2000	-1

central.conf Parameters (Part 12 of 14)

Parameter Description	Valid Range	File Default	Internal Default
`counter_id_max_3`			
The maximum value for this counter based on user ID. Negative one (-1) means nolimit.	INT	3000	-1
`counter_ip_max_1`			
The maximum value for this counter based on IP address. Negative one (-1) means nolimit.	INT	1000	-1
`counter_ip_max_2`			
The maximum value for this counter based on IP address. Negative one (-1) means nolimit.	INT	2000	-1
`counter_ip_max_3`			
The maximum value for this counter based on IP address. Negative one (-1) means nolimit.	INT	3000	-1
`counter_id_reset_interval_1`			
The time interval in seconds before resetting the counter value which is based on the user ID. Negative one (-1) means always reset.	INT	10	-1
`counter_id_reset_interval_2`			
The time interval in seconds before resetting the counter value which is based on the user ID. Negative one (-1) means always reset.	INT	20	-1
`counter_id_reset_interval_3`			
The time interval in seconds before resetting the counter value which is based on the user ID. Negative one (-1) means always reset.	INT	30	-1

central.conf Parameters (Part 13 of 14)

Parameter Description	Valid Range	File Default	Internal Default
`counter_ip_reset_interval_1`			
The time interval in seconds before resetting the counter value which is based on the IP address. Negative one (-1) means always reset.	INT	10	-1
`counter_ip_reset_interval_2`			
The time interval in seconds before resetting the counter value which is based on the IP address. Negative one (-1) means always reset.	INT	20	-1
`counter_ip_reset_interval_3`			
The time interval in seconds before resetting the counter value which is based on the IP address. Negative one (-1) means always reset.	INT	30	-1
`counter_same_rcpt_weight`			
Counter weight for the case in which the sender and the recipient are the same.	INT	3	1
`counter_resend_period`			
The period to determine to use the `counter_resend_weight` value below. Negative one (-1) means to use resend weight below.	TIME	60	-1
`counter_resend_weight`			
Counter weight for the case of "resend".	INT	5	1
`popb4smtp_expiry`			
POP before SMTP expiry.	TIME	600	600

central.conf Parameters (Part 14 of 14)

Parameter Description	Valid Range	File Default	Internal Default
`gms_imapd_conf`			
IMAP is not used in this version of the product. For internal use—do not modify or delete this mandatory value.	STRING	/dev/null	/dev/null

counter_white_black.conf

Path `<A2S_HOME>/1.0.0/etc/counter_white_black.conf`

Purpose This is a restriction counter white and black list.

Max Entry Lines .

Field Delimiter Comma separated values.

The URA ID can be defined in white/black list for the send counter feature.

For the Whilte List

- IP address and URA-ID(not userID) can be defined in the white list.
- If matches, no restriction will happen to the IP address or userID.

For the Black List

- IP address and URA-ID (not userID) can be defined.
- If matches, restriction will always happen to the IP address or userID.

Note If same entry exists in both list, white list will be higher priority.

The following format is used:

```
{ip or uraid, white or black, "Value"}.
```

Example

```
{ip, white, "10.0.0.1"}.
{ip, black, "10.0.0.2"}.
{ip, black, "10.0.0.3"}.
{uraid, white, "uraid1"}.
{uraid, black, "uraid100"}.
```

counter.conf

Path `<A2S_HOME>/1.0.0/etc/counter.conf`

Purpose Configuration for domain based counter weights.

Max Entry Lines

Field Delimiter Colon (:).

The following format is used:

```
domain_name: weight
```

Example

```
a.b.c.com: 1
d.e.f.com: 20
g.h.i.com: 300
```

10 GDSS Administration

This chapter describes administration for the Gemini Distributed Storage Server (GDSS), an Erlang-based server using the database management system. The chapter covers these topics:

Note For overviews of how to configure GDSS features, see *Chapter 6, M2H, A2S, GDSS Configuration*.

If you want to quickly locate the description of a particular setting that you have seen in the GDSS `central.conf` file, you can use *Appendix A, Index of Settings in .properties and .conf Files*.

About the GDSS

There exists a dichotomy in modern storage products. Commodity storage is inexpensive, but unreliable. Enterprise storage is expensive, but reliable. Large capacities are present in both enterprise and commodity class. The problem, then, becomes how to leverage inexpensive commodity hardware to achieve high capacity enterprise class reliability at a fraction of the cost.

This problem space has been researched extensively in academia, especially in the last few years. GDSS uses techniques and algorithms from this research to create a solution which is reliable, cost effective, and scalable.

GDSS Features

GDSS supports a number of advanced features including:

- Key/value object storage system
- Written entirely in Erlang for reliability and scalability
- Contains a web based administration interface, also written in Erlang
- Contains multiple client interfaces
- Chain replication for performance and redundancy

The following diagram shows GDSS nodes in relation to the other components of the GWS.

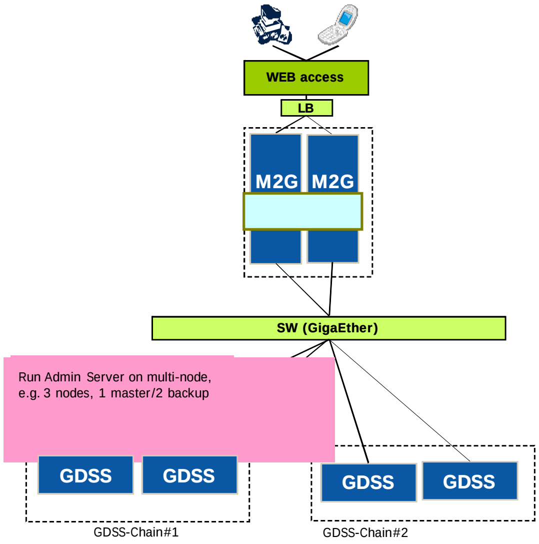


Figure 11 GDSS Node Topology

Brick Access

The decision about which brick to access is made on the M2H. The upper layer application specifies which table to access. The target chain decision is based on hashing result of the key prefix:

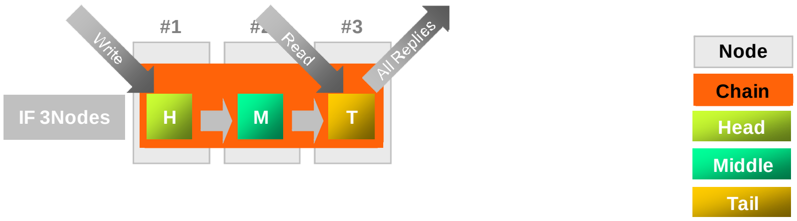
- hashing result of key prefix
- hashing space map (mapping relation between hashing result and chains)

The target brick decision is based on three criteria:

- operation type (read or write)
- brick role (head or tail)
- brick health condition (coordinated by Admin Server)

GDSS Brick Consistency

- All write requests are sent to head brick
- All Read requests are sent to tail brick
- Head brick manage all data replication
- All replies(both read and write) are from tail brick



Chain Replication

A storage replication algorithm divides the key space into a number of chains using hashing. The data for each chain is stored separately, to allow for parallel access of data in separate chains. Data in the chain is replicated to all members of the chain to avoid data loss in the presence of failures. Chain membership is dynamic, and the cluster will operate normally in the presence of multiple failures, given sufficient cardinality of replication

Installing the GDSS

This section describes how to install and set up the Gemini Distributed Storage Server (GDSS). The chapter covers these topics:

- *Installing GDSS on a Single Node, on page 348*
- *Installing GDSS on Multiple Nodes, on page 349*

Installing GDSS on a Single Node

For a single node GDSS deployment, follow the installation steps below.

Note Before performing a GDSS installation or upgrade, read the release notes that come with the package.

1 Install these prerequisite packages:

- TCL
- ERT (platform-specific version: 32-bit or 64-bit)

2 Install the GDSS package. You can run the GDSS installation script in either interactive mode or “silent” mode:

- Run `./installer-gdss.sh` to perform an interactive GDSS installation.
- Run `./installer-gdss.sh -o silent` to perform a “silent” GDSS installation.

3 Run the following command to start the GDSS:

```
/etc/init.d/gdss start
```

4 With the GDSS running, run the following command to complete the single node system initialization process:

```
/etc/init.d/gdss provision-standalone
```

After a few seconds, you should see this message on your console:

```
Bootstrap successful!

Go to the <a href='/'>main page</a> to view the cluster.  You
will be automatically redirected in 5 seconds.
```

In the GDSS Admin Server UI that displays after this re-direct, you can view information about the single-node GDSS “cluster” that has been established.

Installing GDSS on Multiple Nodes

Note Before performing a GDSS installation or upgrade, read the release notes that come with the package.

To install GDSS on multiple nodes, follow the steps below.

1 On each host, install these pre-requisite packages:

- TCL
- ERT (platform-specific version: 32-bit or 64-bit)

2 On each host, install the GDSS package. You can run the GDSS installation script in either interactive mode or “silent” mode:

- Run `./installer-gdss.sh` to perform an interactive GDSS installation.
- Run `./installer-gdss.sh -o silent` to perform a “silent” GDSS installation.

3 On one of the GDSS hosts, start and then stop the GDSS with these commands:

```
/etc/init.d/gdss start
```

```
/etc/init.d/gdss stop
```

4 From the GDSS host that you just started and stopped, copy the file `~mmssys/.erlang.cookie` and paste this file to the same location (`~mmssys/`) on each of the other GDSS hosts. On each host, verify that the file is owned by the `mmssys` user and has permissions 0400.

5 Start each GDSS host with this command:

```
/etc/init.d/gdss start
```

6 From any GDSS host, use a browser to fetch this URL:

```
http://<admin-node>:23080/
```

where `<admin-node>` is the hostname of the GDSS host on which you wish to run the GDSS Admin Server. In the form that displays, edit the following items:

- "Local" should be "false".
- "NodeList" should be a list of all GDSS nodes in the cluster. This list must be formatted like the following example, including the commas and single quotes:

```
'gdss1@hostname-a', 'gdss1@hostname-b', 'gdss1@hostname-c'
```

7 After confirming the “Local” and “NodeList” items, click the "Bootstrap" button to complete the GDSS cluster initialization process.

After a few moments, you should see this message:

```
Bootstrap successful!

Go to the <a href='/'>main page</a> to view the cluster.  You
will be automatically redirected in 5 seconds.
```

In the GDSS Admin Server interface that displays after this re-direct, you can view information about the multi-node GDSS “cluster” that has been established. For further information about this interface, see *Admin Server UI, on page 370*.

IMPORTANT To ensure high availability in a multi-node GDSS cluster, ensure that the GDSS network partition monitor is correctly configured on each node. For guidance on network partition monitor configuration, see *central.conf, on page 386*.

In addition to correctly configuring network partition monitoring, further steps are necessary to ensure high availability in a multi-node GDSS cluster. External users should consult with Gemini Technical Support on multi-node GDSS setup. Gemini personnel can request the internal document `multinode_gdss_install.pdf` from the GDSS development group.

GDSS Initialization Script

The GDSS has an initialization script with which you can start, stop, or restart the server, or check to confirm that it is running, or check the software version number. The start option sets appropriate environment values for the GDSS, and can be integrated into your Linux boot routine.

The GDSS initialization script is run in the `/etc/init.d/` directory by default. You must be the `root` user to run the initialization script. The script syntax is:

```
/etc/init.d/gdss start|version|status|restart|stop
```

Each initialization script option is described in this chapter.

Starting the GDSS

Start the GDSS by using the GDSS initialization script's `start` option.

Do not attempt to start the GDSS by starting up individual processes rather than using the initialization script. The initialization script sets environment variables that are necessary for the GDSS to run properly.

To start the GDSS As `root`, run the GDSS initialization script with the `start` option:

```
/etc/init.d/gdss start
```

The command response should confirm the successful starting of the server.

Example The sample below shows a successful start of the GDSS.

```
>  /etc/init.d/gdss start
Starting gdssd:                                          [  OK  ]
Creating GDSS lock file:                                 [  OK  ]
Gemini Distributed Storage Server (GDSS) start status    [  OK  ]
```

Troubleshooting GDSS Start-Up

If the GDSS fails to start, check the console, the server's application log, and the operating system's `syslog` (`/var/log/messages` by default) for error messages. Potential causes of start-up problems include:

- Inadequate permissions on GDSS log and data directories. These directories must be writable.
- Failure of processes from the server's last running session to die cleanly. Explicitly kill any rogue processes.
- Initialization failures, usually caused by one of these configuration errors:
  - A missing configuration file
  - A missing configuration setting
  - An unrecognized configuration setting
  - Check your `syslog` for messages about failure to initialize.

If you currently have the GDSS application log filtering set at INFO or higher, you may find it helpful to change this setting to DEBUG and then try again to start the server. Depending on the nature of the start-up problem, this setting may result in the server writing additional information into its application log that may be useful in troubleshooting the problem.

Integrating the GDSS Start Script into the Host's Boot Routine

If you wish, you may integrate the GDSS start-up script into the host machine's boot routine so that the GDSS starts automatically when you start the machine. You can integrate GDSS start-up into your Linux host boot routine by using your system's `chkconfig` utility. See the `chkconfig` man page for details.

Changing Chain Length (Redundancy)

The change the chain length, start an Erlang CLI shell. The shell can be running on any node in the cluster, but the argument for the "-remsh" flag must be the name of the node that is running the GDSS admin server.

- The change_chain_length command will require the brick names and node names of each brick in the desired chain.

- ◆ If adding a new brick, the change_chain_length command will automatically start the new brick. However, the GDSS application should already be running on all nodes before modifying the chain.
- ◆ The first brick in the list will be the head brick (if the chain is healthy); the last brick in the list will be the tail brick.
- ◆ At least one brick in the new list must be a member of the chain's current list.

- ■ The recommended naming convention for a brick is: tablename_chN_bM, where N is the chain number and M is the brick number (1 = head).
- ■ After executing the change_chain_length command, use the Admin HTTP server to check the status of the chain.

Example To change the tab1_ch1 chain to be length = 3.

```
# su mmssys -c "/usr/local/gemini/ert/R11B-5/bin/erl -sname
tmp$$ -remsh gdss1@demosv-2"

1> brick_admin:change_chain_length(tab1_ch1,
[{tab1_ch1_b1,'gdss1@machine-a'}, {tab1_ch1_b2,'gdss1@machine-
b'}, {tab1_ch1_b3,'gdss1@machine-c'}]).
```

Adding/Removing Chains

Use the following steps to grow or shrink the GDSS cluster:

- ■ Start an Erlang CLI shell (see above section) on the GDSS admin server.
- ■ You will need to create a "chain description" list. This is a data structure that defines the name of each chain together with the member bricks of each chain. This data structure is an Erlang list, with nested tuples and sub-lists.
  - ◆ If you wish, it may be easier to use an external text editor to create the list, then cut-and-paste it into the Erlang shell. The editor may help you find the matching curly braces and square brackets.
  - ◆ These short instructions cannot tell you everything about using the Erlang shell. Please ask an Erlang developer for assistance, if you hit problems that you do not understand.
- ■ The command brick_admin:get_table_chain_list(TableName?) will return the chain list that's currently in use for the table called TableName.

  A successful return result looks like: {ok, ChainList?}

Note The {ok, ...} wrapper is not part of the chain list.

Example The following shows a step-by-step example.

```
> brick_admin:get_table_chain_list(tab1).

{ok,[{tab1_ch1,[{tab1_ch1_b1,'gdss1@bb2-2'}]}]}
```

The actual chain list is:

```
[{tab1_ch1,[{tab1_ch1_b1,'gdss1@bb2-2'}]}]
```

This list is 1 item long. The item is a 2-tuple, denoted with curly braces.

```
{tab1_ch1,[{tab1_ch1_b1,'gdss1@bb2-2'}]}
```

The first item in the 2-tuple gives the name of the chain, tab1_ch1. The second item in the 2-tuple gives the list of bricks for the healthy chain ... in this case, it is also a list that is 1 item long.

```
[{tab1_ch1_b1,'gdss1@bb2-2'}]
```

The brick's name is also a 2-tuple: the brick's server name (tab1_ch1_b1) and the Erlang virtual machine node name ('gdss1@bb2-2'). The VM node name looks like an email address, but it specifies:

- The instance name of the VM node, gdss1. (It is possible to run multiple GDSS applications on the same machine. If configured to do so, then they should be named gdss1, gdss2, gdss3, and so on.)
- The node's hostname, in this case bb2-2.

For this example, we'll do the following:

- Add a second chain. We'll call the new chain tab1_ch2.
- The new chain will also be of length 1.
- A new machine has had GDSS installed. The machine's hostname is 'ebi'. The Erlang VM instance name is the default 'gdss1', so the Erlang VM node name will be 'gdss1@ebi'.
- We will give the new brick for 'gdss1@ebi' the name tab1_ch2_b1.

  Therefore, the new chain list will be:

  ```
  [{tab1_ch1,[{tab1_ch1_b1,'gdss1@bb2-2'}]},
   {tab1_ch2,[{tab1_ch2_b1,'gdss1@ebi'}]}]
  ```

  Now we're return to our instructions.
- Define a variable that stores the new chain list. The use of newlines and extra whitespace is not important. However, the uppercase/lowercase and all the punctuation is very important.

```
> f(NewChain).

> NewChain = [{tab1_ch1,[{tab1_ch1_b1,'gdss1@bb2-2'}]},

                {tab1_ch2,[{tab1_ch2_b1,'gdss1@ebi'}]}].
```

- Use the start_migration command to define the new chain list and start migrating data to the new chains. (Remember: the table name in this example is 'tab1'.)

  ```
  > brick_admin:start_migration(tab1,
  brick_hash:var_prefix_init(NewChain)).
  ```

- Use the Admin HTTP server to check the status of the table and all of its chains.

Checking the GDSS Version Number

You can check the version number of the GDSS by using the initialization script's `version` option.

To check the GDSS version number

As `root`, run the GDSS initialization script with the `version` option:

```
/etc/init.d/gdss version
```

The command response should indicate the server version number, appended by a version timestamp in format `YYYYMMDDHHMM`.

The sample below shows a response to the `gdss version` option.

```
> /etc/init.d/gdss version
/usr/local/gemini/gdss29b/0.1.0-x: gdss-
1.0.0.GDSS__HEAD.Pl4.29.200810222000
```

Verifying GDSS Running Processes

You can verify GDSS running processes by using the initialization script's `status` option.

To verify GDSS running processes

As `root`, run the GDSS initialization script with the `status` option:

```
/etc/init.d/gdss status
```

The command response should confirm the running GDSS processes.

The sample below shows the `gdss status` response when the GDSS is running.

```
> /etc/init.d/gdss status
gdssd (pid 16145 16119) is running...
```

The sample below shows the `gdss status` response when the GDSS is not running.

```
> /etc/init.d/gdss status
gdssd is stopped
```

Restarting the GDSS

Restart the GDSS by using the initialization script’s `restart` option.

To restart the GDSS

As `root`, run the GDSS initialization script with the `restart` option:

```
/etc/init.d/gdss restart
```

The command response should confirm the successful stopping and then starting of the server.

The sample below shows a successful restart of the GDSS.

```
> /etc/init.d/gdss restart
Stopping gdssd:                                        [  OK  ]
           Removing GDSS lock file:
[  OK  ]
HyperScale Storage Server (GDSS) stop status:          [  OK  ]
Starting gdssd:                                        [  OK  ]
Creating GDSS lock file:                               [  OK  ]
HyperScale Storage Server (GDSS) start status:         [  OK  ]
```

The sample below shows an attempt to restart the GDSS when the GDSS was not already running.

```
> /etc/init.d/gdss restart
Stopping gdssd:                                        [FAILED]
               gdssd was not running so it was not stopped.
Removing GDSS lock file:                               [FAILED]
HyperScale Storage Server (GDSS) stop status:          [FAILED]
Starting gdssd:                                        [  OK  ]
Creating GDSS lock file:                               [  OK  ]
HyperScale Storage Server (GDSS) start status:         [  OK  ]
```

Stopping the GDSS

Shut down the GDSS by using the initialization script's `stop` option.

To shut down the GDSS

As `root`, run the initialization script with the `stop` option:

```
/etc/init.d/gdss stop
```

The command response should confirm the successful stopping of the server.

In this example, the GDSS is successfully stopped.

```
> /etc/init.d/gdss stop
Stopping gdssd:                                         [  OK  ]
Removing GDSS lock file:                                [  OK  ]
HyperScale Storage Server (GDSS) stop status:           [  OK  ]
```

CLI Commands for the GDSS

Below is the list of Command Line Interface (CLI) commands available for the GDSS:

- `show nodes <NODEID>` — check if `<NODEID>` is active or not
- `show nodes` — list all active mnesia nodes
- `show tables <TABLEID>` — show information for `<TABLEID>`
- `show tables` — show information for all mnesia tables
- `show ubf` — show number of ubf connections and its max
- `set loglevel ALERT|WARNG|INFO|DEBUG` — sets application log level
- `exit|logout|q|quit` — exit CLI

The list above is the same for M2H and A2S servers described in the section *Common A2S and M2H CLI Commands List, on page 226*.

Note The TCP port for the GDSS is 7597.

Viewing an Active Node List (show_nodes)

Use the `show_nodes` command to see which nodes in your Erlang-based server cluster are currently active.

The command syntax is as follows:

```
show_nodes [<NODEID>]
```

'show_nodes' Parameters

Parameter	Description
`[<NODEID>]`	Optional node ID. If you specify a node ID, the command will determine whether that node is active or not. If you do not specify a node ID, the command returns a list of active nodes in the cluster. A sample node name is `gdss1@machine1`.

Example

The following example shows that node `gdss1@hotate` is active.

```
CLI> show nodes m2h1@hotate
'gdss1@hotate' exists.
```

Viewing a Database Table List (show_tables)

Use the show_tables command to view high level information about a particular database table, or about the full set of Erlang-based server database tables. The command displays the number of records and number of words in each table, as well as the average number of words per record.

The command syntax is as follows:

```
show_tables [<TABLEID>]
```

'show_tables' Parameters

Parameter	Description
[<TABLEID>]	Optional table ID. If you specify a table ID, the command returns information about that particular table. Valid table IDs are those in the example below.If you do not specify a table ID, the command returns information about all server tables.

Viewing UBF Connections (show_ubf)

Use the `show_ubf` command to view the number of UBF connections for this server

The command syntax is as follows:

```
show_ubf
```

Set Application Log Level (set_loglevel)

Use the set_loglevel command to set the application log level to the lowest severity level of messages to include in the application log.

Each message that the server can generate has an assigned severity level appropriate to the message. You can use the set_loglevel setting to filter the server's application logging so that only messages of your specified level and higher will be logged.

Options are, from highest to lowest level:

- ALERT
  Messages indicating a condition requiring immediate correction.
- WARNG
  Warning messages indicating a potential problem.
- INFO
  Informational messages indicating normal activity.
- DEBUG
  Low level detail messages potentially of use when debugging the application. Setting loglevel to DEBUG will result in a very large number of messages being logged.

For example, with the log level set to INFO, the server will log messages of all levels except DEBUG.

The command syntax is as follows:

```
set_loglevel [ALERT|WARNG|INFO|DEBUG]
```

Example

The following sets the application log level to DEBUG:

```
CLI> set loglevel DEBUG
OK.
```

Terminating the CLI Session (quit)

Use any of the `exit|logout|q|quit` commands to terminate your session with the command line interface. The command syntax is as follows:

```
exit|logout|q|quit
```

There are no arguments to the `exit|logout|q|quit` command.

Example The following command exits the CLI session:

```
CLI> exit
Goodbye!
```

Backup, Fallback, and Disaster Recovery

To backup admin server, copy `bootstrap_copy` folders of one admin server. Use `tar` to back up the `.../var/data` directory of each brick in the cluster is sufficient as long as there are no updates happening while the backup is taking place.

Summary of the GDSS Data API

GDSS Command	Summary	Options
`set`	Set all attributes.	Test-and-set, set 'val' only, set 'flags' only
`add`	Set all attributes, but only if the key already exists.	Test-and-set, set 'val' only, set 'flags' only
`replace`	Set all attributes, but only if the key does not already exist.	Test-and-set, set 'val' only, set 'flags' only
`get`	Fetch all attributes.	Test-and-set
`delete`	Delete a key.	Test-and-set
`get_many`	Get many keys.	Test-and-set, get 'ts' only, include 'val' in results, include 'flags' in results

IMPORTANT The commands will succeed if and only if the timestamp specified in the client's command exactly matches the current 'ts' held by the server.

GDSS Configuration

By default, the GDSS configuration directory is:

```
<GDSS>/1.0.0/etc
```

where `<GDSS_HOME>` is the server's home directory as established during product installation. If during installation you accept the default for `<GDSS_HOME>`, then the GDSS configuration directory is:

```
/usr/local/gemini/gdss/1.0.0/etc
```

The instructions for configuring the GDSS using `central.conf` are the same for the A2S and M2H. These instructions can be found at *Working with the central.conf File, on page 224*.

Adding More Data Nodes to Cluster

You can dynamically add additional data nodes in the cluster using the following steps:

1. Add the new node name to the `network_monitor_monitored_nodes`. See *network_monitor_monitored_nodes, on page 397* for more information about this parameter.
2. Use the `gmt_config_svr:reload_config()` command to reload the configuration.

Note Most of the components cannot be dynamically reloaded in the `central.conf` file. To activate changes that you make to this file, you must restart the GDSS.

Using the GDSS Admin Server UI

This chapter describes how to use the GDSS Admin Server graphical user interface to check the status of your GDSS cluster. The chapter covers these topics:

- *Accessing the Admin Server UI, on page 369*
- *Admin Server UI, on page 370*

Accessing the Admin Server UI

You can access the GDSS Admin Server graphical user interface with a web browser, using this URI:

```
http://<admin-node>:23080/
```

where `<admin-node>` is the hostname of the GDSS host on which the Admin Server is currently running. Use the hostname—*not* the GDSS "node name". For example, if the hostname of the host running the Admin Server is `machine1.provider.com`, and the corresponding GDSS node name is `gdss1@machine1`, then you would access the Admin Server interface at this URI:

```
http://machine1.provider.com:23080/
```

Alternatively, you can use the IP address of the Admin Server host instead of the hostname, such as:

```
http://123.123.321.321:23080/
```

Note 23080 is the default listening port for the GDSS Admin Server interface. In the current release, you cannot change this default port assignment.

The Admin Server runs on only one GDSS host at a time, with other GDSS hosts in the cluster available for fail-over support. If you direct your browser to port 23080 on a GDSS host that is not currently running the Admin Server, the response will be an HTTP 302 re-direct to the correct URI, on the correct host.

Admin Server UI

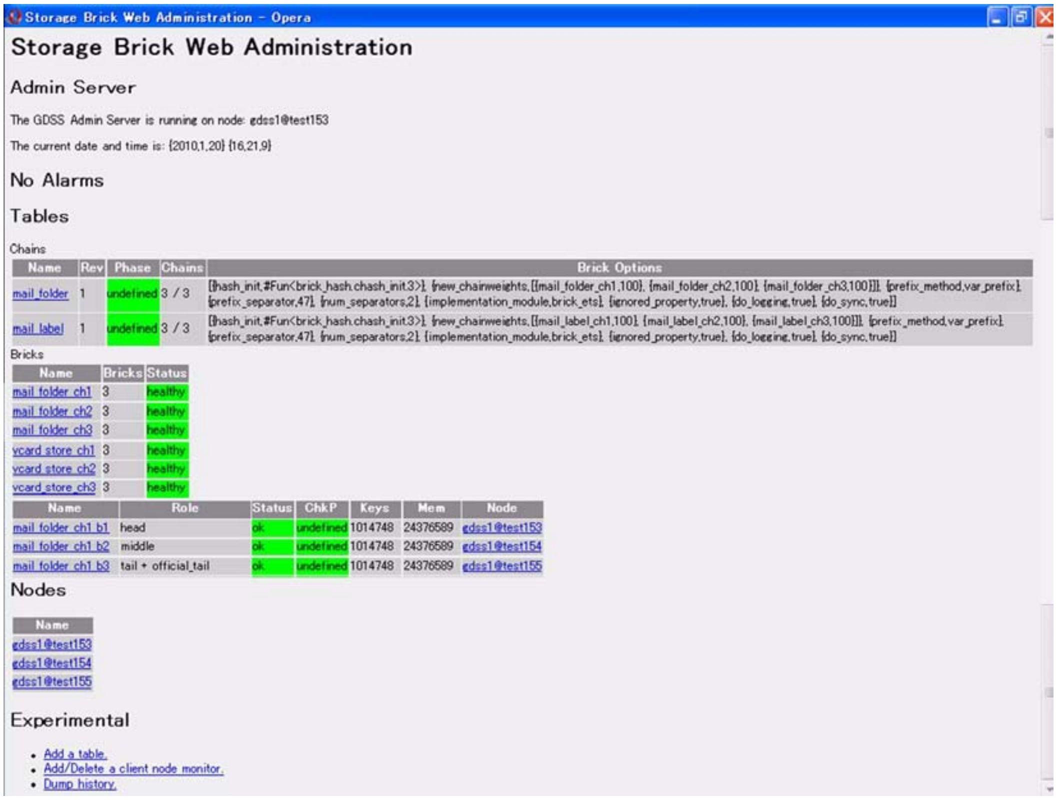
This section shows and describes each of the Admin Server status screens:

- *Storage Brick Web Administration, on page 371*
- *Admin Server, on page 371*
- *Alarms, on page 372*
- *Table, on page 373*
- *Chains, on page 375*
- *Bricks, on page 377*
- *Nodes, on page 377*
- *Experimental, on page 378*

Note This section does not show the initial “Bootstrap” screen that is mentioned in the section *Installing the GDSS, on page 348*. It shows only the screens that you can use to check GDSS cluster status once the cluster is up and running. The status screens described in this section are all read-only.

Storage Brick Web Administration

When you access the Admin Server for an up and running GDSS cluster, the "Storage Brick Web Administration" screen displays. :



Storage Brick Web Administration

Admin Server

The GDSS Admin Server is running on node: gdss1@test153

The current date and time is: {2010,1,20} {16,21,9}

No Alarms

Tables

Chains

Name	Rev	Phase	Chains	Brick Options
mail_folder	1	undefined	3 / 3	[{hash_init,#Fun<brick_hash.chash_init.3>}, {new_chainweights,[{mail_folder_ch1,100}, {mail_folder_ch2,100}, {mail_folder_ch3,100}]}, {prefix_method,var_prefix}, {prefix_separator,47}, {num_separators,2}, {implementation_module,brick_ets}, {ignored_property,true}, {do_logging,true}, {do_sync,true}]
mail_label	1	undefined	3 / 3	[{hash_init,#Fun<brick_hash.chash_init.3>}, {new_chainweights,[{mail_label_ch1,100}, {mail_label_ch2,100}, {mail_label_ch3,100}]}, {prefix_method,var_prefix}, {prefix_separator,47}, {num_separators,2}, {implementation_module,brick_ets}, {ignored_property,true}, {do_logging,true}, {do_sync,true}]

Bricks

Name	Bricks	Status
mail_folder_ch1	3	healthy
mail_folder_ch2	3	healthy
mail_folder_ch3	3	healthy
vcard_store_ch1	3	healthy
vcard_store_ch2	3	healthy
vcard_store_ch3	3	healthy

Name	Role	Status	ChkP	Keys	Mem	Node
mail_folder_ch1_b1	head	ok	undefined	1014748	24376589	gdss1@test153
mail_folder_ch1_b2	middle	ok	undefined	1014748	24376589	gdss1@test154
mail_folder_ch1_b3	tail + official_tail	ok	undefined	1014748	24376589	gdss1@test155

Nodes

Name
gdss1@test153
gdss1@test154
gdss1@test155

Experimental

- Add a table.
- Add/Delete a client node monitor.
- Dump history.

Figure 12   Storage Brick Web Administration Screen

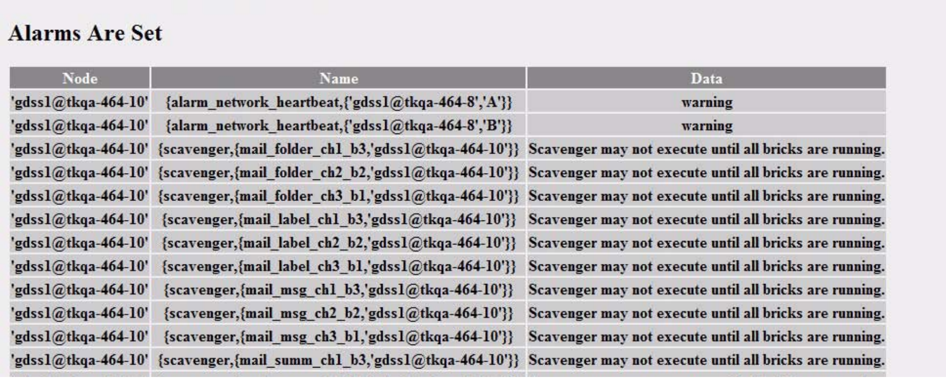
As you can see, the StorageBrick Web Administration screen is divided into seven sections labeled Admin Server, No Alarms/Alarms Are Set, Tables, Chains, Bricks, Nodes, and Experimental.

Admin Server

In the Admin Server section shows the node that the Admin Server is running on and the time and date.

Alarms

If no alarms are set, the screen looks similar to what is shown in *Figure 12, on page 371*. If alarms are set, this section will look similar to what is pictured in *Figure 13, on page 372*.



Alarms Are Set

Node	Name	Data
'gdss1@tkqa-464-10'	{alarm_network_heartbeat,{'gdss1@tkqa-464-8','A'}}	warning
'gdss1@tkqa-464-10'	{alarm_network_heartbeat,{'gdss1@tkqa-464-8','B'}}	warning
'gdss1@tkqa-464-10'	{scavenger,{mail_folder_ch1_b3,'gdss1@tkqa-464-10'}}	Scavenger may not execute until all bricks are running.
'gdss1@tkqa-464-10'	{scavenger,{mail_folder_ch2_b2,'gdss1@tkqa-464-10'}}	Scavenger may not execute until all bricks are running.
'gdss1@tkqa-464-10'	{scavenger,{mail_folder_ch3_b1,'gdss1@tkqa-464-10'}}	Scavenger may not execute until all bricks are running.
'gdss1@tkqa-464-10'	{scavenger,{mail_label_ch1_b3,'gdss1@tkqa-464-10'}}	Scavenger may not execute until all bricks are running.
'gdss1@tkqa-464-10'	{scavenger,{mail_label_ch2_b2,'gdss1@tkqa-464-10'}}	Scavenger may not execute until all bricks are running.
'gdss1@tkqa-464-10'	{scavenger,{mail_label_ch3_b1,'gdss1@tkqa-464-10'}}	Scavenger may not execute until all bricks are running.
'gdss1@tkqa-464-10'	{scavenger,{mail_msg_ch1_b3,'gdss1@tkqa-464-10'}}	Scavenger may not execute until all bricks are running.
'gdss1@tkqa-464-10'	{scavenger,{mail_msg_ch2_b2,'gdss1@tkqa-464-10'}}	Scavenger may not execute until all bricks are running.
'gdss1@tkqa-464-10'	{scavenger,{mail_msg_ch3_b1,'gdss1@tkqa-464-10'}}	Scavenger may not execute until all bricks are running.
'gdss1@tkqa-464-10'	{scavenger,{mail_summ_ch1_b3,'gdss1@tkqa-464-10'}}	Scavenger may not execute until all bricks are running.

Figure 13 Alarms Are Set Section

Alarm information includes the **Node** name that the alarm is set for, the **Name** of the alarm, **Data**, informational messages pertaining to this alarm.

Tables

In the “Tables” section, the “Chains” field displays in X/Y form the number of chains currently mapped to the table, followed by the number of chains in a new mapping that is in progress . Under normal operations these two numbers will be the same; they will differ only if the cluster is in the midst of expanding or contracting. You can ignore the contents of the “Brick Options” field.

In the “Chains” section, the “Status” field will have one of these values:

- `init`
  The initial state of a chain.
- `unknown`
  The state of the chain is unknown. Information regarding chain members is unavailable.
- `unknown_timeout`
  Information regarding one or more chain members remains unavailable after a certain period of time.
- `healthy`
  All members of the chain are running and in sync.

- `degraded`
  One or more members of the chain are either stopped or are in the process of joining the chain and re-syncing the chain's data.

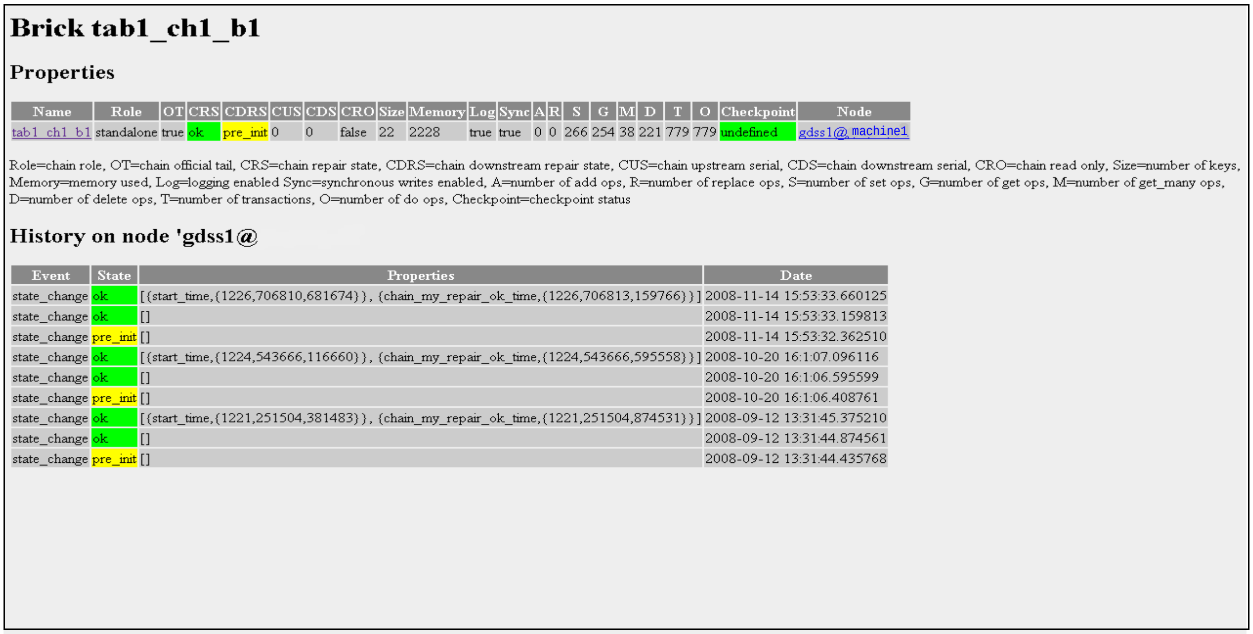
- `stopped`
  All members of the chain are stopped.

For information about fields in the “Bricks” section, see *page 374*.

From the Admin Server’s “Storage Brick Web Administration” screen, you can jump to screens for each individual table (*page 373*), chain (*page 375*), brick (*page 377*), and node (*page 377*).

Table

The Admin Server’s “Table” screen provides information for a selected table. The table name displays in the screen title. The screen lists all the storage chains and bricks associated with the selected table. In the sample below, the table is implemented through only one chain with one brick.



Brick tab1_ch1_b1

Properties

Name	Role	OT	CRS	CDRS	CUS	CDS	CRO	Size	Memory	Log	Sync	A	R	S	G	M	D	T	O	Checkpoint	Node
tab1_ch1_b1	standalone	true	ok	pre_init	0	0	false	22	2228	true	true	0	0	266	254	38	221	779	779	undefined	gdss1@ machine1

Role=chain role, OT=chain official tail, CRS=chain repair state, CDRS=chain downstream repair state, CUS=chain upstream serial, CDS=chain downstream serial, CRO=chain read only, Size=number of keys, Memory=memory used, Log=logging enabled Sync=synchronous writes enabled, A=number of add ops, R=number of replace ops, S=number of set ops, G=number of get ops, M=number of get_many ops, D=number of delete ops, T=number of transactions, O=number of do ops, Checkpoint=checkpoint status

History on node 'gdss1@

Event	State	Properties	Date
state_change	ok	[{start_time,{1226,706810,681674}}, {chain_my_repair_ok_time,{1226,706813,159766}}]	2008-11-14 15:53:33.660125
state_change	ok	[]	2008-11-14 15:53:33.159813
state_change	pre_init	[]	2008-11-14 15:53:32.362510
state_change	ok	[{start_time,{1224,543666,116660}}, {chain_my_repair_ok_time,{1224,543666,595558}}]	2008-10-20 16:1:07.096116
state_change	ok	[]	2008-10-20 16:1:06.595599
state_change	pre_init	[]	2008-10-20 16:1:06.408761
state_change	ok	[{start_time,{1221,251504,381483}}, {chain_my_repair_ok_time,{1221,251504,874531}}]	2008-09-12 13:31:45.375210
state_change	ok	[]	2008-09-12 13:31:44.874561
state_change	pre_init	[]	2008-09-12 13:31:44.435768

Figure 14 Admin Server “Table” Section

Detailed status information is provided for each individual brick, as described in the table on *page 374*.

Admin Server Brick Status Fields (Part 1 of 2)

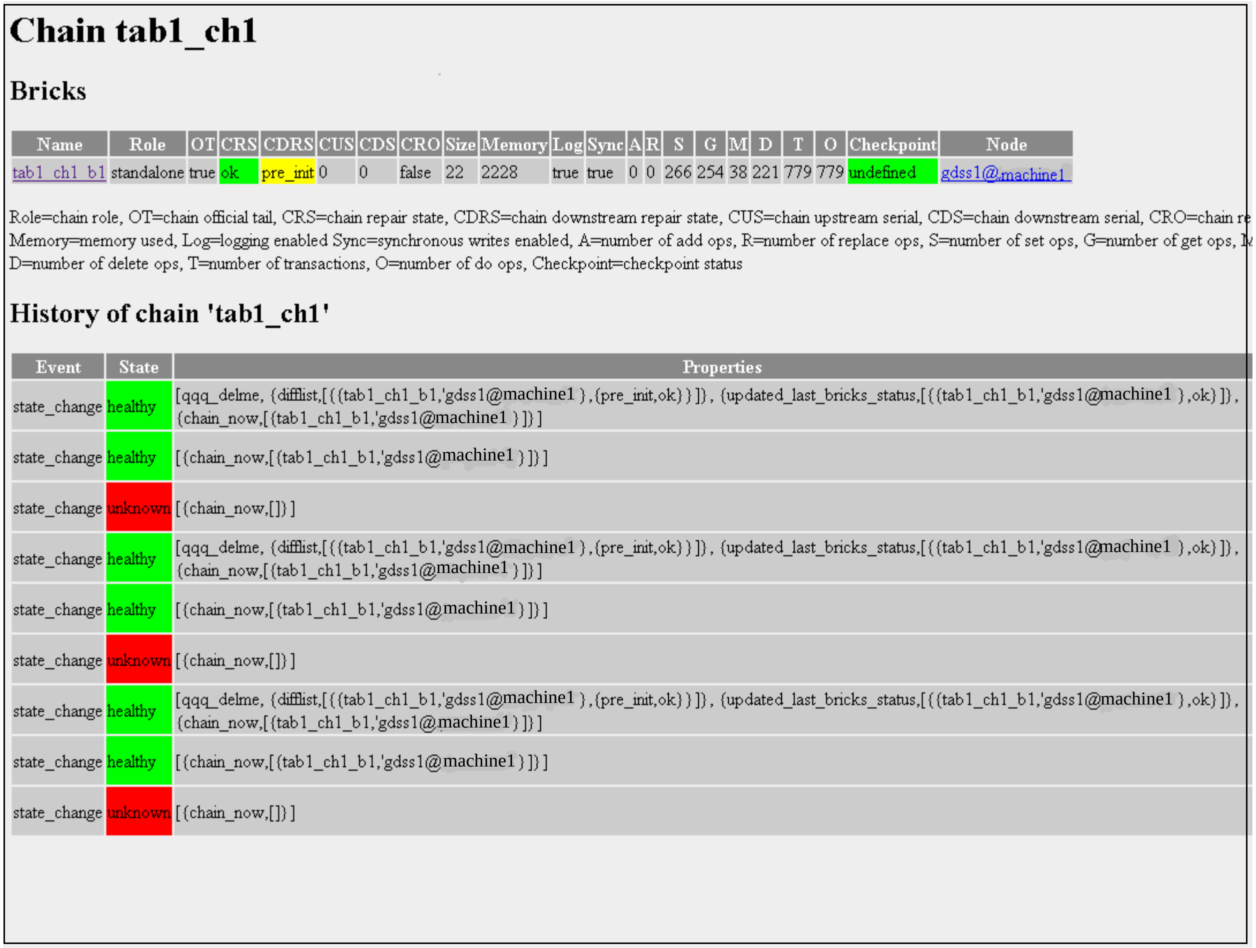
Field	Description
Role	Role of the identified brick within the identified chain. Possible roles are: ◆ `undefined` The role is unknown. This brick should not be handling queries. ◆ `standalone` The brick is operating in a chain of length one. ◆ `head` The brick is operating in a chain of at least length three, and it is the first in the chain. ◆ `middle` The brick is operating in a chain of at least length three, and it is one of the middle bricks in the chain. ◆ `tail` The brick is operating in a chain of at least length three, and it is the last in the chain.
OT	Whether or not the brick is the chain's "official tail", `true` or `false`. This field will display as `true` if the brick is the last brick in the chain and has status OK; or if the brick is not the last brick in the chain but is temporarily acting as the tail while the actual last brick in the chain is in the midst of a repair process.
CRS	The brick's "repair state" within the chain. Possible states are: ◆ `pre_init` The initial state. The brick is running and pingable, but the brick is not in service and the state of the brick's local storage is unknown. ◆ `in_progress` The brick is in the middle of the repair process. ◆ `ok` The brick's data is 100% in sync with other members of the chain.
CDRS	The "repair state" of the next brick downstream from the local brick. Possible states are the same as for CRS above.
CUS	Internal serial number of the next brick upstream from the local brick. Used for replication communications.
CDS	Internal serial number of the next brick downstream from the local brick. Used for replication communications.
CRO	Whether or not the chain is read-only, supporting only data retrieval operations and not data modifying operations. Options are `true` or `false`.
Size	Number of keys within the brick.
Memory	Words allocated to the brick within the Erlang VM.
Log	Whether or not write-ahead transaction logging is enabled for the brick. Options are `true` or `false`.

Admin Server Brick Status Fields (Part 2 of 2)

Field	Description
Sync	Whether or not synchronous writes to disk are enabled for the transaction log. Options are `true` or `false`.
A	Number of “add” operations since the last brick reboot.
R	Number of “replace” operations since the last brick reboot.
S	Number of “set” operations since the last brick reboot.
G	Number of “get” operations since the last brick reboot.
M	Number of “get many” operations since the last brick reboot.
D	Number of “delete” operations since the last brick reboot.
T	Number of transactions since the last brick reboot.
O	Number of “do” operations since the last brick reboot.
Old	Number of “too old” operations since the last brick reboot.
Exp	Number of “expired keys” operations since the last brick reboot.
Checkpoint	Checkpoint status for the brick: either a process ID for the checkpointing process, or `undefined`.
Node	Node on which the brick resides.

Chains

The Admin Server’s “Chain” screen provides information for a selected chain. The chain name displays in the screen title. The screen lists all the storage bricks associated with the selected chain. In the sample below, the chain has only one storage brick in it.



Chain tab1_ch1

Bricks

Name	Role	OT	CRS	CDRS	CUS	CDS	CRO	Size	Memory	Log	Sync	A	R	S	G	M	D	T	O	Checkpoint	Node
tab1_ch1_b1	standalone	true	ok	pre_init	0	0	false	22	2228	true	true	0	0	266	254	38	221	779	779	undefined	gdss1@machine1

Role=chain role, OT=chain official tail, CRS=chain repair state, CDRS=chain downstream repair state, CUS=chain upstream serial, CDS=chain downstream serial, CRO=chain re
Memory=memory used, Log=logging enabled Sync=synchronous writes enabled, A=number of add ops, R=number of replace ops, S=number of set ops, G=number of get ops, M
D=number of delete ops, T=number of transactions, O=number of do ops, Checkpoint=checkpoint status

History of chain 'tab1_ch1'

Event	State	Properties
state_change	healthy	[qqq_delme, {difflist,[{{tab1_ch1_b1,'gdss1@machine1 },{pre_init,ok}}]}, {updated_last_bricks_status,[{{tab1_ch1_b1,'gdss1@machine1 },ok}]}, {chain_now,[{tab1_ch1_b1,'gdss1@machine1 }]}]
state_change	healthy	[{chain_now,[{tab1_ch1_b1,'gdss1@machine1 }]}]
state_change	unknown	[{chain_now,[]}]
state_change	healthy	[qqq_delme, {difflist,[{{tab1_ch1_b1,'gdss1@machine1 },{pre_init,ok}}]}, {updated_last_bricks_status,[{{tab1_ch1_b1,'gdss1@machine1 },ok}]}, {chain_now,[{tab1_ch1_b1,'gdss1@machine1 }]}]
state_change	healthy	[{chain_now,[{tab1_ch1_b1,'gdss1@machine1 }]}]
state_change	unknown	[{chain_now,[]}]
state_change	healthy	[qqq_delme, {difflist,[{{tab1_ch1_b1,'gdss1@machine1 },{pre_init,ok}}]}, {updated_last_bricks_status,[{{tab1_ch1_b1,'gdss1@machine1 },ok}]}, {chain_now,[{tab1_ch1_b1,'gdss1@machine1 }]}]
state_change	healthy	[{chain_now,[{tab1_ch1_b1,'gdss1@machine1 }]}]
state_change	unknown	[{chain_now,[]}]

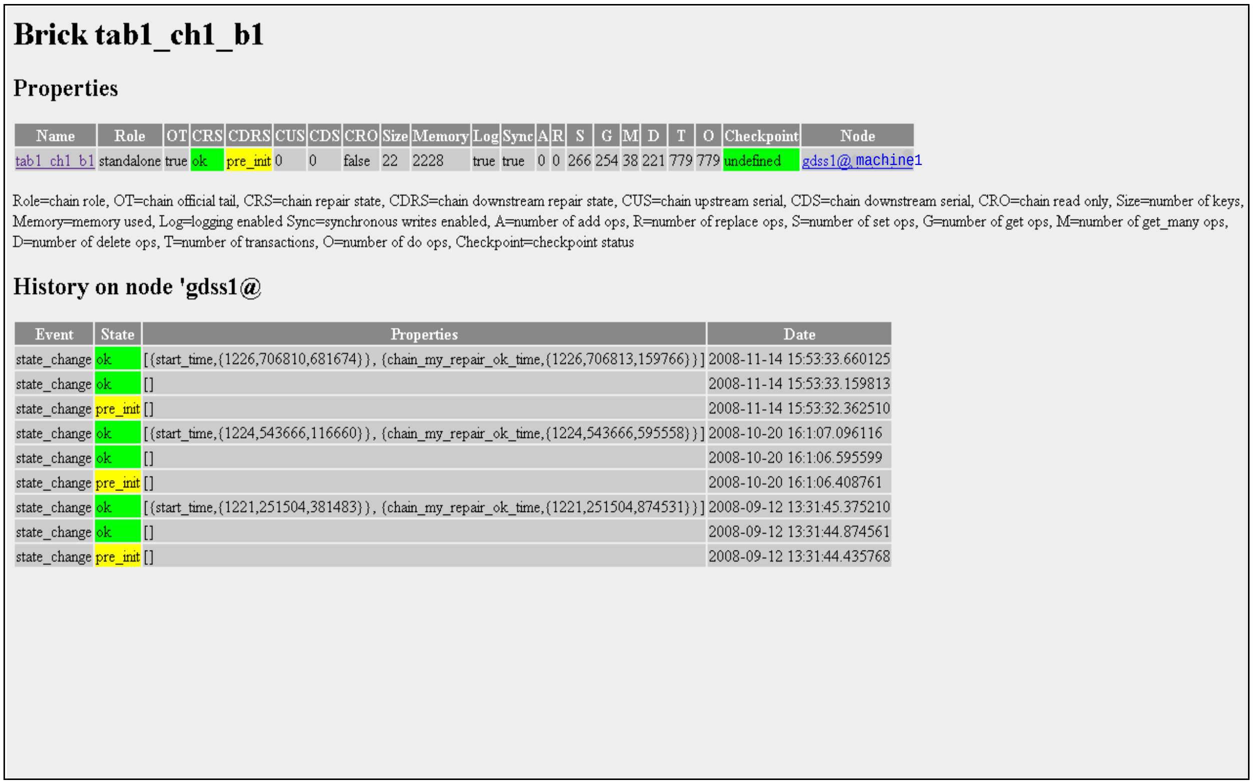
Figure 15 Admin Server "Chain" Section

Detailed status information is provided for each individual brick, as described in the table on *page 374*.

The "Chain" screen also shows a state history for the chain. The possible chain state values are described on *page 371*. You can ignore the "Properties" field.

Bricks

The Admin Server's "Chain" screen provides information for a selected storage brick. The brick name displays in the screen title.



Brick tab1_ch1_b1

Properties

Name	Role	OT	CRS	CDRS	CUS	CDS	CRO	Size	Memory	Log	Sync	A	R	S	G	M	D	T	O	Checkpoint	Node
tab1_ch1_b1	standalone	true	ok	pre_init	0	0	false	22	2228	true	true	0	0	266	254	38	221	779	779	undefined	gdss1@machine1

Role=chain role, OT=chain official tail, CRS=chain repair state, CDRS=chain downstream repair state, CUS=chain upstream serial, CDS=chain downstream serial, CRO=chain read only, Size=number of keys, Memory=memory used, Log=logging enabled Sync=synchronous writes enabled, A=number of add ops, R=number of replace ops, S=number of set ops, G=number of get ops, M=number of get_many ops, D=number of delete ops, T=number of transactions, O=number of do ops, Checkpoint=checkpoint status

History on node 'gdss1@

Event	State	Properties	Date
state_change	ok	[{start_time,{1226,706810,681674}}, {chain_my_repair_ok_time,{1226,706813,159766}}]	2008-11-14 15:53:33.660125
state_change	ok	[]	2008-11-14 15:53:33.159813
state_change	pre_init	[]	2008-11-14 15:53:32.362510
state_change	ok	[{start_time,{1224,543666,116660}}, {chain_my_repair_ok_time,{1224,543666,595558}}]	2008-10-20 16:1:07.096116
state_change	ok	[]	2008-10-20 16:1:06.595599
state_change	pre_init	[]	2008-10-20 16:1:06.408761
state_change	ok	[{start_time,{1221,251504,381483}}, {chain_my_repair_ok_time,{1221,251504,874531}}]	2008-09-12 13:31:45.375210
state_change	ok	[]	2008-09-12 13:31:44.874561
state_change	pre_init	[]	2008-09-12 13:31:44.435768

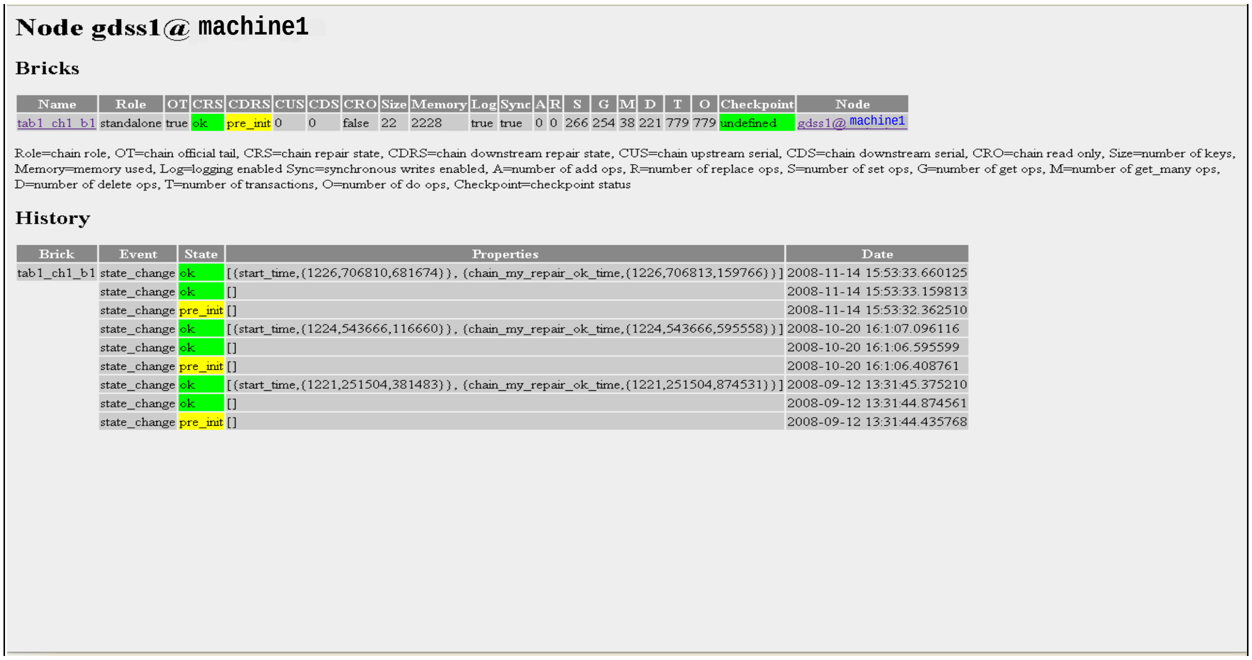
Figure 16   Admin Server "Brick" Section

Detailed status information is provided for the brick, as described in the table on *page 374*.

The "Brick" screen also shows a state history for the brick. The possible brick state values are described on *page 374*. You can ignore the "Properties" field.

Nodes

The Admin Server's "Node" screen provides information for a selected Erlang/GDSS node. The node name displays in the screen title. The screen lists all the storage bricks associated with the selected node. In the sample below, the node has only one storage brick on it.



Node gdss1@ machine1

Bricks

Name	Role	OT	CRS	CDRS	CUS	CDS	CRO	Size	Memory	Log	Sync	A	R	S	G	M	D	T	O	Checkpoint	Node
tab1_ch1_b1	standalone	true	ok	pre_init	0	0	false	22	2228	true	true	0	0	266	254	38	221	779	779	undefined	gdss1@ machine1

Role=chain role, OT=chain official tail, CRS=chain repair state, CDRS=chain downstream repair state, CUS=chain upstream serial, CDS=chain downstream serial, CRO=chain read only, Size=number of keys, Memory=memory used, Log=logging enabled Sync=synchronous writes enabled, A=number of add ops, R=number of replace ops, S=number of set ops, G=number of get ops, M=number of get_many ops, D=number of delete ops, T=number of transactions, O=number of do ops, Checkpoint=checkpoint status

History

Brick	Event	State	Properties	Date
tab1_ch1_b1	state_change	ok	[{start_time,{1226,706810,681674}}, {chain_my_repair_ok_time,{1226,706813,159766}}]	2008-11-14 15:53:33.660125
	state_change	ok	[]	2008-11-14 15:53:33.159813
	state_change	pre_init	[]	2008-11-14 15:53:32.362510
	state_change	ok	[{start_time,{1224,543666,116660}}, {chain_my_repair_ok_time,{1224,543666,595558}}]	2008-10-20 16:1:07.096116
	state_change	ok	[]	2008-10-20 16:1:06.595599
	state_change	pre_init	[]	2008-10-20 16:1:06.408761
	state_change	ok	[{start_time,{1221,251504,381483}}, {chain_my_repair_ok_time,{1221,251504,874531}}]	2008-09-12 13:31:45.375210
	state_change	ok	[]	2008-09-12 13:31:44.874561
	state_change	pre_init	[]	2008-09-12 13:31:44.435768

Figure 17 Admin Server "Node" Section

Detailed status information is provided for each individual brick, as described in the table on *page 374*.

The "Node" screen also shows a state history for each brick. The possible brick state values are described on *page 374*. You can ignore the "Properties" field.

Experimental

The Experimental section of the Admin Server screen has these three options:

- *Add a table, on page 378*
- *Add/Delete a Client Node Monitor, on page 379*
- *Add/Delete a Client Node Monitor, on page 379*

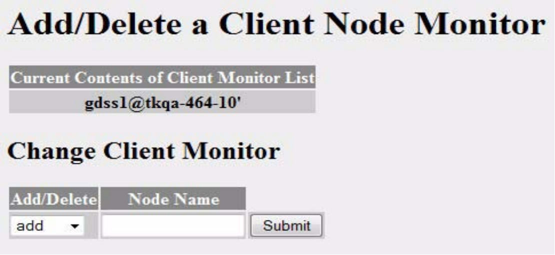
Add a table

Note Add a Table is not used in the NTTR product. All required tables are created by another procedure.

Add/Delete a Client Node Monitor

Use the **Add/Delete a Client Node Monitor** screen to see the list of clients that are monitored (in the example below, the node gdss1@tkqa-464-10 is the only client monitor listed.

Also, you can add or delete a client node monitor using this screen by entering the node name.



Add/Delete a Client Node Monitor

Current Contents of Client Monitor List
gdss1@tkqa-464-10'

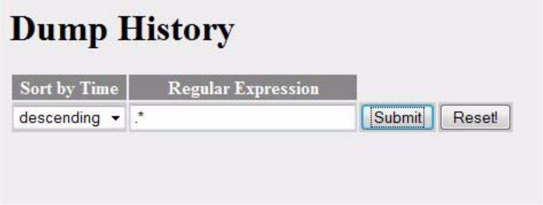
Change Client Monitor

Add/Delete	Node Name
add	

Figure 18 Add/Delete a Client Node Monitor Screen

Dump History

Use the following screen to sort by ascending or descending order a history dump based on matching a string of text with a regular expression.

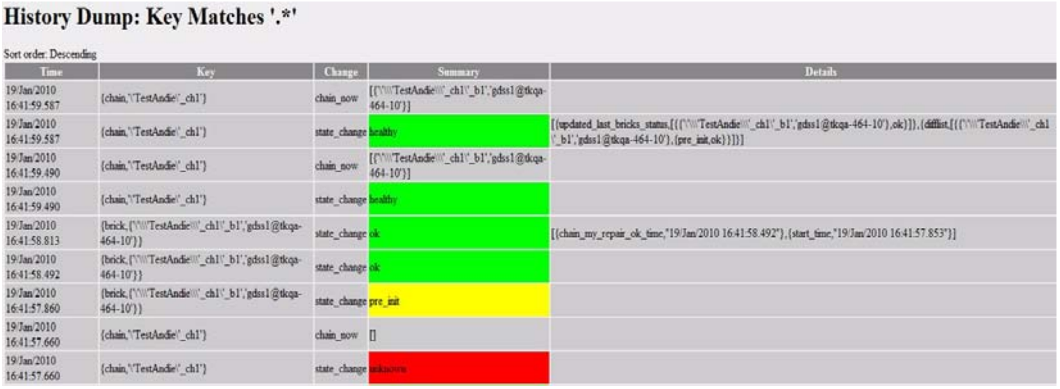


Dump History

Sort by Time	Regular Expression
descending	.*

Figure 19 Dump History Screen

The following screen shows the (partial) results in descending order of the **Dump History** screen when the regular expression, `.*`, was entered.



History Dump: Key Matches '.*'

Sort order: Descending

Time	Key	Change	Summary	Details
19/Jan/2010 16:41:59.587	{chain,'\'TestAndie\'_ch1'}	chain_now	[{'\'\\\\'TestAndie\\\\'_ch1\'_b1','gdss1@tkqa-464-10'}]	
19/Jan/2010 16:41:59.587	{chain,'\'TestAndie\'_ch1'}	state_change	healthy	[{updated_last_bricks_status,[{{'\'\\\\'TestAndie\\\\'_ch1\'_b1','gdss1@tkqa-464-10'},ok}]},{difflist,[{{'\'\\\\'TestAndie\\\\'_ch1\'_b1','gdss1@tkqa-464-10'},{pre_init,ok}}]}]
19/Jan/2010 16:41:59.490	{chain,'\'TestAndie\'_ch1'}	chain_now	[{'\'\\\\'TestAndie\\\\'_ch1\'_b1','gdss1@tkqa-464-10'}]	
19/Jan/2010 16:41:59.490	{chain,'\'TestAndie\'_ch1'}	state_change	healthy	
19/Jan/2010 16:41:58.813	{brick,{'\'\\\\'TestAndie\\\\'_ch1\'_b1','gdss1@tkqa-464-10'}}	state_change	ok	[{chain_my_repair_ok_time,"19/Jan/2010 16:41:58.492"},{start_time,"19/Jan/2010 16:41:57.853"}]
19/Jan/2010 16:41:58.492	{brick,{'\'\\\\'TestAndie\\\\'_ch1\'_b1','gdss1@tkqa-464-10'}}	state_change	ok	
19/Jan/2010 16:41:57.860	{brick,{'\'\\\\'TestAndie\\\\'_ch1\'_b1','gdss1@tkqa-464-10'}}	state_change	pre_init	
19/Jan/2010 16:41:57.660	{chain,'\'TestAndie\'_ch1'}	chain_now	[]	
19/Jan/2010 16:41:57.660	{chain,'\'TestAndie\'_ch1'}	state_change	unknown	

Figure 20   Dump Example Screen

Database Admin Server

An Admin Server is responsible for keeping the cluster schema up-to-date. It coordinates and maintains the health of all chains and bricks.

The common features include repairing, data migration, checkpointing, scavenger, and the supports of those features like bootstrap, client monitor, fast sync and a web interface. An Admin Server performs the following functions:

- Monitors the health of each GDSS chains and bricks
- Maintains GDSS cluster schema (tables, chains and bricks config)
- Coordinates data migration when schema change
- Hosts the web-based UI – basic operations and status of GDSS cluster
- Performs fail-over between multiple GDSS nodes

Fail-over of Admin Server

At any given time, there will only be one active admin server. The rest of them are only standby and be ready to take over when the active fails. See `central.conf` for `admin_server_distributed_nodes`.

When the GDSS's start, one of the nodes in `admin_server_distributed_nodes` will take the active role. You could see which one is currently active by looking at the top of the web interface, for example:

Admin Server Node

The GDSS Admin Server is running on node:

```
gdss1@test189
```

Notice that all other GDSS's will forward their UI to the active. When the active admin server is down and restarted, it will recover itself and may or may not take the active role back.

Scavenging

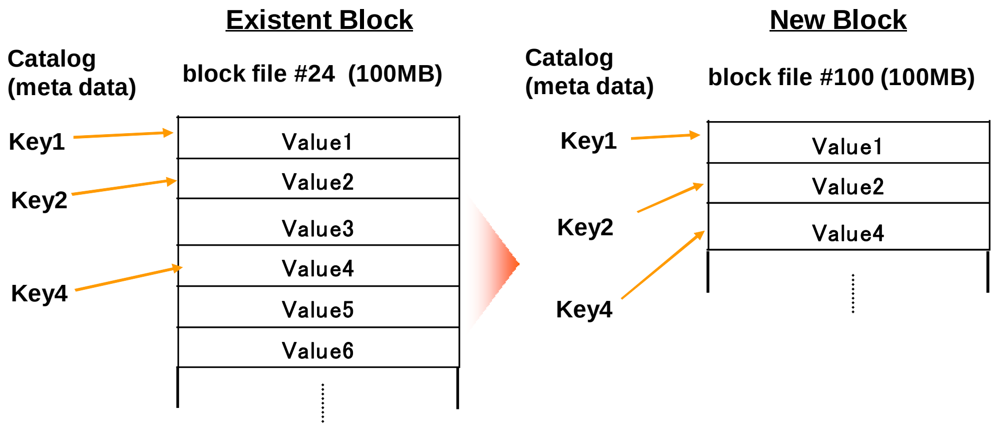
Scavenger reclaims disk space and is generally triggered daily. The following parameters can be configured in `central.conf`.

- `brick_scavenger_start_time: 03:00`
- `brick_skip_live_percentage_greater_than: 90`
- `brick_scavenger_temp_dir: /tmp/gdss_scavenger`

Scavenger can also be triggered manually by CLI using the following command:

```
gmt_hlog_common:start_scavenger_commonlog([])
```

The following diagram shows the scavenger operation set to occur very day at 3:00 a.m.



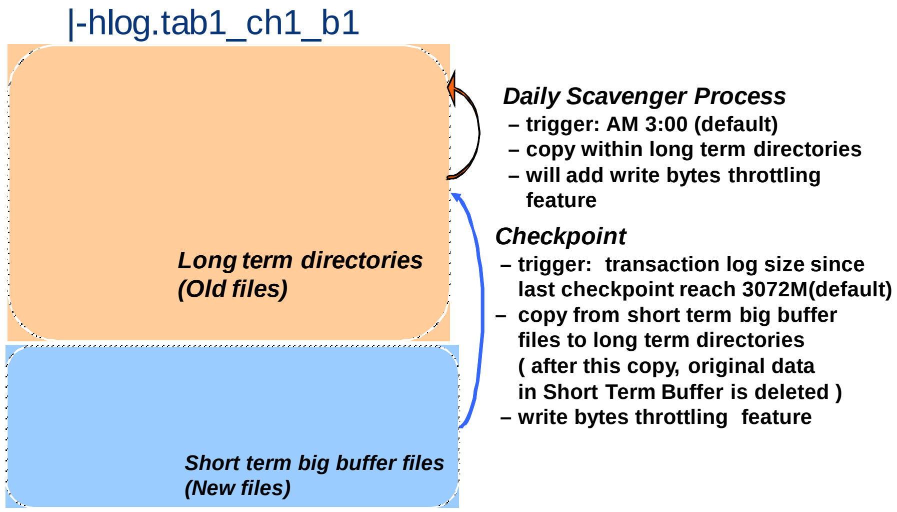
- **e.g. Start at AM 3:00 (default) every day**
  - only non-deleted value data is copied into new block file
  - Existent block data is deleted after the copy
- **Additional disk space (e.g. 30%) is needed because of this mechanism**
- **New block file(block file #100 in above example) size will be the same size - 100MB(default)**

Checkpointing

Periodic checkpointing is to reduce the amount of disk data (and therefore time) required to start a brick after a failure. For values-in-RAM tables, it also has the side-effect of reducing disk usage by breaking the big, short term files into smaller ones.

.

Brick data directory example:



|-hlog.tab1_ch1_b1

Long term directories (Old files)

Short term big buffer files (New files)

Daily Scavenger Process

- **trigger: AM 3:00 (default)**
- **copy within long term directories**
- **will add write bytes throttling feature**

Checkpoint

- **trigger: transaction log size since last checkpoint reach 3072M(default)**
- **copy from short term big buffer files to long term directories (after this copy, original data in Short Term Buffer is deleted)**
- **write bytes throttling feature**

Log Files

All log files are located under `/usr/local/gemini/gdss/var`. The application log and archives are in the subdirectory `log/`. The data logs will reside under the subdirectory `data/`. For each brick, including the bootstrap_copy*, will have their own subdirectory for example:

```
/usr/local/gemini/gdss/var/data/hlog.bootstrap_copy1
```

```
/usr/local/gemini/gdss/var/data/hlog.tab1_ch2_b1
```

In each of these folders there will be a few common things:

- Subdirectory s which holds the working buffer *.HLOG and *.TRK
- sequence_number used by checkpointing to divide short and long term storage

Common Log Server Features

Two tier logging subdivide the long and short term logs into local and common. Meta and blob data will both written to short term common log first. Then meta and blob will then be synched to the local and common in the long term storage respectively using the Scavenger mechanism.

The definitions of meta and blob are:

meta - will be stored in RAM for quick access.

blob - will require disk IO and unable to load in RAM

11 GDSS Configuration Files

This chapter describes the configuration files for the Gemini Distributed Storage Server (GDSS), an Erlang server using the Mnesia database management system. The chapter covers this configuration file:

- *central.conf, on page 386*

IMPORTANT The following files in `<GDS_HOME>/1.0.0/etc/` are for internal use only. DO NOT CHANGE.

```
broker.conf
config.template
imap-host-map
admin.conf
httpdconf
s3.conf
ssl.conf
admin.css
ssl_client.pem
ssl_server.pem
```

Note For overviews of how to configure GDSS features, see *GDSS Configuration, on page 368*.

If you want to quickly locate the description of a particular setting that you have seen in the GDSS `central.conf` file, you can use *Appendix A, Index of Settings in .properties and .conf Files*

central.conf

Path `<GDS_HOME>/1.0.0/etc/central.conf`

Purpose Main GDSS configuration file.

Dynamic Reload You cannot dynamically reload this file. To activate changes that you make to the file, you must restart the GDSS.

The table that follows describes each parameter in the `central.conf` file. For background information about how to work with the `central.conf` file, see *page 295*.

central.conf Parameters (Part 1 of 16)

Parameter Description	Valid Range	File Default	Internal Default
`application_home`			
GDSS top-level directory. File default = set during install (installer defaults to `/usr/local/gemini/gdss`)	STRING	see description	null

central.conf Parameters (Part 2 of 16)

Parameter Description	Valid Range	File Default	Internal Default
`application_nodename`			
First part of the GDSS application node name. Each GDSS server is assigned a “node name”. A node name identifies a specific Linux process on a specific physical machine. Different node names can be used to run multiple GDSS services on the same physical machine, if desired. An GDSS node name has three parts: 1) An application name local to the physical machine. 2) The “@” symbol. 3) The hostname of the physical machine, as shown by the output of the system command `uname -n`. If the hostname as shown by `uname -n` has one or more dots in it (for example `machine1.company.com`) then only the left-most part is used for the GDSS node name (from the example, `machine1`). A sample three-part GDSS node name is `gdss1@machine1`. The first part of the GDSS node name is determined by your `application_nodename` setting—in the sample, `gdss1`.	STRING	gdss1	null
`application_data_dir`			
Data directory for database. File default = set during install (installer defaults to `/usr/local/gemini/gdss/var/data`) IMPORTANT: Do not change the location of the data directory after installation.	STRING	see description	null

central.conf Parameters (Part 3 of 16)

Parameter Description	Valid Range	File Default	Internal Default
`application_app_log_path`			
Path to the GDSS application log file, including file name. File default = <set during install>`/gdss-app.log` (installer defaults to `/usr/local/gemini/gdss/var/log` for the directory path portion)	STRING	see description	dev/null
`application_app_log_level`			
The lowest severity level of messages to include in the application log. Each message that the GDSS can generate has an assigned severity level appropriate to the message. You can use the `application_app_log_level` setting to filter GDSS logging so that only messages of your specified level and higher will be logged. Options are, from highest to lowest level: ◆ `ALERT` Messages indicating a condition requiring immediate correction. ◆ `WARNG` Warning messages indicating a potential problem. ◆ `INFO` Informational messages indicating normal activity. ◆ `DEBUG` Low level detail messages potentially of use when debugging the application. For example, with `application_app_log_level` set to `INFO`, the GDSS will log messages of all levels except `DEBUG`.	ATOM	INFO	INFO
`application_app_log_field_sep`			
Field separator of application log file. Enter either 124 (vertical bar in ASCII code) or 32 (space in ASCII code).	ATOM	124	124

central.conf Parameters (Part 4 of 16)

Parameter Description	Valid Range	File Default	Internal Default
`application_stats_log_path`			
Path to the GDSS statistics log file, including file name. NOTE: The GDSS does not support statistics generation in the current release. Please ignore `application_stats_*` settings.	STRING	/dev/null	dev/null
`application_stats_log_interval`			
Not supported in current release.	TIME	60	NA
`cluster_timeout`			
Enter cluster timeout interval in seconds. WARNING: The `cluster_timeout` value must be larger than the `heartbeat_failure_interval` value, preferably by five (5) seconds or more. File default = <set during install> (installer defaults to 20 seconds for the timeout). Erlang nodes will force a disconnect from each other if this timeout value is exceeded. If there is a network partition (or other failure that will cause network traffic from a node to be dropped or delayed), client protocol operations will hang.	TIME	see description	see description
`vm_swappiness_value`			
Linux virtual memory "swappiness" correction. Should be 0 for all production environments. Red Hat EL4.4 default is 60. http://kerneltrap.org/node/3000 http://www.westnet.com/~gsmith/content/linux-pdflush.htm	see description	0	0
`cli_port`			
TCP port number for the command line interface.	INT	7597	7597

central.conf Parameters (Part 5 of 16)

Parameter Description	Valid Range	File Default	Internal Default
`env_erts_de_busy_limit`			
Value for ERTS_DE_BUSY_LIMIT environment variable. This sets a performance-related tunable parameter within the Erlang virtual machine. It should be changed only upon advice from Gemini support.	INT	4194304	128000
`cli_hello`			
CLI hello message. Default = GDSS CLI Server	TEXT	see description	see description
`cli_prompt`			
CLI prompt.	TEXT	CLI>	CLI>
`brick_max_log_size_mb`			
In MB, the maximum size of any individual file in the transaction write-ahead log.	INT	100	100
`brick_check_checkpoint_max_mb`			
The number of MBs written since the last checkpoint. This is the threshold at which the new checkpoint operation will start. The value should be larger than a single checkpoint dump, which is directly related to the number of keys in the table and the length of each key (to avoid checkpointing every 30 seconds) and smaller than the maximum amount of time to wait for a brick to start, given that the hardware's disks are capable of N MBytes per second and the GDSS can only read some number of MBytes per second.	INT	3072	3072

central.conf Parameters (Part 6 of 16)

Parameter Description	Valid Range	File Default	Internal Default
`brick_check_checkpoint_throttle_bytes`			
It is possible to overwhelm disks with too much I/O checkpoint operations that will interfere with regular operations. This is the number of bytes per second that multiple software bricks executing checkpoints simultaneously will restrict themselves to.	INT	1048576	1000000
`brick_sync_interval_msec`			
A loose upper bound on the interval between brick fsync requests for its main transaction log. WARNING: Do not set this value to be too small.	TIME	200	500
`brick_scavenger_start_time`			
The scavenger daily start time in hh:mm where hh is in hours greater than, or equal to zero and less than 23.	TIME	03:00	03:00
`brick_skip_live_percentage_greater_than`			
For the daily scavenger run, specify threshold for which data files with "live" data greater than this percentage will be ignored. A value of 0 will skip all files, 100 will skip no files.	0 to100	90	90
`brick_scavenger_throttle_bytes`			
Scavenger disk bandwidth throttle in bytes per seconds. File default = 600000000 Internal default = 600000000	INT	see description	see description

central.conf Parameters (Part 7 of 16)

Parameter Description	Valid Range	File Default	Internal Default
`brick_scavenger_temp_dir`			
The temporary directory used by the scavenger for data sorting. It is used by scavenger for temporary swapping. The work directory will be unconditionally removed by `rm -rf` at the start of the scavenger and then created. Any parent directories are not automatically created and therefore must exist. Up to tens of gigabytes of scratch space may be required. File default = `/tmp/gdss_scavenger`	STRING	see description	/tmp
`brick_repair_max_bytes`			
The maximum number of value blob bytes per repair round. File default = 65000000 Internal default = 65000000	INT to INT_MAX	see description	see description
`brick_repair_max_primers`			
Maximum number of parallel repair primer processes.	INT > 0	7	7
`brick_mbox_high_water`			
High water mark for the number of messages queued for a brick's processing. To disable congestion control set this number to zero (0).	INT (0 to INT_MAX)	500	500
`brick_mbox_low_water`			
Low water mark for the number of messages queued for a brick's processing. To disable congestion control set this number to zero (0).	INT (0 to INT_MAX)	100	100

central.conf Parameters (Part 8 of 16)

Parameter Description	Valid Range	File Default	Internal Default
`brick_mbox_repair_high_water`			
High water mark for the number of messages queued for a brick's processing while under repair. Repair will be halted if this high water mark is reached. To disable congestion control set this number to zero (0).	INT (0 to INT_MAX)	1500	1500
`brick_mbox_repair_overload_resume_interval`			
After overload condition, the number (in seconds) to wait before attempting to resume a brick repair.	TIME	300	300
`brick_dirty_buffer_wait`			
The maximum time an OS virtual memory dirty memory page will remain dirty (in seconds). Changes to this attribute or to `/proc/sys/vm/dirty_writeback_centisecs` or to `/proc/sys/vm/dirty_expire_centisecs` or to the XFS-specific VM settings must be coordinated. The default is 60 seconds which is two times the defaults for RedHat EL 4.x and 5.x kernels.	TIME	60	60
`brick_do_op_too_old_timeout`			
The timeout in milliseconds for a brick to consider a client's request "too old". Requests that are too old will be silently ignored.	TIME	3000	3000
`brick_primer_limit`			
Specifies the maximum number of data primer processes that may execute simultaneously. This value should not be changed without recommendation from Gemini Support.	INT > 0	200	200

***central.conf* Parameters (Part 9 of 16)**

Parameter Description	Valid Range	File Default	Internal Default
`brick_primer_limit`			
Specify the maximum number of data primer processes that may execute simultaneously. This value must be greater than 0. **IMPORTANT** Its value should not be changed without recommendation from Gemini Support	INT	200	200
`brick_preprocess_method`			
This parameter is commented out (disabled). Specifies the brick key preprocessing method. If this attribute is not present all bricks use the default table properties list that is recommended by Gemini Technical Support only. If `none`, all bricks use no preprocessors. If `ssf_only`, all bricks use SSF preprocessor only	ATOM	none	none
`brick_expiration_processor`			
This parameter is commented out (disabled). Specifies the brick key expiration method. If an attribute is not present, the default expiration method is the table properties list that is recommended by Gemini Technical Support only.	ATOM	[].	[].
`brick_s3_conf_path`			
If brick_s3_conf_path option is not present, S3 listener will not run. NOTE: The definitive value of brick_s3_tcp_port is really in s3.conf brick_s3_tcp_port: pS3_TCP_PORT File default = <set during install`>/root/conf/s3.conf` (installer defaults to `/root/conf/s3.conf` for the directory path portion).	TEXT	see description	see description

central.conf Parameters (Part 10 of 16)

Parameter Description	Valid Range	File Default	Internal Default
`gdss_ebf_tcp_port`			
TCP port for the GDSS EBF protocol server. File default = <set during install> The installer defaults to `7580.`	TEXT	see description	see description
`gdss_ubf_server_tcp_port`			
TCP port for the GDSS UBF protocol server. File default = <set during install> The installer defaults to `7581.`	TEXT	see description	see description
`gdss_jsf_server_tcp_port`			
TCP port for the GDSS JSF protocol server. File default = <set during install> The installer defaults to `7582.`	TEXT	see description	see description
`gdss_json_rpc_tcp_port`			
TCP port for the GDSS JSON-RPC protocol server. File default = <set during install> The installer defaults to `7598.`	TEXT	see description	see description
`brick_admin_http_tcp_port`			
NOTE: The definitive value of brick_admin_http_tcp_port is really in the admin.conf file. File default = <set during install> (installer defaults to `23080.`)	TEXT	see description	see description

central.conf Parameters (Part 11 of 16)

Parameter Description	Valid Range	File Default	Internal Default
`brick_admin_conf_path`			
If brick_admin_conf_path option is not present, admin HTTP listener will not run. File default = <set during install> (installer defaults to `etc/root/conf/admin.conf`	TEXT	see description	see description
`admin_server_distributed_nodes`			
Comma separated list of nodes eligible to run the Admin Server used by main application startup Tcl script. File default = <set during install> (installer defaults to `'gdss1@machineA'`, `'gdss1@machine-B-with-hyphens'`	TEXT	see description	see description
`network_monitor_enable`			
Enable network partition monitoring. Options are: ◆ `true` Enable network partition monitoring. You can enable network monitoring only if you have set up two networks, A and B, that connect your GDSS nodes. Gemini recommends that A and B be physically separate networks. Network monitoring works by comparing heartbeats from network A and network B. ◆ `false` Disable network partition monitoring. IMPORTANT: For network partition monitoring to function properly, these `central.conf` settings must be assigned identical values on each GDSS node: ◆ `network_monitor_enable` ◆ `network_a_*` ◆ `network_b_*` ◆ `heartbeat_*`	ATOM	set during install (installer defaults to 'false')	false

central.conf Parameters (Part 12 of 16)

Parameter Description	Valid Range	File Default	Internal Default
`network_monitor_monitored_nodes`			
List of all GDSS nodes (without single quotes) that will be monitored. Use this parameter to add a new node then use use the `gmt_config_svr:reload_config()` to reload the configuration. See *Adding More Data Nodes to Cluster, on page 368* for more information. File default = <set during install> (installer defaults to `gdss1@node-a, gdss1@node-b` |	TEXT	see description	see description
`network_a_address`			
IP address for the A network. This network *must* be the same network used by the Erlang network distribution protocol (i.e. the network used for Mnesia replication traffic). File default = set during install (installer defaults to `10.1.1.12`)	STRING	see description	null
`network_a_broadcast_address`			
IP broadcast address for the A network. This network *must* be the same network used by the Erlang network distribution protocol (i.e. the network used for Mnesia replication traffic). File default = set during install (installer defaults to `10.1.1.255`)	STRING	see description	null

central.conf Parameters (Part 13 of 16)

Parameter Description	Valid Range	File Default	Internal Default
`network_a_tiebreaker`			
IP address for the A network to act as a tiebreaker. If the network monitoring application determines that the A network is partitioned and the B network is not partitioned, then if `network_a_tiebreaker` responds to an ICMP echo (a ping), then the local GDSS node is on the "correct" side of the partition. If the local GDSS node is not on the correct side of the partition (if the attempt to ping the tiebreaker address fails), then it shuts down immediately. The `network_a_tiebreaker` address must be extremely reliable and must be as close to the local GDSS node as possible (from a network Layer 1 and 2 point of view) as well as close to all other GDSS nodes. Ideally the tiebreaker should be the address of the Layer 2 switch or Layer 3 router that all Mnesia communications flow through. File default = set during install (installer defaults to `10.1.1.254`)	STRING	see description	null
`network_b_address`			
IP address for the B network. This network should be physically separate from the A network. File default = set during install (installer defaults to `10.10.10.12`)	STRING	see description	null
`network_b_broadcast_address`			
IP broadcast address for the B network. This network should be physically separate from the A network. File default = set during install (installer defaults to `10.10.10.255`)	STRING	see description	null

central.conf Parameters (Part 14 of 16)

Parameter Description	Valid Range	File Default	Internal Default
`heartbeat_beacon_interval`			
Heartbeat beacon interval in milliseconds. At this interval, UDP heartbeart signals are transmitted from the local GDSS node to each other GDSS node in the cluster. The heartbeats are sent out both through network A and through network B. Gemini recommends that this interval be between 250 and 1000 (milliseconds).	INT	1000	1000
`heartbeat_warning_interval`			
Heartbeat alarm interval in seconds. If this interval passes without the local GDSS node receiving a heartbeat signal from a peer GDSS node, an alert is written to the local application log.	INT	set during install (installer defaults to 5)	5
`heartbeat_failure_interval`			
Heartbeat failure interval in seconds. A serious error has occurred if during this interval a heartbeat from a peer GDSS node has been detected on network B but no heartbeat from that node has been detected on network A. The `network_a_tiebreaker` (*page 398*) address will be pinged to determine whether or not the local GDSS node should be shut down to avoid database damage. NOTE: The value of `heartbeat_failure_interval` should be larger than the value of `heartbeat_warning_interval` by a factor of at least 1.5x but preferably 2x or more. Cluster timeout interval. Erlang nodes will force a disconnect from each other if this timeout value is exceeded. If there is a network partition (or other failure that will cause network traffic from a node to be dropped or delayed). Operations will hang. WARNING: The `cluster_timeout` value must be larger than the `heartbeat_failure_interval` value, preferably by 5 seconds or more.	INT	set during install (installer defaults to 15)	15

central.conf Parameters (Part 15 of 16)

Parameter Description	Valid Range	File Default	Internal Default
`heartbeat_status_udp_port`			
UDP port for heartbeat listener	INT	63099	63099
`heartbeat_status_xmit_udp_port`			
UDP port for heartbeat transmitter (base port, actual port may be higher)	INT	63100	63100
`ticket_server_tcp_port`			
Port on which the GDSS listens for requests to its internal ticket broker. In the current release, this listener is not used.	INT	set during install (installer defaults to 2298)	2299
`ticket_server_distributed_nodes`			
List of all GDSS nodes running the distributed ticket server. This node list is required for GDSS start-up. The node list must be identically configured on each of your GDSS nodes. If you are using only one GDSS node, specify just the one node’s name for this setting. NOTE: If a node name includes a hyphen, then you must surround the node name with single quotes. For example: `‘GDSS1@tc-gt22-6’`	STRING	set during install	null

central.conf Parameters (Part 16 of 16)

Parameter Description	Valid Range	File Default	Internal Default
`ticket_maker_reset_timeout`			
This setting works together with the congestion monitoring controls that you establish in the `congestion_watcher.conf` file (*page 317*). When you start or restart the GDSS, or when you dynamically reload the `congestion_watcher.conf` file, the ticket broker will wait for `ticket_maker_reset_timeout` seconds before issuing any messaging tickets. This pause allows time for the congestion monitor to send restriction requests to the ticket broker, if congestion has been detected.	INT	6	6
`gms_imapd_conf`			
For internal use—do not modify.	STRING	/dev/null	/dev/null

12 M2H, A2S and GDSS Logging

This chapter describes A2S, M2H and GDSS server logging. The chapter covers this topic:

Note Statistics logging is not supported in the current release.

Application Logging

All products, the M2G, M2H, A2S and GDSS have an application log. It is an application-layer log for alerts, infos, and errors.

The application log records application-related alerts, warnings, and informational messages, as well as trace messages for debugging. By default the application log is written to this file:

```
<M2H_HOME>/1.0.0/var/log/m2h-app.log
```

or

```
<A2S_HOME>/1.0.0/var/log/a2s-app.log
```

or

```
<GDSS_HOME>/1.0.0/var/log/gdss-app.log
```

Each log entry in the A2S, M2H or GDSS application log is composed of these fields in this order, with vertical bar delimitation:

```
<PID>|<<ERLANGPID>>|<DATETIME>|<MODULE>|<LEVEL>|
<MESSAGECODE>|<MESSAGE>
```

This application log entry format is not configurable.

Each of these application log entry fields is described in the table that follows. The "Position" column indicates the position of the field within a log entry.

GDSS Application Log Fields (Part 1 of 2)

Position	Field	Description
1	`<PID>`	System-assigned process identifier (PID) of the process that generated the log message.
2	`<<ERLANGPID>>`	Erlang process indentifier. Note that the interior brackets are literal.
3	`<DATETIME>`	Timestamp in format `%Y%m%d%H%M%S`, where `%Y` = four digit year; `%m` = two digit month; `%d` = two digit date; `%H` = two digit hour; `%M` = two digit minute; and `%S` = two digit seconds. For example, `20081103230123`.
4	`<MODULE>`	The internal component with which the message is associated. This field is set to a minimum length of 13 characters. If the module name is shorter than 13 characters, spaces will be appended to the module name so that the field reaches the 13 character minimum.

GDSS Application Log Fields (Part 2 of 2)

Position	Field	Description
5	`<LEVEL>`	The severity level of the message. The level will be one of the following: ◆ `ALERT` A condition requiring immediate correction. ◆ `WARNG` A warning message, indicating a potential problem. ◆ `INFO` An informational message indicating normal activity, and requiring no action. ◆ `DEBUG` A highly granular, process-descriptive message potentially of use when debugging the application.
6	`<MESSAGECODE>`	Integer code assigned to all messages of severity level `INFO` or higher. See the *GWS Error Messages Guide* for information.
7	`<MESSAGE>`	The message itself, describing the event that has occurred.

Example The sample below shows a series of GDSS application log entries. Each new entry starts with the PID 19120. The first set of entries records various events associated with GDSS start-up.

```
19120|<0.45.0>|20081103230123|gmt_app        |INFO|2190301|start:
normal []
19120|<0.46.0>|20081103230123|SASL           |INFO|2199999|progress:
[{supervisor,{local,gmt_sup}},{started,[{pid,<0.47.0>},{name,
gmt_config_svr},{mfa,{gmt_config_svr,start_link,["/usr/local/
gemini/gdss/1.0.0/etc/central.conf"]}},{restart_type,permanent},
{shutdown,2000},{child_type,worker}]}]
19120|<0.6.0>|20081103230123|SASL           |INFO|2199999|progress:
[{application,gmt},{started_at,'gdss1@bb2-2'}]
19120|<0.55.0>|20081103230123|SASL           |INFO|2199999|progress:
[{supervisor,{local,inet_gethost_native_sup}},{started,[{pid,<0.56
.0>},{mfa,{inet_gethost_native,init,[[]]}}]}]
19120|<0.31.0>|20081103230123|SASL           |INFO|2199999|progress:
[{supervisor,{local,kernel_safe_sup}},{started,[{pid,<0.55.0>},
{name,inet_gethost_native_sup},{mfa,{inet_gethost_native,start_
link,[]}},{restart_type,temporary},{shutdown,1000},{child_type,
worker}]}]
19120|<0.31.0>|20081103230123|SASL           |INFO|2199999|progress:
[{supervisor,{local,kernel_safe_sup}},{started,[{pid,<0.57.0>},
{name,timer_server},{mfa,{timer,start_link,[]}},{restart_type,
permanent},{shutdown,1000},{child_type,worker}]}]
19120|<0.52.0>|20081103230123|NETWORK_MONITOR|INFO|2190503|
Partition detector: active status on node 'gdss1@bb2-2'
19120|<0.51.0>|20081103230123|SASL         |INFO|2199999|progress:
[{supervisor,{local,partition_detector_sup}},{started,[{pid,
<0.52.0>},{name,partition_detector_server},{mfa,{partition_
detector_server,start_link,[[]]}},{restart_type,permanent},
{shutdown,2000},{child_type,worker}]}]

[...]

19120|<0.6.0>|20081103230124|SASL         |INFO|2199999|progress:
[{application,gdss},{started_at,'gdss1@bb2-2'}]

[...]

19120|<0.6.0>|20081103230125|SASL         |INFO|2199999|progress:
[{application,mnesia},{started_at,'gdss1@bb2-2'}]
19120|<0.99.0>|20081103230125|DEFAULT       |INFO|2199999|
Table ready: external_exclusion
19120|<0.99.0>|20081103230125|DEFAULT       |INFO|2199999|
Table ready: gdict
19120|<0.99.0>|20081103230125|DEFAULT       |INFO|2199999|
Table ready: strmap
19120|<0.99.0>|20081103230125|DEFAULT       |INFO|2199999|
Table ready: schema
19120|<0.99.0>|20081103230125|DEFAULT       |INFO|2199999|
Table ready: gcxKV
19120|<0.99.0>|20081103230125|SUPERVISOR    |INFO|2190701|
External exclusion tables are available

[...]

19120|<0.52.0>|20081103230129|NETWORK_MONITOR|ALERT|2180506|
Alarm SET: network_heartbeat: 'A'
19120|<0.52.0>|20081103230129|NETWORK_MONITOR|ALERT|2180506|
Alarm SET: network_heartbeat: 'B'

[...]
```

Example

The following is an A2S application log example:

```
26597|<0.45.0>|20091209141014|gmt_app      |INFO|2190301|start:
normal []

26597|<0.47.0>|20091209141014|SASL
|INFO|2199999|progress:[{supervisor,{local,gmt_sup}},{started,
[{pid,<0.48.0>},{name,gmt_config_svr},{mfa,{gmt_config_svr,sta
rt_link,["/usr/local/gemini/a2s/1.0.0/etc/
central.conf"]}},{restart_type,permanent},{shutdown,2000},{chi
ld_type,worker}]}]

26597|<0.47.0>|20091209141014|SASL
|INFO|2199999|progress:
[{supervisor,{local,gmt_sup}},{started,[{pid,<0.49.0>},{name,g
mt_cli},{mfa,{gmt_cli,start_link,[7586,"eldap_cli","CLI>","A2S
CLI
Server"]}},{restart_type,permanent},{shutdown,2000},{child_typ
e,worker}]}]

26597|<0.7.0>|20091209141014|SASL
|INFO|2199999|progress:
[{application,gmt},{started_at,a2s1@pollux}]

26597|<0.54.0>|20091209141014|SASL
|INFO|2199999|progress:
[{supervisor,{local,gldaps_sup}},{started,[{pid,<0.55.0>},{nam
e,gldaps},{mfa,{eldap_server,start_link,[eldap_proxy,eldap_pro
xy,7574,10000,connected_clients]}},{restart_type,permanent},{s
hutdown,2000},{child_type,worker}]}]

26597|<0.7.0>|20091209141014|SASL
|INFO|2199999|progress:
[{application,gldaps},{started_at,a2s1@pollux}]

26597|<0.61.0>|20091209141014|NETWORK_MONITOR|WARNG|2180510|Ne
twork monitor is not enabled

26597|<0.60.0>|20091209141014|SASL
|INFO|2199999|progress:
[{supervisor,{local,partition_detector_sup}},{started,[{pid,<0
.61.0>},{name,partition_detector_server},{mfa,{partition_detec
tor_server,start_link,[[]]}},{restart_type,permanent},{shutdow
n,2000},{child_type,worker}]}]

26597|<0.60.0>|20091209141014|SASL
|INFO|2199999|pr?gress:
[{supervisor,{local,partition_detector_sup}},{started,[{pid,<0
.62.0>},{name,partition_detector_mnesia},{mfa,{partition_detec
tor_mnesia,start_link,[[]]}},{restart_type,permanent},{shutdow
n,2000},{child_type,worker}]}]

26597|<0.7.0>|20091209141014|SASL
```

```
|INFO|2199999|progress:
[{application,partition_detector},{started_at,a2s1@pollux}]
```

Example The following is a M2H application log example:

```
26861|<0.70.0>|20091209141030|gmt_app     |INFO|2190301|start:
normal  []

26861|<0.72.0>|20091209141030|SASL
|INFO|2199999|progress:
[{supervisor,{local,gmt_sup}},{started,[{pid,<0.73.0>},{name,g
mt_config_svr},{mfa,{gmt_config_svr,start_link,["/usr/local/
gemini/m2h/1.0.0/etc/
central.conf"]}},{restart_type,permanent},{shutdown,2000},{chi
ld_type,worker}]}]

26861|<0.72.0>|20091209141030|SASL
|INFO|2199999|progress:
[{supervisor,{local,gmt_sup}},{started,[{pid,<0.74.0>},{name,g
mt_cli},{mfa,{gmt_cli,start_link,[7585,"m2h_cli","CLI>","M2H
CLI
Server"]}},{restart_type,permanent},{shutdown,2000},{child_typ
e,worker}]}]

26861|<0.7.0>|20091209141030|SASL
|INFO|2199999|progress:
[{application,gmt},{started_at,m2h1@pollux}]

26861|<0.34.0>|20091209141030|SASL
|INFO|2199999|progress:
[{supervisor,{local,kernel_safe_sup}},{started,[{pid,<0.81.0>}
,{name,timer_server},{mfa,{timer,start_link,[]}},{restart_type
,permanent},{shutdown,1000},{child_type,worker}]}]

26861|<0.79.0>|20091209141030|SASL
|INFO|2199999|progress:
[{supervisor,{local,ticket_server_sup}},{started,[{pid,<0.80.0
>},{name,ticket_stats},{mfa,{ticket_stats,start_link,[]}},{res
tart_type,permanent},{shutdown,2000},{child_type,worker}]}]

26861|<0.79.0>|20091209141030|SASL
|INFO|2199999|progress:
[{supervisor,{local,ticket_server_sup}},{started,[{pid,<0.82.0
>},{name,ticket_maker},{mfa,{ticket_maker,start_link,["/usr/
local/gemini/m2h/1.0.0/etc/
broker.conf"]}},{restart_type,permanent},{shutdown,2000},{chil
d_type,worker}]}]

26861|<0.102.0>|20091209141030|SASL
|INFO|2199999|progress:
[{supervisor,{<0.102.0>,ticket_proto}},{started,[{pid,<0.103.0
>},{mfa,{ticket_proto,init,[[2299,ticket_proto_listener]]}}]}]

26861|<0.79.0>|20091209141030|SASL
|INFO|2199999|progress:
[{supervisor,{local,ticket_server_sup}},{started,[{pid,<0.102.
0>},{name,ticket_proto},{mfa,{supervisor_bridge,start_link,[ti
cket_proto,[2299,ticket_proto_listener]]}},{restart_type,perma
nent},{shutdown,2000},{child_type,worker}]}]

26861|<0.7.0>|20091209141030|SASL
|INFO|2199999|progress:
[{application,ticket_server},{started_at,m2h1@pollux}]
```

The A2S, M2H or GDSS package does not establish a rotation regime for the application log. You can set up rotation for the application log using the operating system's `logrotate` utility. See *Log Rotation and Removal, on page 410* for more information.

Log Rotation and Removal

Rotation of application logs is implemented by the host system's `logrotate` utility. The `logrotate` utility is invoked by cron tab entries established during product installation. The cron tab entries are configured in this file:

```
<GDSS_HOME>/bin/cron.d/gdss-app
```

or

```
<A2S_HOME>/bin/cron.d/a2s-app
```

or

```
<M2H_HOME>/bin/cron.d/m2h-app
```

Default Log Rotation and Removal

By default, the cron tab entries established in the `gdss-app`, `m2h-app` or `a2s-app` file invoke `logrotate` hourly, at the top of the hour, using the `logrotate` configuration specified in this file:

```
<GDSS_HOME>/etc/logrotate-logs/gdss-app-rotate
```

or

```
<A2S_HOME>/etc/logrotate-logs/a2s-app-rotate
```

or

```
<M2H_HOME>/etc/logrotate-logs/m2h-app-rotate
```

The default configuration in the rotate file specifies that when `logrotate` is run each hour, it will rotate the application log either if the log is larger than 10MB, or if the time has arrived for a daily rotation. So rotation occurs at least once per day, and potentially more often depending on the size of the log. Rotated files are gzipped and stored to this directory:

```
<GDSS_HOME>/var/log/archive
```

or

```
<A2S_HOME>/var/log/archive
```

or

```
<M2H_HOME>/var/log/archive
```

The `logrotate` utility is configured to delete archived log files after 30 rotations.

Changing Log Rotation and Removal

To change the timing of logrotate invocation for the application, edit the cron job timing specifications in the <GDSS|A2S|M2H_HOME>/bin/cron.d/*-app file. To change how logrotate implements its rotation of files, edit the logrotate configuration in the file <GDSS|A2S|M2H_HOME>/etc/logrotate-logs/*-app-rotate. For guidance on how to configure logrotate, see the logrotate man page.

Note In the default *-app file there is a find and rm entry that performs an hourly removal of archived logs that are more than 30 days old. This entry is redundant since in the current release there is only the application log, and removal of old archived application logs is handled by logrotate. Note however that if you increase the configured archive period for logrotate (from its default of 30 rotations), you should also edit or delete the find and rm entry in *-app. For example, if in the gdss-app-rotate file you change the logrotate option rotate 30 to rotate 60, then in the gdss-app file you should change the find and rm entry option -mtime +30 to -mtime +60. Alternatively you can simply delete this entry from the file.

A2S and M2H Mnesia Database Logging

The A2S and M2H Mnesia databases use a periodic process to write new transaction data to the per-table data files. If Mnesia is exceptionally busy, new transactions can arrive faster than the table files can be updated. If that situation lasts long enough, the Mnesia transaction log, `LATEST.LOG`, can grow to consume all disk space.

You can configure a limit on the size of `LATEST.LOG` by using the parameter `mnesia_diskmon_latest_log_max` in the A2S `central.conf` file (*page 338*) or in the M2H `central.conf` file (*page 281*).

If the size of `LATEST.LOG` exceeds this configurable limit, then the A2S | M2H logs an alarm message to the application log, and triggers an artificial throttle to all transactions to slow them down, giving Mnesia enough time to finish its maintenance and truncate `LATEST.LOG`. Once truncated, the alarm is cleared (with another application log message) and the throttle on transactions is lifted.

M2H and A2S Transaction Logging

M2H and A2S transaction logging is similar to the M2G's transaction log. They are rotated every 30 minutes. The format of the transaction log for M2H and A2S is different and different than M2G.

M2H has these logging fields in this order, separated by "|"

- pid
- threadid
- date
- type
- proto
- status
- client
- duration
- server
- trid
- gtrid
- yaguid
- uraid
- http_username
- http_req_body_size
- http_req_host
- http_req_line
- http_req_port
- http_req_size
- http_res_body_size
- http_res_code
- http_res_size
- nttr_soid
- nttr_ordertype
- nttr_cmd
- nttr_keyid

- nttr_sts
- ubf_contract
- ubf_req_method
- ubf_req_authinfo

A2S has these fields

- pid
- threadid
- date
- type
- proto
- status
- client
- duration
- server
- trid
- gtrid
- yaguid
- ldap_req_filter
- ubf_contract
- ubf_req_method
- ubf_req_authinfo

A Index of Settings in .properties and .conf Files