



Enterprise Mobile Mail Agent V.3 User Manual

Contents

EMMA Overview	3
Introduction	3
EMMA Architecture	3
EMMA Directory Structure	3
Database Management System (DBMS)	5
EMMA Operations	7
Running EMMA (Shell)	7
Running EMMA (Service)	7
Checking EMMA Status	7
Stopping EMMA (Shell)	7
Stopping EMMA (Service)	7
Logs	7
Sending SMS (Short Message Service)	8
Definition	8
Steps to Send SMS	8
Example	9
Sending LMS (Long Message Service)	10
Definition	10
Steps to Send LMS	10
Appendix	12
EMMA Result Code Table	12

EMMA Overview

Introduction

EMMA is a software that sends/receives text messages to/from end-users' mobile devices. EMMA supports SMS (Short Message Services), LMS (Long Message Services) and MMS (Multimedia Message Services).

EMMA can be installed on both UNIX and Windows OS environments and supports various DBMS such as MySQL, Oracle, DB2 and Microsoft SQL Server.

EMMA Architecture

In concise, EMMA consists of 2 major components: a module and database tables/stored procedures. A module must be installed on server. Database tables and stored procedures must be created on DBMS.

On execution of EMMA, the module calls necessary stored procedures from DBMS to collect and send messages end-users. After sending messages, reports are received from Infobank servers and updated to DBMS.

EMMA Directory Structure

EMMA has a default directory structure as below:

```
charlie@cass:~/EMMA_3_4_1$ ls -al
total 120
drwxrwxr-x 13 charlie charlie 4096 Mar 12 11:26 .
drwxr-xr-x  9 charlie charlie 4096 Apr 12 16:51 ..
drwxr-xr-x  4 charlie charlie 4096 Jan 15 13:36 attach_file
-rwxr-xr-x  1 charlie charlie 7425 Jan 11 16:57 authcipher
drwxr-xr-x  3 charlie charlie 4096 Jan 15 13:36 ban
drwxr-xr-x  2 charlie charlie 4096 Jan 15 13:37 cert
drwxr-xr-x  2 charlie charlie 4096 Jan 11 16:57 classes
drwxr-xr-x  2 charlie charlie 4096 Feb 28 12:18 conf
-rwxr-xr-x  1 charlie charlie 7422 Jan 11 16:57 dbcheck
-rwxr-xr-x  1 charlie charlie 7423 Jan 11 16:57 dbcipher
-rwxr-xr-x  1 charlie charlie 7503 Jan 11 16:57 emma
-rwxr-xr-x  1 charlie charlie 8084 Jan 11 16:57 emmasvc
-rw-r--r--  1 charlie charlie 3974 Jan 11 16:57 faq.txt
-rw-r--r--  1 charlie charlie 6932 Jan 11 16:57 history.txt
drwxrwxr-x  2 charlie charlie 4096 Jan 15 13:36 .install4j
drwxr-xr-x  2 charlie charlie 4096 Jan 15 13:36 lib
drwxr-xr-x  2 charlie charlie 4096 Jan 15 13:50 logs
drwxr-xr-x  2 charlie charlie 4096 Jan 15 13:50 que
-rwxr-xr-x  1 charlie charlie 7475 Jan 11 16:57 regkey
drwxr-xr-x  2 charlie charlie 4096 Jan 15 13:36 sql
drwxr-xr-x  3 charlie charlie 4096 Jan 15 13:36 src
-rwx----- 1 charlie charlie 7982 Jan 11 16:57 uninstall
```

Figure 1. EMMA3 Directory Structure

Important directories and executable are briefly explained as follow:

- attach_file

- Directory where attachments for messages are stored.
- authcipher
Executable to encrypt authentication information within EMMA configuration file.
- ban
Directory where banned data files are stored.
- cert
Directory where private key for certification is stored.
- classes
Directory where callback user classes are stored.
- conf
Directory where configurations for EMMA are stored.
- dbcheck
Executable to check connectivity and permissions with DBMS.
- dbcipher
Executable to encrypt database authentication information within EMMA configuration file.
- emmatest
Executable to start EMMA on shell
- emmasvc
Executable to start EMMA as a background service.
- .install4j
Directory where basic EMMA configurations are stored.
- lib
Directory where necessary libraries to execute EMMA are stored.
- logs
Directory where EMMA logs are stored.
- que
Directory where queues for EMMA execution are stored.
- regkey
Executable to generate and send public/private key to Infobank's server for certification.
- sql
Directory where SQL scripts are stored.
- src
Directory where callback user classes are stored.
- uninstall
Executable to uninstall EMMA from system.

Database Management System (DBMS)

By default, about 13 tables and 51 stored procedures are created on DBMS.

Data to send message (recipient phone number, content, etc.) to end-users are stored in tables. Also, sent message results are updated to log tables. The below is a description of SMS message table for send.

Field	Type	Null	Key	Default	Extra
☐ mt_pr	int(11)	7B NO	PRI	(NULL)	OK auto_increment
☐ mt_refkey	varchar(20)	11B YES		(NULL)	OK
☐ priority	char(2)	7B NO		S	1B
☐ date_client_req	datetime	8B NO		1970-01-01 00:00:00	19B
☐ content	varchar(255)	12B NO		(NULL)	OK
☐ callback	varchar(25)	11B NO		(NULL)	OK
☐ service_type	char(2)	7B NO		(NULL)	OK
☐ broadcast_yn	char(1)	7B NO		N	1B
☐ msg_status	char(1)	7B NO	MUL	1	1B
☐ recipient_num	varchar(25)	11B YES	MUL	(NULL)	OK
☐ date_mt_sent	datetime	8B YES		(NULL)	OK
☐ date_rslt	datetime	8B YES		(NULL)	OK
☐ date_mt_report	datetime	8B YES		(NULL)	OK
☐ mt_report_code_ib	char(4)	7B YES		(NULL)	OK
☐ mt_report_code_ibtype	char(1)	7B YES		(NULL)	OK
☐ carrier	int(5)	6B YES		(NULL)	OK
☐ rs_id	varchar(20)	11B YES		(NULL)	OK
☐ recipient_net	int(5)	6B YES		(NULL)	OK
☐ recipient_npsend	char(1)	7B YES		(NULL)	OK
☐ country_code	varchar(8)	10B NO		82	2B
☐ charset	varchar(20)	11B YES		(NULL)	OK
☐ msg_type	int(11)	7B YES		(NULL)	OK
☐ crypto_yn	char(1)	7B YES		N	1B
☐ ttl	int(5)	6B YES		(NULL)	OK
☐ emma_id	char(2)	7B YES	MUL		0B
☐ reg_date	timestamp	9B NO		CURRENT_TIMESTAMP	17B

Figure 2. EM_SMT_TRAN Table

Some of the most important columns to discuss here are `recipient_num`, `callback`, `content`, `date_client_req` and `msg_status`.

- `recipient_num`
end-user's phone number to receive text message.
- `callback`
source number that sent message to the end-user.
- `content`
Actual message text to send.
- `date_client_req`
Date on which this message should be sent.
Future date can be entered to make this message a reserved message.
- `msg_status`

Current status of this message.

`msg_status` with value of 1 means the message is ready to send and expected to be collected by the EMMA module.

`msg_status` with value of 2 means that the message is currently in the progress of sending the message. Message request to send cannot be cancelled at this point.

`msg_status` with value of 3 means that the message has finished its process.

Explanations on other tables and stored procedures are beyond the scope of this document.

EMMA Operations

TEST Running EMMA (Shell)

EMMA can be executed from shell with command below:

```
$/emmatest
```

Running EMMA (Service)

EMMA can be executed as a service from the shell with command below:

```
$/emmasvc status
The daemon is stopped.
$/emmasvc start
Starting emmasvc
```

Make sure to check the status before starting the service.

Checking EMMA Status

When EMMA is executed on background via `emmasvc` executable, following command can be used to check current running status of EMMA:

```
$/emmasvc status
The daemon is stopped.
```

Stopping EMMA (Shell)

EMMA can be stopped from the shell using OS interrupt. (Usually CTRL+C)

Stopping EMMA (Service)

EMMA service can be stopped with command below:

```
$/emmasvc stop
Shutting down emmasvc
The daemon is not running.
```

Logs

EMMA log files are stored under logs folder situated under EMMA installed directory. Log files are automatically backed up each day having date as a postfix to its filename. Logs can be opened using text editors such as VI or Notepad.

Sending SMS (Short Message Service)

Definition

SMS is a text message with 90 or less bytes long message.

Mandatory data required to send SMS are:

- Recipient phone number
Phone number of the end-user to receive text message
- Date to send
Date to send text message
- Source (Originator) phone number
Phone number of the sender.
- Content
Actual text message to send.

Steps to Send SMS

Steps to send simple single SMS is explained as follow:

1. Insert data on EM_SMT_TRAN table using SQL query as follow:

```
INSERT INTO em_smt_tran (date_client_req, content, callback,
service_type, broadcast_yn, msg_status, recipient_num) VALUES
(SYSDATE(), 'Hello World!', '0298765678', '0', 'N', '1',
'01012345678');
```

mt_pr	mt_refkey	priority	date_client_req	content	callback	service_type	broadcast_yn	msg_status	recipient_num	date_mt_sent	date_rslt	date_mt_report
175	(NULL)	S	2013-04-18 16:28:26	Hello World!	298765678	0	N	1	01012345678	(NULL)	(NULL)	(NULL)

Figure 3. Data tuple inserted on EM_SMT_TRAN

Instead of SYSDATE(), future date can be inserted to make this message a reserved message. For example, if you need to send a text message to end-user on Dec. 25 2013 0900 a.m., you could write a query as follow:

```
INSERT INTO em_smt_tran (date_client_req, content, callback,
service_type, broadcast_yn, msg_status, recipient_num) VALUES
('2013-12-25 09:00:00', 'Merry Christmas!', '0298765678', '0', 'N',
'1', '01012345678');
```

Note that service_type, broadcast_yn and msg_status are fixed to '0', 'N' and '1' respectively.

2. EMMA module will then collect, send and receive report on above data as follow: (printed in log file)

```
[2013/04/18 16:33:58.102] [ INFO] [Thread-0] MySQLSMTDBHandler selectSmtTran SMT QUE:
175/0/01012345678/298765678/2013-04-18 16:28:26.0
[2013/04/18 16:34:00.590] [ INFO] [Thread-0] MTSEnder sendSmsPacket SMT SEND:
```

```
175/1000/newrs5
```

```
[2013/04/18 16:34:02.605] [ INFO] [Thread-0] MySQLRPTDBHandler updateMtResult SMT RPT:  
1/175/01012345678/0
```

3. After receiving a report, a data tuple will be removed from EM_SMT_TRAN table automatically and moved to EM_SMT_LOG_201304 table as follow:

mt_pr	mt_seq	mt_refkey	priority	date_client_req	content	callback	service_type	broadcast_yn	msg_status	recipient_num	change_word!
175		0 (NULL)	S	2013-04-18 16:28:26	Hello World!	298765678	0	N	3	01012345678	(NULL)

Figure 4. Data tuple moved to EM_SMT_LOG_201304

Notice that `msg_status` is 3, indicating that the message has finished its process.

Example

Here is an exemplary scenario where your firm (Office Tel no.: 02-9876-5678) would like to send text message to Bob (Tel no.: 010-1234-5678) saying "Hello World!" immediately. In this scenario, the query would be

```
INSERT INTO em_smt_tran (date_client_req, content, callback,  
service_type, broadcast_yn, msg_status, recipient_num) VALUES  
(SYSDATE(), 'Hello World!', '0298765678', '0', 'N', '1',  
'01012345678');
```

Sending LMS (Long Message Service)

Definition

LMS is a text message with 90 or more bytes long message, containing subject field separately.

Mandatory data required to send SMS are:

- Recipient phone number
Phone number of the end-user to receive text message
- Date to send
Date to send text message
- Source (Originator) phone number
Phone number of the sender.
- Content
Actual text message to send.
- Subject
Subject text of this message.

Steps to Send LMS

Steps to send simple single LMS is explained as follow:

1. Insert data on EM_MMT_TRAN table using SQL query as follow:

```
INSERT INTO em_mmt_tran (date_client_req, subject, content,
attach_file_group_key, callback, service_type, broadcast_yn,
msg_status, recipient_num)
VALUES (SYSDATE(), 'Test Subject', 'Hello World!', '0',
'0298765678', '2', 'N', '1', '01012345678');
```

mt_pr	mt_refkey	priority	msg_class	date_client_req	subject	content_type	content	attach_file_group_key	callback	service_type	broadcast_yn	msg_status	reciple
88	(NULL)	S	1	2013-04-18 17:04:26	Test Subject		0 Hello World!		0 0298765678	2	N	1	01012345678

Figure 5. Data tuple inserted on EM_MMT_TRAN

Instead of SYSDATE(), future date can be inserted to make this message a reserved message. For example, if you need to send a text message to end-user on Dec. 25 2013 0900 a.m., you could write a query as follow:

```
INSERT INTO em_mmt_tran (date_client_req, subject, content,
attach_file_group_key, callback, service_type, broadcast_yn,
msg_status, recipient_num)
VALUES (SYSDATE(), 'Christmas', 'Merry Christmas!', '0',
'0298765678', '2', 'N', '1', '01012345678');
```

Note that attach_file_group_key, service_type, broadcast_yn and msg_status are fixed to '0', '2', 'N' and '1' respectively.

2. EMMA module will then collect, send and receive report on above data as follow:
(printed in log file)

```
[2013/04/18 17:12:15.811] [ INFO] [Thread-0] MySQLMMTDBHandler selectMmtTran MMT QUE:
88/0/01012345678/0298765678/2013-04-18 17:04:26.0

[2013/04/18 17:12:18.556] [ INFO] [Thread-0] MTRPTReceiver SaveToEmmaReportQue MMT
RSLT: 88/0/3015/01012345678/10000/1366272738000/newrs5/ETAG=0000

[2013/04/18 17:12:19.976] [ INFO] [Thread-0] MySQLRPTDBHandler updateMtResult MMT RPT:
1/88/01012345678/0
```

3. After receiving a report, a data tuple will be removed from EM_MMT_TRAN table automatically and moved to EM_MMT_LOG_201304 table as follow:

mt_pr	mt_seq	mt_refkey	priority	msg_class	date_client_req	subject	content_type	content	attach_file_group_key	callback	service_type	broadcast_yn	msg_status
88		0 (NULL)	S	I	2013-04-18 17:04:26	Test Subject		0 Hello World!		0 0298765678	2	N	3

Figure 6. Data tuple moved to EM_MMT_LOG_201304

Notice that `msg_status` is 3, indicating that the message has finished its process.

Appendix

EMMA Result Code Table

There is a column named `mt_report_code_ib` of which its value indicates the result of message delivery. The table below summarizes the possible result code and its meanings.

Code	Result	Explanation
1000	Success	Success
2000	Failure	Time expired
2001	Failure	Failed to send
2002	Failure	Failed to send to device
2003	Failure	Device turned off
2004	Failure	Device message buffer is full
2005	Failure	Out of range
2006	Failure	Deleted message
2007	Failure	Temporary device problem
3000	Invalid	Failed to deliver
3001	Invalid	No end-user
3002	Invalid	Inappropriate content
3003	Invalid	Invalid recipient number format
3004	Invalid	Device service temporarily disabled
3005	Invalid	Failed to process
3006	Invalid	Unsubscribed
3007	Invalid	Not supported device
3008	Invalid	Uncategorized error
3009	Invalid	Message format error
3010	Invalid	MMS Not supported device
3011	Invalid	Server error
3012	Invalid	Spam
3013	Invalid	Service denied
3014	Invalid	Others
3015	Invalid	No path
1001		Server Busy
1002		Invalid recipient number format
1003		Invalid callback number format
1004		Spam
1005		Count exceeded
1006		No attachment

1007		Attachment exists
1008		Failed to save attachment
1009		No client msg key
1010		No content
1011		No callback
1012		No recipient info
1013		No subject
1014		No attachment file key
1015		No attachment file name
1016		No attachment file size
1017		No attachment file content
1018		Not authorized
1019		TTL expired