



Enterprise Mobile Mail Agent V.3

사용자 설명서

Version 3.5.2.1

Enterprise Mobile Mail Agent V.3

사용자 설명서

Version 3.5.2.1

Copyright©InfoBank Corp. All rights reserved. 2008

이 프로그램은 핵, 항공, 대량 수송, 의료 또는 기타 본질적으로 위험한 상황에서 사용하도록 개발된 것이 아닙니다. 따라서 그런 목적으로 사용된 경우, 사용자는 응용 프로그램의 안전한 사용을 보장하기 위한 적절한 안전 조치, 백업, 대비 및 기타 조치를 반드시 취해야 합니다. 프로그램이 이러한 목적으로 사용되었을 경우 인포뱅크(주)는 이러한 프로그램 사용으로 인한 피해를 책임지지 않습니다.

이 프로그램(소프트웨어 설명서 포함)은 저작권법, 특허 및 지적재산권 관련 법규에 의해 보호됩니다. 이 프로그램을 리버스 엔지니어링하거나 분해하거나 또는 역 컴파일 하는 것은 금지되어 있습니다. 또한 이 프로그램에서 사용되는 프로토콜을 리버스 엔지니어링으로 분석하는 것도 금지되어 있습니다.

이 문서의 내용은 사전 공지 없이 변경될 수 있습니다. 이 문서의 어떤 부분도 어떠한 목적이든 인포뱅크(주)의 허가 없이 기계적, 전자적 또는 기타 임의의 수단을 통해 배포되거나 복사할 수 없습니다.

Enterprise Mobile Mail Agent에 관한 궁금한 사항은 인포뱅크(주)로 문의하시기 바랍니다.

변경 이력

버전	일자	내용	작성자
1.1.6	2010-02-01	변경이력 추가	류재훈
1.1.7	2010-09-16	변경이력 추가	이정형
1.1.8	2010-10-20	변경이력 추가	이재희
1.3.0	2011-07-25	변경이력 추가 (EMMA Version : EMMA V3.3.0 이상)	이재희
1.4.0	2012-11-19	변경이력 추가 (EMMA Version : EMMA V3.4.0 이상)	권태진
1.4.1	2014-03-07	crypto_yn CHAR(1) default 'N' → 'Y' 로 변경 em_mmt_file_group --> em_mmt_file 변경	이재희
1.4.2	2014-04-18	LMS/MMS 서비스 검증 추가 (4.5, 4.6)	김정훈
3.4.8.1	2014-06-26	change_word(동보단어) NULL값을 공백(space)으로 처리	김정훈
3.4.8.1	2014-07-01	IBDBSCallback User Library 추가	강덕현
3.4.10.1	2014-08-13	이통사 명칭 수정 (KTF→KT, LGT→LGT+, KT→KTAnn)	이재희
3.4.10.1	2014-09-04	동보단어(change_word) 제약문자 추가 (4.4) → 특수문자 중 pipe(" ") 문자	신현수
3.4.11.1	2014-10-14	동보단어(change_word) 제약문자 추가 (4.4) → "%CHANGEWORD%", "%CHANGEWORD1%", "%CHANGEWORD2%", "%CHANGEWORD3%", "%CHANGEWORD4%", "%CHANGEWORD5%"	신현수
3.4.11.1	2014-10-14	Mysql 표기를 Mysql/Maria로 변경	김정훈
3.4.13.1	2014-06-16	MMS 첨부 파일 SIZE 제약 사항 추가 (결과코드 '3016')	김정훈
3.4.13.1	2014-06-16	Change_word 관련 제약 사항 추가	김정훈
3.4.13.2	2015-08-06	4.15 국제 문자 Oracle Test 쿼리 문 수정 (Sysdate() -> sysdate)	김정훈
3.5.0.0	2015-12-08	사용자 설명서 개편	이재희
3.5.0.0	2015-12-08	msg_key 추가 : 메시지 키 값으로 사용	신현수
3.5.0.0	2015-12-08	TPM 기능 설정 추가	신현수
3.5.0.0	2015-12-08	국제 메시지 발송을 위한 가이드 내역 추가	신현수
3.5.0.0	2015-12-08	PostgreSQL DB 추가	신현수
3.5.0.0	2015-12-08	MSSQL query timeout 발생 시 db 연결을 끊고 재 연결 시도하는 기능 추가	신현수
3.5.0.0	2015-12-08	MMSMT 통신사 포맷 전송 방식 내역 삭제	신현수
3.5.0.0	2015-12-08	MO 릴레이기능 추가(Bridge 사용시)	신현수

3.5.0.0	2015-12-08	분배기능 추가(Bridge 사용시)	신현수
3.5.1.0	2016-02-16	em_banlist(스팸차단) 테이블 내용에 대한 갱신 주기 설정 - db.cf : db.bancheck.interval = 60	권성민
3.5.1.0	2016-02-16	Bypass Module 사용을 위한 설정 추가 - emma.cf : bypass.use = 0 bypass.address0 = 127.0.0.1:3000	권성민
3.5.2.0	2016-03-21	SMT, MMT 개별 메시지 전송 MSSQL, MySQL/Maria 예제 SQL 추가	권성민
3.5.2.1	2016-07-04	메시지 제공 길이에 대한 가이드 변경 - SMS MT : 90 bytes 이하 - LMS MT : 91 bytes 이상	이재희

목차

1. 문서 개요.....	7
1.1. 소개	7
1.2. 용어	7
2. EMMA	8
2.1. 개요	8
2.2. SERVICE ARCHITECTURE	8
2.3. MT 메시지 처리 순서	9
2.4. MO 메시지 처리 순서	10
3. DBMS 연동.....	11
3.1. 사용 TABLE.....	11
3.2. STORED PROCEDURE.....	12
3.3. 테이블에 특정 필드 추가 방법	14
3.4. 로그 테이블 고정 하기	16
4. MT SERVICE 사용.....	17
4.1. SMS MT 개별 메시지 전송	17
4.2. URL MT 개별 메시지 전송	18
4.3. SMS MT 동보 메시지 전송	19
4.4. SMS MT 동보 단어 사용 방법	20
4.5. MMS MT 개별 메시지 전송(첨부파일 미포함)	21
4.6. MMS MT 개별 메시지 전송 (첨부파일 포함).....	22
4.7. MMS MT 동보 메시지 전송	24
4.8. MMS MT 동보 단어 사용 방법	24
4.9. 전송 메시지 암호화 하기	25
4.10. 우선 순위 적용 하여 전송하기	25
4.11. 전송 요청 통신사 지정하여 전송하기	26
4.12. 부서코드(참조)필드 지정하기	27
4.13. 국제 문자 지정하여 전송하기	27
5. MO SERVICE.....	29
5.1. SMS MO 수신 확인	29
5.2. MMS MO 수신 확인	29
6. 전송차단 및 SPAM.....	30
6.1. 인증서버 전송차단 파일	30

6.2.	DB TABLE을 통한 SPAM 체크	30
7.	환경설정.....	31
7.1.	EMMA.CF	31
7.1.1.	인증 정보	31
7.1.2.	프로세스(Thread) 사용여부 설정	31
7.1.3.	프로세스 세션 개수 설정	34
7.1.4.	프로세스 동작 유효 시간 설정	34
7.1.5.	메시지 발송 우선순위 설정	35
7.1.6.	큐 정보 설정	36
7.1.7.	파일 설정	38
7.1.8.	G/W 통신 설정	39
7.1.9.	기타 정보 설정	40
7.1.10.	TPM(Throughput Per Minute) 설정.....	41
7.1.11.	User Library 설정.....	41
7.1.12.	HA(High Availability) 설정	43
7.1.13.	BRIDGE or DIST 모듈 연동 시 사용 되는 설정	44
7.1.14.	Bypass 모듈 연동 시 사용 되는 설정	45
7.2.	DB.CF 설정	45
7.2.1.	데이터베이스 연결 정보	45
7.2.2.	기타 설정	47
7.3.	LOG4J.PROPERTIES.....	49
8.	USER LIBRARY	49
8.1.	USER LIBRARY 종류.....	50
8.1.1.	SMS MT Callback (IBSMTCallback.java)	50
8.1.2.	MMS MT Callback (IBMMTCallback.java).....	52
8.1.3.	SMS MO Callback (IBSMOCallback.java)	54
8.1.4.	MMS MO Callback (IBMMOCallback.java).....	55
8.1.5.	MT Data Transform Callback (IBMTTCallback.java)	56
8.1.6.	MO Data Transform Callback (IBMOTCallback.java)	57
8.1.7.	User Scheduler Callback (IBUSCCallback.java)	58
8.1.8.	DB Security Callback (IBDBSCallback.java)	58
8.2.	USER LIBRARY COMPILE 및 동작방법	58
8.2.1.	SMS MT User Library Compile 방법	58
8.2.2.	SMS MT User Library 설정	60

1. 문서 개요

1.1. 소개

본 문서에는 사용자가 EMMA를 이용하여 메시지를 전송, 수신 하는 방법에 대하여 설명한다.
EMMA의 설치 및 실행 방법은 "EMMA V3 for Java 설치매뉴얼 ~.doc" 을 참조하기 바란다.

참고

Enterprise Mobile Mail Agent(EMMA)란?

인포뱅크(주)의 M& Messaging Service를 이용하기 위하여 고객사의 시스템에 배포되는 클라이언트 프로그램을 말한다. EMMA는 SMS MT, Callback URL, MMS MT를 전송할 수 있고 SMS MO, MMS MO를 수신할 수 있다.

1.2. 용어

- M& Messaging Service : 일반 SMS MT, Callback URL, MMS MT, SMS MO, MMS MO 서비스를 통칭하거나 그 중 일부를 지칭하는 말
- MT(Mobile Terminated) : 통신사의 서버를 통하여 단말기로 전송하는 서비스로 SMS MT, Callback URL, MMS MT를 통칭함
- MO(Mobile Originated) : 이동통신단말기에서 통신사 서버를 통하여 인포뱅크(주) G/W 로 메시지가 수신되는 서비스로 SMS MO, MMS MO를 통칭함
- SMS MT : 휴대폰 단문 메시지 전송 서비스
- Callback URL : 무선인터넷 주소(Callback URL)를 탑재한 메시지 전송 서비스
- MMS MT: 사진, 소리, 동영상 등의 멀티미디어 메시지를 전송하는 서비스
- SMS MO : 휴대폰으로부터 발송된 SMS 메시지를 수신하는 서비스
- MMS MO : 휴대폰으로부터 발송된 MMS 메시지를 수신하는 서비스
- G/W : M& Message Gateway를 지칭하는 말. RS/TS로 구성됨
- 단말기 : 메시지를 수신할 수 있는 모든 종류의 기기
- 메시지 : 일반 SMS MT, Callback URL, MMS MT, SMS MO, MMS MO 를 통칭하거나 그 중 일부를 지칭하는 말
- 개별메시지 : 각각의 수신번호에 각각의 메시지 내용을 전송
- 동보메시지 : 같은 내용의 메시지를 서로 다른 수신자에게 일괄 전송

2. EMMA

이 절에서는 EMMA의 개요 및 서비스 아키텍처에 대해 설명을 한다.

2.1. 개요

문서 개요에서 언급 했듯이 EMMA 는 SMS MT, Callback URL, MMS MT를 전송할 수 있고, SMS MO, MMS MO 을 수신하는 서비스를 제공한다.

- **SMS MT (Short Message Service Mobile Terminated)**
90 bytes 이하의 휴대폰 단문 메시지를 전송하는 서비스이다. 일반적으로 SMS 문자 전송 서비스를 말한다.
- **URL MT(Callback URL Mobile Terminated)**
기존의 문자메시지 서비스(SMS MT)에 무선인터넷 주소(Callback URL)을 탑재한 메시지를 전송하는 서비스이다. 문자메시지를 수신 받은 회원은 통화만 누르면 바로 무선 인터넷에 접속할 수 있는 서비스이다.
- **MMS MT(Multimedia Messaging Service Mobile Terminated)**
기존의 문자메시지 서비스(SMS MT)에 91 bytes 이상의 장문 메시지 및 사진, 소리, 동영상 등의 멀티미디어 메시지를 전송하는 서비스이다.
- **SMS MO(Short Message Service Mobile Originated)**
휴대폰으로부터 발송된 SMS 메시지를 수신하는 서비스를 말한다. 사용자가 특정번호(통신사로부터 부여 받은 SMS 수신 가능번호 예, #2001)로 문자 메시지를 보내면, 실시간으로 받은 문자메시지를 고객 시스템에서 수신할 수 있도록 해 주는 서비스이다.
- **MMS MO(Multimedia Messaging Service Mobile Originated)**
휴대폰으로부터 발송된 MMS 메시지를 수신하는 서비스를 말한다.

2.2. Service Architecture

EMMA는 크게 EMMA, Collector, Sender, Receiver, Distributor, Log Transfer 으로 구성되어 있다. 각 구성 요소에 대해 소개하면 아래와 같다.

- **EMMA Daemon**
인증서버에 접속하여 인증 및 하위 Thread 실행, 하위 Thread 감시 등의 역할을 한다.
- **Collector**
DBMS 메시지 전송 테이블을 검색하여 메시지 전송 큐에 적재하는 역할을 한다. SMS, MMS 두 개의 Collector 로 나뉘어져 있다.
- **Sender**
전송 큐에 있는 메시지를 송신하는 역할을 한다. 서버의 성능 및 이중화 등의 안정성을 위하여 N개로 설정할 수 있다.

- **Receiver**

G/W에서 메시지 전송 결과를 수신하여 리포트 큐에 적재하는 역할을 한다. Sender와 마찬가지로 성능 및 안정성을 위해 N개로 설정할 수 있다.

- **Distributor**

리포트 큐에 적재된 메시지 전송 결과를 DB에 갱신하는 역할을 한다.

- **Log Transfer**

송수신 완료된 메시지를 사용자의 설정에 따라 로그 테이블로 이동한다.

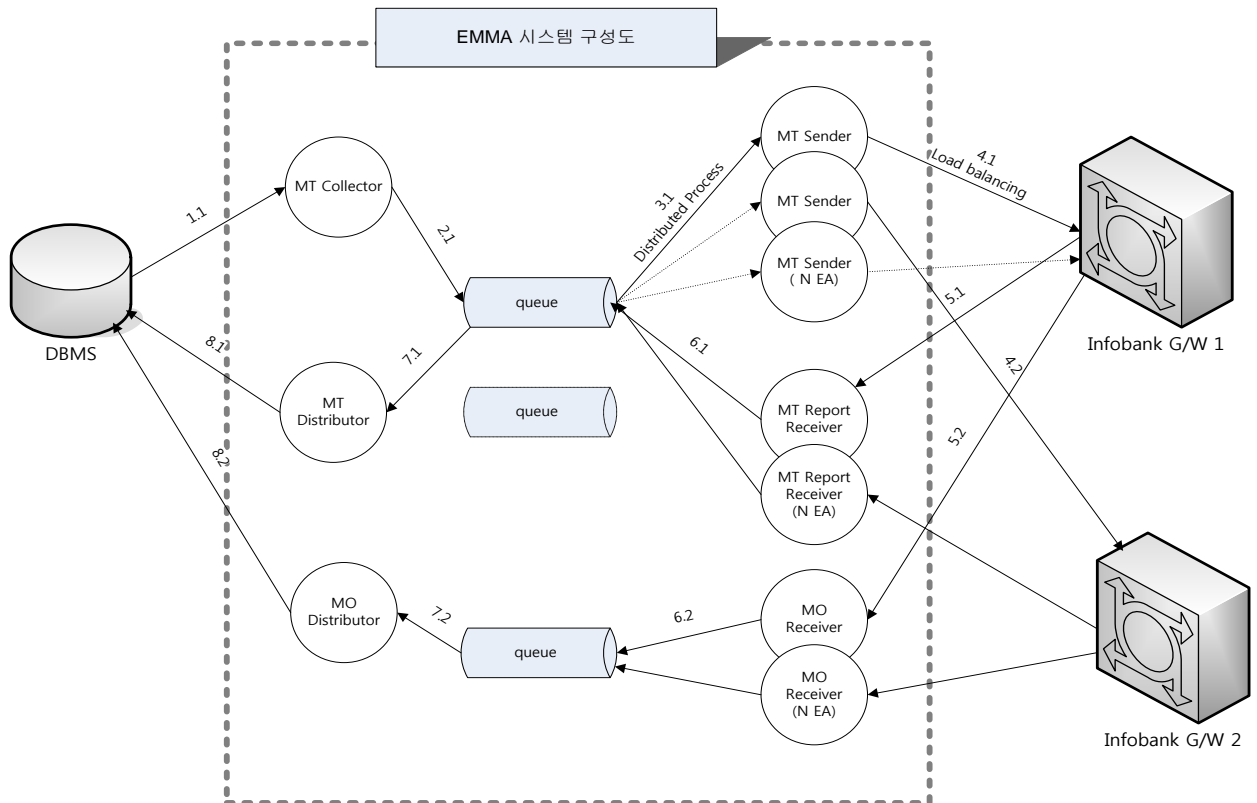


그림 2-1. EMMA 시스템 구성도

2.3. MT 메시지 처리 순서

- 1) MTCollector가 전송테이블을 주기적으로 감시하여 전송할 발송 메시지가 있는지 조회한다.
- 2) 전송할 메시지가 있으면 메시지를 큐에 적재하고, **테이블의 메시지 상태정보(msg_status) 필드를 "2"로 업데이트 한다.**
- 3) MTSender가 메시지 전송 큐에서 전송할 메시지를 꺼내온다.
- 4) MTSender가 꺼내온 메시지를 G/W로 전송한다.
- 5) MTSender가 메시지 전송 후 메시지 전송에 대한 수신 확인을 받는다.
- 6) MTRPTReceiver가 통신사업자가 G/W로 전송한 메시지 전송 결과를 G/W로부터 수신 받는다.
- 7) MTRPTReceiver가 수신한 전송 결과를 Report Queue에 적재한다.

- 8) MTDistributor가 Report Queue에서 메시지 전송 결과를 꺼내온다.
- 9) MTDistributor가 메시지 전송결과를 DB의 전송 테이블에 반영한다. **테이블의 메시지 상태정보 (msg_status) 필드가 "3"으로 업데이트 된다.**

2.4. MO 메시지 처리 순서

- 1) MOrceiver가 통신사업자가 G/W로 전송한 MO 데이터를 G/W로부터 수신 받는다.
- 2) 수신 받은 MO 데이터를 큐에 적재한다.
- 3) MODistributor가 MO Queue에서 MO 메시지를 꺼내온다.
- 4) MO 메시지를 DB의 MO 수신 테이블 또는 로그테이블에 입력한다.

3. DBMS 연동

이 절에서는 EMMA 가 기본적으로 배포하는 표준 테이블 및 Stored Procedure에 대해서 설명한다.

3.1. 사용 Table

아래 테이블 목록은 EMMA 배포 시 Stored Procedure 을 통해 생성되는 테이블 스키마 이다.

EMMA에서 배포한 테이블 이름이나 필드명은 고객이 원할 경우 임의로 바꿀 수 있다. **하지만 Table Schema가 변경되는 경우라도, 같이 배포된 Stored Procedure의 INPUT, OUTPUT 값은 꼭 동일하여야 한다.**

EMMA v3 테이블 목록

구분	이름	설명
공통	em_banlist	SMS MT / MMS MT 전송 시 차단할 수 있는 휴대폰 번호를 등록할 수 있는 테이블
	em_resultcode	엠마 결과 코드 정보 테이블
	em_status	EMMA의 실시간 상태정보가 업데이트 되는 테이블
	em_stat_day_m	일별 통계 테이블(Master)
	em_stat_day_d	일별 통계 상세 테이블(Detail)
SMS MT	em_smt_tran	SMS MT 메시지 전송을 위한 Master 테이블이다. 개별 메시지 전송인 경우, 수신번호 테이블(Detail)에 별도로 입력해 줘야 하는 불편을 해소하기 위해, 수신번호 필드도 추가로 가지고 있다.
	em_smt_client	SMS MT 메시지 전송을 위한 수신자번호 리스트가 있는 Detail 테이블이다. 사용의 편리를 위해 개별 메시지 전송의 경우 사용하지 않고, 동보 메시지 전송인 경우만 사용한다.
	em_smt_log_yyyy, em_smt_log_yyyymm	SMS MT 연별/월별 로그 테이블이다.
MMS MT	em_mmt_tran	MMS MT 메시지 전송을 위한 Master 테이블이다. 개별 메시지 전송인 경우, 수신번호 테이블(Detail)에 별도로 입력해 줘야 하는 불편을 해소하기 위해, 수신번호 필드도 추가로 가지고 있다.
	em_mmt_client	MMS MT 메시지 전송을 위한 수신자번호 리스트가 있는 Detail 테이블이다. 사용의 편리를 위해 개별 메시지 전송의 경우 사용하지 않고, 동보 메시지 전송인 경우만 사용한다.
	em_mmt_file_group	MMS MT 메시지 전송시 첨부파일 그룹 키 정보를 참조하는 테이블이다.

	em_mmt_file	em_mmt_file_group 테이블에서 실제로 참조하는 파일 정보 테이블이다.
	em_mmt_log_yyyy, em_mmt_log_yyyymm	MMS MT 연별/월별 로그 테이블이다.
SMS/MMS MO	em_mo_tran	SMS MO/ MMS MO 메시지 수신을 위한 테이블이다
	em_mo_file	MMS MO 수신 메시지의 첨부파일을 관리하는 테이블이다
	em_mo_log_yyyy, em_mo_log_yyyymm	SMS MO/ MMS MO 연별/월별 로그 테이블이다.

☞ 참고

EMMA가 사용하는 테이블의 상세 설명서는 “EMMA V3 테이블명세서~”.doc 문서를 참조한다.

3.2. Stored Procedure

EMMA V3 는 프로그램 내부에서 직접적으로 SQL을 사용하지 않으며, 서버에 저장된 Stored Procedure를 Call하여 DBMS와 인터페이스 한다. 이는 테이블 확장이나 변경이 용이하게 하기 위함이다. EMMA 초기 설치 후 Table Schema가 변경되는 경우, 같이 배포된 Stored Procedure도 꼭 변경해 줘야 한다.

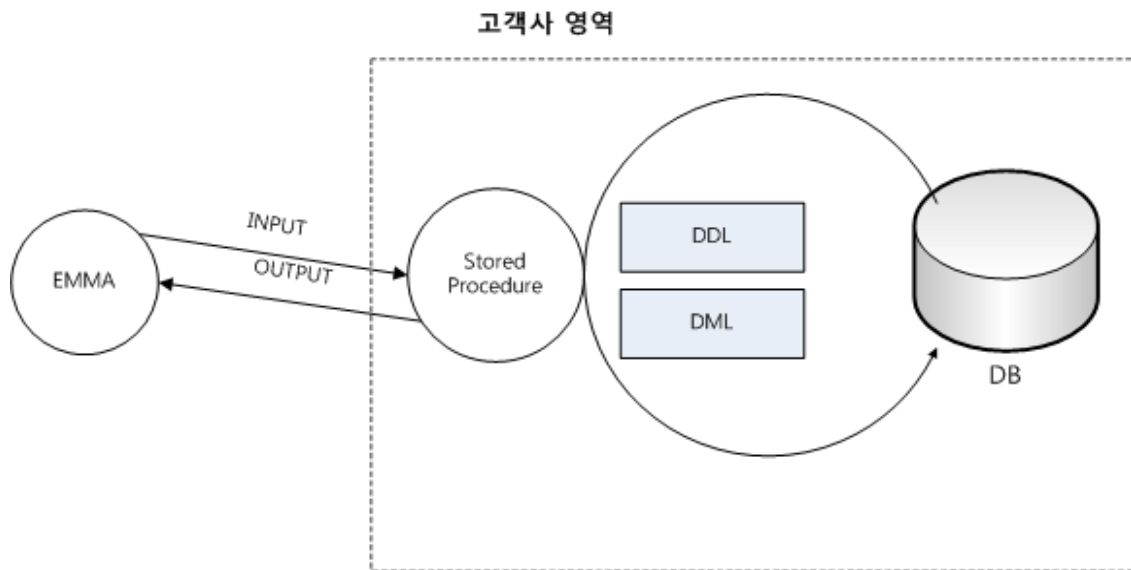


그림 3-1. DB 연동 구성도

공통 Stored Procedure 목록

필드	설명
sp_em_common_create	em_banlist, resultcode 테이블 등 공통으로 사용하는 테이블을 생성한다.
sp_em_common_banlist	전송차단번호 목록을 검색한다.

SMS MT Stored Procedure 목록

필드	설명
sp_em_smt_create	SMS MT 서비스 관련 테이블을 생성한다.
sp_em_smt_log_create	SMS MT 서비스에서 로그테이블을 사용할 경우 로그 테이블을 생성한다.
sp_em_smt_tran_select	SMS 개별 메시지 전송 데이터를 검색한다.
sp_em_smt_client_select	SMS MT 동보 메시지 전송 데이터를 검색한다. 동보 메시지를 사용할 경우 테이블이 Master/Detail 구조로 되어 있다.
sp_em_smt_update	SMS 메시지를 큐에 적재한 후 적재 결과를 갱신한다.
sp_em_smt_tran_rslt_update	개별 메시지 전송 결과를 갱신한다.
sp_em_smt_client_rslt_update	동보 메시지 전송 결과를 갱신한다.
sp_em_smt_tran_log_move	송수신 완료된 메시지를 로그 테이블로 이동한다.
sp_em_smt_client_log_move	송수신 완료된 동보 메시지를 로그 테이블로 이동한다.
sp_em_smt_tran_log_move_past	유효기간이 지난 메시지를 로그 테이블로 이동한다.
sp_em_smt_client_log_move_past	유효기간이 지난 동보 메시지를 로그 테이블로 이동한다.

MMS MT Stored Procedure 목록

필드	설명
sp_em_mmt_create	MMS MT 서비스 관련 테이블을 생성한다.
sp_em_mmt_log_create	MMS MT 서비스에서 로그테이블을 사용할 경우 로그 테이블을 생성한다.
sp_em_mmt_tran_select	MMS 개별 메시지 전송 데이터를 검색한다.
sp_em_mmt_client_select	MMS MT 동보 메시지 전송 데이터를 검색한다. 동보 메시지를 사용할 경우 테이블이 Master/Detail 구조로 되어 있다.
sp_em_mmt_file_select	MMT발송 시 첨부파일이 있는 경우 해당하는 첨부파일을 조회
sp_em_mmt_file_update	MMT발송 시 첨부파일에 대한 AttachFileKey를 update
sp_em_mmt_update	MMS 메시지를 큐에 적재한 후 적재 결과를 갱신한다.
sp_em_mmt_tran_rslt_update	개별 메시지 전송 결과를 갱신한다.
sp_em_mmt_client_rslt_update	동보 메시지 전송 결과를 갱신한다.
sp_em_mmt_tran_log_move	송수신 완료된 메시지를 로그 테이블로 이동한다.
sp_em_mmt_client_log_move	송수신 완료된 동보 메시지를 로그 테이블로 이동한다.
sp_em_mmt_tran_log_move_past	유효기간이 지난 메시지를 로그 테이블로 이동한다.
sp_em_mmt_client_log_move_past	유효기간이 지난 동보 메시지를 로그 테이블로 이동한다.

SMS MO Stored Procedure 목록

필드	설명
sp_em_smo_create	SMS MO 서비스 관련 테이블을 생성한다.

sp_em_smo_log_create	SMS MO 서비스에서 로그테이블을 사용할 경우 로그 테이블을 생성한다.
sp_em_smo_tran_insert	수신된 SMS MO 메시지를 SMO Tran 테이블에 입력한다.

MMS MO Stored Procedure 목록

필드	설명
sp_em_mmo_create	MMS MO 서비스 관련 테이블을 생성한다.
sp_em_mmo_log_create	MMS MO 서비스에서 로그테이블을 사용할 경우 로그 테이블을 생성한다.
sp_em_mmo_tran_insert	수신된 MMS MO 메시지를 SMO Tran 테이블에 입력한다.
sp_em_mmo_file_insert	MMS MO 메시지에 첨부된 파일을 테이블에 입력한다.

3.3. 테이블에 특정 필드 추가 방법

이 절에서는 고객이 EMMA 테이블에 특정 필드를 추가하는 방법에 대해서 설명한다. EMMA는 모듈 수정 없이 프로시저 수정만으로 필드 추가 및 수정이 가능하다. 필드 추가 방법은 각 서비스 별로 비슷하며, 여기서는 SMS(Oracle) 기준 **TRAN_ETC1 필드를 추가**하는 방법을 설명한다. 각 전체 내용은 일부를 발췌 한 것이며, 적색으로 표시된 부분이 변경된 사항이며, 작업 시 Comma(,)에 유의하기 바란다.

프로시저 명	변경 내용
sp_em_smt_create	<pre> create table em_smt_tran (mt_pr NUMBER(11) NOT NULL, msg_key VARCHAR2(20), 중략 reg_date TIMESTAMP default sysdate, mt_res_cnt NUMBER(5), TRAN_ETC1 VARCHAR2(160), CONSTRAINT pk_em_smt_tran PRIMARY KEY (mt_pr) using index tablespace TS_IDX_IMDS101) tablespace TS_DAT_IMDS101 '; create global temporary table em_smt_log_temp (mt_pr NUMBER(11) NOT NULL,</pre>

	msg_key VARCHAR2(20), 중략 reg_date TIMESTAMP default sysdate, mt_res_cnt NUMBER(5), TRAN_ETC1 VARCHAR2(160)) on commit delete rows ';
sp_em_smt_log_create	sp_em_smt_create 와 같은 방식으로 적용
sp_em_smt_tran_log_move	INSERT INTO em_smt_log_temp SELECT mt_pr, msg_key, 중략 sysdate, mt_res_cnt, tran_etc1 FROM em_smt_tran WHERE msg_status = '3' AND (broadcast_yn = 'N' OR broadcast_yn IS NULL) AND TO_CHAR (date_client_req, 'yyyymmdd') = v_date_client_req AND ROWNUM <= 2000; sql_string := ' INSERT INTO em_smt_log_' v_log_table ' SELECT mt_pr, msg_key, 중략 sysdate, mt_res_cnt, tran_etc1 FROM em_smt_log_temp ';
sp_em_smt_client_log_move	sp_em_smt_tran_log_move 와 같은 방식으로 적용
sp_em_smt_tran_log_move_past	sp_em_smt_tran_log_move 와 같은 방식으로 적용
sp_em_smt_client_log_move_past	sp_em_smt_tran_log_move 와 같은 방식으로 적용

3.4. 로그 테이블 고정 하기

이 절에서는 EMMA 로그 테이블을 고정 또는 연별 로그테이블로 적용하는 방법에 대해 설명한다. 필드 추가 방법은 각 서비스 별로 비슷하며, 여기서는 SMS(MySQL) 기준 **em_smt_log_table** 으로 **고정하여 사용**하는 방법을 설명한다. 적색으로 표시된 부분이 변경된 사항이며, 작업 시 Comma(,)에 유의하기 바란다.

프로시저 명	변경 내용
sp_em_smt_log_create	<pre> /* create em_smt_log_table table */ IF p_log_table <> " THEN SET @sql = CONCAT(" CREATE TABLE IF NOT EXISTS em_smt_log_table (mt_pr int(11) NOT NULL, mt_seq int(11) NOT NULL, 중략 </pre>
sp_em_smt_log_create	sp_em_smt_create 와 같은 방식으로 적용
sp_em_smt_tran_log_move	<pre> INSERT INTO em_smt_log_temp SELECT mt_pr, 0 AS mt_seq, 중략 SET @sql_insert = CONCAT(" INSERT INTO em_smt_log_table SELECT mt_pr, mt_seq, 중략 </pre>
sp_em_smt_client_log_move	sp_em_smt_tran_log_move 와 같은 방식으로 적용
sp_em_smt_tran_log_move_past	sp_em_smt_tran_log_move 와 같은 방식으로 적용
sp_em_smt_client_log_move_past	sp_em_smt_tran_log_move 와 같은 방식으로 적용

4. MT Service 사용

EMMA 사용자는 M& Messaging Service를 이용하기 위해서 간단히 DB Table에 데이터를 입력하거나 조회하면 된다. 이 절에서는 각 서비스별로 MT를 발송하거나, MO를 수신하는 방법에 대해 설명한다.

4.1. SMS MT 개별 메시지 전송

SMS MT를 보내기 위해서는 '전송예약시간', '수신자번호', '회신번호', '메시지 내용', '메시지 상태'를 다음 예시와 같이 입력한다. 이때 서비스 타입(service_type) 값은 '0' 이다.

입력 SQL

▶ Oracle의 경우

```
insert into em_smt_tran (mt_pr, date_client_req, content, callback, service_type, broadcast_yn, msg_status, recipient_num) values(sq_em_smt_tran_01.nextval, sysdate, 'Test Message 입니다', '01191540006', '0', 'N', '1', '01188880001');
```

▶ MSSQL의 경우

```
insert into em_smt_tran (date_client_req, content, callback, service_type, broadcast_yn, msg_status, recipient_num) values(getdate(), 'Test Message 입니다', '01191540006', '0', 'N', '1', '01188880001');
```

▶ MySQL/Maria의 경우

```
insert into em_smt_tran (date_client_req, content, callback, service_type, broadcast_yn, msg_status, recipient_num) values(sysdate(), 'Test Message 입니다', '01191540006', '0', 'N', '1', '01188880001');
```

☞ 중요

DB 전송 시 status 변화

msg_status : 1 (전송요구), 2 (큐에 적재, 결과대기 중), 3 (송수신완료)

date_client_req : 전송 요청시간, 이 값이 현재시간과 같거나 과거이며, msg_status='1' 인 것이 전송 대상이 된다. 큐에 적재 후 msg_status 는 바로 '2' 가 된다.

서비스 타입(service_type)

EMMA에서 메시지 송수신시 서비스 타입은 아래와 같다.

0-SMS MT, 1-CALLBACK URL, 2-MMS MT, 3-LMS, 4-SMS MO, 5-MMS MO

예를 들어 SMS MT전송을 위해서는 service_type 은 '0'이 되어야 한다.

☞ 참고

Sequence에 대하여

전송 테이블의 필드 중에 mt_pr(순번) 은 Integer 타입의 자동 증가 필드이다. 보통은 이런 자동 증가 값을 Sequence라 한다.

▶ Oracle

```
CREATE SEQUENCE sq_em_smt_tran_01 INCREMENT BY 1 START WITH 1 NOCYCLE NOCACHE
```

▶ MSSQL

```
mt_pr int identity(1,1) NOT NULL
```

▶ MySQL/Maria

```
mt_pr int(11) NOT NULL auto_increment,
```

☞ 참고

Content 필드 줄 바꿈(Line Break) 입력 방법

메시지내용(content) 필드에 줄 바꿈을 넣는 방법은 DBMS별로 아래와 같다.

▶ Oracle의 경우

```
'안녕하세요'||chr(13)||chr(10)||'좋은 하루 되십시오'
```

▶ MSSQL의 경우

```
'안녕하세요'+char(13) + char(10) + '좋은 하루 되십시오'
```

▶ MySQL/Maria의 경우

```
'안녕하세요\r\n좋은 하루 되십시오'
```

☞ 참고

DBMS 별로 SQL이 유사하므로, Oracle 입력 SQL만을 명시한다. DBMS별로 상이한 부분은 Sequence 및 함수 관련 SQL이다.

4.2. URL MT 개별 메시지 전송

URL MT를 보내기 위한 필드들은 기본적으로 SMS MT 필드와 같다. 그러나 URL전송임을 구분하기 위해서, 추가로 service_type 필드를 조정해 주어야 한다. 앞서서와 마찬가지로 mt_pr은 자동 증가 하는 Sequence이고, msg_status 는 '1' 로 한다. 그리고 date_client_req는 sysdate(getdate) 함수 값을 입력하거나, 현재 시간, 혹은 예약 전송 시 메시지를 전송하고자 하는 시간을 입력해야 한다. URL 전송을 위한 service_type 는 '1' 이다.

입력 SQL

▶ Oracle의 경우

```
insert into em_smt_tran (mt_pr, date_client_req, content, callback, service_type, broadcast_yn, msg_status, recipient_num) values(sq_em_smt_tran_01.nextval, sysdate, 'http://wap.test.co.kr 테스트', '01191540006', '1', 'N', '1', '01188880001');
```

📌 중요

URL전송을 위한 주의사항

URL전송을 위해서는 특정한 포맷을 맞추어서 DB에 입력하여야 한다. 즉 URL 전송의 경우도 data 자체는 content 필드에 입력된다. URL은 보통 Title과 URL 두 부분으로 이루어져 있기 때문에, 이를 지원하려면 "URL TITLE" 즉, URL이 먼저 나오고, 중간에 스페이스 하나로 분리한 후, 타이틀을 써주면 된다. URL 이후 최초로 나오는 스페이스가 구분자가 된다.

즉 <http://wap.test.co.kr>을 "테스트"라는 이름으로 전송하고 싶다면 "<http://wap.test.co.kr> 테스트"라고 입력하면 된다.

4.3. SMS MT 동보 메시지 전송

EMMA V3에서는 SMS MT 전송 시 동일 메시지를 여러 사람에게 보내는 동보 메시지 전송 기능을 제공한다. 동보 전송은 대량이 비슷한 내용의 메시지를 서로 다른 고객에게 전송할 때 적합하다. 동보 메시지는 한 패킷에 N개의 고객정보를 그룹으로 전송하는 형태로 기존의 일반 개별 메시지 전송에 비해 분당 20배 이상의 전송량을 처리할 수 있다.

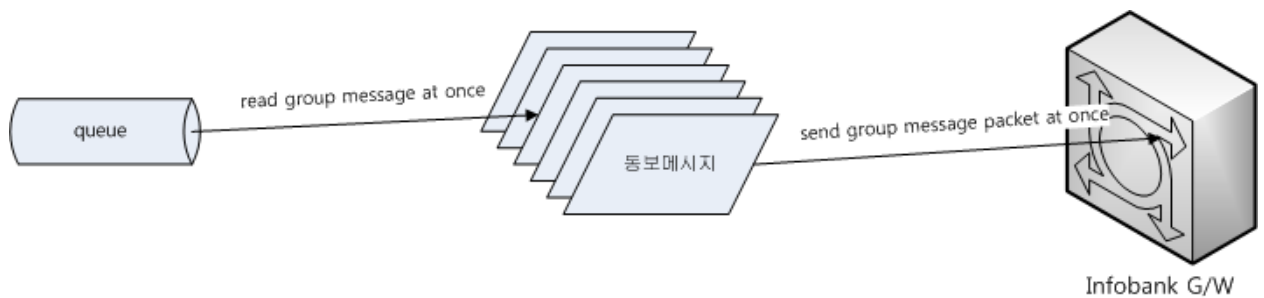


그림 4-1. 동보 메시지 전송

입력 SQL

▶ Oracle의 경우

1) Master 테이블 입력(em_smt_tran)

```
insert into em_smt_tran (mt_pr, date_client_req, content, callback, service_type, broadcast_yn, msg_status)
values(sq_em_smt_tran_01.nextval, sysdate, 'Test Message 입니다', '01191540006', '0', 'Y', '9');
```

2) 최근에 입력된 tran_pr 구하기

```
@max = select max(mt_pr) from em_smt_tran;
```

3) Detail 테이블 입력(em_smt_client)

```
insert into em_smt_client (mt_pr, mt_seq, msg_status, recipient_num) values(@max, 1, '1', '01199990001');
insert into em_smt_client (mt_pr, mt_seq, msg_status, recipient_num) values(@max, 2, '1', '01199990001');
```

4) 메시지 전송을 위한 상태(msg_status)값 '1'로 변경

```
update em_smt_tran set msg_status = '1' where mt_pr = @max;
```

☞ 중요

동보메시지 여부(broadcast_yn)

개별 메시지 전송 : N, 동보 메시지 전송 : Y

EMMA 내부적으로 broadcast_yn 이 'Y' 일 경우만 수신번호테이블(em_smt_client) 테이블을 조회한다.

☞ 중요

동보 전송 시 Master 테이블의 recipient_num 값의 의미

동보 전송 시 Master 테이블의 recipient_num, recipient_net, recipient_npsend, country_code 값들은 참조되지 않는다. 즉 마스터에 들어가는 값은 넣어도 전송되지 않는다는 것을 다시 한번 명심하기 바란다.

4.4. SMS MT 동보 단어 사용 방법

동보 단어는 동보 메시지 전송 시 각 수신번호 별로 본문 단어를 치환할 단어이다. 예를 들어 content 필드에 '%CHANGEWORD%님의 결제금액 %CHANGEWORD%원이 입금되었습니다' 입력하고, change_word# 필드에 '홍길동, 10,000, 홍길동'을 차례대로 넣는다면, 단말기에는 '홍길동님의 결제금액 10,000원이 입금되었습니다'로 표출된다.

동보 단어 표출방식은 %CHANGEWORD%, %CHANGEWORD#% 두 가지가 존재한다.

입력 SQL(%CHANGEWORD% 방식)

▶ Oracle의 경우

1) Master 테이블 입력(em_smt_tran)

```
insert into em_smt_tran (mt_pr, date_client_req, content, callback, service_type, broadcast_yn, msg_status)
values(sq_em_smt_tran_01.nextval, sysdate, '%CHANGEWORD%님의 결제금액 %CHANGEWORD%원이 입금되었습니다. %CHANGEWORD%님 감사합니다', '01191540006', '0', 'Y', '9');
```

2) 최근에 입력된 tran_pr 구하기

```
@max = select max(mt_pr) from em_smt_tran;
```

3) Detail 테이블 입력(em_smt_client)

```
insert into em_smt_client (mt_pr, mt_seq, msg_status, recipient_num, change_word1, change_word2, change_word3, change_word4, change_word5) values(@max, 1, '1', '01199990001', '홍길동', '10000', '홍길동',
```

```
NULL, NULL);
```

4) 메시지 전송을 위한 상태(msg_status)값 '1'로 변경

```
update em_smt_tran set msg_status = '1' where mt_pr = @max;
```

입력 SQL(%CHANGEWORD#% 방식)

▶ Oracle의 경우

1) Master 테이블 입력(em_smt_tran)

```
insert into em_smt_tran (mt_pr, date_client_req, content, callback, service_type, broadcast_yn, msg_status)
values(sq_em_smt_tran_01.nextval, sysdate, '%CHANGEWORD1%님의 결제금액 %CHANGEWORD2%원이 입금
되었습니다. %CHANGEWORD1%님 감사합니다', '01191540006', '0', 'Y', '9');
```

2) 최근에 입력된 tran_pr 구하기

```
@max = select max(mt_pr) from em_smt_tran;
```

3) Detail 테이블 입력(em_smt_client)

```
insert into em_smt_client (mt_pr, mt_seq, msg_status, recipient_num, change_word1, change_word2,
change_word3, change_word4, change_word5) values(@max, 1, '1', '01199990001', '홍길동', '10,000', NULL,
NULL, NULL);
```

📌 중요

%CHANGEWORD#% 형태를 사용 하는 경우(#은 숫자를 의미함)

- 메시지 내용에 %CHANGEWORD#%은 Detail TABLE의 change_word#값으로 치환한다.
- 메시지 내용에 %CHANGEWORD1%부터 순서대로 사용하여야 한다. 순서대로 사용하지 않는 경우, 정상적으로 치환되지 않을 수 있으니 주의 바란다.

%CHANGEWORD#%, %CHANGEWORD%를 혼용해서 사용할 수 없다.

동보 단어 내용에 사용할 수 없는 내용

- 특수문자 pipe("|"), "%CHANGEWORD%", "%CHANGEWORD1%", "%CHANGEWORD2%", "%CHANGEWORD3%", "%CHANGEWORD4%", "%CHANGEWORD5%"

4.5. MMS MT 개별 메시지 전송(첨부파일 미포함)

첨부파일이 없을 경우에는 em_mmt_tran테이블의 **attach_file_group_key** 필드를 '0'을 입력하면 첨부파일이 없다는 것을 의미한다. 보통 첨부파일이 없는 장문 메시지 전송을 **LMS 서비스**라고 한다.

입력 SQL

▶ Oracle의 경우

```
insert into em_mmt_tran (mt_pr, date_client_req, subject, content, attach_file_group_key, callback,
service_type, broadcast_yn, msg_status, recipient_num) values(sq_em_mmt_tran_01.nextval, sysdate, 'Test 제목
입니다', 'Test Message 입니다', '0', '01191540006', '3', 'N', '1', '01188880001');
```

▶ MSSQL의 경우

```
insert into em_mmt_tran (date_client_req, subject, content, attach_file_group_key, callback, service_type,
broadcast_yn, msg_status, recipient_num) values(getdate(), 'Test 제목입니다', 'Test Message 입니다', '0',
'01191540006', '3', 'N', '1', '01188880001');
```

▶ MySQL/Maria의 경우

```
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 제목입니다', 'Test Message 입니다', '0',
'01191540006', '3', 'N', '1', '01188880001');
```

☞ 중요

LMS(service_type=3)인 경우 반드시 첨부파일이 없어야 한다.

em_mmt_tran테이블의 attach_file_group_key필드값이 0이 아니면 오류 처리된다.(결과코드 'E920')

4.6. MMS MT 개별 메시지 전송 (첨부파일 포함)

일반적으로 첨부파일을 포함하여 전송하면 MMS 서비스라고 말한다. 첨부파일을 전송하기 위해서는 em_mmt_file테이블의 attach_file_group_key를 참조하여 첨부파일을 전송한다. EMMA는 성능향상을 위해 같은 파일은 재전송하지 않는다. 같은 파일 판단 기준은 attach_file_seq, attach_file_subpath, attach_file_name이 모두 같으면 서로 같은 파일로 인식한다. 만약 파일 이름은 같은데, 서로 다른 파일이라면 attach_file_seq 값을 증가시켜서 전송해 줘야 한다.

입력 SQL(파일 1개 전송)

▶ Oracle의 경우

1) 첨부파일 마스터 테이블 입력(em_mmt_file)

```
insert into em_mmt_file (attach_file_group_key, attach_file_group_seq, attach_file_seq, attach_file_subpath,
attach_file_name) values(1, 1, 1, NULL, 'test1.jpg');
```

2) 전송테이블 입력(em_mmt_tran)

```
insert into em_mmt_tran (mt_pr, date_client_req, subject, content, attach_file_group_key, callback,
service_type, broadcast_yn, msg_status, recipient_num) values(sq_em_mmt_tran_01.nextval, sysdate, 'Test 제목
입니다', 'Test Message 입니다', '1', '01191540006', '2', 'N', '1', '01188880001');
```

입력 SQL(파일 2개 전송)

▶ Oracle의 경우

1) 첨부파일 마스터 테이블 입력(em_mmt_file)

```
insert into em_mmt_file (attach_file_group_key, attach_file_group_seq, attach_file_seq, attach_file_subpath, attach_file_name) values(2, 1, 1, NULL, 'test1.jpg');
```

```
insert into em_mmt_file (attach_file_group_key, attach_file_group_seq, attach_file_seq, attach_file_subpath, attach_file_name) values(2, 2, 2, NULL, 'test2.jpg');
```

2) 전송테이블 입력(em_mmt_tran)

```
insert into em_mmt_tran (mt_pr, date_client_req, subject, content, attach_file_group_key, callback, service_type, broadcast_yn, msg_status, recipient_num) values(sq_em_mmt_tran_01.nextval, sysdate, 'Test 제목 입니다', 'Test Message 입니다', '2', '01191540006', '2', 'N', '1', '01188880001');
```

입력 SQL(test1.jpg가 기존에 보냈던 파일과 다를 경우)

▶ Oracle의 경우

1) 첨부파일 마스터 테이블 입력(em_mmt_file)

```
insert into em_mmt_file (attach_file_group_key, attach_file_group_seq, attach_file_seq, attach_file_subpath, attach_file_name) values(3, 1, 3, NULL, 'test1.jpg');
```

2) 전송테이블 입력(em_mmt_tran)

```
insert into em_mmt_tran (mt_pr, date_client_req, subject, content, attach_file_group_key, callback, service_type, broadcast_yn, msg_status, recipient_num) values(sq_em_mmt_tran_01.nextval, sysdate, 'Test 제목 입니다', 'Test Message 입니다', '3', '01191540006', '2', 'N', '1', '01188880001');
```

☞ 중요

attach_file_carrier 필드의 Bit masking 값

NULL : ALL,

1(0x01) : SKT,

2(0x02) : KT,

4(0x04) : LGU+,

8(0x08) : KTAAnn

예> SKT/KT 로만 전송하겠다 하면 해당 값을 0x01 + 0x02 => 3(0x03) 으로 지정해 주면 된다.

☞ 중요

MMS MT(service_type=2)인 경우 반드시 첨부파일이 있어야 한다.

em_mmt_tran 테이블의 attach_file_group_key 필드값이 존재하며, em_mmt_file 테이블에도 em_mmt_tran 테이블과 연관된 필드 값이 필요하다.

em_mmt_tran테이블의 attach_file_group_key필드값이 없을 경우 오류 처리 된다.(결과코드 'E921')

📌 중요

MMS 발송 시 첨부 파일 SIZE는 50K이하, 1개 파일에 대한 전송을 권장한다.

MMS 발송 시 첨부 파일 SIZE가 총 1MB 이상이거나 파일 1개당 300K 이상인 경우, 실패 처리되므로 주의 바란다. (결과코드 '3016: 첨부파일 사이즈 제한 실패')

4.7. MMS MT 동보 메시지 전송

MMS MT 서비스도 4.3절 SMS MT 동보 메시지 전송 과 같은 기능을 제공한다. 자세한 내용은 [4.3절](#) 을 다시 한번 참조하기 바란다.

입력 SQL

▶ Oracle의 경우

1) Master 테이블 입력(em_mmt_tran)

```
insert into em_mmt_tran (mt_pr, date_client_req, subject, content, attach_file_group_key, callback,
service_type, broadcast_yn, msg_status, recipient_num) values(sq_em_mmt_tran_01.nextval, sysdate, 'Test 제목
입니다', 'Test Message 입니다', '0', '01191540006', '2', 'Y', '9', '01188880001');
```

2) 최근에 입력된 tran_pr 구하기

```
@max = select max(mt_pr) from em_mmt_tran;
```

3) Detail 테이블 입력(em_mmt_client)

```
insert into em_mmt_client (mt_pr, mt_seq, msg_status, recipient_num, change_word1, change_word2,
change_word3, change_word4, change_word5) values(@max, '1', '1', '01199990001', NULL, NULL, NULL, NULL,
NULL);
```

```
insert into em_mmt_client (mt_pr, mt_seq, msg_status, recipient_num, change_word1, change_word2,
change_word3, change_word4, change_word5) values(@max, '2', '1', '01199990001', NULL, NULL, NULL, NULL,
NULL);
```

4) 메시지 전송을 위한 상태(msg_status)값 '1'로 변경

```
update em_mmt_tran set msg_status = '1' where mt_pr = @max;
```

4.8. MMS MT 동보 단어 사용 방법

MMS MT 서비스도 4.4절 SMS MT 동보 단어 사용방법과 같은 기능을 제공한다. 자세한 내용은

[4.4절](#) 을 다시 한번 참조하기 바란다.

4.9. 전송 메시지 암호화 하기

EMMA는 인증서버 인증하는 과정에서 EMMA가 생성하여 교환하는 대칭키로 메시지를 암호화하여 전송하는 기능을 제공한다. SMS, MMS 전송테이블의 암호화 여부(crypto_yn)을 'Y'로 설정(default : Y)하면, AES 또는 Seed 알고리즘으로 암호화하여 송수신 한다. 대칭키 암호화 알고리즘은 인포뱅크(주) 와 M& Messaging Service 계약 시 인증서버에 설정할 수 있다.

입력 SQL

▶ SMS(Oracle)

```
insert into em_smt_tran (mt_pr, date_client_req, content, callback, service_type, broadcast_yn, msg_status, recipient_num, crypto_yn) values(sq_em_smt_tran_01.nextval, sysdate, 'Test Message 입니다', '01191540006', '0', 'N', '1', '01188880001', 'Y');
```

▶ MMS(Oracle)

```
insert into em_mmt_tran (mt_pr, date_client_req, subject, content, attach_file_group_key, callback, service_type, broadcast_yn, msg_status, recipient_num, crypto_yn) values(sq_em_mmt_tran_01.nextval, sysdate, 'Test 제목입니다', 'Test Message 입니다', '0', '01191540006', '2', 'N', '1', '01188880001', 'Y');
```

4.10. 우선 순위 적용 하여 전송하기

EMMA V3는 메시지 전송 시 기본적으로 3종류(Slow/Fast/Very Fast)의 우선순위를 적용하여 발송할 수 있다. 우선순위는 메시지 전송테이블(em_smt_tran, em_mmt_tran) 테이블의 우선순위(priority) 필드를 'S', 'F', 'VF' 로 설정하여 전송하면 된다. VF는 가장 최우선으로 보내는 메시지이며, F, S의 순서대로 전송하게 된다. 우선순위를 적용하지 않고 일반적으로 보낼 때는 기본값인 'S' 으로 설정하여 보내면 된다.

입력 SQL

▶ Very Fast

```
insert into em_smt_tran (mt_pr, priority, date_client_req, content, callback, service_type, broadcast_yn, msg_status, recipient_num) values(sq_em_smt_tran_01.nextval, 'VF', sysdate, 'Test Message 입니다', '01191540006', '0', 'N', '1', '01188880001');
```

▶ Fast

```
insert into em_smt_tran (mt_pr, priority, date_client_req, content, callback, service_type, broadcast_yn, msg_status, recipient_num) values(sq_em_smt_tran_01.nextval, 'F', sysdate, 'Test Message 입니다', '01191540006', '0', 'N', '1', '01188880001');
```

▶ Slow (기본설정)

```
insert into em_smt_tran (mt_pr, priority, date_client_req, content, callback, service_type, broadcast_yn,
msg_status, recipient_num) values(sq_em_smt_tran_01.nextval, 'S', sysdate, 'Test Message 입니다',
'01191540006', '0', 'N', '1', '01188880001');
```

☞ 중요

emma.cf 파일에서의 priority 관련 설정

우선순위 사용여부 기본값은 false 이다. 우선순위를 적용하고자 한다면 emma.cf에서 아래 값을 변경 해 줘야 한다.

```
VF사용 : priority.use.sms.veryfast = 1
         priority.use.mms.veryfast = 1
F 사용 : priority.use.sms.fast = 1
         priority.use.mms.fast = 1
```

발송비율

VF, F, S 발송비율 설정 (2:1:1로 설정할 경우)

```
priority.cnt.sms.veryfast = 2
priority.cnt.sms.fast = 1
priority.cnt.sms.normal = 1
priority.cnt.mms.veryfast = 2
priority.cnt.mms.fast = 1
priority.cnt.mms.normal = 1
```

4.11. 전송 요청 통신사 지정하여 전송하기

EMMA V3는 메시지 발송 시 특정 통신사를 지정하여 전송할 수 있는 기능을 제공한다. 전송 테이블의 recipient_net 필드에 Bit masking 값을 입력하여 원하는 통신사를 지정한다. 전송 요청 통신사에 값이 들어 있는 경우 번호이동 수신 시 재 전송 할건지 여부(recipient_npsend)도 설정할 수 있다.

☞ 중요

recipient_net 필드의 Bit masking 값

```
NULL : ALL,
1(0x01) : SKT,
2(0x02) : KT,
4(0x04) : LGU+,
8(0x08) : KTAAnn
```

예> SKT/KT 로만 전송하겠다 하면 해당 값을 0x01 + 0x02 => 3(0x03) 으로 지정해 주면 된다.

recipient_npsend 필드 값 설정

Y-번호이동처리를 위한 재전송요청,

N-번호이동처리를 위한 재전송요청 안 함.

입력 SQL

▶ 전송 이통사 지정하기(Oracle)

//SKT/KT로만 전송하고, 결과가 번호이동 시 재전송

```
insert into em_smt_tran (mt_pr, date_client_req, content, callback, service_type, broadcast_yn, msg_status,
recipient_num, recipient_net, recipient_npsend) values(sq_em_smt_tran_01.nextval, sysdate, 'Test Message 입
니다', '01191540006', '0', 'N', '1', '01188880001', '3', 'Y');
```

4.12. 부서코드(참조)필드 지정하기

EMMA V3는 메시지 발송 시 부서별 메시지 구분을 위한 참조 필드를 지정할 수 있다. 만약 참조 키 필드로 정산이 필요할 경우 **mt_refkey** 필드를 지정하여 발송 한다.

입력 SQL

▶ 부서별 정산을 위한 발송(Oracle)

```
insert into em_smt_tran (mt_pr, mt_refkey, date_client_req, content, callback, service_type, broadcast_yn,
msg_status, recipient_num) values(sq_em_smt_tran_01.nextval, 'deptA', sysdate, 'Test Message 입니다',
'01191540006', '0', 'N', '1', '01188880001');
```

4.13. 국제 문자 지정하여 전송하기

EMMA V3는 국제 문자 메시지를 발송 할 수 있다. 메시지 발송 시 메시지 종류(msg_type)와 국가 코드(country_code) 필드를 지정하여 발송 한다.

1) 개별 메시지 전송

입력 SQL

▶ 국제 문자 발송(Oracle)

```
insert into em_smt_tran (mt_pr, date_client_req, content, callback, service_type, broadcast_yn, msg_status,
msg_type, country_code, recipient_num) values (sq_em_smt_tran_01.nextval, sysdate, 'This is test 국제
```

```
message.', '15882189', '0', 'N', '1', '1004', '86', '13260312345');
```

2) 동보 메시지 전송

입력 SQL

▶ 국제 문자 발송(Oracle)

1) Master 테이블 입력(em_smt_tran)

```
insert into em_smt_tran (mt_pr, date_client_req, content, callback, service_type, broadcast_yn, msg_status,
msg_type) values (sq_em_smt_tran_01.nextval, sysdate, 'This is test 국제message.', '15882189', '0', 'Y', '1',
'1004');
```

2) 최근에 입력된 tran_pr 구하기

```
@max = select max(mt_pr) from em_smt_tran;
```

3) Detail 테이블 입력(em_smt_client)

```
insert into em_smt_client (mt_pr, mt_seq, msg_status, country_code, recipient_num) values (@max, 1, '1',
'86', '13260312345');
```

```
insert into em_smt_client (mt_pr, mt_seq, msg_status, country_code, recipient_num) values (@max, 2, '1',
'86', '13260312345');
```

4) 메시지 전송을 위한 상태(msg_status)값 '1'로 변경

```
update em_smt_tran set msg_status = '1' where mt_pr = @max;
```

☞ 중요

charset 설정

/conf/emma.cf 설정 파일에서 gw.charset = UTF-8 로 설정해 주면 된다.

주요 필드 설명(필수 입력 사항)

필드명	필드값	예시
country_code	국가코드	13260312345
recipient_num	수신자 전화번호(국가코드 제외)	86
msg_type	1004(국제문자)	1004

5. MO Service

5.1. SMS MO 수신 확인

할당된 특번으로 SMS MO 전송 후, SMS MO 테이블을 확인한다.

☞ 테이블의 상세 설명은 “EMMA V.3 DB명세서~.doc”파일을 참조한다.

5.2. MMS MO 수신 확인

할당된 특번으로 MMS MO 전송 후, MMS MO 테이블을 확인한다.

☞ 테이블의 상세 설명은 “EMMA V.3 DB명세서~.doc”파일을 참조한다.

6. 전송차단 및 SPAM

EMMA V.3 는 인증서버파일, 고객 파일, 고객 SPAM Table을 통하여 메시지 발송 시 전송 차단 및 SPAM 체크 기능을 제공한다. 이 절에서는 3가지 SPAM 체크 기능에 대하여 설명한다. 전송차단 및 SPAM 기능을 사용하지 않는다면 이 절을 건너뛰어도 상관 없다.

6.1. 인증서버 전송차단 파일

EMMA 는 최초 실행될 때, 인증서버로부터 업데이트 된 전송차단 파일을 자동으로 수신 받는다. 그리고 메시지 전송 시 이 전송차단 파일을 체크하여 Block된 Number에 대해 전송 차단 기능을 제공한다. 인증서버 전송차단파일은 SMS, URL, MMS 수신번호 차단 정보를 가지고 있다. 인증서버 전송차단은 자동으로 처리되므로 고객이 별도로 설정할 사항은 없다.

6.2. DB Table을 통한 SPAM 체크

차단번호 리스트테이블에 등록된 데이터에 대해 서비스 별 수신번호, 회신번호, 메시지 내용에 대해 전송 차단 기능을 제공한다.

em_banlist 차단 예시

▶ 수신번호 차단

```
insert into em_banlist (service_type, ban_seq, ban_type, content, send_yn, allow_yn, ban_status_yn, reg_date, update_date) values('0','1','R','01188880001','N','Y','Y',sysdate(),sysdate());
```

▶ callback number 정규식 체크

```
insert into em_banlist (service_type, ban_seq, ban_type, content, send_yn, allow_yn, ban_status_yn, reg_date, update_date) values('0','2','C','^02[- ]?700[- ]?[0-9]{4,4}($[^0-9])','N','Y','Y',sysdate(),sysdate());
```

▶ message 정규식 체크

```
insert into em_banlist (service_type, ban_seq, ban_type, content, send_yn, allow_yn, ban_status_yn, reg_date, update_date) values('0','3','T','bibirace.com','N','Y','Y',sysdate(),sysdate());
```

중요

수신번호 차단의 경우 전송메시지 검색 Stored Procedure에서 Outer Join 으로 정보를 가져 오므로, EMMA 을 재 기동 안 해도 자동으로 반영된다. callback이나 content체크에 대해서는 EMMA 을 재 기동 해야 한다.

7. 환경설정

7.1. emma.cf

emma.cf 파일은 EMMA 서비스 및 기본 설정 정보를 포함한다.

7.1.1. 인증 정보

인포뱅크(주)의 M& Messaging Service를 이용하기 위해서는 EMMA는 실행될 때 인증서버로부터 인증을 받아야 정상적으로 서비스를 제공 받을 수 있다. 인증서버 정보는 인포뱅크(주)에서 계약 완료 후 전달 받을 수 있다. 만약 EMMA를 설치하는 서버에 방화벽이 있을 경우, 아래 접속정보에 대해 방화벽 오픈 작업을 해줘야 한다.(outbound)

Properties

환경변수	의미
auth.host	인포뱅크(주) 인증서버 접속 IP 또는 URL
auth.port	인증서버의 접속 Port
auth.id	인증서버 접속 클라이언트 ID
auth.password	인증키 생성을 위한 초기 패스워드
auth.encrypted	인증정보(auth.id, auth.password) 암호화 여부 (0:암호화 안되어 있는 경우, 1:암호화 되어 있는 경우)

설정 용례

```
auth.host      = auth.infobank.net
auth.port      = 12001
auth.id        = infobank
auth.password  = DMJ7XB92TSTYI295TU35
auth.encrypted = 0
```

7.1.2. 프로세스(Thread) 사용여부 설정

EMMA V.3는 SMS MT / MMS MT / SMS MO / MMS MO 4가지 서비스가 가능한 모듈이다. 인포뱅크(주)에서 발급된 고객 ID에 따라 서비스의 사용여부가 각각 다르게 되며, 사용 여부에 따라 아래 설정을 변경해 줘야 한다. 사용하지 않는 서비스를 사용으로 설정하면, 동작 오류로 인해 실제 사용하는 서비스에도 영향을 미치게 된다.

따라서 사용하는 ID의 계약 현황에 따라 아래 사용여부를 설정해 줘야 한다. 사용하는 서비스에는 '1'로 표기한다.

Properties

환경변수	의미
process.use.mtsender	메시지를 G/W로 전송하는 프로세스의 사용여부를 설정한다. SMS MT / MMS MT 서비스를 사용할 경우 값을 꼭 '1' 로 설정해야 한다.
process.use.mtreceiver	메시지 송신 결과를 G/W 에서 수신하는 프로세스의 사용여부를 설정한다. SMS MT / MMS MT 서비스를 사용할 경우 값을 꼭 '1' 로 설정해야 한다.
process.use.smoreceiver	SMS MO 서비스를 수행하는 프로세스의 사용여부를 설정한다. SMS MO 서비스를 사용할 경우 값을 꼭 '1' 로 설정해야 한다
process.use.mmoreceiver	MMS MO 서비스를 수행하는 프로세스의 사용여부를 설정한다. MMS MO 서비스를 사용할 경우 값을 꼭 '1' 로 설정해야 한다
process.use.smtcollector	메시지를 큐에 저장하는 프로세스의 사용여부를 설정한다. SMS MT 서비스를 사용하는 경우 꼭 '1' 로 설정해야 한다
process.use.mmtcollector	메시지를 큐에 저장하는 프로세스의 사용여부를 설정한다. MMS MT 서비스를 사용하는 경우 꼭 '1' 로 설정해야 한다
process.use.mtdistributor	G/W에서 수신한 송신 결과를 DB에 갱신하는 프로세스의 사용여부를 설정한다. SMS MT / MMS MT 서비스를 사용하는 경우 꼭 '1' 로 설정해야 한다
process.use.modistributor	수신한 MO 메시지를 DB에 갱신하는 프로세스의 사용여부를 설정한다. SMS MO / MMS MO 서비스를 사용하는 경우 꼭 '1' 로 설정해야 한다
process.use.logtransfer	송수신 완료된 메시지를 로그 테이블로 이동하는 프로세스의 사용여부를 설정한다. SMS MT MMS MT 서비스에서 로그테이블의 사용할 경우 '1' 로 설정해야 한다
process.use.jobscheduler	EMMA에서 주기적으로 실행되는 스케줄러이다. 기본 설정 시 작업은 하루에 한번(AM 04:00) 수행되며, 이때 수행하는 작업은 전송 요청 일자가 10일 이전인 데이터를 로그 테이블로 이동시킨다. 또한 설정에 따라 로그 파일을 삭제할 수도 있다
process.use.statscheduler	EMMA에서 주기적으로 로그테이블에서 일별로 통계를 생성하는 Daily Statistics Scheduler를 수행하는 프로세스의 사용여부를 설정한다.
process.use.sessionmonitor	EMMA 상태정보를 모니터링하는 프로세스의 사용여부를 설정한다.

설정 용례(SMS MT 사용시)

process.use.mtsender	= 1
process.use.mtreceiver	= 1
process.use.smoreceiver	= 0
process.use.mmoreceiver	= 0
process.use.smtcollector	= 1
process.use.mmtcollector	= 0
process.use.mtdistributor	= 1
process.use.modistributor	= 0
process.use.logtransfer	= 1

설정 용례(MMS MT 사용시)

process.use.mtsender	= 1
process.use.mtreceiver	= 1
process.use.smoreceiver	= 0
process.use.mmoreceiver	= 0
process.use.smtcollector	= 0
process.use.mmtcollector	= 1
process.use.mtdistributor	= 1
process.use.modistributor	= 0
process.use.logtransfer	= 1

설정 용례(SMS MO 사용시)

process.use.mtsender	= 0
process.use.mtreceiver	= 0
process.use.smoreceiver	= 1
process.use.mmoreceiver	= 0
process.use.smtcollector	= 0
process.use.mmtcollector	= 0
process.use.mtdistributor	= 0
process.use.modistributor	= 1
process.use.logtransfer	= 0

설정 용례(MMS MO 사용시)

process.use.mtsender	= 0
process.use.mtreceiver	= 0
process.use.smoreceiver	= 0
process.use.mmoreceiver	= 1

```
process.use.smtcollector    = 0
process.use.mmtcollector    = 0
process.use.mtdistributor   = 0
process.use.modistributor   = 1
process.use.logtransfer     = 0
```

7.1.3. 프로세스 세션 개수 설정

EMMA V.3는 시스템 장애 대비 및 성능개선을 위해 G/W 연결 세션을 2개 이상 유지할 수 있다. 그러므로 한쪽 G/W 장애 시에도 정상적으로 메시지 송수신이 가능하며, 고객사의 서버 성능, 네트워크 속도 등에 따라 송수신 프로세스를 N개로 분산하여 전송할 수 있다.

Properties

환경변수	의미
session.cnt.mtsender	MT 서비스에서 메시지 전송 프로세스의 세션 개수
session.cnt.mtreceiver	MT 서비스에서 메시지 전송 결과 수신 프로세스의 세션 개수
session.cnt.smoreceiver	SMS MO 서비스 수신 프로세스 세션 개수
session.cnt.mmoreceiver	MMS MO 서비스 수신 프로세스 세션 개수

설정 용례(각 서비스 별로 2개의 세션 사용 시)

```
session.cnt.mtsender      = 2
session.cnt.mtreceiver    = 2
session.cnt.smoreceiver   = 2
session.cnt.mmoreceiver   = 2
```

7.1.4. 프로세스 동작 유효 시간 설정

EMMA V.3는 설정된 프로세스 별로 동작할 유효 시간대를 설정할 수 있다. 이 설정의 목적은 원하는 시간대의 발송 금지를 목적으로 시작하였으며, 또한 프로세스 별로 동작 시간대를 분산 처리 함으로써 안정성 및 성능을 더욱 높일 수 있다. 아래 설정 용례의 경우 전송 메시지를 큐에 적재하는 프로세스는 08:00부터 시작하며, 큐에 있는 메시지를 전송하는 프로세스는 09:00부터 시작하며, 송수신이 완료된 메시지를 로그 테이블로 이동하는 프로세스는 새벽 00:00 에 시작함으로써 분산하여 처리하여 대용량 송수신시 적합하다.

Properties

환경변수	의미
runtime.collector.overtimefail	발송제한시간인 경우 메시지 재발송 기능 (Collector만 해당, 0:재발송함, 1:재발송안함)
runtime.collector.start	Collector(DBMS 메시지 전송 테이블을 검색하여 메시지 전송

	큐 적재) 동작 시작 시간
runtime.collector.end	Collector 동작 종료 시간
runtime.sender.start	Sender(전송 큐에 있는 메시지 송신) 동작 시작 시간
runtime.sender.end	Sender 동작 종료 시간
runtime.logtransfer.start	Log Transfer(송수신 완료된 메시지 로그 테이블 이동) 동작 시작 시간
runtime.logtransfer.end	Log Transfer 동작 종료 시간
runtime.jobscheduler.start	Job Scheduler 및 Statistics Scheduler 동작 시간
runtime.userscheduler.start	User Scheduler 동작 시간

설정 용례

```
# 발송제한시간인 경우 메시지 재발송 기능(Collector만 해당, 0:재발송함, 1:재발송안함)
runtime.collector.overtimefail = 0

# Collector 동작 시작 시간
runtime.collector.start = 00:00

# Collector 동작 종료 시간
runtime.collector.end = 24:00

# Sender 동작 시작 시간
runtime.sender.start = 00:00

# Sender 동작 종료 시간
runtime.sender.end = 24:00

# Log Transfer 동작 시작 시간
runtime.logtransfer.start= 00:00

# Log Transfer 동작 종료 시간
runtime.logtransfer.end = 24:00

# Job Scheduler 동작 시간
runtime.jobscheduler.start = 04:00

# User Scheduler 동작 시간
runtime.userscheduler.start = 05:00
```

7.1.5. 메시지 발송 우선순위 설정

EMMA V.3는 메시지 서비스 타입(SMS MT, MMS MT)별 발송 우선순위 기능을 강화하여, 다양한 우선순위에 따라 메시지가 전송 가능하도록 한다. 또한 발송비율에 따라 최우선(VF) 메시지, 우선(F) 메시지, 일반(S) 메시지 비율을 설정할 수 있다. 아래의 설정 용례의 값에 따르면 VF : F : S 비율은 2: 1: 1의 비율로 메시지 큐에 저장된다.

Properties

환경변수	의미
priority.use.sms.veryfast	SMS MT 서비스 최우선 메시지 사용 유무 (0:사용안함, 1:사용함)
priority.use.mms.veryfast	MMS MT 서비스 최우선 메시지 사용 유무 (0:사용안함, 1:사용함)
priority.use.sms.fast	SMS MT 서비스 일반 메시지 보다 우선 순위가 높은 메시지 사용 유무(0:사용안함, 1:사용함)
priority.use.mms.fast	MMS MT 서비스 일반 메시지 보다 우선 순위가 높은 메시지 사용 유무(0:사용안함, 1:사용함)
priority.cnt.sms.veryfast	SMS MT 서비스 최우선 메시지 발송 비율
priority.cnt.sms.fast	SMS MT 서비스 일반 메시지 보다 우선 순위가 높은 메시지 발송 비율
priority.cnt.sms.normal	SMS MT 서비스 일반 메시지 발송 비율
priority.cnt.mms.veryfast	MMS MT 서비스 최우선 메시지 발송 비율
priority.cnt.mms.fast	MMS MT 서비스 일반 메시지 보다 우선 순위가 높은 메시지 발송 비율
priority.cnt.mms.normal	MMS MT 서비스 일반 메시지 발송 비율

설정 용례

```

priority.use.sms.veryfast = 1
priority.use.mms.veryfast = 1
priority.use.sms.fast     = 1
priority.use.mms.fast     = 1

priority.cnt.sms.veryfast = 2
priority.cnt.sms.fast     = 1
priority.cnt.sms.normal   = 1
priority.cnt.mms.veryfast = 2
priority.cnt.mms.fast     = 1
priority.cnt.mms.normal   = 1

```

7.1.6. 큐 정보 설정

EMMA V.3 는 서비스 사용시 내부적으로 큐 파일을 사용한다. 큐 파일 사용유무를 설정하여 서비스를 사용하거나 혹은 일시적으로 사용 안 할 수 있다.

Properties

환경변수	의미
que.use.smsmt	MTSender 에서 SMS MT Queue 사용 유무, SMS MT 서비스를 사용할 경우 값을 꼭 '1' 로 설정해야 한다.
que.use.mmsmt	MTSender 에서 MMS MT Queue 사용 유무, MMS MT 서비스를 사용할 경우 값을 꼭 '1' 로 설정해야 한다.
que.use.mtrpt	MTDistributor 프로세스에서 큐 사용유무, MT 서비스를 사용할 경우 값을 꼭 '1' 로 설정해야 한다.
que.use.smsmo	MODistributor 프로세스에서 SMS MO 큐 사용유무, SMS MO 서비스를 사용할 경우 값을 꼭 '1' 로 설정해야 한다. 반대로 서비스를 사용하지 않는데, 값을 '1'로 설정하고, MMS MO 서비스를 운영하는 경우에는 성능에 나쁜 영향을 미칠 수 있다.
que.use.mmsmo	MODistributor 프로세스에서 MMS MO 큐 사용유무, MMS MO 서비스를 사용할 경우 값을 꼭 '1' 로 설정해야 한다.
que.cnt.smsmo	MODistributor에서 SMS MO 수신 비율
que.cnt.mmsmo	MODistributor에서 MMS MO 수신 비율
que.pollinginterval	Queue에 데이터가 없는 경우, 다음 데이터를 조회하기 전에 쉬는 sleep time으로 단위는 초(second) 이다.
que.name.smsmt.s	SMS MT 서비스 일반 메시지인 경우, Queue Name 을 설정해야 한다. (설정 안할 경우, 'send_smsmt' 으로 생성된다.)
que.name.smsmt.f	SMS MT 서비스 일반 메시지 보다 우선 순위가 높은 메시지인 경우, QueueName 을 설정해야 한다. (설정 안할 경우, 'send_smsmt_fast' 으로 생성된다.)
que.name.smsmt.v	SMS MT 서비스 최우선 메시지인 경우, QueueName 을 설정해야 한다. (설정 안할 경우, 'send_smsmt_real' 으로 생성된다.)
que.name.mmsmt.s	MMS MT 서비스 일반 메시지인 경우, QueueName 을 설정해야 한다. (설정 안할 경우, 'send_smsmt' 으로 생성된다.)
que.name.mmsmt.f	MMS MT 서비스 일반 메시지 보다 우선 순위가 높은 메시지인 경우, QueueName 을 설정해야 한다. (설정 안할 경우, 'send_smsmt_fast' 으로 생성된다.)
que.name.mmsmt.v	MMS MT 서비스 최우선 메시지인 경우, QueueName 을 설정해야 한다. (설정 안할 경우, 'send_smsmt_real' 으로 생성된다.)
que.name.mtrpt	MT 결과에 대한 메시지인 경우, QueueName 을 설정해야 한다. (설정 안할 경우, 'recv_rslt' 으로 생성된다.)
que.name.smsmo	SMS MO 수신에 대한 메시지인 경우, QueueName 을 설정해야 한다. (설정 안할 경우, 'recv_smsmo' 으로 생성된다.)
que.name.mmsmo	MMS MO 수신에 대한 메시지인 경우, QueueName 을 설정해야 한다.

	한다. (설정 안할 경우, 'recv_mmsmo' 으로 생성된다.)
que.name.sess	EMMA 상태정보를 모니터링하는 프로세스에서 사용되며, QueueName 을 설정해야 한다. (설정 안할 경우, 'sess_info.que' 로 생성된다.)

설정 용례

```

que.use.smsmt = 1
que.use.mmsmt = 1

que.use.mtrpt = 1
que.use.smsmo = 1
que.use.mmsmo = 1

que.cnt.smsmo = 1
que.cnt.mmsmo = 1

que.pollinginterval = 2

que.name.smsmt.s = send_smsmt
que.name.smsmt.f = send_smsmt_fast
que.name.smsmt.v = send_smsmt_real

que.name.mmsmt.s = send_mmsmt
que.name.mmsmt.f = send_mmsmt_fast
que.name.mmsmt.v = send_mmsmt_real

que.name.mtrpt = recv_rslt

que.name.smsmo = recv_smsmo
que.name.mmsmo = recv_mmsmo

que.name.sess = sess_info.que

```

7.1.7. 파일 설정

EMMA V.3 에서 사용하는 파일과 관련된 정보를 설정한다. 첨부파일 및 수신거부 차단파일 등이 이에 해당한다.

Properties

환경변수	의미
file.type.mmsmt	MMS MT 파일 위치 타입(local, http)
file.path.mmsmt	MMS MT 서비스 사용시 실제 첨부파일이 존재하는 디렉토리 경로
file.path.mmsmo	MMS MO 서비스 사용시 실제 첨부파일이 존재하는 디렉토리 경로
file.path.auth.ban	인증서버로부터 받은 수신거부차단파일을 저장할 디렉토리 경로

설정 용례

```
file.type.mmsmt = local
file.path.mmsmt = ./attach_file/mmsmt/
file.path.mmsmo = ./attach_file/mmsmo/
file.path.auth.ban = ./ban/auth/
```

7.1.8. G/W 통신 설정

인증서버, G/W 서버에 통신 장애 시 최대 연결 시도 회수 및 상태정보 송수신 주기를 설정한다.

Properties

환경변수	의미
retry.auth.maxcount	infobank 인증서버 최대 연속 재 연결 횟수
retry.immg.maxcount	infobank g/w 최대 연속 재 연결 횟수
retry.interval	재시도하기 위해 sleep 타임
ping.interval	연결 상태를 확인하기 위한 ping 전송 주기
logtransfer.interval	전송완료된 메시지 로그 테이블 이동 주기
sessionmonitor.interval	EMMA 모니터링 관리자 갱신 주기
authban.interval	인증서버로 부터 차단파일을 수신하여 갱신하는 주기(단위:분)
jobscheduler.interval	Job Scheduler 동작 주기(단위:분)
statscheduler.interval	통계 생성을 위한 동작 주기(단위:분) 엠마가 기동되는 시점에 최초 실행하여 데이터를 수집하며, 엠마 기동 이후 interval 주기에 따라 데이터 수집을 반복
userscheduler.interval	User Scheduler 동작 주기(단위 : 분)
gw.charset	G/W 가 이통사로 전송 시 보낼 메시지의 문자셋 (EUC-KR, UTF-8)

설정 용례

```
retry.auth.maxcount = 28800
retry.immg.maxcount = 28800
retry.interval = 10
ping.interval = 60
```

```
logtransfer.interval = 60
sessionmonitor.interval = 6
authban.interval = 1440
jobscheduler.interval = 1440
statscheduler.interval = 30
userscheduler.interval = 1440
gw.charset = EUC-KR
```

7.1.9. 기타 정보 설정

EMMA 서비스 관련 기타 정보를 설정한다. 중복발송방지를 설정할 경우 처리 속도는 전송 속도는 다소 늦어질 수 있다.

Properties

환경변수	의미
dupcheck.smsmt	SMSMT 중복발송 차단 체크 여부(0:사용안함, 1:사용함)
dupcheck.smsmt.cacheinterval	SMSMT 중복발송 차단 체크 시 최근전송이력 cache time (단위:초)
dupcheck.mmsmt	MMSMT 중복발송 차단 체크 여부(0:사용안함, 1:사용함)
dupcheck.mmsmt.cacheinterval	MMSMT 중복발송 차단 체크 시 최근전송이력 cache time (단위:초)
mmsmo.tagstrip	MMSMO Html 및 XML 태그 삭제 여부(0:삭제안함, 1:삭제함)
mtsender.exceed.fail	MT 사용건수 초과시 재전송 사용 여부 (0:재전송함, 1:재전송안함)
log.file.path	로그파일 삭제 시 로그 파일이 저장되는 디렉토리
log.file.delete	이전에 생성된 로그 파일 삭제 여부(0:삭제안함, 1:삭제함)
log.file.duration	보관할 로그 파일 기간 (단위:일)
log.use.personalinfo.hide	로그 파일에 개인정보 노출차단 사용유무 (0:개인정보 숨김 사용안함, 1:개인정보 숨김 사용)

설정 용례

```
# SMSMT 중복발송 차단 체크 여부(0:사용안함, 1:사용함)
dupcheck.smsmt = 1
# SMSMT 중복발송 차단 체크 시 최근전송이력 cache time
dupcheck.smsmt.cacheinterval = 60
# MMSMT 중복발송 차단 체크 여부(0:사용안함, 1:사용함)
dupcheck.mmsmt = 0
# MMSMT 중복발송 차단 체크 시 최근전송이력 cache time(단위:초)
```



```

dupcheck.mmsmt.cacheinterval = 120
# MMSMO Html 및 XML 태그 삭제 여부(0:삭제안함, 1:삭제함)
mmsmo.tagstrip = 1
# 사용건수초과시 실패 처리 여부(0:재전송함, 1:재전송안함)
mtsender.exceed.fail = 1
# 로그 파일이 저장되는 디렉토리
log.file.path = ./logs/
# 이전에 생성된 로그 파일 삭제 여부(0:삭제안함, 1:삭제함)
log.file.delete = 0
# 보관할 로그 파일 기간 (단위:일, 2년)
log.file.duration = 730
# 로그 파일에 개인정보 노출차단 사용유무(0:개인정보 숨김 사용안함, 1:개인정보 숨김 사용)
log.use.personalinfo.hide = 0

```

7.1.10. TPM(Throughput Per Minute) 설정

EMMA가 전송하는 메시지의 발송속도를 TPM설정을 이용해 제한 할 수 있다.

Properties

환경변수	의미
tpm.count	TPM Control 사용 유무(설정값: EMMA가 분당 발송하는 메시지의 최대 개수)

설정 용례

```
tpm.count = 0
```

7.1.11. User Library 설정

User Library는 EMMA 가 서비스를 수행하는 동안 특정 프로세스 수행 후 사용자 Library 를 호출한다. 고객사는 메시지 송수신 후 실시간으로 고객사의 특정 업무를 수행할 수 있으며, 특정 처리를 위해 해당 DBMS를 주기적으로 polling 할 필요가 없다.

Properties

환경변수	의미
callback.use.smt	SMS MT User Library 사용 여부(0:사용안함, 1:사용함)
callback.class.smt	SMS MT Interface를 구현한 User Class Name
callback.use.mmt	MMS MT User Library 사용 여부(0:사용안함, 1:사용함)
callback.class.mmt	MMS MT Interface를 구현한 User Class Name
callback.use.smo	SMS MO User Library 사용 여부(0:사용안함, 1:사용함)

callback.class.smo	SMS MO Interface를 구현한 User Class Name
callback.use.mmo	MMS MO User Library 사용 여부(0:사용안함, 1:사용함)
callback.class.mmo	MMS MO Interface를 구현한 User Class Name
callback.use.mtt	DB에서 MT데이터를 가져와 MT발송 전 데이터를 조작하기 위한 User Library 사용 여부(0:사용안함, 1:사용함)
callback.class.mtt	DB에서 MT데이터를 가져와 MT발송 전 데이터를 조작하기 위한 Interface를 구현한 User Class Name
callback.use.mot	수신받은MO 데이터를 DB에 넣기 전 데이터를 조작하기 위한 User Library 사용 여부(0:사용안함, 1:사용함)
callback.class.mot	수신받은 MO데이터를 DB에 넣기 전 데이터를 조작하기 위한 Interface를 구현한 User Class Name
callback.use.userscheduler	User Scheduler 사용 여부(0:사용안함, 1:사용함)
callback.class.userscheduler	User Scheduler Interface를 구현한 User Class Name
callback.use.dbs	DB패스워드 조회 User Library Call사용여부(0:사용안함, 1:사용함)
callback.class.dbs	DB 패스워드 조회 Interface를 구현한 User Class Name

설정 용례

```
# SMS MT User Library Call 사용 유무(0:사용안함, 1:사용함)
callback.use.smt = 0
callback.class.smt = ib.emma.callback. SampleSMTCallback

# MMS MT User Library Call 사용 유무(0:사용안함, 1:사용함)
callback.use.mmt = 0
callback.class.mmt = ib.emma.callback. SampleMMTCallback

# SMS MO User Library Call 사용 유무(0:사용안함, 1:사용함)
callback.use.smo = 0
callback.class.smo = ib.emma.callback. SampleSMOCallback

# MMS MO User Library Call 사용 유무(0:사용안함, 1:사용함)
callback.use.mmo = 0
callback.class.mmo = ib.emma.callback. SampleMMOCallback

# MT Data Transform Library Call 사용 유무(0:사용안함, 1:사용함)
callback.use.mtt = 0
callback.class.mtt = ib.emma.callback.SampleMTTCallback

# MO Data Transform Library Call 사용 유무(0:사용안함, 1:사용함)
callback.use.mot = 0
callback.class.mot = ib.emma.callback.SampleMOTCallback
```

```
# UserScheduler User Library Call 사용 유무(0:사용안함, 1:사용함)
callback.use.userscheduler = 0
callback.class.userscheduler = ib.emma.callback.SampleUSCCallback

# DB 패스워드 조회 User Library Call 사용 유무(0:사용안함, 1:사용함)
callback.use.dbs = 0
callback.class.dbs = ib.emma.callback.SampleDBSCallback
```

7.1.12. HA(High Availability) 설정

HA 서비스는 장애가 발생하더라도, 중단 없는 서비스를 위해 EMMA 서버를 이중화하는 것을 말한다. EMMA는 Active-Active mode, Active-Standby mode의 HA 서비스를 모두 제공한다. Active-Active방식은 두 대의 서버에 설치된 EMMA가 동시에 송수신하는 방식이며, Active-Standby Primary 서버의 EMMA가 메인으로 동작하며, Primary 서버 장애 시 Secondary 서버가 자동으로 장애를 감지하여, 이어서 전송하는 것을 말한다.

Properties

환경변수	의미
ha.mode	HA 사용 방식 (none : 사용안함, active-active, active-standby)
ha.emma.id	설치된 EMMA 고유 ID (E1, E2, E3...)로 중복될 수 없다.
ha.status	EMMA 설정 상태(primary, secondary), active-standby 모드에서만 사용
ha.primary.host	Primary EMMA서버 접속 IP, active-standby 모드에서만 사용
ha.primary.port	Primary EMMA서버 접속 포트, active-standby 모드에서만 사용

설정 용례(Active-Active)

```
# Server #1 설정
ha.mode = active-active
ha.emma.id = E1
#ha.status = primary
#ha.primary.host = localhost
#ha.primary.port = 4000

# Server #2 설정
ha.mode = active-active
ha.emma.id = E2
#ha.status = primary
#ha.primary.host = localhost
```

```
#ha.primary.port = 4000
```

설정 용례(Active-Standby)

```
# Server #1 설정
ha.mode = active-standby
ha.emma.id = E1
ha.status = primary
ha.primary.host = localhost
ha.primary.port = 4000 (Listen Port)

# Server #2 설정
ha.mode = active-standby
ha.emma.id = E2
ha.status = secondary
ha.primary.host = 10.10.121.137 (Primary 접속 IP 또는 host)
#ha.primary.port = 4000(primary 서버 접속 포트)
```

7.1.13. BRIDGE or DIST 모듈 연동 시 사용 되는 설정

BRIDGE or DIST 모듈 연동 시 EMMA 에서 사용하는 설정 정보이다.

Properties

환경변수	의미
que.use.distribute	분배 서비스 사용 유무 (1:사용함, 0:사용안함)
que.name.distribute.smsmt	SMSMT / URLMT 의 분배 큐 이름
que.name.distribute.mmsmt	MMSMT / LMSMT 의 분배 큐 이름
que.use.smrelay	SMSMO Relay 사용 유무 (0:사용안함, 1:사용함)
que.use.mmorelay	MMSMO Relay 사용 유무 (0:사용안함, 1:사용함)

설정 용례

```
# 분배 서비스 사용유무 (0:사용안함, 1:사용함)
que.use.distribute = 0

# 분배 큐 이름 설정
que.name.distribute.smsmt = dist_que_smsmt
que.name.distribute.mmsmt = dist_que_mmsmt

# MO Relay 사용 유무 (0:사용안함, 1:사용함)
que.use.smrelay = 0
que.use.mmorelay = 0
```

7.1.14. Bypass 모듈 연동 시 사용 되는 설정

Bypass 모듈 연동 시 EMMA 에서 사용하는 설정 정보이다.

Properties

환경변수	의미
bypass.use	Bypass 서비스 사용여부 (0:사용안함, 1:사용함)
bypass.address0	Bypass 서비스 target 주소 설정(N개 설정 가능)

설정 용례

```
# Bypass 서비스 사용유무 (0:사용안함, 1:사용함)
```

```
bypass.use = 0
```

```
# Bypass 서비스 target 주소 설정
```

```
bypass.address0 = 127.0.0.1:3000
```

```
bypass.address1 = 127.0.0.1:4000
```

```
bypass.address2 = 10.10.121.137:5000
```

```
.....
```

7.2. db.cf 설정

db.cf 파일은 EMMA 가 연결하는 데이터베이스 관련 정보 설정 파일이다.

7.2.1. 데이터베이스 연결 정보

EMMA가 JDBC로 DB연결 시 사용하는 정보를 설정한다. EMMA V3에서는 Multi-Database Instance 동시 접속이 가능하다. Index는 실제로 연결할 Database의 순번에 해당된다. 다중 데이터베이스 연결을 할 경우 **db.count** 개수를 사용할 데이터베이스 개수에 맞게끔 설정해 준다.

Properties

환경변수	의미
db.count	1
db0.type	접속 데이터베이스 타입 (1-Oracle, 2-MSSQL, 3-MySQL, 5-PostgreSQL, 6-DB2, 7-Tibero, 8-Maria, 9-Sybase)
db0.driver	JDBC Driver 이름
db0.url	JDBC 연결 URL
db0.user	DB user

db0.password	User Password
db0.monum	DB에 매핑시킬 MO번호 (M0번호에 따라 다중 데이터베이스 연결 시 사용)

설정 용례(Oracle)

```
db0.type=1
db0.driver=oracle.jdbc.driver.OracleDriver
db0.url=jdbc:oracle:thin:@localhost.:1521:imds
db0.user=imds
db0.password=psimds
db0.monum=
```

설정 용례(MSSQL)

```
db0.type=2
db0.driver=com.microsoft.jdbc.sqlserver.SQLServerDriver
db0.url=jdbc:microsoft:sqlserver://localhost.49:1433;SelectMethod=cursor;DatabaseName=imds
db0.user=imds
db0.password=psimds
db0.monum=
```

설정 용례(MySQL)

```
db0.type=3
db0.driver=org.gjt.mm.mysql.Driver
db0.url=jdbc:mysql://localhost.141:3306/imds?useUnicode=true&characterEncoding=euc_kr
db0.user=imds
db0.password=psimds
db0.monum=
```

설정 용례(PostgreSQL)

```
db0.type=5
db0.driver=org.postgresql.Driver
db0.url=jdbc:postgresql://localhost.141:5432/imds
db0.user=imds
db0.password=psimds
db0.monum=
```

설정 용례(DB2)

```
db0.type=6
db0.driver=com.ibm.db2.jcc.DB2Driver
db0.url=jdbc:db2://localhost:50000/dbname
db0.user=imds
db0.password=psimds
db0.monum=
```

설정 용례(Tibero)

```
db0.type=7
db0.driver=com.tmax.tibero.jdbc.TbDriver
db0.url= jdbc:tibero:thin:@localhost:8629:dbname
db0.user=imds
db0.password=psimds
db0.monum=
```

설정 용례(Maria)

```
db0.type=8
db0.driver=org.mariadb.jdbc.Driver
db0.url= jdbc:mariadb://localhost:3306/imds?useUnicode=true&noAccessToProcedureBodies=true
db0.user=imds
db0.password=psimds
db0.monum=
```

설정 용례(Sybase)

```
db0.type=9
db0.driver=net.sourceforge.jtds.jdbc.Driver
db0.url= jdbc:jtds:sybase://localhost:3344/imds
db0.user=imds
db0.password=psimds
db0.monum=
```

7.2.2. 기타 설정

Properties

환경변수	의미
db.ttl	전송 유효 시간 설정 (단위, 분) 아래 설정은 7월 13일 12:00:00로 예약된 메시지에 대해 14일 12:00:00 까지만 발송을 가능하게 한다. 이 설정은 고객 DB 시간에 한해 적용된다.

db.pollinginterval	전송할 메시지들이 DB에 존재하는지 여부를 조회하는 주기를 설정할 수 있다. '1'이라는 설정은 테이블 조회 후, 1초를 sleep한다는 의미이다. 그러나 전송할 메시지들이 DB에 존재하여 최근에 전송한 레코드가 존재한다면 아래 설정은 무시되어 sleep time 없이 다음 전송을 위해 테이블을 조회한다.
db.checkprivilege	생성 (조회, 수정, 삭제 포함) 권한을 체크 한다.
db.spprefix	다중 EMMA를 사용할 경우 각 EMMA가 Call하는 Stored Procedure를 구분하기 위한 Prefix이다.
db.bancheck	em_banlist(스팸차단) 테이블 사용 여부(0:사용안함, 1:사용함)
db.bancheck.interval	em_banlist(스팸차단) 테이블 정보 갱신 주기(단위:초)
db.encrypted	db.cf DB연결정보(db0.user, db0.password) 암호화 여부 (0:암호화 안되어 있는 경우, 1:암호화 되어 있는경우)
db.querytimeoutcheck	Oracle 쿼리 수행 시간에 제한을 두기 위해 사용한다. Oracle 11g의 경우 query 수행 도중 멈추는 현상이 있을 수 있으므로 사용하기를 권장한다. Default는 '0'(사용하지 않음) '1' (사용함)
db.querytimeout	Oracle 쿼리 수행 시간에 제한 둘 경우, 이 설정으로 timeout 시간을 설정한다. 단위는 '초'이다.

설정 용례(Oracle)

```
# 전송 유효 시간 설정 (단위, 분)
# 전송 예약 시간부터 1440 분 (24 시간) 이전 데이터만 발송이 가능하다.
db.ttl = 1440

# 전송 테이블 조회 주기 설정 (단위, 초)
db.pollinginterval = 10

# 테이블 생성 (조회, 수정, 삭제 포함) 권한을 체크한다.
db.checkprivilege = 1

# Stored Procedure Prefix
# EMMA 가 다중 설치되어 있을 경우 사용하는 Stored Procedure 구분 prefix
#db.spprefix = e1_

# em_banlist(스팸차단) 테이블 사용 여부(0:사용안함, 1:사용함)
db.bancheck = 1

# em_banlist(스팸차단) 테이블 정보 갱신 주기(단위:초)
db.bancheck.interval = 60
```



```
# database information encryption Y/N
db.encrypted = 0

# oracle 쿼리 수행 제한 시간 설정(단위 : 초)
db.querytimeoutcheck = 1
db.querytimeout = 10
```

7.3. log4j.properties

EMMA는 어플리케이션에서 발생하는 여러 가지 상황들을 로그(log)를 통해 알려준다. EMMA는 기본적으로 Apache Logging Services인 log4j 를 사용한다. 단 다른 Logging Library 적용이 가능하도록 Simple Logging Façade 인 SLF4J 을 사용하여 쉽게 변경이 가능하다.

log4j.properties파일의 기본 설정을 변경하고자 할 경우, log4j 홈페이지(<http://logging.apache.org/>)를 참조하기 바란다.

rootlogger 설정 용례(일반운영 시)

```
#set rootlogger
log4j.rootLogger=INFO, IBCON, IBINFO
```

rootlogger 설정 용례(DEBUG)

```
#set rootlogger
log4j.rootLogger=DEBUG, IBCON, IBINFO, IBDEBUG
```

rootlogger 설정 용례 TRACE)

```
#set rootlogger
log4j.rootLogger=TRACE, IBCON, IBINFO, IBTRACE
```

8. User Library

EMMA 는 각 서비스 별로 특정 프로세스 수행 후 고객이 작성한 User Library 를 호출하는 기능을 제공한다. EMMA V.3 이전 버전에서는 메시지 송수신 후 특정 Job 수행을 위해 Trigger등을 사용하거나 별도로 DB를 Polling하는 Application을 개발했지만, EMMA V.3 이후 버전에서는 User Library 을 사용하기를 권고한다.

이 절에서는 User Library를 개발, 사용법에 대해 설명한다. User Library Call 기능을 사용하지 않는다면 이 절을 건너뛰어도 상관 없다.

📌 중요

EMMA는 User Library 기능을 위한 Interface를 정의하여 고객은 이 Interface를 구현하는 Class를 작성하면 된다. 각 클래스 Sample은 EMMA 배포 시 \$EMMA_HOME/src 폴더에 같이 배포된다.

8.1. User Library 종류

EMMA에서 제공 되는 User Library는 아래와 같다.

8.1.1. SMS MT Callback (IBSMTCallback.java)

SMS MT 전송 과정에서 메시지 전송 또는 전송결과 수신 후 수행 후 호출 되는 프로세스를 정의한 Interface이다. 주로 결과 수신 후 특정 Job을 수행하고자 할 경우 사용된다.

```
● Void doAfterEnque(  
    int dbIndex,  
    String key,  
    int seq,  
    String type,  
    String recipientNum,  
    String callback,  
    String msg  
)
```

설명	• SMS MT 를 DB 테이블에서 조회하여 전송 큐에 enqueue 후 호출되는 메소드
Parameter	• dbIndex : 연결할 Database Index. 하나의 EMMA가 2개 이상의 DB에 연결할 때 사용한다. • key : 메시지 키 (mt_pr). • seq : 메시지 키의 순번. 일반적으로는 0이며, 동보 전송을 할때 순번으로 사용된다. • type : 메시지 타입 (SMT : 0, URL : 1) • recipientNum : 수신자 번호 • callback : 콜백 번호 (발신자 번호) • msg : 송신 메시지
Return	• void

```
● Void doAfterSend(  
    String key,  
    String code  
)
```

설명	• SMS MT 전송 큐에서 dequeue 하여 인포뱅크 G/W 송신 후 호출되는 메소드
Parameter	<ul style="list-style-type: none"> • key : 메시지 키 (mt_pr). • code : IB 송신 결과 코드
Return	• void

```

● Void doAfterReceive(
    String key,
    int seq,
    String recipientNum,
    String code,
    int carrier,
    long date
)

```

설명	• SMS MT 이통사 전송 결과를 수신하여 리포트 큐에 enqueue 후 호출되는 메소드
Parameter	<ul style="list-style-type: none"> • key : 메시지 키 (mt_pr). • seq : 메시지 키의 순번. 일반적으로는 0이며, 동보 전송을 할때 순번으로 사용된다. • recipientNum : 수신자 번호 • code : 이통사 결과코드 • carrier : 최종 착신망 정보 (SKT/KT/LGU+) • date : 단말기 도착 시간
Return	• void

```

● Void doAfterDequeue(
    int dbIndex,
    String key,
    int seq,
    String recipientNum,
    String code,
    int carrier,
    long date
)

```

설명	• SMS MT 이통사 전송 결과를 리포트 큐에서 dequeue 후 호출되는 메소드 (모든 전송 과정이 끝난 후 호출되는 메소드)
Parameter	<ul style="list-style-type: none"> • dbIndex : 연결할 Database Index. 하나의 EMMA가 2개 이상의 DB에 연결할 때 사용한다. • key : 메시지 키 (mt_pr). • seq : 메시지 키의 순번. 일반적으로는 0이며, 동보 전송을 할때 순번으로 사용된다. • recipientNum : 수신자 번호

- code : 이통사 결과코드
- carrier : 최종 착신망 정보 (SKT/KT/LGU+)
- date : 단말기 도착 시간

Return • void

8.1.2. MMS MT Callback (IBMMTCallback.java)

MMS MT 전송 과정에서 메시지 전송 또는 전송결과 수신 후 수행 후 호출 되는 프로세스를 정의한 Interface이다. 주로 결과 수신 후 특정 Job을 수행하고자 할 경우 사용된다.

- **Void doAfterEnque (**
 int dbIndex,
 String key,
 int seq,
 String recipientNum,
 String callback,
 String msg
)

설명 • MMS MT 를 DB 테이블에서 조회하여 전송 큐에 enqueue 후 호출되는 메소드

Parameter • dbIndex : 연결할 Database Index. 하나의 EMMA가 2개 이상의 DB에 연결할 때 사용한다.
 • key : 메시지 키 (mt_pr).
 • seq : 메시지 키의 순번. 일반적으로는 0이며, 동보 전송을 할때 순번으로 사용된다.
 • recipientNum : 수신자 번호
 • callback : 콜백 번호 (발신자 번호)
 • msg : 송신 메시지

Return • void

- **Void doAfterSend(**
 String key,
 String code
)

설명 • MMS MT 전송 큐에서 dequeue 하여 인포뱅크 G/W 송신 후 호출되는 메소드

Parameter • key : 메시지 키 (mt_pr).
 • code : IB 송신 결과 코드

Return • void

- **Void doAfterReceive(**

String key,
int seq,
String recipientNum,
String code,
int carrier,
long date
)

설명	• MMS MT 이통사 전송 결과를 수신하여 리포트 큐에 enqueue 후 호출되는 메소드
Parameter	<ul style="list-style-type: none"> • key : 메시지 키 (mt_pr). • seq : 메시지 키의 순번. 일반적으로는 0이며, 동보 전송을 할때 순번으로 사용된다. • recipientNum : 수신자 번호 • code : 이통사 결과코드 • carrier : 최종 착신망 정보 (SKT/KT/LGU+) • date : 단말기 도착 시간
Return	• void

- Void doAfterDequeue(
int dbIndex,
String key,
int seq,
String recipientNum,
String code,
int carrier,
long date
)

설명	• MMS MT 이통사 전송 결과를 리포트 큐에서 dequeue 후 호출되는 메소드 (모든 전송 과정이 끝난 후 호출되는 메소드)
Parameter	<ul style="list-style-type: none"> • dbIndex : 연결할 Database Index. 하나의 EMMA가 2개 이상의 DB에 연결할 때 사용한다. • key : 메시지 키 (mt_pr). • seq : 메시지 키의 순번. 일반적으로는 0이며, 동보 전송을 할때 순번으로 사용된다. • recipientNum : 수신자 번호 • code : 이통사 결과코드 • carrier : 최종 착신망 정보 (SKT/KT/LGU+) • date : 단말기 도착 시간
Return	• void

8.1.3. SMS MO Callback (IBSMOCallback.java)

SMS MO 메시지 수신 후 수행 후 호출 되는 프로세스를 정의한 Interface이다.

- **Void doAfterReceive(
 String key,
 String moRecipient,
 String moOriginator,
 String moCallback,
 String msg,
 int carrier,
 long date,
 int extraValue
)**

설명	• SMS MO를 수신하여 enqueue 후 호출되는 메소드
-----------	-----------------------------------

Parameter	<ul style="list-style-type: none">• key : MO Key• moRecipient : MO 특번 (MO 번호)• moOriginator : MO 발신 단말기 번호• moCallback : MO 발신시 발신자가 수동으로 입력한 번호 (기본적으로 단말기 번호)• msg : MO 메시지 내용• carrier : 최종 착신망 정보 (SKT/KT/LGU+)• date : MO 발생 시간• extraValue : LGU+ EMS MO 구분 순번. LGU+ EMS MO 인 경우 사용. LGU+ 일부 단말에서 sms mo를 80바이트 이상 장문으로 보내면 80바이트씩 나누어 보내주는 경우에 사용되는 value 값. 메시지 내용의 앞부분 1byte 값이 sequence 이며, 1byte 값을 받음.
------------------	---

Return	• void
---------------	--------

- **Void doAfterDequeue(
 int dbIndex,
 String key,
 String moRecipient,
 String moOriginator,
 String moCallback,
 String msg,
 int carrier,
 long date,
 int extraValue
)**

설명	• SMS MO 수신 후 큐에서 dequeue하여 DB Insert 후 호출되는 메소드
Parameter	<ul style="list-style-type: none"> • dbIndex : 연결할 Database Index. 하나의 EMMA가 2개 이상의 DB에 연결할 때 사용한다. • key : MO Key • moRecipient : MO 특번 (MO 번호) • moOriginator : MO 발신 단말기 번호 • moCallback : MO 발신시 발신자가 수동으로 입력한 번호 (기본적으로 단말기 번호) • msg : MO 메시지 내용 • carrier : 최종 착신망 정보 (SKT/KT/LGU+) • date : MO 발생 시간 • extraValue : LGU+ EMS MO 구분 순번. LGU+ EMS MO 인 경우 사용. LGU+ 일부 단말에서 sms mo를 80바이트 이상 장문으로 보내면 80바이트씩 나누어 보내주는 경우에 사용되는 value 값. 메시지 내용의 앞부분 1byte 값이 sequence 이며, 1byte 값을 받음.
Return	• void

8.1.4. MMS MO Callback (IBMMOCallback.java)

MMS MO 메시지 수신 후 수행 후 호출 되는 프로세스를 정의한 Interface이다.

- **Void doAfterReceive(**
String key,
String moRecipient,
String moOriginator,
String moCallback,
String subject,
String msg,
int carrier,
long date
)

설명	• MMS MO를 수신하여 enqueue 후 호출되는 메소드
Parameter	<ul style="list-style-type: none"> • key : MO Key • moRecipient : MO 특번 (MO 번호) • moOriginator : MO 발신 단말기 번호 • moCallback : MO 발신시 발신자가 수동으로 입력한 번호 (기본적으로 단말기 번호) • subject : MO 메시지 제목 • msg : MO 메시지 내용 • carrier : 최종 착신망 정보 (SKT/KT/LGU+) • date : MO 발생 시간
Return	• void

- **Void doAfterDequeue(**
 int dbIndex,
 String key,
 String moRecipient,
 String moOriginator,
 String moCallback,
 String subject,
 String msg,
 int carrier,
 long date,
 ArrayList<String> attachFileList
)

설명	• MMS MO 수신 후 큐에서 dequeue하여 DB Insert 후 호출되는 메소드
Parameter	<ul style="list-style-type: none"> • dbIndex : 연결할 Database Index. 하나의 EMMA가 2개 이상의 DB에 연결할 때 사용한다. • key : MO Key • moRecipient : MO 특번 (MO 번호) • moOriginator : MO 발신 단말기 번호 • moCallback : MO 발신시 발신자가 수동으로 입력한 번호 (기본적으로 단말기 번호) • subject : MO 메시지 제목 • msg : MO 메시지 내용 • carrier : 최종 착신망 정보 (SKT/KT/LGU+) • date : MO 발생 시간 • attachFileList : 첨부 파일 목록
Return	• void

8.1.5. MT Data Transform Callback (IBMTTCallback.java)

MT 메시지 전송을 위해 DB에서 Record 검색 후 특정 필드의 값을 가공하기 위해 호출되는 프로세스를 정의한 Interface이다. 주로 휴대번호, 메시지가 암호화하여 사용하거나, 값을 변경하여 전송하고자 할 경우 사용된다.

- **Boolean doSmtTransform(**
 IBRefObject refObj
)

설명	• SMS MT 데이터 조회 후 변환된 데이터를 return 받기 위해 호출되는 메소드
Parameter	• refObj : IB 참조 개체. 수신번호, 회신번호, 메시지, 동보단어를 포함하고 있다. 그 외에 여분의 필드 4개가 있다.
Return	• 변환 성공 : true

- 변환 실패 : false

- **Boolean doMmtTransform(
IBRefObject refObj
)**

설명	• MMS MT 데이터 조회 후 변환된 데이터를 return 받기 위해 호출되는 메소드
Parameter	• refObj : IB 참조 개체. 수신번호, 회신번호, 메시지, 동보단어를 포함하고 있다. 그 외에 여분의 필드 4개가 있다.
Return	<ul style="list-style-type: none"> • 변환 성공 : true • 변환 실패 : false

8.1.6. MO Data Transform Callback (IBMOTCallback.java)

MO 메시지 수신 후 DB에 insert 하기 전에 필드의 값을 가공하기 위해 호출되는 프로세스를 정의한 Interface이다. 주로 휴대번호, 메시지를 암호화하여 DB에 Insert 하고자 할 경우에 사용된다.

- **Boolean doSmoTransform(
IBRefObjectMO refObjMO
)**

설명	• SMS MO 데이터 수신 후 변환된 데이터를 return 받기 위해 호출되는 메소드
Parameter	• refObjMO : IB 참조 개체. MO 번호, 발신자 실제 단말기 번호, 발신자가 수동으로 입력한 번호, 메시지, 제목을 포함하고 있다. 발신자가 수동으로 입력한 번호는 기본적으로 실제 단말기 번호이다.
Return	<ul style="list-style-type: none"> • 변환 성공 : true • 변환 실패 : false

- **Boolean doMmoTransform(
IBRefObjectMO refObjMO
)**

설명	• MMS MO 데이터 수신 후 변환된 데이터를 return 받기 위해 호출되는 메소드
Parameter	• refObjMO : IB 참조 개체. MO 번호, 발신자 실제 단말기 번호, 발신자가 수동으로 입력한 번호, 메시지, 제목을 포함하고 있다. 발신자가 수동으로 입력한 번호는 기본적으로 실제 단말기 번호이다.
Return	<ul style="list-style-type: none"> • 변환 성공 : true • 변환 실패 : false

8.1.7. User Scheduler Callback (IBUSCCallback.java)

사용자 정의 Scheduler 프로세스를 정의한 Interface이다. 주로 특정 주기로 Job을 수행하고자 할 경우 사용된다.

- **Void run(
 int dbIndex,
 int dbCount
)**

설명	• 설정된 시간마다 동작하게 되어있는 사용자 콜백 함수
Parameter	• dbIndex : 연결할 Database Index (기본적으로 0) • dbCount : 전체 Database Count (기본적으로 1)
Return	• void

8.1.8. DB Security Callback (IBDBSCallback.java)

JDBC 드라이버가 DB 연결 시도할 때, 사용할 DB 사용자에게 대한 비밀번호를 가공하기 위해 사용하기 위한 Interface이다. 기본적으로 db.cf의 db0.password의 값으로 DB 연결을 시도하나 이 Interface를 사용하면 고객센터의 필요에 따라 DB 패스워드에 대한 보안 관리를 수행할 수 있다.

- **String getDBPassword(
 int dbIndex,
 String driver,
 String url,
 String dbUserId)**

설명	• DB 패스워드를 리턴하는 콜백 함수
Parameter	• dbIndex : 연결할 Database Index (기본적으로 0) • driver : DB Driver class 명 • url : JDBC URL 주소 • dbUserId : DB 접근 사용자 ID
Return	• String : EMMA의 특정 dbIndex에 대한 DB 패스워드를 리턴. Null을 리턴할 경우, db.cf의 db[index].password를 사용함.

8.2. User Library Compile 및 동작방법

각 서비스 별 User Library 컴파일 방법 및 동작 설정 방법은 유사하므로, 이 절에서는 SMS MT 서비스 기준으로 설명한다.

8.2.1. SMS MT User Library Compile 방법

예제로 제공되는 \$EMMA_HOME/src/ib/emma/callback/SampleSMTCallback.java을 클래스 이름 및

모듈을 고객의 환경에 맞게 수정한 후 javac 명령어를 이용하여 컴파일 한다. 아래 예제는 메시지 전송 후 결과 수신 시 특정 테이블을 업데이트 하는 로직을 수행하는 User Callback 이다.

참고

나머지 User Library 적용 방법은 동일하며, Method 파라미터 및 처리 방법은 실제 예제를 참조 하기 바란다.

샘플 Source 수정

```
public void doAfterDequeue(int dbIndex, String key, int seq,
    String recipientNum, String code, int carrier, long date) {

    Connection conn = null;
    PreparedStatement pstmt = null;
    String strQuery = null;
    strQuery = "insert into user_table"
        + "( mt_pr, msg_status, result_code )"
        + "values( ?, '3', ? );";

    try {
        DBConnector dbConnector = new DBConnector();
        conn = dbConnector.getConnection(0);

        pstmt = conn.prepareStatement(strQuery);
        pstmt.setString(1, key);
        pstmt.setInt(2, carrier);
        int ret = pstmt.executeUpdate();
        logger.info("USER_TABLE INSERT: {}/{}/{}", new Object[]{ret, key, carrier });
    } catch (SQLException e) {
        logger.error("SQLException occured when inserting user_table", e);
    } catch (Exception e) {
        logger.error("Exception occured when inserting user_table", e);
    } finally {
        try { pstmt.close(); } catch (Exception e) { }
        try { conn.close(); } catch (Exception e) { }
    }
}
```

컴파일

▶ java version 확인

```
imds@zeus:~/Infobank/EMMA$ java -version
java version "1.6.0_12"
Java(TM) SE Runtime Environment (build 1.6.0_12-b04)
Java HotSpot(TM) Server VM (build 11.2-b01, mixed mode)
```

▶ Compile

```
imds@zeus:~/Infobank/EMMA$ cd src
imds@zeus:~/Infobank/EMMA/src$ javac -classpath ../lib/emma-3.1.3.jar:../lib/slf4j-api-1.5.6.jar
ib/emma/callback/SampleSMTCallback.java
```

8.2.2. SMS MT User Library 설정

8.2.1에서 생성된 클래스를 \$EMMA_HOME/ib/emma/callback 밑에 복사 후 emma.cf 를 설정한 후 EMMA 을 재 기동 하여 테스트 한다.

적용 예

▶ 컴파일 된 클래스 복사

```
imds@zeus:~/Infobank/EMMA$ cp SampleSMTCallback.class
$EMMA_HOME/classes/ib/emma/callback/
```

▶ emma.cf 설정

```
callback.use.smt = 1
callback.class.smt = ib.emma.callback.SampleSMTCallback
```

▶ emma restart

```
imds@zeus:~/Infobank/EMMA$ ./emmasvc stop; ./emmasvc start
```