



Rich Talk Agent(RTA)

사용자 설명서 V1.0.4

Last Modified: 22 Jan, 2025

Rich Talk Agent

사용자 설명서

Version 1.0.4

Copyright©InfoBank Corp. All rights reserved. 2021

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㈜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㈜의 허가 없이 기계적, 전자적 또는 기타 임의의 수단을 통해 배포되거나 복사할 수 없습니다.

Rich Talk Agent에 관한 궁금한 사항은 인포뱅크㈜로 문의하시기 바랍니다.

변경 이력

[illegible]

목차

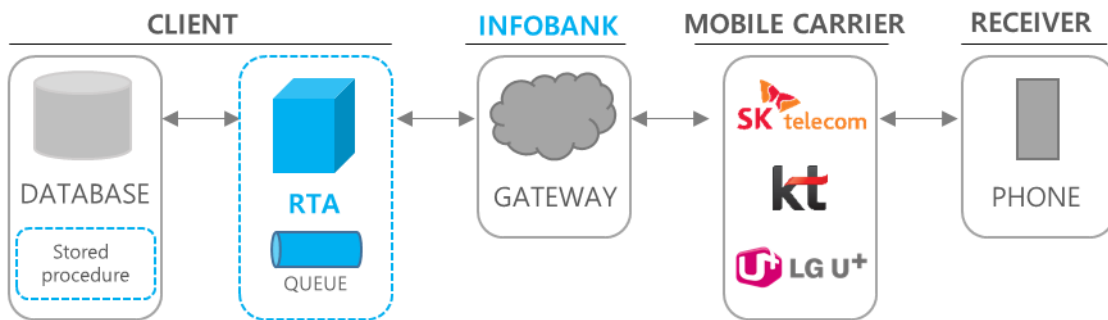
1. 개요	6
2. 사전 준비사항.....	7
3. 설치 및 제거.....	7
3.1. 시스템 요구사항	7
3.2. RTA 설치.....	8
3.2.1. Linux/Unix 버전.....	8
3.2.2. Windows 버전	12
3.3. 데이터베이스 프로시저.....	19
3.3.1. 일반 프로시저(MySQL, MariaDB, MSSQL, PostgreSQL)	19
3.3.2. Package 프로시저(Oracle, Tibero).....	20
4. 환경설정	21
4.1. RTA 설정파일(RTA.CF)	21
4.2. 데이터베이스 관련 설정파일(DB.CF)	25
4.3. LOG 설정파일(LOGBACK.XML)	28
5. 데이터베이스.....	30
5.1. 테이블 구성 및 명세서	30
5.1.1. 테이블 구성.....	30
5.1.2. 테이블 명세서(MySQL).....	30
5.2. 프로시저 구성	34
5.3. 기타 객체	35
5.3.1. 시퀀스(Oracle, Tibero, PostgreSQL).....	35
5.3.2. 함수(MySQL, MariaDB).....	35
6. 발송 서비스.....	36
6.1. RCS 메시지 종류.....	36
6.2. 메시지 상태 구분	36
6.3. RCS 메시지 전송.....	37
6.4. RCS 이미지 파일 업로드(FILE API).....	44
6.5. RCS 버튼(액션 버튼)	47
6.5.1. RCS버튼(액션 버튼) 구성.....	47
6.5.2. RCS버튼(액션 버튼) 예제.....	47
6.6. RCS CAROUSEL 구조	52
6.7. TEMPLATE 구조	54
6.8. RCS 템플릿 전송.....	54

6.8.1.	<i>LMS 템플릿</i>	54
6.8.2.	<i>정보성 템플릿</i>	60
6.8.3.	<i>이미지 템플릿</i>	62
7.	부가기능	70
7.1.	이중화(HA)	70
7.2.	발송 차단	71
7.3.	중복 메시지 차단	71
7.4.	우선 순위 발송	72
7.5.	BYPASS 모듈 사용	73
7.6.	로그파일 개인정보 미 표시	73
7.7.	사용자 함수	74
7.7.1.	<i>RCS Callback(IBCSCallback.java)</i>	75
7.7.2.	<i>Message Data Transform Callback(IBMDTCallback.java)</i>	77
7.7.3.	<i>User Scheduler Callback(IBUSCCallback.java)</i>	78
7.7.4.	<i>Database Security Callback(IBDBSCallback.java)</i>	78
7.8.	FAILOVER 발송	78
8.	RCS 코드표	81
9.	FAQ	82
9.1.	RTA 관련 FAQ	82
9.2.	데이터베이스 관련 FAQ	82
9.3.	발송 서비스 관련 FAQ	83
10.	문의	83

1. 개요

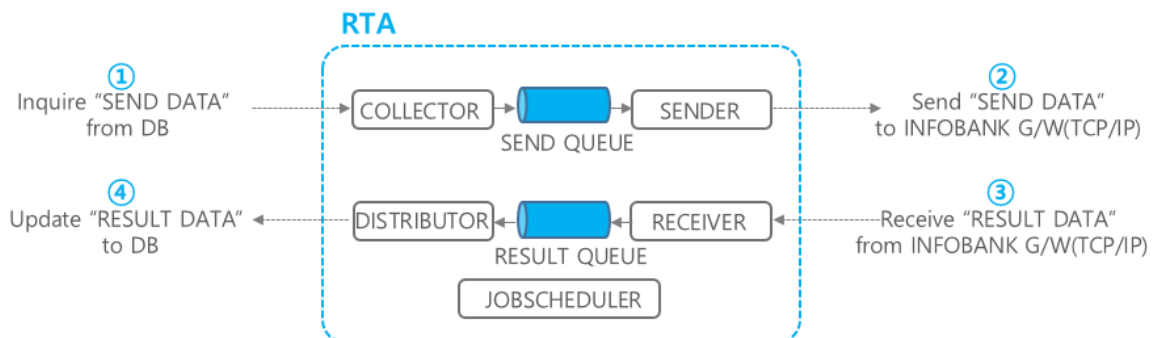
● Rich Talk Agent(RTA)란?

- 인포뱅크의 RCS Messaging Service를 이용하기 위하여 고객사의 시스템에 배포되는 클라이언트 프로그램을 말합니다. RTA는 모든 종류의 RCS메시지들을 전송할 수 있습니다.
- 지정된 데이터베이스 테이블에 발송 데이터를 입력하면 이를 감지한 RTA가 인포뱅크 G/W에 송신하여 이동통신사에 발송 요청 후 결과를 수신 받게 됩니다.



● RTA의 구성 및 원리

- 발송 및 결과 수신에 대한 구성요소는 다음과 같습니다.
 - **COLLECTOR:** 데이터베이스를 조회 하여 발송 대상 데이터를 SEND QUEUE에 저장
 - **SENDER:** SEND QUEUE의 데이터를 인포뱅크 G/W로 송신
 - **RECEIVER:** 인포뱅크 G/W의 결과 데이터를 수신하여 RESULT QUEUE에 저장
 - **DISTRIBUTOR:** RESULT QUEUE의 데이터로 데이터베이스를 갱신 후 LOG 테이블로 이관
 - **JOBSCHEDULER:** 결과데이터가 갱신 되지 못한 발송데이터를 LOG 테이블로 이관
- 데이터 정합 및 운용 안정성을 고려하여 각 기능 사이에 파일 큐를 사용합니다.



2. 사전 준비사항

● RTA 계정 및 설치 파일

- 영업 담당자로부터 RTA 접속 계정 정보를 제공받아야 합니다.
- 인포뱅크 비즈플러스 사이트(<https://www.ibizplus.co.kr>)에 접속하여 사용자 시스템에 맞는 설치파일을 다운로드 합니다.

● RCS Biz Center 가입 및 정보 등록

- RCS 발송을 위해 RCS Biz Center¹ 사이트(<https://www.rcsbizcenter.com>)에 접속하여 가입한 후, 기업, Brand²정보, 발신번호, 대행사(인포뱅크) 등의 정보를 등록해야 합니다.
- 가입 시 영업 담당자에 문의하여 주시기 바랍니다.
- RCS 템플릿³ 발송을 위해서는 템플릿 등록 요청 후 승인이 필요합니다.

● 데이터베이스 접속 계정 및 권한

- RTA가 설치되는 서버에서 접속 가능 한 데이터베이스 계정 정보와 저장 프로시저의 실행 권한(EXECUTE), 테이블 권한(SELECT/UPDATE/DELETE/CREATE)이 필요합니다.

● 네트워크 방화벽 허용

- RTA가 접속하는 인포뱅크 G/W⁴ 에 대해 방화벽 허용(Outbound)이 필요합니다.
- 인포뱅크 G/W는 Any로 열려 있습니다.

3. 설치 및 제거

3.1. 시스템 요구사항

● SYSTEM

구분	설치 시 필요 환경
Linux/Unix	JAVA(JDK/JRE) 6.0 (1.6.0_01) 이상
Windows	

※JAVA설치가 안되 있을 경우 JAVA가 포함된 "RTA withJRE" 버전으로 설치합니다.

¹ 이동통신사가 제공하는 RCS 계정 관리 사이트 입니다.

² 발신 번호와 매칭되어 스마트폰 메시지앱에서 프로필형태로 보여지는 정보입니다.

³ RCS 템플릿에 대한 자세한 사항은 '[6.1 RCS 메시지 종류](#)'를 참고하시기 바랍니다.

⁴ '[3.1 시스템 요구사항](#)' Network를 참고하시기 바랍니다.

● DATABASE

구분	지원 버전
Oracle	9i 이상
Microsoft SQL Server	2000 이상
MySQL	5.0 이상 (engine : InnoDB 사용)
PostgreSQL	9.3 이상
MariaDB	5.5 이상
Tibero	5 이상

※RTA는 데이터베이스(저장 프로시저) 기반으로 동작합니다.

● Network

구분	주소	포트
인증 서버	auth.infobank.net (203.248.251.225)	12001
IB G/W Server#1		3000
IB G/W Server#2		3000

※인포뱅크는 Any로 허용되어 있습니다. 고객사 내부 보안 정책이 있을 경우 접근 허용 (Outbound)이 필요합니다.

3.2. RTA 설치

3.2.1. Linux/Unix 버전

● 설치

① 업로드한 설치 파일(RTA_unix_1_x_x.sh)에 실행 권한을 부여합니다.

```
shell> chmod 775 RTA_unix_1_x_x.sh
```

② 매개변수 "-c"를 입력하여 실행 후 설치마법사를 따라 설치합니다.

```
shell> ./RTA_unix_1_x_x.sh -c
```

```
Starting Installer ...
```

```
This will install RTA on your computer.
```

OK [o, Enter], Cancel [c] 설치여부확인

Where should RTA be installed?

[설치경로입력]

Which components should be installed?

1: Oracle files

2: Mssql files

3: Mysql files

4: MariaDB files

Please enter a comma-separated list of the selected values or [Enter] for the default selection:

[3]

데이터베이스 종류 선택

Create symlinks?

Yes [y, Enter], No [n] RTA 실행 쉘의 심볼릭링크 생성 여부

In this step, you configure the RTA authentication information

AUTH.Host :

[auth.infobank.net] 인증서버 URL

AUTH.Port :

[12001] 인증서버 포트

AUTH.ID :

[RTA계정ID]

AUTH.Password :

RTA계정암호

In this step, you configure the RTA service

Database

Oracle [1, Enter]

MSSQL2000 [2]

MSSQL2005 [3]

MySQL [4]

MariaDB [5]

PostgreSQL [6]

Tibero [7]

데이터베이스 종류 선택

In this step, you configure the RTA service

DB IP:

[데이터베이스 접속 주소]

DB Port:

[데이터베이스 접속 포트번호]

DB Name:

[데이터베이스 명]

DB User :

[데이터베이스 접속 계정 명]

DB Password :

데이터베이스 접속계정암호

...설치...

※ 모든 항목들은 설치 완료 후 \$RTA_HOME/conf/rta.cf 또는 db.cf에서 변경 가능합니다. 자세한 설정 방법은 '[4. 환경설정](#)' 을 참조 하시기 바랍니다.

③설치된 경로에서 ./regkey를 실행하여 인증 키를 등록합니다.

```
shell> ./regkey
```

※실행 후 \$RTA_HOME/cert/에 인증서파일이 생성됩니다.

※정상적으로 인증서 파일이 생성된 RTA계정은 다시 regkey.sh을 실행하여도 인증서파일이 재생성⁵되지 않습니다.

④(선택사항)설정 파일에 저장된 접속정보들에 대해 노출을 방지하고자 암호화 합니다.

```
shell> ./authcipher.sh RTA 계정정보(계정 명/암호) 암호화
```

```
shell> ./dbcipher.sh 데이터베이스 접속 계정정보(계정 명/암호) 암호화
```

⁵ 재생성을 위해서는 인포뱅크 영업담당자에 요청 해주시기 바랍니다. 재생성이 될 경우 이전 인증서파일은 사용할 수 없습니다.

- 서비스 명령

- 실행(start)

```
shell> ./rtasvc start  
Starting rtasvc
```

- 상태 확인(status)

```
shell> ./rtasvc status  
The daemon is running. 또는 The daemon is stopped.
```

- 종료(stop)

```
shell> ./rtasvc stop  
Shutting down rtasvc  
Stopped.
```

※종료 명령 이후 “Stopped.” 문구가 정상 출력되지 않으면 비정상적인 상태로 종료가 되지 않았을 수 있습니다. 따라서 종료 이후엔 상태 확인 명령어로 확인 작업이 필요합니다.

※종료가 지속적으로 되지 않을 경우 리눅스 강제 종료 명령어(kill)를 이용하여 종료한 뒤 인포뱅크에 문의 하시기 바랍니다.

- 제거

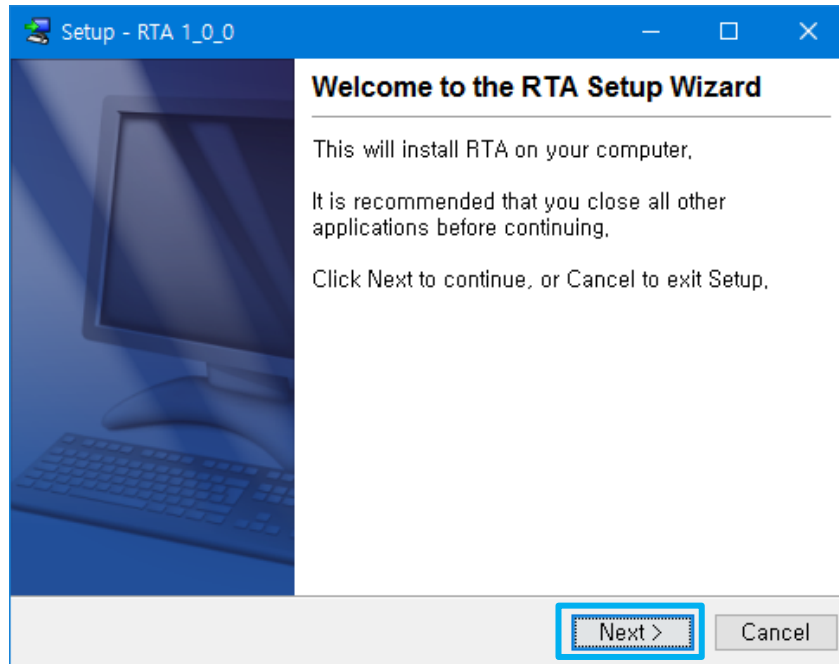
①uninstall.sh에 매개변수 “-c”를 입력하여 실행 후 제거마법사를 따라 제거합니다.

```
shell> ./uninstall -c  
This launcher was created with an evaluation version of install4j  
Are you sure you want to completely remove RTA and all of its components?  
Yes [y, Enter], No [n] 제거 여부 확인  
  
Uninstalling RTA 1_0_0...
```

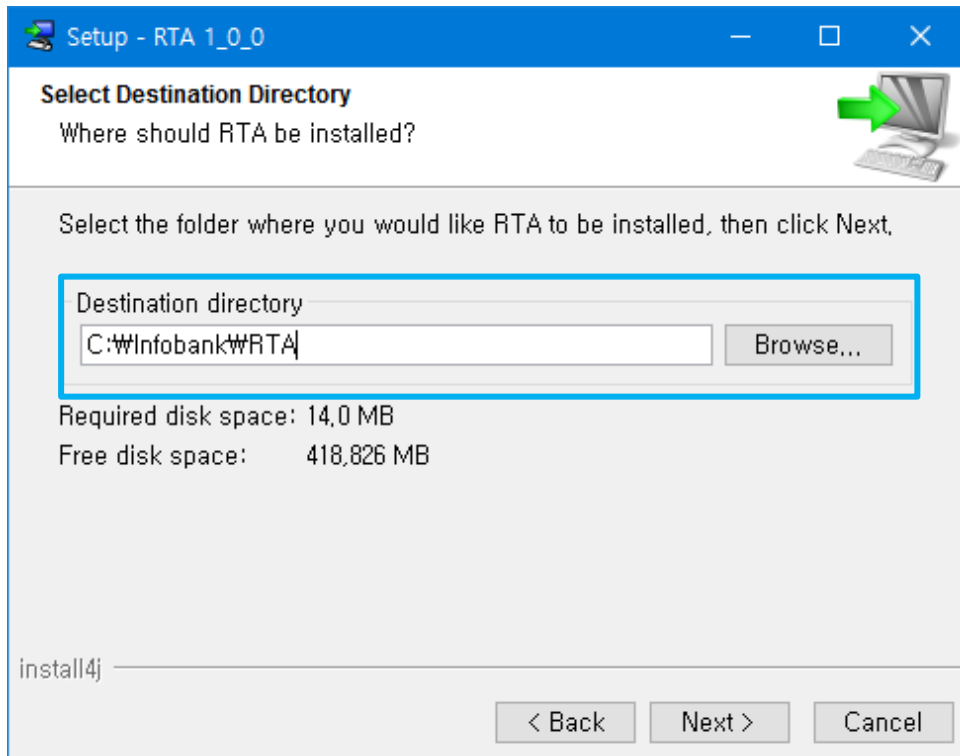
3.2.2. Windows 버전

- 설치

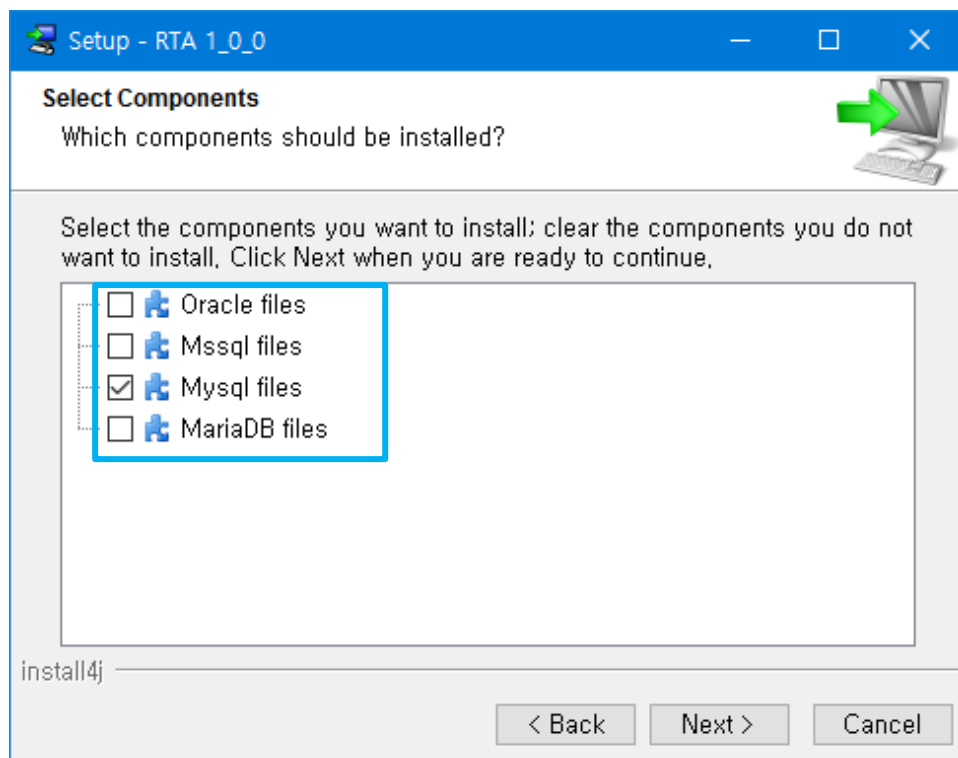
① 업로드한 설치 파일(RTA_unix_1_x_x_64bit.exe)를 더블 클릭하여 실행된 마법사를 따라 설치합니다.



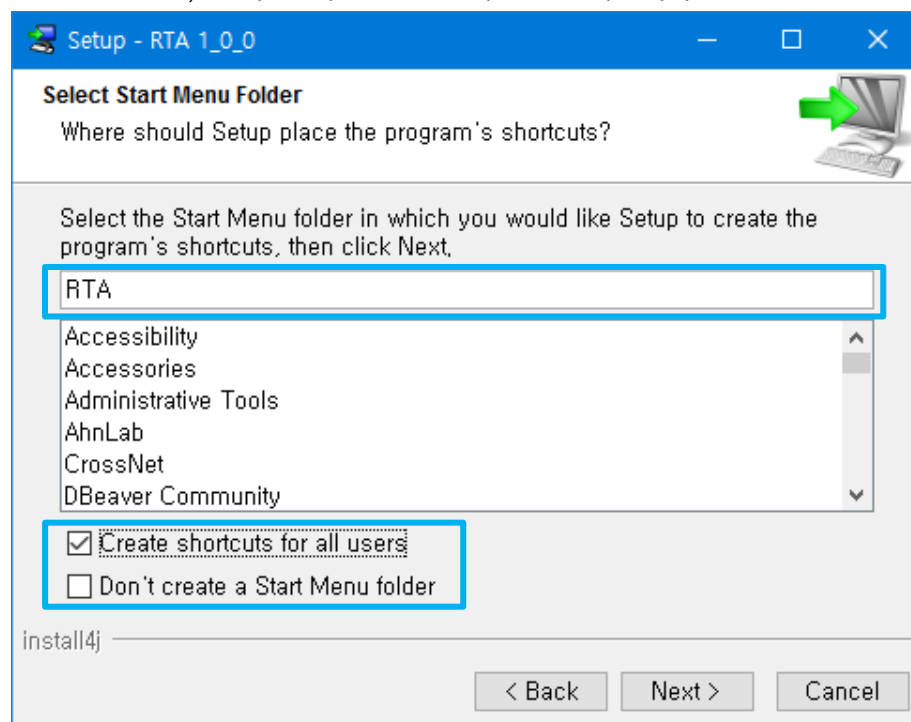
② RTA가 설치될 경로를 설정하고 'Next >' 버튼을 클릭합니다.



③ 접속할 데이터베이스 종류를 선택 하여 체크 활성화 후 'Next >' 버튼을 클릭합니다.



④ 바로 가기 버튼 생성(Create shortcuts for all users)할지, 시작 메뉴 폴더 생성을 안할지(Don't create a Start Menu folder) 선택 한 후 'Next >' 버튼을 클릭합니다.



⑤ 인증 서버 정보 및 RTA 계정정보를 입력하고 'Next >' 버튼을 클릭합니다. (해당 부분 입력하지 않아도 추후 ~rta설치폴더\wconf\wrtacf 파일에서 수정 가능합니다.)

Setup - RTA 1_0_0

RTA Authentication Configuration

In this step, you configure the RTA authentication information

AUTH,Host : rcs-auth.infobank.net

AUTH,Port : 12001

AUTH,ID :

AUTH>Password :

install4j

< Back Next > Cancel

⑥ 접속할 데이터베이스 종류를 선택 하여 체크 활성화 후 'Next >' 버튼을 클릭합니다.

Setup - RTA 1_0_0

RTA Service Configuration

In this step, you configure the RTA service

Database MySQL

install4j

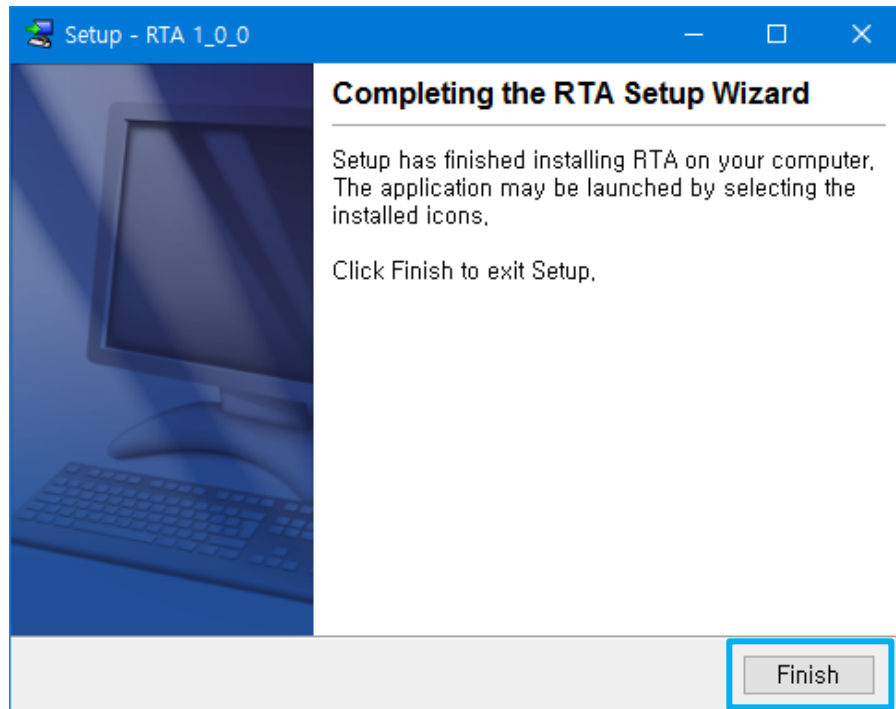
< Back Next > Cancel

⑦ 접속할 데이터베이스 정보들을 입력 한 후 'Next >' 버튼을 클릭합니다. (해당 부분 입력하지 않아도 추후 ~rta설치폴더\conf\wdb.cf 파일에서 수정 가능합니다.)

⑧ Window 서비스에 등록⁶(Install service "rtasvc")할지, 부팅 시 시작(Start on bootup)할지 선택 한 후 'Next >' 버튼을 클릭합니다.

⁶ 서비스 수동 등록 및 삭제를 위해서는 CMD에서 sc create, sc delete 명령어를 이용하여 주시기 바랍니다.

⑨ 설치가 진행되고 'Finish' 버튼을 클릭하여 완료합니다.



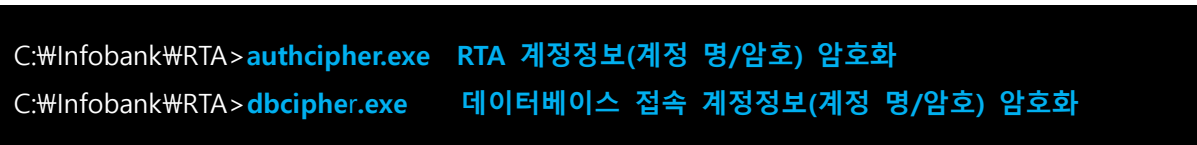
⑩ RTA계정정보가 rta.cf에 정상적으로 입력된 상태에서 설치된 경로에서 regkey.exe를 실행(탐색기에서 더블 클릭 또는 CMD에서 실행)하여 인증 키를 등록합니다.



※실행 후 WRTA_HOME\cert\에 인증서파일이 생성됩니다.

※정상적으로 인증서 파일이 생성된 RTA계정은 다시 regkey.exe를 실행하여도 인증서파일이 재 생성⁷되지 않습니다.

⑪(선택사항)설정 파일에 저장된 접속정보들에 대해 노출을 방지하고자 암호화 합니다.

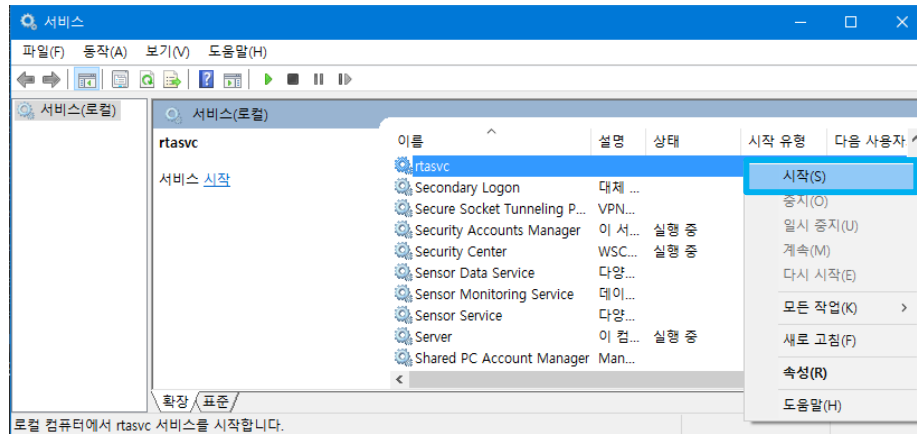


● 서비스 명령

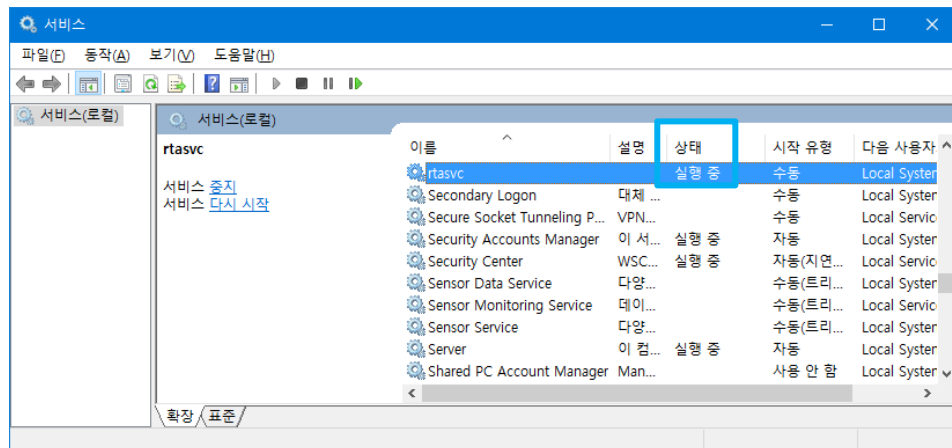
Windows의 서비스(제어판>관리도구>서비스, 실행>services.msc)를 사용합니다.

- 실행(start)
 - rtasvc 서비스에서 오른쪽 버튼 클릭 후 '시작(S)' 을 클릭합니다.

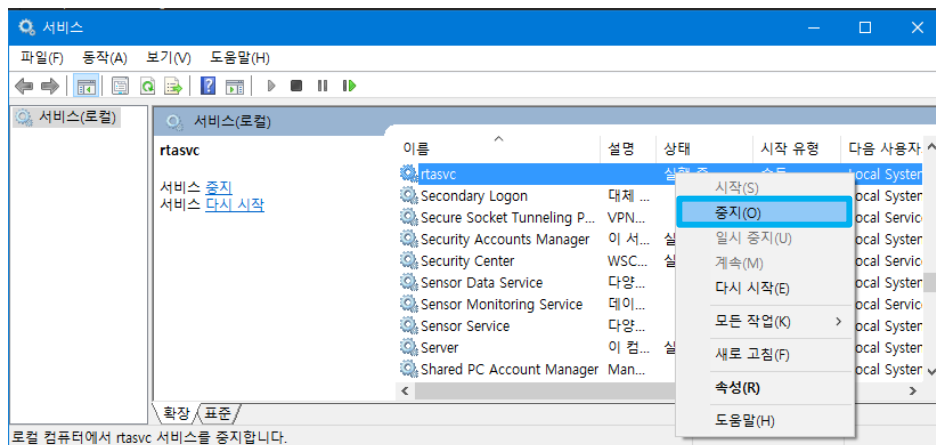
⁷ 재생성을 위해서는 인포뱅크 영업담당자에 요청 해주시기 바랍니다. 재생성이 될 경우 이전 인증서파일은 사용할 수 없습니다.



- 상태 확인(status)
 - rtasvc 서비스의 상태 영역을 확인합니다.
 - 상태가 변경되지 않는 경우 rtasvc 서비스에서 오른쪽 버튼 클릭 후 '새로 고침(F)'을 클릭합니다.



- 종료(stop)
 - rtasvc 서비스에서 오른쪽 버튼 클릭 후 '중지(O)'를 클릭합니다.



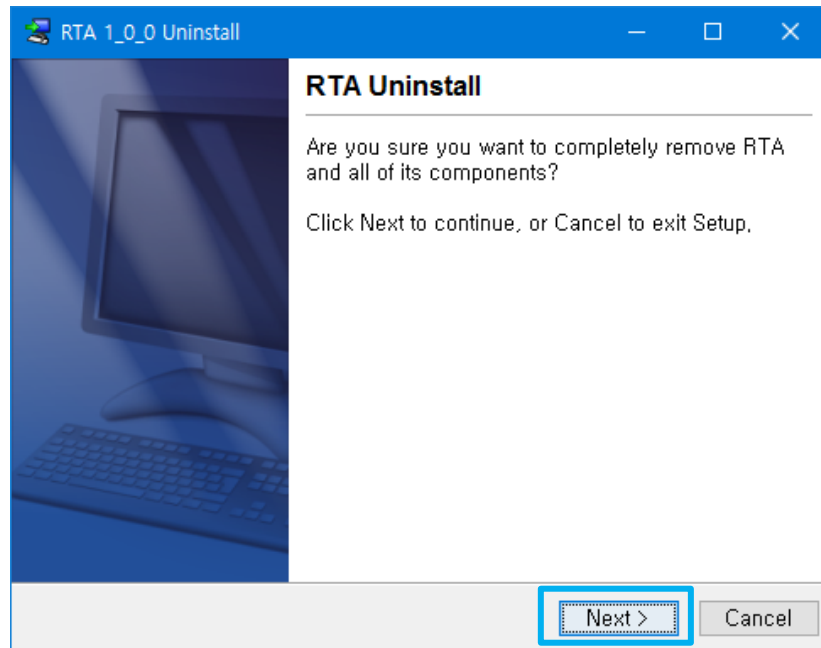
※종료 명령에도 비정상적인 상태로 종료가 되지 않았을 수 있습니다. 따라서 종료 이후엔 상태

영역을 새로고침하여 확인이 필요합니다.

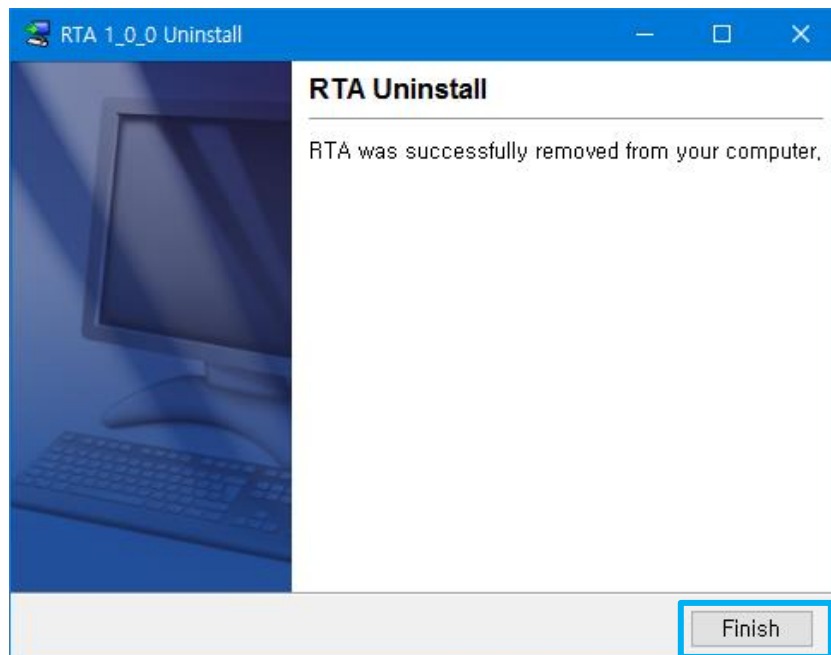
※종료가 지속적으로 되지 않을 경우 Windows의 작업관리자 메뉴를 통하여 '작업 끝내기' 명령으로 종료한 뒤 인포뱅크에 문의 하시기 바랍니다.

- **제거**

- ① 설치된 경로에서 uninstall.exe를 더블클릭하여 실행된 마법사를 따라 삭제 합니다.



- ② 삭제가 진행되고 'Finish' 버튼을 클릭하여 완료합니다.



3.3. 데이터베이스 프로시저

3.3.1. 일반 프로시저(MySQL, MariaDB, MSSQL, PostgreSQL)

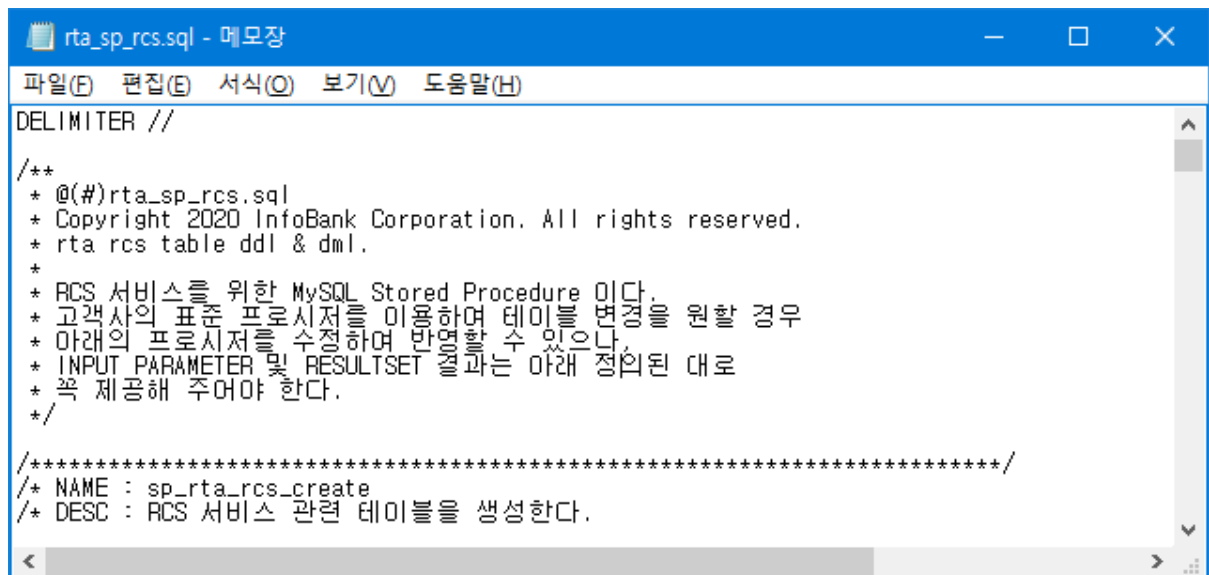
- 생성 SQL 목록

구분	파일명	내용	비고
공통	rta_sp_com.sql	RTA 공통 프로시저	-
RCS발송	rta_sp_rcs.sql	RCS 발송 관련 프로시저	

※MySQL, MariaDB는 rta_sp_rcs.sql에 함수(ss_rta_rcs_get_seq)⁸ 생성 문이 포함되어 있습니다.

- 설치

- ① 텍스트 뷰어를 통해 생성 SQL을 확인 또는 환경에 맞게 수정 후 생성 SQL을 복사합니다.



```
DELIMITER //

/**
 * @(#)rta_sp_rcs.sql
 * Copyright 2020 InfoBank Corporation. All rights reserved.
 * rta rcs table ddl & dml.
 *
 * RCS 서비스를 위한 MySQL Stored Procedure 이다.
 * 고객사의 표준 프로시저를 이용하여 테이블 변경을 원할 경우
 * 아래의 프로시저를 수정하여 반영할 수 있으나,
 * INPUT PARAMETER 및 RESULTSET 결과는 아래 정의된 대로
 * 꼭 제공해 주어야 한다.
 */

/*****
/* NAME : sp_rta_rcs_create
/* DESC : RCS 서비스 관련 테이블을 생성한다.
*****/
```

- ② 복사된 SQL문을 데이터베이스 도구 프로그램 또는 데이터베이스 콘솔 창에 접속하여 실행합니다.

⁸ 함수 기능과 관련하여 자세한 사항은 '[5.3.1 함수](#)'를 참조해 주시기 바랍니다.

3.3.2. Package⁹ 프로시저(Oracle, Tiberio)

- 생성 SQL 목록

구분	파일명	내용	비고
공통	rta_sp_com.pks	RTA 공통 기능(명세)	테이블, 인덱스 스페이스 수정 필요
	rta_sp_com.pkb	RTA 공통 기능(본문)	
RCS발송	rta_sp_rcs.pks	RCS 발송 기능(명세)	
	rta_sp_rcs.pkb	RCS 발송 기능(본문)	

- 설치

- ① 텍스트 뷰어를 통해 생성 SQL 중 공통, RCS발송 패키지 본문 에서 테이블 및 인덱스 스페이스를 수정합니다.

```

, failback_kko_sender_key VARCHAR2(4000)
, etc_text_1 VARCHAR2(100)
, etc_text_2 VARCHAR2(100)
, etc_text_3 VARCHAR2(100)
, etc_num_1 NUMBER
, etc_num_2 NUMBER
, etc_num_3 NUMBER
, etc_date_1 TIMESTAMP
, CONSTRAINT pk_rta_rcs_tran PRIMARY KEY (seq)
using index tablespace TS_IDX_IMDS101
)
tablespace TS_DAT_IMDS101';

EXECUTE IMMEDIATE sql_string ;

/** index create */
sql_string := ' CREATE INDEX ix_rta_rcs_tran_01 ON rta_rcs_tran(msg_status,

```

※기본 테이블 스페이스명은 'TS_DAT_IMDS101', 인덱스 스페이스명은 'TS_IDX_IMDS101' 로 지정되어 있습니다.

- ② 수정된 생성 SQL을 복사하여 데이터베이스 도구 프로그램 또는 데이터베이스 콘솔 창에 접속하여 실행합니다.

※패키지 명세(.pks) 생성 SQL 실행 이후에 패키지 본문(.pkb) 생성 SQL을 실행 합니다.

⁹ 연관성이 있는 프로시저 또는 함수들의 집합입니다.

4. 환경설정

~RTA 설치경로/conf/ 내의 환경설정 파일들을 수정하여 상세 기능 설정 및 인증정보들을 변경할 수 있습니다.

4.1. RTA 설정파일(rta.cf)

- 인증 정보 설정

```
#####
# authentication Configuration Information                                     #
#####

#-----
# Infobank authentication server information
# auth.encrypted(0:Non-encrypted, 1:Encrypted)
#-----

auth.host      = auth.infobank.net      인포뱅크 G/W 인증 서버 주소
auth.port      = 12001                  인포뱅크 G/W 인증 서버 포트
auth.id        = 계정 명                 RTA계정 명
auth.password   = 비밀번호              RTA계정 비밀번호
auth.encrypted  = 0                     계정 정보의 암호화 여부10 (0-미사용, 1-사용)

rcs.agency.id   = infobank              RCS 중계사 ID11
```

- 기능 설정

```
#####
# Process Configuration Information                                           #
#####

#-----
# Whether to use the detailed function(0:Not use ,1:Use)
```

¹⁰ authcipher를 실행하여 접속계정정보 텍스트들이 암호화 되었을 때 '1'로 변경됩니다. 자세한 사항은 '[3.2 RTA 설치](#)'를 참조해 주시기 바랍니다.

¹¹ 이동통신사에 전달되는 RCS메시지 중계사의 ID입니다. 특이사항 없을 경우 변경하지 않습니다.

```

#-----
process.use.rcs.collector    = 1      COLLECTOR기능 사용 유무(0-미사용, 1-사용)
process.use.rcs.sender      = 1      SENDER기능 사용 유무(0-미사용, 1-사용)
process.use.rcs.receiver    = 1      RECEIVER기능 사용 유무(0-미사용, 1-사용)
process.use.rcs.distributor = 1      DISTRIBUTOR기능 사용 유무(0-미사용, 1-사용)
process.use.rcs.trandeleter  = 1      TRANDELETER기능 사용 유무(0-미사용, 1-사용)
process.use.jobscheduler    = 1      JOBSCHEDULER 기능 사용 유무(0-미사용, 1-사용)

#-----
# Set session connection number by RTA process
#-----
# RCSSender session count
session.cnt.rcs.sender      = 2      SENDER기능의 세션 수 설정

# RPTReceiver session count
session.cnt.rcs.receiver    = 2      RECEIVER기능의 세션 수 설정

#-----
# Operation valid time setting
#-----
# Set message resend function in case of timeout(only Collector 0:Send, 1:Not
send)
runtime.collector.overtimefail = 0    COLLECTOR 유희시간으로 중지된 메시지들 이후
                                     재 발송 여부(0:발송, 1:발송안함)

# Collector start time
runtime.collector.start = 00:00      COLLECTOR 기능 시작 시간

# Collector end time
runtime.collector.end = 24:00        COLLECTOR 기능 종료 시간

# Sender start time
runtime.sender.start = 00:00         SENDER 기능 시작 시간

# Sender end time
runtime.sender.end = 24:00           SENDER 기능 종료 시간

```

```

# Tran_Deleter start time
runtime.trandeleter.start= 00:00      TRANDELETER 기능 시작 시간

# Tran_Deleter end time
runtime.trandeleter.end = 24:00      TRANDELETER 기능 종료 시간

# Job Scheduler operating time
runtime.jobscheduler.start = 04:00    JOBSCHEDULER 기능 호출 시간

# User Scheduler operating time
runtime.userscheduler.start = 05:00   USERSCHEDULER 기능 호출 시간

#-----
# Message Sending Priority Use(0:Not use ,1:Use)
#-----
# whether to use 'Very Fast'
priority.use.rcs.veryfast = 0         우선순위 발송(VF) 사용여부(0:미사용, 1:사용)

# whether to use 'Fast'
priority.use.rcs.fast      = 0         우선순위 발송(F) 사용여부(0:미사용, 1:사용)

#-----
# Set priority number of sending message
#-----
priority.cnt.rcs.veryfast = 0  우선순위 별 발송 비율 설정
priority.cnt.rcs.fast     = 0
priority.cnt.rcs.normal   = 1

```

● 파일 큐 관련 설정

```

#####
# RTA Queue Configuration Information                                     #
#####

#-----
# Set transmission queue lookup frequency(second)

```

```
#-----
que.pollinginterval = 2      파일 큐 읽기 주기 설정(초)

#-----
# Queue Name setting
#-----
que.name.rcs.s = send_rcs      SEND 큐 파일명 설정(우선순위 S)
que.name.rcs.f = send_rcs_fast  SEND 큐 파일명 설정(우선순위 F)
que.name.rcs.v = send_rcs_veryfast  SEND 큐 파일명 설정(우선순위 VF)

que.name.rcs.rpt = recv_rcs_rslt  RESULT 큐 파일명 설정
```

- 인포뱅크 G/W 접속 관련 설정

```
#####
# RTA GW Connection Configuration Information                                     #
#####

#-----
# Infobank authentication server connection retries count
#-----
retry.auth.maxcount = 28800      인포뱅크 G/W 인증 실패 시 재시도 수

#-----
# Infobank g/w connection retries count
#-----
retry.mmgp.maxcount = 28800      인포뱅크 G/W 접속 실패 시 재시도 수

#-----
# Retry connection interval(second)
#-----
retry.interval = 10              인포뱅크 G/W 재시도 주기 설정(초)

#-----
# Ping send interval to check connection status(second)
#-----
```

```

ping.interval = 10                인포뱅크 G/W 연결 중 네트워크 상태 확인 주기 설정(초)

#-----
# Completion message tran delete cycle(second)
#-----

trandeleter.interval = 10        TRAN DELETER 기능 주기 설정(초)

#-----
# Job Scheduler operation cycle(minute)
#-----

jobscheduler.interval = 1440     JOB SCHEDULER 기능 주기 설정(분, 1440m=1day)

#-----
# User Scheduler operation cycle(minute)
#-----

userscheduler.interval = 1440   USER SCHEDULER 기능 주기 설정(분, 1440m=1day)

```

4.2. 데이터베이스 관련 설정파일(db.cf)

- 데이터베이스 접속정보 설정

```

#*****
# Database Connection Information
#*****

db.count = 1                접속할 데이터베이스 수12 설정
                             (다수일 경우 하단 DBMS 세팅 시 인덱스(db0, db1) 순차적으로 증가
                             필요)

#-----
# DBMS Setting
# (db0.type: 1-Oracle, 2-MSSQL, 3-MySQL, 5-PostgreSQL, 7-Tibero, 8-Maria)
#-----

db0.type      = 1            데이터베이스 타입 코드
db0.driver    = oracle.jdbc.driver.OracleDriver  데이터베이스 JDBC Driver 클래스

```

¹² RTA는 1개의 AGENT로 다수의 데이터베이스와 접속하여 동작할 수 있습니다.

```
db0.url      = jdbc:oracle:thin:@접속주소:접속포트번호:DB명   데이터베이스 접속 정보
db0.user     = 계정 명                                       데이터베이스 접속 계정
db0.password = 비밀번호                                       데이터베이스 접속 계정 비밀번호
```

● 데이터베이스 동작 관련 설정

```
*****
# Database ETC Information
*****
#-----
# database information encryption(0:Non-encrypted, 1:Encrypted)
#-----
db.encrypted = 0          데이터베이스 접속 정보 암호화 여부13(0-미사용, 1-사용)

#-----
# Send Time to limit Setting(minute)
#-----
db.ttl = 1440             발송 요청 시간 최대 이전 값14 설정(분, 1440m=1day)

#-----
# DB Table(rta_rcs_tran) query interval setting(second)
#-----
db.pollinginterval = 2    발송 대상 조회 주기15

#-----
# Check the privilege(create, drop, delete) to table(0:Not use ,1:Use)
#-----
db.checkprivilege = 1     접속한 데이터베이스 계정에 권한 확인 기능(0:미사용, 1:사용)

#-----
# Stored Procedure Prefix (If RTA procedure is installed in multiple)
#-----
#db.spprefix = e1_        데이터베이스 프로시저 접두어 설정
```

¹³ dbcipher를 실행하여 데이터베이스 접속 정보 텍스트들이 암호화 되었을 때 '1'로 변경됩니다. 자세한 사항은 ['3.2 RTA 설치'](#)를 참조해 주시기 바랍니다.

¹⁴ 설정 값이 클수록 이전에 발송 요청한 데이터들이 발송 대상에 포함됩니다.

¹⁵ 발송이 없을 때는 해당 주기만큼 조회하며 발송데이터가 있을 경우 주기 없이 바로 재 조회합니다.

```

#-----
# Spam blocking Setting
#-----
# DB Table(rta_banlist)(0:Not use, 1:Use)
#db.bancheck = 0          발송 차단 기능 사용 여부16(0:미사용, 1:사용)

# DB Table(rta_banlist) query interval setting(second)
db.bancheck.interval = 60  발송 차단 데이터 조회 주기 설정(초)

#-----
# Set query execution timeout(oracle, mssql, sybase)
#-----
# whether to use(0:Not use, 1:Use)
db.querytimeoutcheck = 0   데이터베이스 작업 타임아웃 설정 여부(0:미사용, 1:사용)

# Query timeout settings(second)
db.querytimeout = 10       데이터베이스 작업 타임아웃 시간 설정(초)

#-----
# Set Validity Query
#-----
# whether to use(0:Not use, 1:Use)
db.validationquerycheck = 1 데이터베이스 세션 유효성 체크 사용 여부(0:미사용, 1:사용)

# validation query interval setting(second)
db.validationqueryinterval = 5 데이터베이스 세션 유효성 체크 주기 설정(초)

```

● 데이터베이스 프로시저 명 설정

```

#-----
# Oracle, Tiberio Package Name  데이터베이스 패키지 명 설정(Oracle, Tiberio)
#-----
pkgname.com      = rta_pkg_com
pkgname.rcs      = rta_pkg_rcs

```

¹⁶ 발송 차단 수신번호는 해당 사용여부와 상관 없이 발송 대상 조회 시 조건에 포함됩니다.

```
#-----
# Common Stored Procedure Name 데이터베이스 공통 프로시저 명 설정
#-----
spname.common_create          =sp_rta_common_create
spname.common_banlist         =sp_rta_common_banlist
spname.common_checkprivilege  =sp_rta_common_checkprivilege
#-----
# RCS Service Stored Procedure Name 데이터베이스 RCS발송 프로시저 명 설정
#-----
spname.rcs_create              =sp_rta_rcs_create
spname.rcs_log_create          =sp_rta_rcs_log_create
spname.rcs_tran_select         =sp_rta_rcs_tran_select
spname.rcs_update              =sp_rta_rcs_update
spname.rcs_tran_rslt_update    =sp_rta_rcs_tran_rslt_update
spname.rcs_failover_mt_rslt_update =sp_rta_failover_mt_rslt_update
spname.rcs_tran_rslt_delete    =sp_rta_rcs_tran_rslt_delete
spname.rcs_tran_log_move_past  =sp_rta_rcs_tran_log_move_past
```

4.3. LOG 설정파일(logback.xml)

RTA는 어플리케이션에서 발생하는 여러 가지 상황들을 로그를 통해 알려줍니다. RTA는 기본적으로 자바 오픈 소스 로깅 프레임워크인 logback을 사용합니다. logback.xml파일의 기본 설정을 변경하고자 할 경우, logback 홈페이지(<https://logback.qos.ch/>)를 참조하시기 바랍니다.

- LOG level 설정

- 일반 설정

```
<root level="INFO"> #root level = INFO로 설정
  <appender-ref ref="IBCON" />
  <appender-ref ref="IBINFO" />
</root>
<!-- CONSOLE appender --> Console appender, INFO appender 만 주석해제 처리
<appender name="IBCON" class="ch.qos.logback.core.ConsoleAppender">
  ....생략
</appender>
<!-- INFO appender -->
```

```
<appender name="IBINFO" class="ch.qos.logback.core.rolling.RollingFileAppender">
    ....생략
</appender>
```

- DEBUG 설정

```
<root level="DEBUG"> root level = DEBUG 설정
    <appender-ref ref="IBCON" />
    <appender-ref ref="IBINFO" />
    <appender-ref ref="IBDEBUG"/>
</root>

<!-- INFO level 주식 처리
<root level="INFO">
    <appender-ref ref="IBCON" />
    <appender-ref ref="IBINFO" />
</root>
-->

<!-- #DEBUG appender(When debug mode) --> DEBUG appender 주식해제 처리
<appender name="IBDEBUG" class="ch.qos.logback.core.rolling.RollingFileAppender">
    ....생략
</appender>
```

● LOG 파일 삭제 주기 및 압축 설정

기본적으로 .gz 형식으로 로그파일을 일단위로 압축해주고 있으며, 로그 파일 삭제 주기는 730(2년)으로 설정되어 있습니다. 압축 파일 형식 및 로그 파일 삭제 주기는 logback.xml 파일 내 아래 설정을 통해 변경 가능합니다.

```
<rollingPolicy class="ch.qos.logback.core.rolling.TimeBasedRollingPolicy">
    <!-- rollover daily -->
    <fileNamePattern> ./logs/rta.info.log.%d{yyyy-MM-dd}.gz</fileNamePattern>
    <maxHistory>730</maxHistory> fileNamePattern에 따라 일단위 삭제, 분단위 삭제, 초단위 삭제
</rollingPolicy>
```

5. 데이터베이스

5.1. 테이블 구성 및 명세서

5.1.1. 테이블 구성

구분	테이블 명	설명
공통	rta_banlist	발송 차단 정보(수신번호/메시지 내용)를 등록할 수 있는 테이블입니다.
RCS	rta_rcs_tran	RCS 메시지 전송 테이블 입니다.
	rta_rcs_log_yyyymm	RCS 메시지 월별 로그 테이블 입니다.

5.1.2. 테이블 명세서(MySQL)

● rta_banlist (수신거부 테이블)

Field	Type	Null	Default	Description
ban_seq*	int(11)	NN		순번
ban_type	char(1)	NN		스팸 차단 구분 (수신 번호, 회신 번호, 메시지 내용) ¹⁷
content	varchar(45)	NN		스팸 번호
send_yn	char(1)	NN	'N'	스팸 문구 전송 여부
ban_status_yn	char(1)	NN	'Y'	전송 차단 여부
reg_date	datetime	NN	sysdate()	등록 시간
reg_user	varchar(20)			등록 자
update_date	datetime	NN	'1970-01-01 00:00:00'	수정 시간
update_user	varchar(20)			수정 자

● rta_rcs_tran (RCS 발송 테이블)

Field	Type	Null	Default	Description
seq*	bigint	NN		메시지 고유 아이디, auto_increment ¹⁸

¹⁷ R: 수신 번호, C: 회신 번호, T: 메시지 내용

¹⁸ 필드 자동 증가 값으로 지원하지 않는 데이터베이스는 시퀀스를 사용하여 입력합니다. 자세한 내용은 '[5.3.1 시퀀스](#)'를 참조해 주시기 바랍니다.

refkey	varchar(20)			부서 코드 (참조용 필드) ¹⁹
priority	char(2)	NN	'S'	메시지 우선 순위 ²⁰
subject	varchar(1000)	NN		메시지 제목
content	text	NN		전송 메시지
callback	varchar(25)	NN		발신자 전화 번호(회신 번호)
service_type	char(2)	NN		서비스 메시지 전송 타입
msg_type	int(11)			메시지 종류
msg_status	char(1)	NN	'1'	메시지 상태 ²¹
country_code	varchar(8)	NN	'82'	국가 코드
recipient_num	varchar(25)			수신자 전화 번호
date_client_req	datetime	NN	'1970-01-01 00:00:00'	전송 예약 시간
date_sent	datetime			인포뱅크 G/W 접수 시간
date_rslt	datetime			이통사가 수신자번호로 전송한 시간
date_report	datetime			인포뱅크로부터 결과 수신한 시간
report_code	varchar(10)			전송 결과 ²²
carrier	int(5)			착신망 정보 ²³
rs_id	varchar(20)			전송된 인포뱅크 G/W 정보
agent_id	char(2)		' '	AGENT 이중화시 사용되는 ID
reg_date	timestamp ²⁴		current_timestamp	데이터 등록일자
rsc_group_id	varchar(32)			RCS Group ID ²⁵
rsc_format_id	varchar(40)	NN		RCS 메시지포맷 ID(템플릿코드) ²⁶
rsc_expiryoption	char(1)		'1'	RCS 재시도 방식 ²⁷
rsc_copyallowed	char(1)		'Y'	RCS 복사 ²⁸ 가능 여부
rsc_header	char(1)	NN	'1'	RCS 상단 '광고' 표출 여부
rsc_footer	varchar(10)			RCS 수신거부 번호
rsc_btn_info	varchar(4000)			RCS 버튼정보
rsc_img_url	varchar(1000)			RCS 이미지 URL

¹⁹ 정산을 구별하기 위해 사용하는 필드입니다. 특이사항이 없을 경우 값을 넣지 않습니다. G/W전달되지 않는 참조 용 필드는 전용 참조 필드(etc)를 사용합니다.

²⁰ 우선 순위 발송을 위한 필드로 자세한 설명은 '7.4 우선 순위 발송'을 참조해 주시기 바랍니다.

²¹ 상태 코드에 대해서는 '6.2 메시지 상태 구분' 을 참조해 주시기 바랍니다.

²² 발송 결과 코드는 '9. 발송 결과 코드표' 를 참조해 주시기 바랍니다.

²³ SKT:20001, KT:20002, LGU:20003, ETC:20000

²⁴ mysql(mariaDB) 호환성을 고려한(특정 버전 이상부터 datetime 타입에 default값 설정 가능) default값 설정을 위해 timestamp로 선언되었습니다.

²⁵ 이동통신사에서 특정 메시지 집합을 구별하기 위한 값입니다. 특정 메시지 집합의 분석이 필요한 경우 사용합니다.

²⁶ RCS 메시지 포맷 ID는 '8. RCS 코드표' 를 참조해 주시기 바랍니다.

²⁷ 결과 수신까지의 timeout 설정 옵션입니다. (1:3일간 시도 후 타임아웃, 2:10초간 시도 후 타임아웃)

²⁸ 스마트폰 메시지 어플리케이션에서 해당 메시지 내용의 복사 기능 활성화 여부입니다.

rsc_brand_id	varchar(64)			RCS 브랜드 ID
failover_yn	char(1)	NN	'N'	Failover 발송 여부
failover_service_type	char(2)			Failover 데이터 (공통-메시지 전송 타입)
failover_subject	varchar(40)			Failover 데이터 (공통-메시지 제목)
failover_content	varchar(4000)			Failover 데이터 (공통-메시지 내용)
failover_img	varchar(200)			Failover 데이터 (공통-이미지 URL/파일 키)
failover_prefix	varchar(40)			Failover 처리 시 메시지 내용 앞에 Prefix 추가
failover_timeout_send	char(1)		'1'	Failover 전송 후 timeout 발생 시 failover 발송 여부
failover_date_sent	datetime			Failover 인포뱅크 G/W 접수 시간
failover_date_rslt	datetime			Failover 이통사가 수신자번호로 전송한 시간
failover_date_report	datetime	NN		Failover 인포뱅크로부터 결과 수신한 시간
failover_report_code	varchar(10)			Failover 전송 결과
failover_carrier	int(5)			Failover 착신망 정보
etc_text_1	varchar(100)			유저 기타필드 ²⁹
etc_text_2	varchar(100)			유저 기타필드
etc_text_3	varchar(100)			유저 기타필드
etc_num_1	int(11)			유저 기타필드
etc_num_2	int(11)			유저 기타필드
etc_num_3	int(11)			유저 기타필드
etc_date_1	datetime			유저 기타필드
rsc_brand_key	varchar(64)			RCS 브랜드키
rsc_agency_key	Varchar(100)			RCS 대행사키

● rta_rsc_log_yyyymm (RCS 로그 테이블)

- 발송 테이블(rta_rsc_tran)에서 전송 완료된(msg_status = '3') 데이터들에 대해 해당 테이블로 이관되므로 발송테이블과 동일한 구조입니다. 기본은 발송 테이블 요청 시간(date_client_req)

²⁹ 사용자 편의를 위해 생성해 놓은 필드 입니다.

에 따라 월별 테이블로 생성되어 이관됩니다.

Field	Type	Null	Default	Description
seq*	bigint	NN		메시지 고유 아이디, auto_increment
refkey	varchar(20)			부서 코드 (참조용 필드)
priority	char(2)	NN	'S'	메시지 우선 순위 ³⁰
subject	varchar(1000)	NN		메시지 제목
content	text	NN		전송 메시지
callback	varchar(25)	NN		발신자 전화 번호(회신 번호)
service_type	char(2)	NN		서비스 메시지 전송 타입
msg_type	int(11)			메시지 종류
msg_status	char(1)	NN	'1'	메시지 상태
country_code	varchar(8)	NN	'82'	국가 코드
recipient_num	varchar(25)	NN		수신자 전화 번호
date_client_req	datetime	NN	'1970-01-01 00:00:00'	전송 예약 시간
date_sent	datetime			인포뱅크 G/W 접수 시간
date_rslt	datetime			이통사가 수신자번호로 전송한 시간
date_report	datetime			인포뱅크로부터 결과 수신한 시간
report_code	varchar(10)			전송 결과
carrier	int(5)			착신망 정보
rs_id	varchar(20)			전송된 인포뱅크 G/W 정보
agent_id	char(2)		' '	AGENT 이중화시 사용되는 ID
reg_date_tran	datetime			tran 테이블 데이터 등록일자
reg_date	timestamp		current_timestamp	log 테이블 데이터 등록일자
rsc_group_id	varchar(32)			RCS Group ID
rsc_format_id	varchar(40)	NN		RCS 메시지포맷 ID(템플릿코 드)
rsc_expiryoption	char(1)		'1'	RCS 재시도 방식
rsc_copyallowed	char(1)		'Y'	RCS 복사가능 여부
rsc_header	char(1)	NN	'1'	RCS 상단 '광고' 표출 여부
rsc_footer	varchar(10)			RCS 수신거부 번호
rsc_btn_info	varchar(4000)			RCS 버튼정보

³⁰ 우선 순위 발송을 위한 필드로 자세한 설명은 ['7.4 우선 순위 발송'](#)을 참조해 주시기 바랍니다.

rcs_img_url	varchar(1000)			RCS 이미지 URL
rcs_brand_id	varchar(64)			RCS 브랜드 ID
failover_yn	char(1)	NN	'N'	Failover 발송 여부
failover_service_type	char(2)			Failover 데이터 (공통-메시지 전송 타입)
failover_subject	varchar(40)			Failover 데이터 (공통-메시지 제목)
failover_content	varchar(4000)			Failover 데이터 (공통-메시지 내용)
failover_img	varchar(200)			Failover 데이터 (공통-이미지 URL/파일 키)
failover_prefix	varchar(40)			Failover 처리 시 메시지 내용 앞에 Prefix 추가
failover_timeout_sen d	char(1)		'1'	Failover 전송 후 timeout 발생 시 failover 발송 여부
failover_date_sent	datetime			Failover 인포뱅크 G/W 접수 시간
failover_date_rslt	datetime			Failover 이통사가 수신자번호로 전송한 시간
failover_date_report	datetime			Failover 인포뱅크로부터 결과 수신한 시간
failover_report_code	varchar(10)			Failover 전송 결과
failover_carrier	int(5)			Failover 착신망 정보
etc_text_1	varchar(100)			유저 기타필드
etc_text_2	varchar(100)			유저 기타필드
etc_text_3	varchar(100)			유저 기타필드
etc_num_1	int(11)			유저 기타필드
etc_num_2	int(11)			유저 기타필드
etc_num_3	int(11)			유저 기타필드
etc_date_1	datetime			유저 기타필드
rcs_brand_key	varchar(64)			RCS 브랜드키
rcs_agency_key	Varchar(100)			RCS 대행사키

5.2. 프로시저 구성

구분	프로시저 명	설명
----	--------	----

공통 (rta_pkg_com)	sp_rta_common_create	공통으로 사용하는 테이블을 생성합니다.
	sp_rta_common_banlist	발송 차단 정보를 조회합니다.
	sp_rta_common_checkprivilege	테이블 생성 (조회, 수정, 삭제 포함) 권한을 체크합니다.
RCS (rta_pkg_rcs)	sp_rta_rcs_create	RCS 서비스 관련 테이블을 생성합니다.
	sp_rta_rcs_log_create	RCS 서비스에서 로그테이블을 사용할 경우 로그 테이블을 생성합니다.
	sp_rta_rcs_tran_select	RCS 메시지 전송 데이터를 조회합니다.
	sp_rta_rcs_update	RCS 메시지를 큐에 적재한 후 적재 결과를 갱신합니다.
	sp_rta_rcs_tran_rslt_update	RCS 메시지 전송 결과를 갱신합니다.
	sp_rta_failover_mt_rslt_update	Failover 메시지 전송 결과를 갱신합니다.
	sp_rta_rcs_tran_rslt_delete	로그 테이블로 이동된 데이터를 삭제합니다. (MySQL, MariaDB만 해당)
	sp_rta_rcs_tran_log_move_past	유효기간이 지난 메시지를 로그 테이블로 이동합니다.

※Oracle, Tiberio는 프로시저가 패키지로 묶여 있습니다.

※Oracle, Tiberio는 sp_rta_rcs_create에 시퀀스(sq_rta_rcs_tran_01) 생성 문이 포함되어 있습니다.

5.3. 기타 객체

5.3.1. 시퀀스(Oracle, Tiberio, PostgreSQL)

RCS 발송테이블의 키 필드인 seq에 중복을 방지하기 위해 사용됩니다. 자동증가값을 제공하지 않는 데이터베이스(Oracle, Tiberio, PostgreSQL)에 해당되며 발송 요청 데이터 입력 시 시퀀스 값을 증가시킨 순번 값으로 입력이 필요합니다.

구분	시퀀스 명	설명
RCS	sq_rta_rcs_tran_01	nextval 을 이용하여 다음 순번 반환

5.3.2. 함수(MySQL, MariaDB)

RCS 발송테이블의 키 필드 자동 증가 값(auto_increment) 기능은 데이터베이스 서버가 재 시작 될 경우 값을 초기화 합니다. 따라서 이를 방지하기 위해 발송테이블에 최종 순번 데이터를 남기도록 이관 조건에 사용되는 함수 입니다.

구분	함수 명	설명
RCS	ss_rta_rcs_get_seq	RCS 최종 순번을 리턴

6. 발송 서비스

6.1. RCS 메시지 종류

- **RCS SMS (Rich Communication Services Short Message Service)**
100자 이하, 최대 버튼 1개의 RCS메시지를 전송하는 서비스
- **RCS LMS (Rich Communication Services Long Message Service)**
1,300자 이하, 최대 버튼 3개의 RCS메시지를 전송하는 서비스
- **RCS MMS (Rich Communication Services Multimedia Message Service)**
1,300자 이하, 최대 버튼 2개, 이미지가 포함된 RCS메시지를 전송하는 서비스
- **RCS MMS (Carousel)**
1,300자 이하, 최대 카드 6개, 카드 별 최대 버튼 2개의 다중 카드의 RCS 메시지를 전송하는 서비스
- **RCS 템플릿(Template)**
사전 등록된 지정된 형태의 RCS 메시지를 전송하는 서비스(버튼 등록 포함)
 - Free 템플릿: RCS Biz Center를 통한 계정 생성 시 바로 사용 가능한 템플릿
 - Description 템플릿: 등록된 형태에 맞게 전체 메시지내용을 기술하여 요청하는 템플릿
 - Cell 템플릿: 등록된 블록 구조에 맞게 key, value를 JSON형태로 요청하는 템플릿
 - Image 템플릿: 광고 혹은 정보성 메시지를 이미지를 포함하여 구성한 템플릿

RCS						
구분	일반				템플릿	
메시지 종류	SMS	MMS	MMS	Carousel	Free, Description, Image	Cell
서비스 타입	20	23	22	26	24	25

※자세한 사양은 '[8. RCS 코드표](#)' 를 참조해주시기 바랍니다.

6.2. 메시지 상태 구분

RTA가 일련의 작업 중 메시지 상태 값을 다음과 같이 구분합니다.

메시지 상태					
구분	발송 요청	발송 중	결과 수신	Failover 결과 수신	처리 완료
코드 값	1	2	3	4	5
설명	최초 등록	파일 큐에 적재 상태	발송 결과가 수신된 상태	Failover 발송 결과가 수신된 상태	이관 대상 또는 이관 완료

6.3. RCS 메시지 전송

- RCS를 보내기 위해서는 '전송예약시간', '수신 번호', '발신번호', '메시지 제목', '메시지 내용', '메시지 상태', 'RCS포맷ID'등의 입력이 필요합니다.
- RCS MMS는 이미지파일을 직접 첨부하여 발송하지 않고 사전에 별도 RCS 이미지파일업로드 API를 통해 발급한 img_url 값을 사용하여 발송합니다. img_url을 생성하기 위해서는 '[6.4 RCS 이미지 파일업로드\(FILE API\)](#)' 를 참조해주시기 바랍니다.
- 일부 RCS메시지(RCS MMS(Carousel), 템플릿) 및 버튼이 포함된 RCS 전송을 위해 JSON형태로 입력이 필요합니다.
- RCS포맷 ID는 '[8.RCS 코드표](#)' 를 참조해주시기 바랍니다.

● RCS SMS 예제

▶ Oracle

```
insert into rta_rcs_tran (seq, date_client_req, subject, content, callback, service_type,
rsc_format_id, msg_status, recipient_num, rcs_header) values(sq_rta_rcs_tran_01.nextval, sysdate,
'Test 제목', 'Test Message 입니다', '01000000000', '20', 'SS000000', '1', '01000000000', '0');
```

▶ MSSQL

```
insert into rta_rcs_tran (date_client_req, subject, content, callback, service_type, rcs_format_id,
msg_status, recipient_num, rcs_header) values(getdate(), 'Test 제목', 'Test Message 입니다',
'01000000000', '20', 'SS000000', '1', '01000000000', '0');
```

▶ MySQL/MariaDB

```
insert into rta_rcs_tran (date_client_req, subject, content, callback, service_type, rcs_format_id,
msg_status, recipient_num, rcs_header) values(sysdate(), 'Test 제목', 'Test Message 입니다',
'01000000000', '20', 'SS000000', '1', '01000000000', '0');
```

● RCS LMS 예제

▶ Oracle

```
insert into rta_rcs_tran (seq, date_client_req, subject, content, callback, service_type,
rsc_format_id, msg_status, recipient_num, rcs_header, rcs_btn_info)
values(sq_rta_rcs_tran_01.nextval, sysdate, 'Test 제목', 'Test Message 입니다', '01000000000', '23',
```

```
' SL000000', '1', '01000000000', '0', '[{"suggestions": [{"action": {"urlAction": {"openUrl": {"url": "URL정보"}}, "displayText": "버튼명", "postback": {"data": "set_by_chatbot_open_url"}}, {"action": {"composeAction": {"composeTextMessage": {"phoneNumber": "01000000000", "text": "테스트"}}, "displayText": "버튼명", "postback": {"data": "set_by_chatbot_compose_text_message"} } } ]}];
```

▶ MSSQL

```
insert into rta_rcs_tran (date_client_req, subject, content, callback, service_type, rcs_format_id, msg_status, recipient_num, rcs_header, rcs_btn_info) values(getdate(), 'Test 제목', 'Test Message 입니다', '01000000000', '23', ' SL000000', '1', '01000000000', '0', '[{"suggestions": [{"action": {"urlAction": {"openUrl": {"url": "URL정보"}}, "displayText": "버튼명", "postback": {"data": "set_by_chatbot_open_url"}}, {"action": {"composeAction": {"composeTextMessage": {"phoneNumber": "01000000000", "text": "테스트"}}, "displayText": "버튼명", "postback": {"data": "set_by_chatbot_compose_text_message"} } } ]}];
```

▶ MySQL/MariaDB

```
insert into rta_rcs_tran (date_client_req, subject, content, callback, service_type, rcs_format_id, msg_status, recipient_num, rcs_header, rcs_btn_info) values(sysdate(), 'Test 제목', 'Test Message 입니다', '01000000000', '23', ' SL000000', '1', '01000000000', '0', '[{"suggestions": [{"action": {"urlAction": {"openUrl": {"url": "URL정보"}}, "displayText": "버튼명", "postback": {"data": "set_by_chatbot_open_url"}}, {"action": {"composeAction": {"composeTextMessage": {"phoneNumber": "01000000000", "text": "테스트"}}, "displayText": "버튼명", "postback": {"data": "set_by_chatbot_compose_text_message"} } } ]}];
```

● RCS MMS 예제

▶ Oracle

```
insert into rta_rcs_tran (seq, date_client_req, subject, content, callback, service_type, rcs_format_id, msg_status, recipient_num, rcs_header, rcs_img_url) values(sq_rta_rcs_tran_01.nextval, sysdate, 'Test 제목', 'Test Message 입니다', '01000000000', '22', 'SMwThT00', '1', '01000000000', '0', 'maapfile://rest_srv_test_951eb68af1894ed98a66f0226fed5f20');
```

▶ MSSQL

```
insert into rta_rcs_tran (date_client_req, subject, content, callback, service_type, rcs_format_id, msg_status, recipient_num, rcs_header, rcs_img_url) values(getdate(), 'Test 제목', 'Test Message 입니다', '01000000000', '22', 'SMwThT00', '1', '01000000000', '0', 'maapfile://rest_srv_test_951eb68af1894ed98a66f0226fed5f20');
```

▶ MySQL/MariaDB

```
insert into rta_rcs_tran (date_client_req, subject, content, callback, service_type, rcs_format_id,
msg_status, recipient_num, rcs_header, rcs_img_url) values(sysdate(), 'Test 제목', 'Test Message 입니다', '01000000000', '22', 'SMwThT00', '1', '01000000000', '0',
'maapfile://rest_srv_test_951eb68af1894ed98a66f0226fed5f20');
```

● RCS MMS(Carousel) 예제

※RCS Carousel에 대한 JSON 형태는 '[6.6 RCS Carousel 구조](#)' 를 참조하시기 바랍니다.

▶ Oracle

```
insert into rta_rcs_tran(seq, date_client_req, subject, content, callback, service_type, msg_status,
recipient_num, rcs_format_id, rcs_header, rcs_img_url, rcs_btn_info)
values(sq_rta_rcs_tran_01.nextval, sysdate, '{"subject":["제목 입니다.", "제목2 입니다.", "제목3 입니
다."]}', '{"content":["메시지 내용 입니다.", "메시지 내용 2 입니다.", "메시지 내용 3 입니다."]}',
'01000000000', '26', '1', '01000000000', 'CMwMhM0300',
'0', '{"img_url":["maapfile://rest_srv_test_951eb68af1894ed98a66f0226fed5f20", "", "maapfile://rest_sr
v_test_951eb68af1894ed98a66f0226fed5f20"]}'
, '{"suggestions": [{"action": {"urlAction": {"openUrl": {"url": "URL정보"}}, "displayText": "버튼명
", "postback": {"data": "set_by_chatbot_open_url"}}, {"action": {"composeAction":
{"composeTextMessage": {"phoneNumber": "01000000000", "text": "테스트"}, "displayText": "버튼
명", "postback": {"data": "set_by_chatbot_compose_text_message"} } } } ],
{"suggestions": [{"action": {"urlAction": {"openUrl": {"url": "URL정보"}}, "displayText": "버튼명
", "postback": {"data": "set_by_chatbot_open_url"}}, {"action": {"composeAction":
{"composeTextMessage": {"phoneNumber": "01000000000", "text": "테스트"}, "displayText": "버튼
명", "postback": {"data": "set_by_chatbot_compose_text_message"} } } } ],
{"suggestions": [{"action": {"urlAction": {"openUrl": {"url": "URL정보"}}, "displayText": "버튼명
", "postback": {"data": "set_by_chatbot_open_url"}}, {"action": {"composeAction":
{"composeTextMessage": {"phoneNumber": "01000000000", "text": "테스트"}, "displayText": "버튼
명", "postback": {"data": "set_by_chatbot_compose_text_message"} } } } ]}');
```

▶ MSSQL

```
insert into rta_rcs_tran(date_client_req, subject, content, callback, service_type, msg_status,
recipient_num, rcs_format_id, rcs_header, rcs_img_url, rcs_btn_info) values(getdate(), '{"subject":["
제목 입니다.", "제목2 입니다.", "제목3 입니다."]}', '{"content":["메시지 내용 입니다.", "메시지 내용
2 입니다.", "메시지 내용 3 입니다."]}', '01000000000', '26', '1', '01000000000', 'CMwMhM0300',
'0', '{"img_url":["maapfile://rest_srv_test_951eb68af1894ed98a66f0226fed5f20", "", "maapfile://rest_sr
v_test_951eb68af1894ed98a66f0226fed5f20"]}'
, '{"suggestions": [{"action": {"urlAction": {"openUrl": {"url": "URL정보"}}, "displayText": "버튼명
", "postback": {"data": "set_by_chatbot_open_url"}}, {"action": {"composeAction":
{"composeTextMessage": {"phoneNumber": "01000000000", "text": "테스트"}, "displayText": "버튼
명", "postback": {"data": "set_by_chatbot_compose_text_message"} } } } ],
{"suggestions": [{"action": {"urlAction": {"openUrl": {"url": "URL정보"}}, "displayText": "버튼명
", "postback": {"data": "set_by_chatbot_open_url"}}, {"action": {"composeAction":
{"composeTextMessage": {"phoneNumber": "01000000000", "text": "테스트"}, "displayText": "버튼
명", "postback": {"data": "set_by_chatbot_compose_text_message"} } } } ]}';
```

```

{"suggestions": [{"action": {"urlAction": {"openUrl": {"url": "URL정보"}}, "displayText": "버튼명"}, {"postback": {"data": "set_by_chatbot_open_url"}}, {"action": {"composeAction": {"composeTextMessage": {"phoneNumber": "01000000000", "text": "테스트"}, "displayText": "버튼명"}, {"postback": {"data": "set_by_chatbot_compose_text_message"} } } ]}];

```

▶ MySQL/MariaDB

```

insert into rta_rcs_tran(date_client_req, subject, content, callback, service_type, msg_status,
recipient_num, rcs_format_id, rcs_header, rcs_img_url, rcs_btn_info) values(sysdate(), '{"subject":["
제목 입니다.", "제목2 입니다.", "제목3 입니다."]}', '{"content":["메시지 내용 입니다.", "메시지 내용
2 입니다.", "메시지 내용 3 입니다."]}', '01000000000', '26', '1', '01000000000', 'CMwMhM0300',
'0', '{"img_url":["maapfile://rest_srv_test_951eb68af1894ed98a66f0226fed5f20"], "", "maapfile://rest_sr
v_test_951eb68af1894ed98a66f0226fed5f20"}'
, '{"suggestions": [{"action": {"urlAction": {"openUrl": {"url": "URL정보"}}, "displayText": "버튼명"}, {"postback": {"data": "set_by_chatbot_open_url"}}, {"action": {"composeAction": {"composeTextMessage": {"phoneNumber": "01000000000", "text": "테스트"}, "displayText": "버튼명"}, {"postback": {"data": "set_by_chatbot_compose_text_message"} } } ]} },
{"suggestions": [{"action": {"urlAction": {"openUrl": {"url": "URL정보"}}, "displayText": "버튼명"}, {"postback": {"data": "set_by_chatbot_open_url"}}, {"action": {"composeAction": {"composeTextMessage": {"phoneNumber": "01000000000", "text": "테스트"}, "displayText": "버튼명"}, {"postback": {"data": "set_by_chatbot_compose_text_message"} } } ]} },
{"suggestions": [{"action": {"urlAction": {"openUrl": {"url": "URL정보"}}, "displayText": "버튼명"}, {"postback": {"data": "set_by_chatbot_open_url"}}, {"action": {"composeAction": {"composeTextMessage": {"phoneNumber": "01000000000", "text": "테스트"}, "displayText": "버튼명"}, {"postback": {"data": "set_by_chatbot_compose_text_message"} } } ]}];

```

● RCS 템플릿(free, description) 예제.

※사전에 등록된 템플릿에서 문자메시지 내용은 description을 사용하고, 변수형 버튼 나머지 사용자 정의 key와 value로 구성된 JSON형태로 content작성이 필요합니다.

※free는 버튼을 사용하지 않습니다.

※버튼의 key/value 정보도 content에 작성하며, rcs_btn_info는 사용하지 않습니다.

※ [6.7 Template](#) 구조 참조 바랍니다.

▶ Oracle

```

insert into rta_rcs_tran(seq, date_client_req, content, callback, service_type, msg_status,
recipient_num, rcs_format_id, rcs_header) values(sq_rta_rcs_tran_01.nextval, sysdate,
'{"description":"템플릿(프리)메시지입니다."}', '01000000000', '24', '1', '01000000000',
'UBR.JSUUE931f5-GG000F', '0');

```

▶ MSSQL

```
insert into rta_rcs_tran(date_client_req, content, callback, service_type, msg_status,
recipient_num, rcs_format_id, rcs_header) values(getdate(), '{"description": "템플릿(프리)메시지입니다."}', '01000000000', '24', '1', '01000000000', 'UBR.JSUUE931f5-GG000F', '0');
```

▶ MySQL/MariaDB

```
insert into rta_rcs_tran(date_client_req, content, callback, service_type, msg_status,
recipient_num, rcs_format_id, rcs_header) values(sysdate(), '{"description": "템플릿(프리)메시지입니다."}', '01000000000', '24', '1', '01000000000', 'UBR.JSUUE931f5-GG000F', '0');
```

● RCS 템플릿(cell) 예제.

※사전에 등록된 템플릿의 key와 value로 구성된 JSON형태로 content작성이 필요합니다.

※버튼의 key/value 정보도 content에 작성하며, rcs_btn_info는 사용하지 않습니다.

※ [6.7 Template](#) 구조 참조 바랍니다.

▶ Oracle

```
insert into rta_rcs_tran(seq, date_client_req, content, callback, service_type, msg_status,
recipient_num, rcs_format_id, rcs_header)
values(sq_rta_rcs_tran_01.nextval, sysdate, '{"customName": "고객명", "productName": "상품명",
"date": "2020년 1월 9일"}', '01000000000', '25', '1', '01000000000', 'UBR.JSUUE931f5-
ProductOrder', '0');
```

▶ MSSQL

```
insert into rta_rcs_tran(date_client_req, content, callback, service_type, msg_status,
recipient_num, rcs_format_id, rcs_header)
values(getdate(), '{"customName": "고객명", "productName": "상품명", "date": "2020년 1월 9일"}',
'01000000000', '25', '1', '01000000000', 'UBR.JSUUE931f5-ProductOrder', '0');
```

▶ MySQL/MariaDB

```
insert into rta_rcs_tran(date_client_req, content, callback, service_type, msg_status,
recipient_num, rcs_format_id, rcs_header)
values(sysdate(), '{"customName": "고객명", "productName": "상품명", "date": "2020년 1월 9일"}',
'01000000000', '25', '1', '01000000000', 'UBR.JSUUE931f5-ProductOrder', '0');
```

● RCS 템플릿(image) 예제.

※사전에 등록된 템플릿에서 문자메시지 내용은 description을 사용하고, 변수형 버튼 나머지 사용자 정의 key와 value로 구성된 JSON형태로 content작성이 필요합니다.

※버튼의 key/value 정보도 content에 작성하며, rcs_btn_info는 사용하지 않습니다.

※ [6.7 Template](#) 구조 참조 바랍니다.

▶ Oracle

```
INSERT INTO rta_rcs_tran(seq, date_client_req, subject, content,
callback, service_type, msg_status, recipient_num, rcs_format_id, rcs_header)
VALUES(sq_rta_rcs_tran_01.nextval, sysdate, ' ', '{"title":"홍길동 고객님", "subTitle1":"행사기간", "subDesc1":"
오늘부터 31일 까지", "subTitle2":"제공혜택", "subDesc2":"신규 가입고객으로 프로모션 가격으로 이용이 가
능합니다."}',
'0316281688', '24', '1', '01022142653', 'IBR.nGSbatra95-783LVFgOUL4VVe0L2LN6mhYR4', '0');
```

▶ MSSQL

```
INSERT INTO rta_rcs_tran(date_client_req, subject, content,
callback, service_type, msg_status, recipient_num, rcs_format_id, rcs_header)
VALUES(GETDATE(), '', '{"title":"홍길동 고객님", "subTitle1":"행사기간", "subDesc1":"오늘부터 31일
까지", "subTitle2":"제공혜택", "subDesc2":"신규 가입고객으로 프로모션 가격으로 이용이 가능합
니다."}',
'0316281688', '24', '1', '01022142653', 'IBR.nGSbatra95-783LVFgOUL4VVe0L2LN6mhYR4', '0');
```

▶ MySQL/MariaDB

```
INSERT INTO rta_rcs_tran(date_client_req, subject, content,
callback, service_type, msg_status, recipient_num, rcs_format_id, rcs_header)
VALUES(SYSDATE(), '', '{"title":"홍길동 고객님", "subTitle1":"행사기간", "subDesc1":"오늘부터 31일
까지", "subTitle2":"제공혜택", "subDesc2":"신규 가입고객으로 프로모션 가격으로 이용이 가능합
니다."}',
'0316281688', '24', '1', '01022142653', 'IBR.nGSbatra95-783LVFgOUL4VVe0L2LN6mhYR4', '0');
```



6.4. RCS 이미지 파일 업로드(FILE API)

RCS 메시지에 이미지를 포함시키기 위해서 사전에 해당 API를 통해 업로드 하여 발급된 fileKey를 사용해야 합니다. API는 HTTP 통신이며 1회 호출에 1개의 이미지를 업로드 할 수 있습니다.

아래 규격은 FILE API 통해 발급한 FileKey를 RTA에서 활용하는 부분의 설명이며, 자세한 API 규격은 Infobank RCS REST API 규격서를 참조합니다.

● 인증(토큰 발급) API

- ID와 Password를 이용하여, API를 호출할 수 있는 토큰(Access Token)을 발급 받을 수 있습니다.
- X-IB-Client-Id : 인포뱅크를 통해 발급받은 ID
- X-IB-Client-Passwd : 인포뱅크를 통해 발급받은 Password

[Request URL]

```
POST https://rcs-auth.ibizplus.co.kr:7000/v1/auth/token
```

[Request Header]

Name	Type	Desc
X-IB-Client-Id	String	인포뱅크를 통해 발급받은 ID
X-IB-Client-Passwd	String	인포뱅크를 통해 발급받은 Password
Accept	String	application/json

Request Sample (cURL)

```
curl -X POST -H "X-IB-Client-Id:ID" -H "X-IB-Client-Passwd:Password" -H "Accept: application/json" https://rcs-auth.ibizplus.co.kr:7000/v1/auth/token
```

[Response Header]

Parameter	Type	Desc
Content-Type	String	application/json; charset=UTF-8

[Response Body]

Parameter	Type	Desc
accessToken	String	인증 토큰정보
schema	String	Bearer
expired	String (Date-Time)	토큰 만료 시각

Response Sample (cURL) - Success

```
{
  "accessToken":"eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJleHAiOiJlODI4OTkxODksImF1ZCI6Im1tcyl6Iklk4iLCJzbXMiOiJZliwicmVwJoiTiJ9.LEo0A411Yi0T7YDV7V8FGaD6DXzdl J5TeVf_8d2h3JQ", "schema":"Bearer", "expired":"20201228132629"
}
```

Response Sample (cURL) - Fail

```
{
  "errorCode":"A200",
  "errorText":"Invaild authentication information"
}
```

● 파일 업로드 API

- 파일을 업로드하여 RCS MMS / FAILOVER(MT) MMS 전송 시 사용할 파일 키를 발급합니다.
- 파일은 확장자(RCS: bmp, gif, jpeg, jpg, png, FAILOVER MT: jpg/jpeg), 크기(RCS: 1 Mbyte 이하, FAILOVER MT: 300kb 이하) 제한³¹이 있습니다.
- Multipart/form-data 방식만 지원 합니다.
- 요청 시 파일 업로드 수는 1개로 제한하며 분당 최대 30회 업로드 가능합니다.
- 파일명과 크기가 같은 이미지의 경우 1시간에 1건만 등록³²할 수 있습니다.
 - 발급 된 키는 만료 시간 전까지 재사용 가능합니다.
- 만료일까지 파일 키를 사용할 수 있으며 만료일 후 파일은 삭제됩니다.
- 파일키를 설정할 경우 유니크 하게 생성 되어 합니다.
{고객ID}_{사용자지정코드} 예)infobank_00000001
- SSL TLS1.2를 사용하여 업로드 합니다.

○ Request URL

POST https://rcs-file.ibizplus.co.kr:7010/v1/rcs/file

○ Request Header

Name	Type	Description
Authorization	String	발급받은 schema + " " + 발급받은 accessToken
Content-Type	String	multipart/form-data; boundary=구분값

³¹ 국내 통신사에서 수용할 수 있는 RCS/MT 이미지 사양입니다.

³² 동일한 파일의 불필요한 업로드를 막고 공격을 차단하기 위한 보안 정책입니다.

Accept	String	application/json
--------	--------	------------------

o Request Body

Name	Type	Required	Description
fileData	binary[]	YES	업로드할 파일 바이너리 Content-Disposition에 filename 포함
fileKey	String	NO	파일 키 사용자 지정 ³³ (최대 길이 64자)
fileType	String	NO	파일 종류 (Default: 0) (0: RCS전용, 1: FAILOVER(MT)전용, 2: 모두)

● Request Example(cURL)

```
curl -X POST
-H "Authorization:Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJleHAiOiJlE0ODI4 ..."
-H "Accept: application/json"
-F fileData=@/data/test.jpg
-F fileType=0
https://rcs-file.ibizplus.co.kr:7010/v1/rcs/file
```

※jpg가 아닌 이미지 파일에 대해서는 type명시 필요
(-F "fileData=@/data/test.png;type=image/png")

o Response Header

Parameter	Type	Description
Content-Type	String	application/json; charset=UTF-8

o Response Body

Parameter	Type	Description
fileKey	String	파일 키
expired	String	파일 만료 시간

● Response Example

➡ Success (성공)

```
{
  "fileKey":"infobank_4ebb7d294f5645e0a79e37990952a178",
  "expired":"20200428133909"
}
```

³³ 유니크하게 생성 되어 합니다. {고객ID}_{사용자지정코드} 예)infobank_00000001

6.5. RCS 버튼(액션 버튼)

- 총 9가지 형태의 버튼을 제공합니다.
- 각 버튼에 맞는 JSON형태의 입력이 필요합니다.
- JSON Array의 Index가 버튼의 순서대로 매핑 됩니다. (Array Index 0일경우 첫번째 버튼으로 인식)

6.5.1. RCS버튼(액션 버튼) 구성

버튼 구분	설명
URL 연결	Web page 또는 App으로 이동할 수 있습니다.
지도/위치 (지도 보여주기, 지도검색, 위치 전송)	미리 지정된 지도의 위치를 보여주거나 사용자의 현재 위치를 서버로 전송 할 수 있습니다.
일정 등록	사용자의 캘린더에 특정 일정을 등록할 수 있습니다.
복사하기	지정된 내용을 클립보드로 복사합니다.
대화방 열기 (문자/음성, 영상)	다른 번호로 메시지를 보낼 수 있도록 대화방을 엽니다.
전화 연결	특정 전화번호로 전화를 걸 수 있습니다.

※RCS 메시지 액션 버튼 구성 및 규격은 이동통신사 사정에 의해 변경될 수 있습니다.

6.5.2. RCS버튼(액션 버튼) 예제

● URL 연결(urlAction)

매개변수	예제
URL정보 버튼명	<pre>[{"suggestions":[{"action":{"urlAction":{"openUrl":{"url":"URL정보"}},},{"displayText":"버튼명",},{"postback":{"data": "set_by_chatbot_open_url"}},}]}</pre>

	<pre> } } }</pre>
--	-------------------------

● 지도 보여주기(mapAction)

매개변수	예제
위도(latitude) 경도(longitude) 표시될 라벨 명(label) 지도URL 버튼 명	<pre> [{"suggestions":[{" "action":{" "mapAction": { "showLocation": { "location": { "latitude": 위도(예시:37.4220041) , "longitude": 경도(예시:-122.0862515), "label": "라벨 명" }, "fallbackUrl": "지도URL " (예시: "https://www.google.com/maps/@37.4219162,-122.078063,15z") } }, "displayText":"버튼 명", "postback":{" "data": "set_by_chatbot_open_map" } } }] }</pre>

● 지도 검색(mapAction)

매개변수	예제
지도검색어 지도URL 버튼명	<pre> [{"suggestions":[{" "action":{" "mapAction": { "showLocation": {</pre>

	<pre> "location": { "query": "검색어" }, "fallbackUrl": "지도URL" (예시: "https://www.google.com/maps/search") } }, "displayText": "버튼명", "postback": { "data": "set_by_chatbot_open_map" } } } } } </pre>
--	--

● 내 위치 전송하기(mapAction)

매개변수	예제
버튼 명	<pre> [{"suggestions": [{ "action": { "mapAction": { "requestLocationPush": {} }, "displayText": "버튼 명", "postback": { "data": "set_by_chatbot_open_map" } } }] }] </pre>

● 일정 등록(calendarAction)

매개변수	예제
------	----

일정 시작 시간 일정 종료 시간 일정 제목 일정 내용 버튼 명	<pre>[{"suggestions":[{"action":{"calendarAction":{"createCalendarEvent":{"startTime": "시작시간 (예: "2020-03-14T00:00:00Z"), endTime": "종료시간 (예: "2020-03-14T23:59:59Z"), title": "일정 제목", description": "일정 내용" }} }, "displayText": "버튼명", "postback":{"data": "set_by_chatbot_create_calendar_event" }} }]</pre>
---	--

● 복사하기(clipboardAction)

매개변수	예제
복사될 텍스트 버튼 명	<pre>[{"suggestions":[{"action":{"clipboardAction":{"copyToClipboard":{"text": "복사될 텍스트" }}, "displayText": "버튼 명", "postback":{"data": "user_copied_data1" }} }]</pre>

● 대화방열기-문자(composeAction)

매개변수	예제
수신번호 메시지 내용 버튼 명	<pre> [{"suggestions":[{" "action":{" "composeAction": { "composeTextMessage": { "phoneNumber": 수신번호 (예: "+8215881234"), "text": 메시지내용 } } }, "displayText":"버튼명", "postback":{" "data": "set_by_chatbot_compose_text_message" } }] }] </pre>

● 대화방열기(음성, 영상)(composeAction)

매개변수	예제
수신번호 버튼 명	<pre> [{"suggestions":[{" "action":{" "composeAction": { "composeRecordingMessage": { "phoneNumber": 수신번호 (예: "+8215881234"), "type": "VIDEO" } } }, "displayText":"버튼명", "postback":{" "data": "set_by_chatbot_compose_recording_message" } }] }] </pre>

	<pre> } } } } </pre>
--	--

● 전화 연결(dialerAction)

매개변수	예제
전화번호 버튼 명	<pre> [{"suggestions":[{" "action":{" "dialerAction": { "dialPhoneNumber": { "phoneNumber": "전화번호" } }, "displayText":"버튼명", "postback":{" "data": "set_by_chatbot_dial_phone_number" } } }]}]} </pre>

6.6. RCS Carousel 구조

- 여러 장의 카드로 구성되 있는 Carousel형태의 메시지를 보내기 위해서 JSON형태의 입력이 필요합니다.
- 메시지 제목(subject), 메시지 내용(content), 이미지URL(rcs_img_url), 버튼 정보(rcs_btn_info)들에 해당됩니다.
- rcs_img_url은 발급받은 FILE_KEY를 'maapfile://파일키' 형태로 사용하며, 'mappfile://'은 생략 가능합니다.
- JSON Array의 Index가 Carousel 카드순서대로 매핑되며, 최대 6장까지 가능합니다. (Array Index 0일경우 첫번째 카드로 인식)
- JSON Array 중 카드에 입력할 값이 없으면 Index를 맞추기 위해 공백("")을 입력합니다.
(예시: {"content":["메시지 내용", "", "메시지 내용3"]})

- 카드 별로 메시지 제목, 메시지 내용, 이미지URL 중 최소 1개는 입력 되어 합니다.

● 메시지 제목 JSON형태

NAME	VALUE	예제
subject	Array	<pre>{ "subject": ["제목 입니다.", "제목2 입니다.", "제목3 입니다."] }</pre>

● 메시지 내용 JSON형태

NAME	VALUE	예제
content	Array	<pre>{ "content": ["내용 입니다.", "내용2 입니다.", "내용3 입니다."] }</pre>

● 이미지URL JSON형태

NAME	VALUE	예제
img_url	Array	<pre>{ "img_url": ["maapfile://rest_srv_test_951eb68af1894ed98a66f0226fed5f20 ", "maapfile://rest_srv_test_951eb68af1894ed98a66f0226fed5f23 "] }</pre>

● 버튼정보 JSON형태

NAME	VALUE	예제
-	Array	<pre>[{ "suggestions": [{ "action": { ...중략... } }], }, { "suggestions": [{ "action": { ...중략... } }] }]</pre>

		}]
--	--	--------

6.7. Template 구조

- RCS Content 필드의 정보 중, 템플릿 타입에 사용되는 콘텐츠 정보입니다.
- RTA_RCS_TRAN 테이블의 content 필드에 Json String 형태로 제공되어야 합니다.
- Template ID에 따라서 사용법이 달라질 수 있습니다.

● Content Json 정의

NAME	VALUE	예제
title	String	템플릿 제목
description	String	템플릿 내용
{key}	String	사전에 등록된 Key, value(JSON
subTitle1	String	소제목1
subDesc1	String	소본문1
subTitle2	String	소제목2
subDesc2	String	소본문2
subTitle3	String	소제목3
subDesc3	String	소본문3

6.8. RCS 템플릿 전송

6.8.1. LMS 템플릿

기본정보

Name	Value
groupId	RBC 메시지 발송 통계용. 동일 messagebase ID/동일 group ID/동일 날짜 기준으로 집계
header *	"0" - 비 광고 "1" - (광고) 표기
footer	"header" : "1" 인 경우 필수, "000-000-000"의 무료수신거부 전화번호 입력
copyAllowed	"true" - 수신자가 글자 복사 가능하도록 허용 "false" - 수신자가 글자 복사할 수 없음

- CLL00001 (명세서A형)

Body

Name	contentType	note
mTitle	title	최대 17자
mTitleMedia	maapfile	<u>로고 또는 브랜드 로고</u>
title	description	최대 30자
description	description	최대 1300자
cellTitle1	description	최대 30자
cellLeft1	description	
cellRight1	description	
cellLeft2~10	description	
cellRight2~10	description	
cellTitle2	description	최대 30자
cellLeft11	description	
cellRight11	description	
cellLeft12~20	description	
cellRight12~20	description	
customParam *	etc or url	버튼 내 정의한 변수

⚠ contentType이 description인 항목의 글자수 합이 1300자 이하여야 합니다.

i mTitleMedia, itemMedia는 템플릿 등록 시점에 지정한 상태로 메시지 발송시점에 변경/추가할 수 없습니다.

- CLL00002 (문단형)

Body

Name	contentType	note
mTitle	title	최대 17자
mTitleMedia	maapfile	<u>로고 또는 브랜드 로고</u>
title1	description	최대 30자
description1	description	최대 1300자
title2	description	최대 30자
description2	description	최대 1300자
customParam *	etc or url	버튼 내 정의한 변수

⚠ contentType이 description인 항목의 글자수 합이 1300자 이하여야 합니다.

ℹ bold로 표기한 파라미터에 값이 없으면 같은 색상의 파라미터도 실제 수신시 보이지 않습니다.

- CLL00003 (타이틀 강조 기본형)

Body

Name	contentType	note
mTitle	title	최대 17자
mTitleMedia	maapfile	<u>로고 또는 브랜드 로고</u>
title	description	최대 30자
description	description	최대 1300자
customParam *	etc or url	버튼 내 정의한 변수

⚠ contentType이 description인 항목의 글자수 합이 1300자 이하여야 합니다.

ℹ itemMedia는 템플릿 등록 시점에 지정한 상태로 메시지 발송시점에 변경/추가할 수 없습니다.

- CLL00004 (기본형)

Body

Name	contentType	note
mTitle	title	최대 17자
mTitleMedia	maapfile	<u>로고 또는 브랜드 로고</u>
title	description	최대 30자
description	description	최대 1300자
customParam *	etc or url	버튼 내 정의한 변수

⚠ contentType이 description인 항목의 글자수 합이 1300자 이하여야 합니다.

ℹ mTitleMedia는 템플릿 등록 시점에 지정한 상태로 메시지 발송시점에 변경/추가할 수 없습니다.

● CLL00005 (명세서B형)

Body

Name	contentType	note
mTitle	title	최대 17자
mTitleMedia	maapfile	<u>로고 또는 브랜드 로고</u>
title1	description	최대 30자
description1	description	총 글자수 내 자유
title2	description	최대 30자
description2	description	총 글자수 내 자유
cellTitle1	description	최대 30자
cellLeft1	description	
cellRight1	description	
cellLeft2~10	description	
cellRight2~10	description	
title3	description	최대 30자
description3	description	총 글자수 내 자유
notice	description	총 글자수 내 자유
customParam *	etc or url	버튼 내 정의한 변수

⚠ contentType이 description인 항목의 글자수 합이 1300자 이하여야 합니다.

i mTitleMedia는 템플릿 등록 시점에 지정한 상태로 메시지 발송시점에 변경/추가할 수 없습니다.

● CLL00006 (명세서C형)

Body

Name	contentType	note
mTitle	title	최대 17자
mTitleMedia	maapfile	<u>로고 또는 브랜드 로고</u>
title1	description	최대 30자
description1	description	총 글자수 내 자유
cellTitle1	description	최대 30자
cellLeft1	description	
cellRight1	description	
cellLeft2~10	description	
cellRight2~10	description	
title2	description	최대 30자
description2	description	총 글자수 내 자유
title3	description	최대 30자
description3	description	총 글자수 내 자유
notice	description	총 글자수 내 자유
customParam *	etc or url	버튼 내 정의한 변수

⚠ contentType이 description인 항목의 글자수 합이 1300자 이하여야 합니다.

i mTitleMedia는 템플릿 등록 시점에 지정한 상태로 메시지 발송시점에 변경/추가할 수 없습니다.

[LMS 템플릿 MySQL 예제]

-- LMS 템플릿 기본형

```
insert into rta_rcs_tran (date_client_req, subject, content, callback, service_type, rcs_format_id, msg_status,
recipient_num, rcs_header)
values(sysdate(), ' ', '{"mTitle":"mTitle 내용", "title":"title 내용", "description":"description 내용"}',
'발신번호', '24', '템플릿 ID', '1', '수신번호', '0');
```

-- LMS 템플릿 문단형

```
insert into rta_rcs_tran (date_client_req, subject, content, callback, service_type, rcs_format_id, msg_status, recipient_num, rcs_header)
```

```
values(sysdate(), ' ', '{"mTitle":"mTitle 내용", "title1":"title1 내용", "description1":"description1 내용", "title2":"title2 내용", "description2":"description2 내용"}',
```

```
'발신번호', '24', '템플릿 ID', '1', '수신번호', '0');
```

6.8.2. 정보성 템플릿

기본정보

Name	Value
groupId	RBC 메시지 발송 통계용. 동일 messagebase ID/동일 group ID/동일 날짜 기준으로 집계
header *	"0" - 비 광고
copyAllowed	"true" - 수신자가 글자 복사 가능하도록 허용 "false" - 수신자가 글자 복사할 수 없음

- CLT00002 (아이템 강조형 서술)

Body

Name	contentType	note
itemTitle	description	최대 20자
itemDesc	description	최대 18자
itemMedia	maapfile	<u>로고 또는 브랜드 로고</u>
title	description	최대 30자
description	description	최대 90자
customParam *	etc or url	버튼 내 정의한 변수

⚠ contentType이 description인 항목의 글자수 합이 90자 이하여야 합니다.

ℹ itemMedia는 템플릿 등록 시점에 지정한 상태로 메시지 발송시점에 변경/추가할 수 없습니다.

- CLT00001 (아이템 강조형 테이블)

Body

Name	contentType	note
itemTitle	description	최대 20자
itemDesc	description	최대 18자
itemMedia	maapfile	<u>로고 또는 브랜드 로고</u>
cellTitle	description	최대 30자
cellLeft1	description	
cellRight1	description	
cellLeft2~5	description	
cellRight2~5	description	
customParam *	etc or url	버튼 내 정의한 변수

⚠ contentType이 description인 항목의 글자수 합이 90자 이하여야 합니다.

- TATA001D (타이틀 자유형)

Body

Name	contentType	note
title1 *	title	최대 8자 발송 시, 사용하지 않는 변수
customParam	title	title2 로 등록한 변수 최대 20자
description *	description	최대 90자
customParam *	etc	버튼 내 정의한 변수

⚠ 정보성 메시지만 발송 가능합니다.

⚠ 템플릿 등록 시, {{변수}}+고정 메시지의 패턴을 맞춰서 발송합니다.

- 타이틀 선택형

Body

Name	contentType	note
description *	description	최대 90자
customParam *	etc	버튼 내 정의한 변수

❗ 정보성 메시지만 발송 가능합니다.

❗ 템플릿 등록 시, {{변수}}+고정 메시지의 패턴을 맞춰서 발송합니다.

6.8.3. 이미지 템플릿

기본정보

Name	Value
groupId	RBC 메시지 발송 통계용. 동일 messagebase ID/동일 group ID/동일 날짜 기준으로 집계
header *	"0" - 비 광고 "1" - (광고) 표기
footer	"header" : "1" 인 경우 필수, "000-000-000"의 무료수신거부 전화번호 입력
copyAllowed	"true" - 수신자가 글자 복사 가능하도록 허용 "false" - 수신자가 글자 복사할 수 없음

- 아이템 상세형

Body

Name	contentType	note
<code>media *</code>	maapfile	900 × 600 px의 jpg, png, gif (1MB 이하) 발송 시, 사용하지 않는 변수
<code>mediaUrl</code>	url	media 클릭 시, 이동되는 웹 경로 발송 시, 사용하지 않는 변수
<code>itemTitle</code>	description	최대 20자
<code>itemDesc</code>	description	최대 18자
<code>itemMedia</code>	maapfile	<u>로고 또는 브랜드 로고</u>
<code>title</code>	description	최대 30자
<code>description</code>	description	최대 500자
<code>cellTitle</code>	description	최대 30자
<code>cellLeft1</code>	description	
<code>cellRight1</code>	description	
<code>cellLeft2~5</code>	description	
<code>cellRight2~5</code>	description	
<code>customParam *</code>	etc or url	버튼 내 정의한 변수

⚠ contentType이 description인 항목의 글자수 합이 500자 이하여야 합니다.

i bold로 표기한 파라미터에 값이 없으면 같은 색상의 파라미터도 실제 수신시 보이지 않습니다.

- 슬라이드형 (medium)

Body

Name	contentType	note
<code>media1</code>	maapfile	696 * 504 px의 jpg, png, gif (1MB 이하) 발송 시, 사용하지 않는 변수
<code>title1</code>	title	최대 30자
<code>description1</code>	description	
<code>media2</code>	maapfile	696 * 504 px의 jpg, png, gif (1MB 이하) 발송 시, 사용하지 않는 변수
<code>title2</code>	title	최대 30자
<code>description2</code>	description	
<code>media3~6</code>	maapfile	696 * 504 px의 jpg, png, gif (1MB 이하) 발송 시, 사용하지 않는 변수
<code>title3~6</code>	title	최대 30자
<code>description3~6</code>	description	
<code>customParam *</code>	etc or url	버튼 내 정의한 변수

 contentType이 description인 항목의 글자수 합이 500자 이하여야 합니다.

- 썸네일형 (세로)

Body

Name	contentType	note
media *	maapfile	900 * 560 px의 jpg, png, gif (1MB 이하) 발송 시, 사용하지 않는 변수
mediaUrl	url	media 클릭 시, 이동되는 웹 경로 발송 시, 사용하지 않는 변수
title	title	최대 30자
description	description	최대 150자
subMedia1 *	maapfile	300 * 300 px의 jpg, png, gif (1MB 이하) 발송 시, 사용하지 않는 변수
subMediaUrl1	url	발송 시, 사용하지 않는 변수
subDesc1 *	description	최대 33자
subMedia2 *	maapfile	300 * 300 px의 jpg, png, gif (1MB 이하) 발송 시, 사용하지 않는 변수
subMediaUrl2	url	발송 시, 사용하지 않는 변수
subDesc2 *	description	최대 33자
subMedia3 *	maapfile	300 * 300 px의 jpg, png, gif (1MB 이하) 발송 시, 사용하지 않는 변수
subMediaUrl3	url	발송 시, 사용하지 않는 변수
subDesc3 *	description	최대 33자
customParam *	etc or url	버튼 내 정의한 변수



contentType이 maapfile인 항목의 파일 용량 합이 1MB 이하여야 합니다.

- 썸네일형 (가로)

Body

Name	contentType	note
media *	maapfile	900 * 560 px의 jpg, png, gif (1MB-이하) 발송 시, 사용하지 않는 변수
mediaUrl	url	media 클릭 시, 이동되는 웹 경로 발송 시, 사용하지 않는 변수
title	title	최대 30자
description	description	최대 150자
subMedia1 *	maapfile	300 * 300 px의 jpg, png, gif (1MB-이하) 발송 시, 사용하지 않는 변수
subMediaUrl1	url	발송 시, 사용하지 않는 변수
subMedia2 *	maapfile	300 * 300 px의 jpg, png, gif (1MB-이하) 발송 시, 사용하지 않는 변수
subMediaUrl2	url	발송 시, 사용하지 않는 변수
subMedia3 *	maapfile	300 * 300 px의 jpg, png, gif (1MB-이하) 발송 시, 사용하지 않는 변수
subMediaUrl3	url	발송 시, 사용하지 않는 변수
customParam *	etc or url	버튼 내 정의한 변수

ⓘ contentType이 maapfile인 항목의 파일 용량 합이 1MB 이하여야 합니다.

- 이미지 강조형 (3:4)

Body

Name	contentType	note
media *	maapfile	900 * 1200 px의 jpg, png, gif (1MB 이하) 발송 시, 사용하지 않는 변수
mediaUrl	url	media 클릭 시, 이동되는 웹 경로 발송 시, 사용하지 않는 변수
title	title	최대 30자
description *	description	최대 150자

- 이미지 강조형 (1:1)

Body

Name	contentType	note
media *	maapfile	900 * 900 px의 jpg, png, gif (1MB 이하) 발송 시, 사용하지 않는 변수
mediaUrl	url	media 클릭 시, 이동되는 웹 경로 발송 시, 사용하지 않는 변수
title	title	최대 30자
description *	description	최대 150자

- SNS형

Body

Name	contentType	note
media *	maapfile	900 * 900 px의 jpg, png, gif (1MB 이하) 발송 시, 사용하지 않는 변수
mediaUrl	url	media 클릭 시, 이동되는 웹 경로 발송 시, 사용하지 않는 변수
title	title	최대 30자
description *	description	최대 150자

- SNS형 (중간버튼)

Body

Name	contentType	note
media *	maapfile	900 * 560 px의 jpg, png, gif (1MB 이하) 발송 시, 사용하지 않는 변수
mediaUrl	url	media 클릭 시, 이동되는 웹 경로 발송 시, 사용하지 않는 변수
title	title	최대 30자
description *	description	최대 150자

- 이미지&타이틀 강조형 (1:1)

Body

Name	contentType	note
media *	maapfile	900 * 900 px의 jpg, png, gif (1MB 이하) 발송 시, 사용하지 않는 변수
mediaUrl	url	media 클릭 시, 이동되는 웹 경로 발송 시, 사용하지 않는 변수
title	title	최대 30자
subTitle1 *	description	최대 4자
subDesc1 *	description	최대 33자
subTitle2	description	최대 4자
subDesc2	description	최대 33자
subTitle3	description	최대 4자
subDesc3	description	최대 33자

- 이미지&타이틀 강조형 (3:4)

Body

Name	contentType	note
media *	maapfile	900 * 1200 px의 jpg, png, gif (1MB-이하) 발송 시, 사용하지 않는 변수
mediaUrl	url	media 클릭 시, 이동되는 웹 경로 발송 시, 사용하지 않는 변수
title	title	최대 30자
subTitle1 *	description	최대 4자
subDesc1 *	description	최대 33자
subTitle2	description	최대 4자
subDesc2	description	최대 33자
subTitle3	description	최대 4자
subDesc3	description	최대 33자

[이미지 템플릿 MySQL 예제]

-- 이미지 템플릿 아이템 상세형

```
insert into rta_rcs_tran (date_client_req, subject, content, callback, service_type, rcs_format_id, msg_status,
recipient_num, rcs_header)
values(sysdate(), ' ', '{"itemTitle":"itemTitle 내용", "title":"title 내용", "description":"description 내용",
"cellTitle":"cellTitle 내용", "cellLeft1":"cellLeft1 내용", "cellLeft2":"cellLeft2 내용",
"cellRight1":"cellRight1 내용", "cellRight2":"cellRight2 내용"}',
'발신번호', '24', '템플릿 ID', '1', '수신번호', '0');
```

-- 이미지 템플릿 SNS형 (중간버튼)

```
insert into rta_rcs_tran (date_client_req, subject, content, callback, service_type, rcs_format_id, msg_status,
recipient_num, rcs_header)
values(sysdate(), ' ', '{"title":"title 내용", "description":"description 내용"}',
'발신번호', '24', '템플릿 ID', '1', '수신번호', '0');
```

-- 이미지 템플릿 이미지 강조형(1:1)

```
insert into rta_rcs_tran (date_client_req, subject, content, callback, service_type, rcs_format_id, msg_status,
recipient_num, rcs_header)
```

```
values(sysdate(), ' ', '{"title":"title 내용", "description":"description 내용"}',  
'발신번호', '24', '템플릿 ID', '1', '수신번호', '0');
```

7. 부가기능

7.1. 이중화(HA³⁴)

RTA는 운영의 안정성을 보장하기 위해 이중화를 제공합니다. 이중화 된 RTA가 모두 작동하는 Active-Active 모드와 Primary와 Secondary로 구별하여 서로간의 장애 감지³⁵를 통해 이어서 동작하게 되는 Active-Standby 모드 등 2가지 방식을 제공합니다. RTA설정파일(rta.cf)를 수정하여 적용할 수 있습니다.

```
#####  
# RTA Server HA(High Availability) Configuration Information #  
#####  
  
#-----  
# HA Mode (none, active-active, active-standby)  
#-----  
ha.mode = none    이중화 사용설정여부(none, active-active, active-standby)  
  
#-----  
# HA Id  
#-----  
ha.agent.id = A1    이중화 시 사용될 해당 RTA의 ID(고유한 ID필요)  
  
#-----  
# HA Status (only active-standby mode)(primary, secondary)  
#-----  
ha.status = primary    active-standby 이중화 시 primary, secondary 설정  
  
#-----  
# HA Active Server Infomation(only active-standby mode)  
#-----
```

³⁴ High Availability(고 가용성: 서버와 네트워크 등이 오랜기간동안 지속적으로 정상 운영이 가능한 성질)

³⁵ 특정 포트를 통해 Health Check하는 방식입니다. Primary가 Listen 하게 되며 Check가 3회이상 되지 않았을 때 Secondary RTA기능이 활성화 됩니다.

```
ha.primary.host = localhost active-standby 이중화 시 장애감지를 위한 접속정보  
ha.primary.port = 4000
```

※설정파일(rta.cf) 수정 후에 RTA 재 기동이 필요합니다.

7.2. 발송 차단

RTA는 rta_banlist 테이블을 통하여 수신 번호, 발신번호, 메시지 내용 3가지 항목에 대해 메시지 발송 시 차단하는 기능을 제공합니다.

수신 번호는 rta_banlist 테이블에 입력 즉시 전송대상에서 차단되게 되지만 발신번호, 메시지내용에 대해서는 설정파일 수정이 필요하며, RTA 메모리에 올리는 주기가 존재하여 일정 시간 이후에 차단 됩니다.

자세한 설정 방법은 ['4.2 데이터베이스 관련 설정파일\(db.cf\)'](#) 을 참조하시기 바랍니다.

rta_banlist 입력 예시 (mysql)

▶ 수신 번호 차단

```
insert into rta_banlist (ban_seq, ban_type, content, send_yn, ban_status_yn, reg_date,  
update_date) values('1', 'R', '01000000000', 'N', 'Y', sysdate(), sysdate());
```

▶ 발신번호 차단 (정규식 또는 지정)

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

▶ 메시지 내용 차단 (정규식 또는 단어 지정)

```
insert into rta_banlist (ban_seq, ban_type, content, send_yn, ban_status_yn, reg_date,  
update_date) values('3', 'T', '지정문구', 'N', 'Y', sysdate(), sysdate());
```

※전송 차단으로 판단된 메시지는 결과 코드 'E917'로 처리됩니다.

7.3. 중복 메시지 차단

일정시간동안에 발송된 메시지와 동일한 수신 번호와 메시지를 다시 발송 요청 했을 때 중복메시지라 판단하여 차단해주는 기능을 제공합니다. RTA설정파일(rta.cf)를 수정하여 적용할 수 있습니다.

설정파일(rta.cf) 수정

```
#-----  
# Duplicate Message Blocking Settings(0:Not use, 1:Use)  
#-----  
dupcheck.rcs = 0    중복메시지 기능 사용여부(0:미사용, 1:사용)  
  
#-----  
# Duplicate Message Blocking Recent Message Cache Delete Interval(second)  
#-----  
dupcheck.rcs.cacheinterval = 60    중복메시지 구분을 위한 메모리 삭제 주기(초)
```

- ※cacheinterval 값을 과도하게 크게 설정하면 발송 성능저하 또는 리소스 부하가 나타날 수 있습니다.
- ※중복메시지로 판단된 메시지는 결과 코드 'E915'로 처리됩니다.
- ※설정파일(rta.cf) 수정 후에 RTA 재 기동이 필요합니다.

7.4. 우선 순위 발송

RTA는 메시지 전송 시 3종류(Slow/Fast/Very Fast)의 우선순위를 적용하여 발송할 수 있습니다. 우선순위는 RTA설정파일(rta.cf)를 수정하여 RTA를 재 기동한 이후에 전송 테이블(rta_rcs_tran) 테이블의 우선순위(priority) 필드를 'S', 'F', 'VF' 로 입력하여 사용할 수 있습니다. VF는 가장 최우선으로 보내는 메시지이며, F, S의 순서대로 전송하게 됩니다. 우선순위를 적용하지 않고 일반적으로 보낼 때는 기본값인 'S' 으로 설정하여 입력해 주시기 바랍니다.

설정파일(rta.cf) 수정

```
#-----  
# Message Sending Priority Use(0:Not use ,1:Use)  
#-----  
# whether to use 'Very Fast'  
priority.use.rcs.veryfast = 0    우선순위(VF)발송 기능 사용여부(0:미사용, 1:사용)  
  
# whether to use 'Fast'  
priority.use.rcs.fast      = 0    우선순위(F)발송 기능 사용여부(0:미사용, 1:사용)  
  
#-----  
# Set priority number of sending message 우선순위 구별 간 발송 비율 설정  
#-----
```

```
priority.cnt.rcs.veryfast = 0
priority.cnt.rcs.fast      = 0
priority.cnt.rcs.normal   = 1
```

※설정파일(rta.cf) 수정 후에 RTA 재 기동이 필요합니다.

7.5. Bypass 모듈 사용

보안 및 환경적인 이유로 인포뱅크 G/W와 직접통신하지 못할 때 Bypass 접속정보를 입력하여 기존 G/W정보를 변경하는 기능을 제공합니다. RTA설정파일(rta.cf)를 수정하여 적용할 수 있습니다.

설정파일(rta.cf) 수정

```
auth.host    = auth.infobank.net    인증서버 주소 (bypass ip)
auth.port    = 12001                인증서버 포트 (bypass port)

#####
# Bypass module Configuration Information                                #
#####

# Whether to use Bypass module(0:Not use, 1:Use)
bypass.use = 0                    Bypass 기능 사용여부(0:미사용, 1:사용)

# Bypass target IP address setting  Bypass ip/port 설정
bypass.address0 = 127.0.0.1:3000
bypass.address1 = 127.0.0.1:3000
```

※설정파일(rta.cf) 수정 후에 RTA 재 기동이 필요합니다.

7.6. 로그파일 개인정보 미 표시

개인정보 보호를 위해 RTA LOG파일의 수신 번호 정보를 마스킹 처리 또는 삭제해주는 기능을 제공합니다. RTA설정파일(rta.cf)를 수정하여 적용할 수 있습니다.

설정파일(rta.cf) 수정

```
#-----
```

```
# Use of personal information log file filtering
(0:Not used, 1:Used(Delete), 2:Used(Change only recipient number to asterisk(*))
#-----
log.use.personalinfo.hide = 0 수신번호 필터 기능 설정(0:미사용, 1:삭제, 2:마스킹처리(*))
```

※설정파일(rta.cf) 수정 후에 RTA 재 기동이 필요합니다.

7.7. 사용자 함수

RTA 는 발송 및 결과처리 등 기능 사이에 사용자 함수를 호출 할 수 있도록 인터페이스를 정의하여 제공하고 있습니다.

JAVA로 컴파일 된 class 파일을 적용 경로에 복사 한 후 RTA설정파일(rta.cf)를 수정하여 적용할 수 있습니다.

● 예제파일 및 적용 경로 정보

▶ 예제파일

~RTA 설치경로/src

▶ 적용경로

~RTA 설치경로/classes/ib/rta/callback/

● 적용방법

- ① 추가할 사용자 함수 구간의 예제 파일을 수정한 후 JAVA를 통해 컴파일 합니다.
- ② 컴파일 된 class파일을 적용 경로에 복사합니다.
- ③ RTA설정파일을 수정하여 기능을 설정합니다.

```
#####
# RTA User Library(Callback) Configuration Information                                     #
#####
#-----
# RCS User Library Call
#-----
callback.use.rcs = 0 RCS CALLBACK 사용여부(0:미사용, 1:사용)
callback.class.rcs = ib.rta.callback.SampleRCSCallback RCS CALLBACK 클래스 명
```

```

#-----
# Message Data Transform Library Call
#-----
callback.use.mdt = 0          MDT CALLBACK 사용여부(0:미사용, 1:사용)
callback.class.mdt = ib.rta.callback.SampleMDTCallback MDT CALLBACK 클래스 명

#-----
# UserScheduler User Library Call
#-----
callback.use.userscheduler = 0    USC CALLBACK 사용여부(0:미사용, 1:사용)
callback.class.userscheduler = ib.rta.callback.SampleUSCCallback USC CALLBACK
클래스명

#-----
# DBS User Library Call
#-----
callback.use.dbs = 0          DBS CALLBACK 사용여부(0:미사용, 1:사용)
callback.class.dbs = ib.rta.callback.SampleDBSCallback DBS CALLBACK 클래스 명

```

7.7.1. RCS Callback(IBRCSCallback.java)

RCS 전송 과정에서 메시지 전송 또는 전송 결과 수신 후 수행 후 호출 되는 프로세스를 정의한 인터페이스 입니다. 결과 수신 후 특정 작업을 수행할 때 사용될 수 있습니다.

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

설명	RCS 를 DB 테이블에서 조회하여 전송 큐에 enqueue 후 호출되는 메소드
-----------	--

Parameter	• dbIndex : 연결할 Database Index (2개 이상의 DB에 연결 시 사용) • key : 메시지 키 (발송테이블의 seq).
------------------	--

- seq : 메시지 순번(사용안함)
- type : 메시지 타입
- recipientNum : 수신번호
- callback : 발신번호
- msg : 메시지내용

Return void

Class#method ib.frame.util.StringUtil#getMaskingPersonalInfo(String) :
LOG에 수신번호 표출 시 마스킹 처리

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

설명 RCS 전송 큐에서 dequeue 하여 인포뱅크 G/W 송신 후 호출되는 메소드

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

Return void

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

설명 RCS 이통사 전송 결과를 수신하여 리포트 큐에 enqueue 후 호출되는 메소드

Parameter • key : 메시지 키 (발송테이블의 seq).
• seq : 메시지 순번(사용안함)
• recipientNum : 수신번호
• code : 결과코드
• carrier : 착신망 정보 (SKT/KT/LGU+)
• date : 단말기 도착 시간

Return void

Class#method ib.frame.util.StringUtil#getMaskingPersonalInfo(String) :
LOG에 수신번호 표출 시 마스킹 처리

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

설명	RCS 이통사 전송 결과를 리포트 큐에서 dequeue 후 호출되는 메소드 (모든 전송 과정이 끝난 후 호출되는 메소드)
Parameter	• dbIndex : 연결할 Database Index. (2개 이상의 DB에 연결 시 사용) • key : 메시지 키 (발송테이블의 seq). • seq : 메시지 순번(사용 안함) • recipientNum : 수신번호 • code : 결과 코드 • carrier : 착신망 정보 (SKT/KT/LGU+) • date : 단말기 도착 시간
Return	void
Class#method	ib.frame.util.StringUtil#getMaskingPersonalInfo(String) : LOG에 수신번호 표출 시 마스킹 처리

7.7.2. Message Data Transform Callback(IBMDTCallback.java)

메시지 전송을 위해 DB에서 Record 검색 후 특정 필드의 값을 가공하기 위해 호출되는 프로세스를 정의한 인터페이스 입니다.. 주로 수신번호, 메시지가 암호화하여 사용하거나, 값을 변경하여 전송하고자 할 경우 사용 할 수 있습니다.

- **Boolean doRcsTransform(**
 IBRefObject refObj
)

설명	RCS 데이터 조회 후 변환된 데이터를 return 받기 위해 호출되는 메소드
Parameter	• refObj : IB 참조 개체. 수신번호, 발신번호, 메시지, 버튼정보를 포함하며 그 외에 여분의 필드 4개
Return	boolean
class#method	ib.frame.util.StringUtil#getMaskingPersonalInfo(String) : LOG에 수신번호 표출 시 마스킹 처리

7.7.3. User Scheduler Callback(IBUSCCallback.java)

사용자 정의 스케줄러 프로세스를 정의한 인터페이스 입니다. 주기적인 작업을 호출할 때 사용 할 수 있습니다. 호출 시간³⁶은 RTA설정파일(rta.cf)를 수정하여 설정할 수 있습니다. 자세한 사항은 ['4.1 RTA 설정파일\(rta.cf\)'](#) 을 참조해 주시기 바랍니다.

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

설명	설정된 시간마다 동작되는 메소드
Parameter	• dbIndex : 연결할 Database Index (2개 이상의 DB에 연결 시 사용) • dbCount : 전체 Database Count (기본적으로 1)
Return	void

7.7.4. Database Security Callback(IBDBSCallback.java)

JDBC 드라이버가 데이터베이스 연결 시도할 때, 접속 계정에 대한 비밀번호를 가공하기 위해 사용 하기 위한 인터페이스 입니다.

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

설명	데이터베이스 패스워드를 리턴하는 콜백 함수
Parameter	• dbIndex : 연결할 Database Index (2개 이상의 DB에 연결 시 사용) • driver : DB Driver class 명 • url : JDBC URL 주소 • dbUserId : DB 접근 사용자 ID
Return	String (접속 계정 암호, Null일 경우 db.cf의 db[index].password를 사용)

7.8. Failover 발송

RCS 메시지 발송이 실패한 경우 지정한 Failover 메시지(문자메시지(SMS, LMS, MMS))를 발송하는 기능을 제공합니다.

³⁶ 기본은 05:00로 설정되어 있습니다.

- Failover 메시지 종류

Failover			
메시지 종류	SMS	MMS	LMS
서비스 타입	0	2	3

Failover 사용을 위해서는 RCS 메시지를 발송할 때 failover_yn을 'Y'로 하고 Failover 관련 데이터들을 입력해주셔야 합니다.

Failover 관련 필드는 아래와 같으며 더 자세한 내용은 ['5.1.2 테이블 명세서'](#) 를 참조해 주시기 바랍니다.

- Failover 관련 필드

매개변수 명	설명
failover_yn	Failover 발송 여부
failover_service_type	Failover 데이터 (공통-메시지 전송 타입)
failover_subject	Failover 데이터 (공통-메시지 제목)
failover_content	Failover 데이터 (공통-메시지 내용)
failover_img	Failover 데이터 (공통-이미지 URL/파일 키)
failover_prefix	Failover 처리 시 메시지 내용 앞에 Prefix 추가
failover_timeout_send	Failover 전송 후 timeout 발생 시 failover 발송 여부
failover_date_sent	Failover 인포뱅크 G/W 접수 시간
failover_date_rslt	Failover 이동사가 수신자번호로 전송한 시간
failover_date_report	Failover 인포뱅크로부터 결과 수신한 시간
failover_report_code	Failover 전송 결과
failover_carrier	Failover 착신망 정보

- Failover 예제

▶ Oracle

```
insert into rta_rcs_tran(seq, date_client_req, subject, content, callback, service_type, msg_status,
recipient_num, rcs_format_id, rcs_header,
failover_yn, failover_service_type, failover_subject, failover_content, failover_prefix)
values( SQ_RTA_RCS_TRAN_01.nextval, SYSDATE, 'RCS SMS 제목', '테스트 SMS문자 입니다.',
'01000000000', '20', '1', '01000000000', 'SS000000', '0',
'Y', '0', 'failover 제목', 'failover SMS입니다.', '[재발신]');
```

▶ MSSQL

```
insert into rta_rcs_tran(date_client_req, subject, content, callback, service_type, msg_status,  
recipient_num, rcs_format_id, rcs_header,  
failover_yn, failover_service_type, failover_subject, failover_content, failover_prefix)  
values( getdate(), 'RCS SMS 제목', '테스트 SMS문자 입니다.', '01000000000', '20', '1',  
'01000000000', 'SS000000', '0',  
'Y', '0', 'failover 제목', 'failover SMS입니다.', '[재발신]);
```

▶ **MySQL/MariaDB**

```
insert into rta_rcs_tran(date_client_req, subject, content, callback, service_type, msg_status,  
recipient_num, rcs_format_id, rcs_header,  
failover_yn, failover_service_type, failover_subject, failover_content, failover_prefix)  
values( sysdate(), 'RCS SMS 제목', '테스트 SMS문자 입니다.', '01000000000', '20', '1',  
'01000000000', 'SS000000', '0',  
'Y', '0', 'failover 제목', 'failover SMS입니다.', '[재발신]);
```

8. RCS 코드표

구분	서비스 타입	포맷 ID	메시지 타입	메시지 사양						
				이미지	제목	내용	카드수	버튼		
SMS	20	SS000000	Stanalone	-	-	최대 100 자	1	1 (최대 17 자)		
LMS	23	SL000000		-	최대 30 자	최대 1300 자	1	3 (최대 17 자)		
MMS	22	SMwThT00	Standalone	Tall (568*528)			1	2 (최대 17 자)		
		SMwThM00	Media Top	Medium (568*336)			1			
MMS (CAROUSEL)	26	CMwMhM0300	Carousel Medium	Medium (696*504)			최대 30 자	최대 1300 자	3	2 (최대 13 자)
		CMwMhM0400						4		
		CMwMhM0500						5		
		CMwMhM0600			6					
		CMwShS0300	Carousel Small	Short (360*336)	최대 1300 자	3		2 (최대 5 자)		
		CMwShS0400			4					
		CMwShS0500			5					
		CMwShS0600			6					
TEMPLATE (FREE)	24	발급받은 TEMPLATE ID	Free	등록된 형태	-	최대 90 자	1	-		
TEMPLATE (DESCRIPTION)			Description					2 (최대 17 자)		
TEMPLATE (IMAGE)			Image							
TEMPLATE (CELL)	25		Cell							

※RCS 템플릿 코드 및 RCS 사양은 이동통신사 사정에 의해 변경될 수 있습니다.

9. FAQ

9.1. RTA 관련 FAQ

- **인증키 파일을 찾지 못합니다.**

→ RTA 설치 후 최초 실행 시 아래와 같은 log가 출력된다면, 인증 키를 생성하지 않아 발생한 오류입니다. 인증 키는 \$RTA_HOME/regkey 를 실행하여 생성할 수 있습니다. regkey 수행 시 success가 출력되지 않는다면, RTA ID, 비밀번호가 정확한지 확인해 주시기 바랍니다.

```
RTA startDaemon [%18s] ALERT: init failure, {} ib.frame.cf: .WcertWxxxx_pri.der (지정된 파일을 찾을 수 없습니다)
```

- **Infobank G/W 연결 시 오류가 발생합니다.**

→ RTA 실행 시 아래와 같은 log가 출력된다면 인포뱅크 G/W 연결 오류 입니다. 이 때는 사용자 환경의 RTA가 설치되는 서버 방화벽을 체크해 주시기 바랍니다.

```
connectMMG NetException occurred when processing connect IB G/W ib.a.e.j: fail to connect authentication server ip: xxx.xxx.xxx.xxx port: 3000||Connection timed out
```

9.2. 데이터베이스 관련 FAQ

- **DB Stored Procedure 오류가 발생합니다.**

→ RTA 실행 시 아래와 같은 로그가 출력된다면, DB Package 또는 Stored Procedure가 설치되어 있지 않아 발생한 오류입니다. 같이 제공되는 DB 프로시저를 생성해 주시기 바랍니다.

```
java.sql.SQLException: ORA-06550: line 1, column 7: PLS-00201: identifier 'RTA_PKG_COM.SP_RTA_COMMON_CHECKPRIVILEGE' must be declared
```

- **Oracle 계정 권한 설정 오류가 발생합니다.**

→ RTA 실행 시 아래와 같은 메시지가 발생하는 경우는 오라클 계정의 테이블 생성 권한이 없어서 발생하는 경우 입니다. 이때는 해당 계정에 테이블 생성 권한을 부여해 주시기 바랍니다.

```
java.sql.SQLException: ORA-01031: 권한이 불충분합니다 ORA-06512: "RTA.RTA_PKG_COM", 줄 152에서 ORA-06512: 줄 1에서 at oracle.jdbc.driver.DatabaseError.throwSQLException(DatabaseError.java:112)
```

- **Content필드 입력 시 Line break(줄 바꿈)는 어떻게 입력하나요?**

→ Message 본문 입력 시 줄 바꿈을 입력하기 위해서는 DB별로 아래처럼 처리해주시기 바랍니다.

▶ **Oracle**

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

▶ **MSSQL**

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

▶ **MySQL/MariaDB**

'안녕하세요\r\n좋은 하루 되십시오'

9.3. 발송 서비스 관련 FAQ

- **전송 테이블에 데이터를 입력했는데, 발송이 되지 않습니다.**

→ 전송 조건(WHERE)에 적합한 데이터를 입력하였는지 프로시저(sp_rta_rcs_tran_select) 내용을 검토하여 확인해 주시기 바랍니다.

10. 문의

본 문서와 관련된 기술 문의는 아래 메일 주소로 연락 주시면 됩니다.

Email : support@infobank.net