D'Amo DA

사용자 설명서

Products Name: DA v4.0

Mintenance No.: 70.0

2020. 10. 23

디아모 개발부

Penta SECURITY

D'Amo DA v4.0.70

이 문서는 제품의 개발사인 당사의 임직원, 당사와 NDA 체결된 파트너사 및 EULA 체결된 고객사의 임직원을 대상으로 작성한다.

매뉴얼 안내

펜타의 제품 매뉴얼은 다음과 같이 분류하며 각 문서의 보안 등급에 따라 필요한 정보를 제공한다.

문서명	문서 보안 등급	정의
사용자 설명서	SL4	• 제품과 함께 고객에게 제공되는 기본 설명서이다. • 제품 기능에 대한 개요(Concept) 및 각 기능에 대해 설명한다. • 제품(H/W, S/W)를 설치하기 위한 절차 및 동작 확인을 위한 최소한의 설정과 서비스의 이용 방법을 설명한다.
파트너 안내서	SL3	• 파트너사의 엔지니어에게 제공된다. • 고객에게 공개되지 않는 기능의 '설정/운영 방법'을 각 절차에 따라 개조식으로 설명한다.
특정 고객용 설명서	SL2~SL4	사이트 패키지 등과 같이 특정 사이트의 고객에게 제공한다.
개발자 안내서	SL2~SL4	• 고객의 개발자 또는 협력사에 제공한다. • 제품과 함께 제공되는 API를 이용하여 응용 서비스 개발이 가능한 정보를 제공한다.

유지 보수에 대한 정보는 별도 유지 보수 계약서에 명시되므로, 본 매뉴얼에서는 제공하지 않는다.

매뉴얼 표기법

이 문서는 다음과 같은 표기법에 준하여 작성한다.

표기법

표기법	설명	예
ex)	예제 표기	ex) 회사(기관명), 전산 담당 부서명 등을 입력
${}	사용자 환경에 따라 입력되는 값이 상이할 경우	${설치 디렉터리}₩ABC.exe 파일을 입력해 주세요.
[ ]	Command Line 명령어에서 사용자가 선택적으로 입력하는 옵션 정보	도스 창에서 DIR[₩W]를 입력하세요.
	메뉴/기능 버튼 이름	[시작] - [모든 프로그램] - [관리도구]
" "	용어 정의에서 정의한 고유 명사	“scpdb_agent”으로 입력하고,
' '	강조하는 말이나 글 앞/뒤에 사용	'admin' 계정은 WAPPLES에 추가되어 있는

표기법	설명	예
-	특정 기능을 수행하기 위해 진입하는 메뉴 또는 기능의 이름을 분리할 때	[키 관리] - [데이터 암호화 키] - [변경]

회사 안내

본사

서울특별시 영등포구 국제금융로2길 25 ㈜유수홀딩스 20층
TEL: 02.780.7728 FAX: 02.786.5281
www.pentasecurity.com
펜타시큐리티시스템㈜

일본 지사

도쿄도 신주쿠구 요츠야 욘쵸메 3-20, 이찌고 요츠야 욘초메 빌딩 3층
TEL: 81.3.5573.8191 FAX: 81.3.5573.8193
www.pentasecurity.co.jp
Penta Security Systems K.K

만든 사람들

작성 및 편집: 디아모 개발부
표지 및 디자인: 디자인 팀
교정교열: 품질부

DA_제품 개요

D'Amo DA는, DBMS Application Encryption의 약자로 **API 형태의 DB 서버 암호화 제품**이다.

DA는 애플리케이션 서버에 암/복호화 API를 삽입하여 **DBMS의 부담을 최소화** 함으로써 효과적인 **DB 보안** 기능을 제공한다.

주요 기능 및 특징

본 제품은, 다음과 같은 기능과 특징을 갖는다.

보안성

강력한 키 관리와 관리자 인증

- 하이브리드 암호화 방식을 사용하여 **암/복호화 키를 이중 암호화** 한다.
- 별도의 하드웨어를 통해 **강력한 키 관리 기능**을 지원한다.(D'Amo KMS 적용 시)

편의성

다양한 알고리즘과 GUI를 통한 편의성 증대

- 관리자 시스템에서 키 관리와 설정(D'Amo KMS 적용 시)을 가능하게 함으로써 **관리자 편의성**을 제공한다.
- 관리도구를 통해 **정책 설정** 및 **로그/시스템 현황** 정보를 제공한다.

- 직관적인 GUI와 CLI를 제공한다.

확장성

다양한 알고리즘과 환경 지원

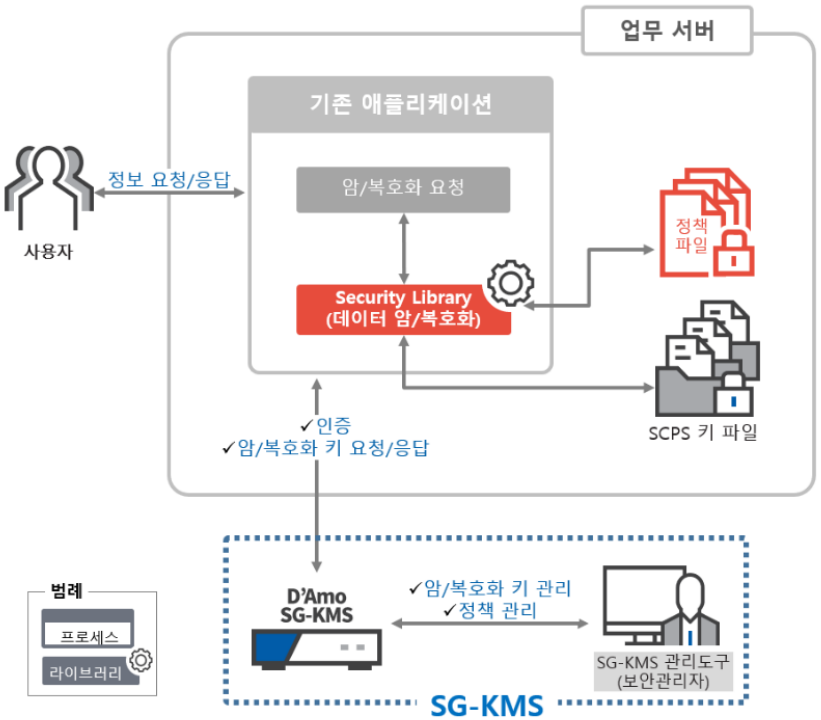
- **국내/외 표준 암호화 알고리즘**을 지원한다.
- **이기종(Heterogeneous) DBMS간 데이터 연동** 시, 데이터 암호 키가 다른 경우에도 안전한 연동이 가능하다.
- **관리 대상 DBMS 추가 시** D'Amo SCP Agent 설치만으로 **기존 암호화 관리 체계와 통합** 가능하다.

제품 구성

DA 시스템의 구성

DA를 구성하는 시스템의 각 요소와 주변 제품과의 관계를 아래에 나타낸다.

DA 시스템의 구성도



각 시스템의 구성 요소와 그 역할은 다음과 같다.

DA의 구성 요소와 역할

구성 요소	개요
정책 파일	권한 정책 파일로써 DBMS의 User를 기준으로 설정한다.
SCPS 키 파일	암호화 정책 파일로써 알고리즘, 운영 모드, 데이터 키 등 암호화에 필요한 정보가 저장된다.
SG-KMS	SG-KMS 관리도구와 통신 가능한 서버로서 암호화 정책을 관리한다.
SG-KMS 관리도구	SG-KMS와 통신할 수 있는 서버로서 해당 서버를 관리하는 GUI[a]를 제공한다.

a 그래픽 사용자 인터페이스(graphical user interface, GUI)

키 시스템

DA 키 시스템의 구조

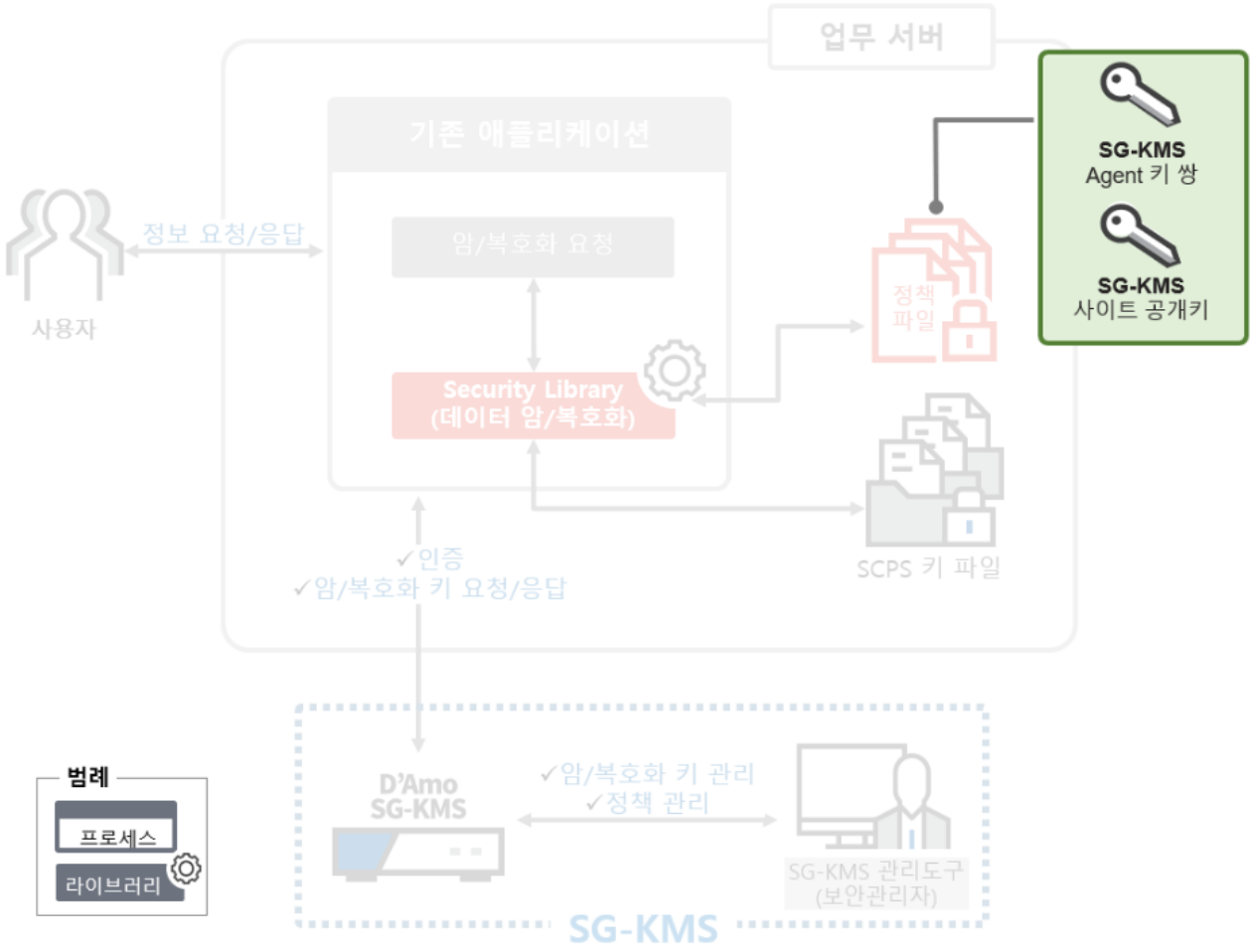
DA는 안전한 사용자 인증과 데이터 보호를 위해 **PKI 기반의 다양한 키를 생성/관리**하며, 그 구조는 다음과 같다.

DA 키 시스템의 구조도

DA 키 시스템의 구성

DA 사용을 위한 키의 위치는 다음과 같다.

DA 키 시스템의 구성도



각 키 시스템의 구성 요소와 역할은 다음과 같다.

DA 키 시스템의 구성 요소와 특징

구성 요소	개요
SG-KMS Agent 키 쌍	SG-KMS와 DA간(1:1) 인증을 위한 키 쌍이다.
SG-KMS 사이트 키 쌍	SG-KMS와 DA(1:n)가 연동하는 데에 필요한 인증 키 쌍이다.

DA의 동작 과정

DA의 설치부터 암/복호화까지의 대략적인 절차는 다음과 같다.

DA의 동작

파트 I.

설치 및 설정 안내서(SL4)

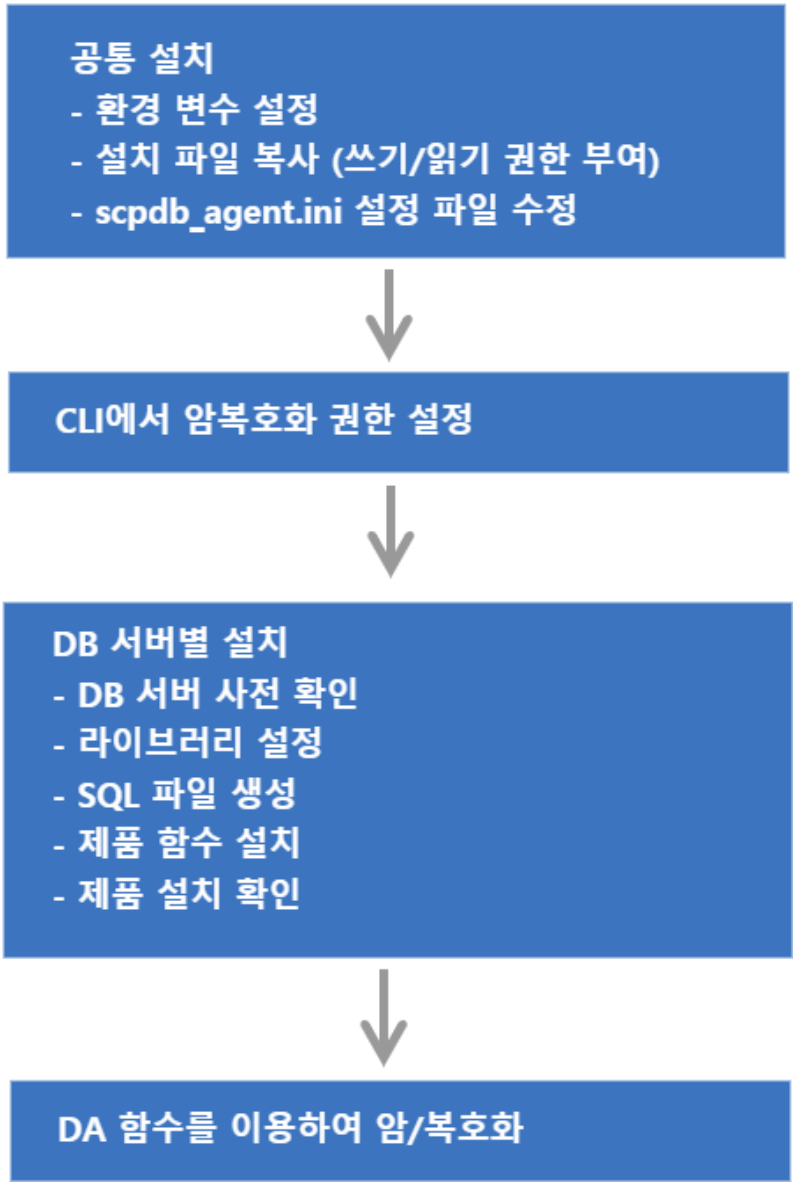
여기에서는 고객(또는 엔지니어)가 제품(소프트웨어, 하드웨어)를 설치하기 위한 간략한 절차가 기술되며, 제품의 동작을 확인하기 위한 최소한의 설정 방법을 포함한다.

1.

Oracle

DA는 DB 서버의 종류에 따라서 설치 및 운용 방법이 각각 다르다.

그림 1-1 설치 개념도



1.1 지원 운영체제 및 DB서버

DA에서 지원하는 운영체제 및 DB서버는 아래와 같다. 단, 특정 환경은 지원되지 않을 수도 있으므로, 제품 설치 전에 상세한 지원 가능 여부는 펜타시큐리티시스템으로 문의한다.

표 1-1 제품이 지원하는 운영체제 및 DB 서버 정보

구분	설명
운영체제	Windows, AIX, HP IA, HP PA-RISC, Linux, SUN, TRU64
DB 서버	ORACLE 8i ~, SQL Server 2012 ~, DB2 9.x ~, Tibero 4 SP 1 ~, MySQL 5.x ~, MariaDB 5.5, 10.0, 10.1, Cache DB 2009.1 ~, Informix IDS 9.x ~ Sybase ASE 15.7 SP61 ~, Sybase IQ 15.4 ~, CUBRID 2008 R1.3 ~ , PostgreSQL 9.4 ~

1.2 설치 파일의 구성 확인

DA의 설치 파일은 아래와 같은 명칭으로 압축 파일(zip) 형태로 제공된다.

- DA 설치파일: Install_DAmo_DA_v{버전}.zip
  - 압축을 해제 한 뒤, 설치 대상 DB 서버, OS 및 bit에 맞는 설치 파일을 준비한다.

설치 파일에 제품을 사용할 수 있는 '라이선스'는 포함되어 있지 않다.
펜타시큐리티시스템에 문의하여 '라이선스' 파일은 별도로 준비한다.

표 1-2 설치 파일(Install_DAmo_DA_v{버전}.zip)을 압축 해제 시, 디렉터리의 구성

구성	설명
_SampleScpsFiles	SG-KMS 연동 없이 암복호화를 테스트할 수 있는 테스트 키 파일
_TestAgentKeyPair	CLI에서 사용할 수 있는 테스트 키 쌍
Altibase	DA-ALT 제품의 설치 바이너리 폴더
Cache	DA-CDB 제품의 설치 바이너리 폴더
Cubrid	DA-CUB 제품의 설치 바이너리 폴더
DB2	DA-DB2 제품의 설치 바이너리 폴더
Informix	DA-IFX 제품의 설치 바이너리 폴더
MySQL	DA-MYQ 제품의 설치 바이너리 폴더
Oracle	DA-ORA 제품의 설치 바이너리 폴더
Postgres	DA-PGS 제품의 설치 바이너리 폴더
SQL Server	DA-MSQ 제품의 설치 바이너리 폴더

구성	설명
SybaseASE	DA-SYB 제품의 설치 바이너리 폴더
SybaseIQ	DA-SIQ 제품의 설치 바이너리 폴더
Tibero	DA-TIB 제품의 설치 바이너리 폴더

1.2.1 DB 서버 및 운영체제 별 SQL파일 구성

각 DB 서버 및 운영체제 별 SQL파일 구성은 다음과 같다. 다음 장에서 각 DB별로 SQL파일 설치 방법을 설명한다.

표 1-3 DB 서버 및 운영체제 별 SQL파일

DB 서버 종류	Linux 일 경우	Windows 일 경우
Oracle	000.da_user.pkg 000.da_user.sql 001.inner_function.ora(c 버전 설치 시) 002.user_interface.ora(JAVA 버전 설치 시) 002.user_interface_java.ora 003.grant_execute_functions.sql(JAVA 버전 설치 시) 005.securej_privilege.sql install_make.sh	000.da_user.pkg 000.da_user.sql 001.inner_function.ora(c 버전 설치 시) 002.user_interface.ora(JAVA 버전 설치 시) 002.user_interface_java.ora 003.grant_execute_functions.sql(JAVA 버전 설치 시) 005.securej_privilege.sql 009.da_test.sql
MYSQL	001.inner_function.mys 002.user_interface. mys 003.grant_execute_functions.sql 009.da_test.sql install_make.sh	001.inner_function.mys 002.user_interface.mys 003.grant_execute_functions.sql
TIBERO	001.inner_function.tbs(c 버전 설치 시) 001.inner_function_java.tbs(JAVA 버전 설치 시) 002.user_interface.tbs(c 버전 설치 시) 002.user_interface_java.tbs(JAVA 버전 설치 시) 003.grant_execute_functions.sql install_make.sh	001.inner_function_java.tbs 002.user_interface_java.tbs 003.grant_execute_functions.sql
INFORMIX	001.inner_function.ifx 002.user_interface. ifx 003.grant_execute_functions.sql 009.da_test.sql install_make.sh	해당 없음
POSTGRESQL	001.inner_function.post 002.user_interface.post 003.grant_execute_functions.post	해당 없음

DB 서버 종류	Linux 일 경우	Windows 일 경우
DB2	001.inner_function.db2 002.user_interface.db2 003.grant_execute_functions.sql install_make.sh	001.inner_function.db2 002.user_interface.db2 003.grant_execute_functions.sql
CUBRID	001.inner_function.sql 002.user_interface.sql	001.inner_function.sql 002.user_interface.sql
SYBASE	001.inner_function.sybase 002.user_interface.sybase 003.grant_execute_functions.sql install_make.sh	해당 없음
SYBASE IQ	001.inner_function.sybiq 002.user_interface. sybiq 003.grant_execute_functions.sql install_make.sh	해당 없음
SQL Server	해당 없음	001.inner_function.sql 002.user_interface. sql 003.grant_execute_functions.sql install_make.bat
Cache DB	SCP.xml install_make.sh	SCP.xml
Altibase	해당 없음	000.da_user.pkg 000.da_user.sql 001.inner_function.sql 002.user_interface. sql install_make.bat

1.2.2 공통 설치 파일 구성

환경에 관계 없이 공통적으로 사용하는 설치 파일은 아래와 같다.

표 1-4 공통 설치 파일 (sql파일 제외)

파일 분류	파일 명	파일 용도
Library 파일	libdamoscpdb.{so|a|sl|dll}	DA 메인 라이브러리. 주로 DBMS External Interface를 담당
	libdamocm-4.0.{so|a|sl|dll}	공통 모듈 라이브러리
	liblogw-0.2.{so|a|sl|dll}	로그를 기록하는 라이브러리
	libcis_cc-3.3.{so|a|sl|dll}	암호화, 복호화 기능을 제공하는 라이브러리
	libcis_ce-3.3.{so|a|sl|dll}	암호화, 복호화를 제외한 부가적인 기능을 제공하는 라이브러리(예: Base64, 인증서 관리, 특성 유지 암호화 등)

파일 분류	파일 명	파일 용도
설정 파일	scpdb_agent.ini	DA 구동시 실행에 필요한 설정정보를 참조
License 파일	damo_lic.cer	DA 구동 시 제품의 유효성을 검증하는데 사용
Agent key 파일	damo_agt_site.cer	SG-KMS 연동, CLI 프로그램에서 사용하는 인증서 쌍
	damo_agt.cer	
	damo_agt.key	
접근제어 파일	acl_cli 파일	DB 의 USER 별로 암·복호 권한을 설정하는데에 사용
	privilege.damo	권한 파일
JAVA class 파일 (Oracle, Tibero, Cubrid 설치 가능)	ScpAgentException.class	예외 처리 Class
	ScpCryptData.class	암호화 복호화 Class
SQL 파일	아래 새로운 표에 DB별로 표기함	

Agent Key 파일은 SG-KMS 연동에 필요한 키 발급를 참고하여 발급 받는다.

DA-PGS(PostgreSQL)의 경우, DB 서버 버전에 따라 libdamoscpdb.so 라이브러리 선택

- libdamoscpdb94.so (Postgres 9.4)
- libdamoscpdb95.so (Postgres 9.5)
- libdamoscpdb95AS.so (EDB Postgres 9.5)
- libdamoscpdb96AS.so (EDB Postgres 9.6)
- libdamoscpdb10.so (Postgres 10)

1.3 환경변수 설정

DA를 설치할 운영체제에 환경변수 DA_INST_HOME를 설정한다. 이 매뉴얼에서는 제품 설치 경로를 아래와 같이 가정하여 설명한다.

- Linux 환경일 경우: /home/dbms_api
- Windows 환경일 경우: E:₩dbms_api

주의) DA_INST_HOME 설정 시, 주의 사항

- 리눅스의 경우 "/root" 디렉토리로 설정을 권장하지 않는다.
- 윈도우의 경우 "바탕화면"으로 설정을 권장하지 않는다.

위의 경로로 설정할 경우 접근 권한 등의 이유로 문제가 발생할 가능성이 존재한다.

1.3.1 환경변수 설정 - Linux 환경일 경우

DA_INST_HOME 환경변수에 DA의 설치 디렉터리를 설정한다.

```
.profile을 사용하는 경우
export DA_INST_HOME=/home/dbms_api


.cshrc를 사용하는 경우
setenv DA_INST_HOME=/home/dbms_api

export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$DA_INST_HOME
```

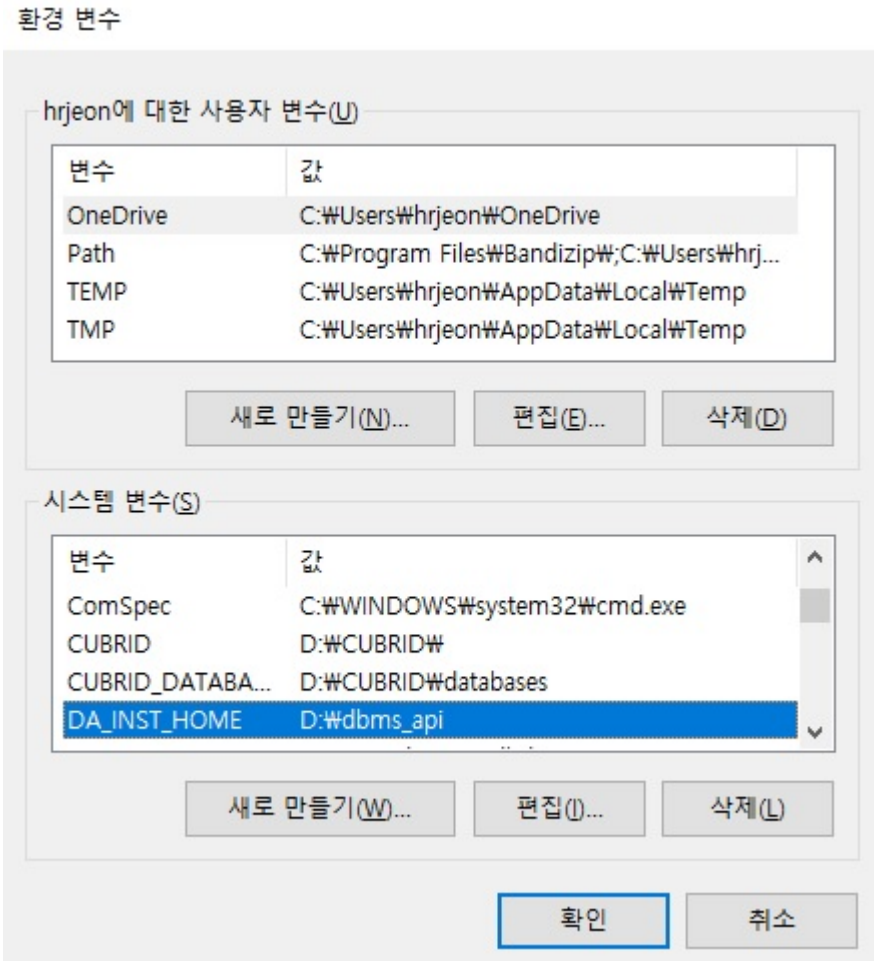
표 1-5 운영 체제별 라이브러리 PATH 명칭

운영 체제	라이브러리 PATH 명칭
HP_UX	SHLIB_PATH
AIX	LIBPATH
LINUX, SUN	LD_LIBRARY_PATH

1.3.2 환경변수 설정 - Windows 환경일 경우

[탐색기->내컴퓨터->등록정보->고급 탭->환경변수] 를 선택하여 나타나는 환경변수 설정 다이얼로그에서 시스템변수 DA_INST_HOME 변수와 값을 추가한다.

그림 1-2 DA_INST_HOME



1.4 DA의 설치 파일 복사

$DA_INST_HOME 디렉터리에 위 [설치 파일의 구성 확인]에서 나열된 파일들을 복사한다.

1.5 라이선스 파일 설정

$DA_INST_HOME 디렉토리에 라이선스 파일(damo_lic.cer)을 복사한다.

라이선스 파일명이 damo_lic.cer가 아닌 다른 이름으로 저장되어 있다면 변경해야 한다.

1.6 설치 파일의 접근 권한 부여 (Linux 설치 시)

Linux 사용자 계정에 DA 설치 파일(라이브러리, 실행 파일, 디렉토리)의 접근 권한을 부여한다. Windows에서 제품을 설치 할 경우, 이 과정은 생략한다.

```
$> cd $DA_INST_HOME
$> chmod 755 lib* acl_cli sql/install_make.sh
```

1.7 (SG-KMS 연동 시) SG-KMS에서 DA 정보 등록 및 연동에 필요한 키 내보내기

DA는 데이터를 암·복호화하기 위해 '암호화 키'가 필요하다. '암호화 키'를 얻기 위해서는 SG-KMS를 연동하거나 SCPS라는 암호화 키 파일을 이용할 수 있는데,
다음 설명은 SG-KMS 연동을 위해 SG-KMS에서 DA 정보를 등록하고 연동에 필요한 키를 내보내는 방법에 대해서 설명한다.

1.7.1 사전 준비

1.7.1.1 지원 SG-KMS 버전

DA와 연동 가능한 SG-KMS 버전은 다음과 같다.

표 1-6 연동 가능한 SG-KMS 버전

SG-KMS Major version	SG-KMS Minor version
SG-KMS v3.0	v3.0.9.0 이상 연동 가능
SG-KMS v4.0	v4.0.104.5 이상 연동 가능

SG-KMS v2.3 연동은 미지원한다.

1.7.1.2 연동 전 점검 항목

SG-KMS 연동을 위해서는 다음과 같이 사전 준비가 필요하다.

- SG-KMS 관리도구
- SG-KMS 관리도구에 접속할 수 있는 ID와 비밀번호
- SG-KMS 매뉴얼

이 매뉴얼에서는 SG-KMS의 키 종류 및 SG-KMS 관리도구 사용 방법에 대한 내용은 다루지 않으므로 **[SG-KMS 매뉴얼]**을 참고한다.

1.7.1.3 SG-KMS 연동에 필요한 키 발급

DA와 SG-KMS의 연동을 위해서는 먼저 SG-KMS에 DA를 Agent로 등록해야한다. 등록하는 절차는 SG-KMS 사용자설명서의 '대칭키를 사용하는 D'Amo Agent 등록 안내서'를 참고한다.

[D'Amo Agent 키] 파일, [Agent ID(CN)]와 [서비스 ID]정보는 DA와 SG-KMS 연동을 위한 설정 과정에서 필요하다. SG-KMS 관리자는 DA 설치 엔지니어에게 안전하게 전달한다.

SG-KMS 관리자에게 받은 D'Amo Agent의 인증서 및 키 파일 명을 다음과 같이 변경후 $DA_INST_HOME/key 디렉터리에 복사한다.

표 1-7 SG-KMS 연동에 필요한 키 목록

키 구분	생성된 키 파일명	변경할 키 파일명
사이트 키	damo-site_{발행기관/부서명}-SITE_V3.cer	damo_agt_site.cer
Agent 키	damo-scp_SITE_V3-{Agent 이름}.cer	damo_agt.cer
	damo-scp_SITE_V3-{Agent 이름}.key	damo_agt.key
	damo-scp_SITE_V3-{Agent 이름}.spin	damo_agt.spin

Agent 키 경로 지정, 키 이름 변경은 반드시 필요한 작업은 아니지만 관리를 위해 변경하는 것을 권장한다.

1.8 설정 파일(scpdb_agent.ini) 수정

DA의 운용을 위해 사용하는 scpdb_agent.ini 설정 파일 수정 방법에 대해 설명한다. DA는 SG-KMS를 이용하거나 SCPS 파일을 이용하여 암호화 키를 얻기 때문에 고객의 환경에 맞게 설정해야 한다.

$DA_INST_HOME 디렉터리에 있는 scpdb_agent.ini 설정 파일에서 아래 항목의 값을 수정한다.

1. 설정 파일의 [KEYINFO] 항목 - [KEY1]에 암호화 키 정보를 입력한다.

```
[KEYINFO]
KEY1=암복호화 하려는 암호화 키 정보를 입력한다.
//키 정보는 다음과 같은 값을 입력할 수 있다.
///ServiceID: SG-KMS에 생성한 서비스 ID를 입력한다.
///SCP_FilePath: SG-KMS에서 서비스 내보내기를 통해 발급한 SCPS 파일의 절대 경로 및 파일명, 확장자를 입력한다.

//*예제 - Windows 경우
KEY1=DA_AES256
KEY2=C:\DA\Policy\S_AES128.SCPS\DA_AES256.scps
KEY3=DA_AES256,C:\DA\Policy\S_AES128.SCPS\DA_AES256.scps

//*예제 - Linux 또는 Unix 경우
KEY1=DA_AES256
KEY2=/home/dbms_api/key/DA_AES256.scps
KEY3=DA_AES256,/home/dbms_api/key/DA_AES256.scps
```

ServiceID를 입력 할 경우 SG-KMS와 통신을 통해 암호화 정책을 가져와 데이터 암복호화를 수행한다.

SCP_FilePath를 입력 할 경우 KMS와 통신하지 않고 서버의 SCP 파일을 통해 암호화 정책을 가져와 데이터 암복호화를 수행한다.

ServiceID와 SCPS 파일명을 동시에 입력 시, SG-KMS와 네트워크 연결 실패 하면 SCPS 파일을 이용하여 암호화 한다.
ServiceID와 SCPS 파일명 사이에 공백이 있으면 SCPS 파일을 읽을 수 없다. 따라서 예제와 같이 띄어쓰기를 하지 않고 ServiceID와 SCPS 파일 경로를 붙여서 입력한다.

2. 설정 파일의 [Server], [Server2] 항목을 수정한다.

```
[Server]
ServerIP: SG-KMS의 IP를 입력한다.
ServerPort: SG-KMS의 포트번호를 입력한다. 기본값은 2525이다.

//*예제 - 모든 OS 공통
```

```
6  ServerIP=192.168.22.25
7  ServerPort=2525
```

```
1  [Server2]
2  ServerIP: 이중화를 위한 2번 SG-KMS의 IP를 입력한다.
3  ServerPort: 이중화를 위한 2번 SG-KMS의 포트번호를 입력한다. 기본값은 2525이다.
4
5  //*예제 - 모든 OS 공통
6  ServerIP=192.168.22.26
7  ServerPort=2525
```

ServerIP는 최소 1개 ~ 최대 10개까지 등록이 가능하다.

3. 설정 파일의 [AGENT] 항목을 설정한다.

```
1  [AGENT]
2  AgentID=SG-KMS 관리도구에서 설정한 D'Amo Agent의 Agent ID
3  LogDir=로그가 저장될 디렉터리 위치
4  LogLevel=로그가 남는 수준
5  SiteCertFilePath=SG-KMS 장비에서 설정한 해당 장비의 사이트 공개키(.cer)
6  CertFilePath=SG-KMS 관리도구에서 설정한 D'Amo Agent의 공개키(.cer)
7  KeyFilePath=SG-KMS 관리도구에서 설정한 D'Amo Agent의 비공개키(.key)
8  SPIN=SG-KMS 관리도구에서 설정한 D'Amo Agent의 SPIN으로, damo-scp_SITE_V3-[Agent이름].spin
파일의 값
9
10 //[예제]
11 AgentID=DA
12 LogDir=/home/dbms_api/log
13 LogLevel=4
14 SiteCertFilePath=/home/dbms_api/key/damo_agt_site.cer
15 CertFilePath=/home/dbms_api/key/damo_agt.cer
16 KeyFilePath=/home/dbms_api/key/damo_agt.key
17 SPIN=XaMh1ylXUh123XUh
```

로그가 남는 수준(LogLevel)에는 아래 5가지 숫자 입력이 가능하며, 각 값의 설정은 다음과 같다.
- 0: 아무 로그도 남기지 않을 경우
- 2: 경고 로그를 파일에 기록
- 4: 에러 로그와 경고 로그를 파일에 기록

- 6: 정보 로그, 에러 로그, 경고 로그를 파일에 기록
- 8: 디버그, 정보 로그, 에러 로그, 경고 로그를 파일에 기록

제품 운영 중 scpdb_agent.ini 파일을 수정하면 CONFIG_REINIT() 함수를 호출해야 변경된 내용이 적용된다.

1.9 CLI에서 권한 설정

$DA_INST_HOME 디렉터리에서 acl_cli 파일을 실행하여 USER 단위로 암/복호화 권한을 설정한다. USER 는 DB의 소유자명이고, KEY는 설정 파일(scpdb_agent.ini)의 [KEYINFO] 중 ALIAS 값(예: KEY1)이다.

CLI 명령어를 자세히 보려면 help 명령어를 실행한다.

```
CLI 실행 방법
$> cd $DA_INST_HOME
$> ./acl_cli  - start
Enter the PIN of CLI-key. : damo_agt.key 의 비밀번호
```

```
권한 추가할 경우
D'Amo > SET PRIV ENC [USER]"[KEY]"1"1
D'Amo > SAVE ALL
D'Amo > SHOW ALL

예제)
D'Amo > SET PRIV ENC SCOTT"KEY1"1"1
D'Amo > SET PRIV ENC SCOTT"KEY2"1"1
```

scpdb_agent.ini 설정 파일이 아래 예제와 같을 경우, CLI에서 권한 설정할 때 입력해야 하는 2번째 [KEY] 인자 값에는 KEY1을 입력해야 한다. (※ARIA256을 입력하는 것이 아님)

#scpdb_agent.ini 설정 파일 예제
[KEYINFO]
KEY1=ARIA256

```
권한 삭제할 경우
D'Amo > DEL PRIV ENC [USER]"[KEY]
D'Amo > SAVE ALL
D'Amo > SHOW ALL
```

CLI 에서 권한을 추가하거나 삭제 한 경우 반드시 SAVE ALL 명령어를 실행하며 SHOW ALL 명령어를 이용하여 적용 여부를 확인 한다.

1.10 DB 서버 사전 확인

DA를 Oracle 환경에서 설치 하기 전에 다음과 같은 사항들을 확인한다.

- DB 서버 재시작 또는 DB 서버의 listener 재시작
- DB 엔진 설치 계정에 폴더 생성
- DB 사용자로 SCP 계정 생성 (권고사항)
- DB 서버의 DBA 권한 계정
- DB 엔진 설치 계정의 . profile 파일 수정 (Linux 일 경우)

1.11 라이브러리 설정

1.11.1 라이브러리 설정 (Linux / Unix 환경)

Oracle 환경은 JAVA 버전 또는 C버전으로 설치가 가능하다. $ORACLE_HOME/lib 디렉터리에 DA의 library 파일을 symbolic link 로 생성한다.

```
$> cd $ORACLE_HOME/lib
$> ln -s $DA_INST_HOME/libcis_cc-3.3.{so|a|sl} libcis_cc-3.3.{so|a|sl}
$> ln -s $DA_INST_HOME/libcis_ce-3.3.{so|a|sl} libcis_ce-3.3.{so|a|sl}
$> ln -s $DA_INST_HOME/liblogw-0.2.{so|a|sl} liblogw-0.2.{so|a|sl}
$> ln -s $DA_INST_HOME/libdamocm-4.0.{so|a|sl} libdamocm-4.0.{so|a|sl}
$> ln -s $DA_INST_HOME/libdamoscpdb.{so|a|sl} libdamoscpdb.{so|a|sl}
```

1.11.2 라이브러리 설정 (Windows 환경)

Oracle 환경은 JAVA 버전 또는 C버전으로 설치가 가능하다. C버전으로 설치 할 경우 이 과정을 생략한다. Windows의 JAVA 버전으로 설치할 경우, 설치 파일 중 아래 5개 파일을 %ORACLE_HOME%₩BIN 폴더에 복사한다.

- cis_cc-3.3.dll
- cis_ce-3.3.dll
- logw-0.2.dll
- damocm-4.0.dll
- damoscpdb.dll

1.12 sql 파일 생성

1. $DA_INST_HOME/sql 디렉터리에서 install_make.sh을 이용하여 설치할 sql 파일을 생성한다. D_INI 는 설정 파일(scpdb_agent.ini)의 경로다.

```
[Linux 환경일 경우]
$> cd $DA_INST_HOME/sql
$> ./install_make.sh D_INI

[예제]
$> ./install_make.sh /home/dbms_api
D_INI_PATH is replaced by /home/dbms_api
```

```
[Windows 환경일 경우 - cmd창에서]
cmd> cd $DA_INST_HOME/sql
cmd> install_make.bat D_INI

[예제]
cmd> install_make.bat C:\da
D_INI_PATH is replaced by C:\da
```

2. 아래 3개의 .sql 파일이 생성되었는지 확인한다.

- 001.inner_function.ora.sql
- 002.user_interface.ora.sql
- 002.user_interface_java.ora.sql

1.13 DB 서버에서 제품을 사용할 계정 생성 (권장사항)

DB에서 DA 함수를 사용할 계정를 생성하는 것을 권고한다. 이 매뉴얼에서는 'SCP' 계정을 생성하였다고 가정한다. 'SCP' 계정 생성 시, CONNECT, RESOUCE, CREATE LIBRARY 권한을 부여한다.

```
SQL> CREATE USER SCP IDENTIFIED BY [password];
User created.
SQL> GRANT CONNECT, RESOURCE, CREATE LIBRARY TO SCP;
Grant succeeded.
SQL>
```

1.14 제품 함수 설치

1. $DA_INST_HOME/sql 위치에서 DB의 'SCP' 계정으로 접속하여, DA의 LIBRARY 와 함수를 설치한다.

```
SQL> CREATE LIBRARY SECURE_SCP_LIB AS '/ fullpath_of_DA_INST_HOME/libdamoscpdb.{so¦a¦sl}';
/
SQL> START 000.da_user.pkg
Package created.
SQL> START 000.da_user.sql
Package body created.
SQL> START 001.inner_function.ora.sql
Function created.
□
Function created.
```

2. JAVA 버전 또는 C 버전 설치에 따라, 002 파일은 각각 다른 것을 실행해야 한다.

- JAVA 버전 설치 시: 002.user_interface_java.ora.sql 파일 실행
- C 버전 설치 시: 002.user_interface.ora.sql 파일 실행

```
[JAVA 버전일 경우]
SQL> START 002.user_interface_java.ora.sql
Function created.
□
Function created.
```

```
[C 버전일 경우]
SQL> START 002.user_interface.ora.sql
Function created.
□
Function created.
```

3. DB의 'SCP' 계정에게 함수 실행 권한을 부여한다. 모든 계정에게 함수 실행 권한을 부여할 때는 003.grant_execute_functions.sql 파일을 실행한다.

```
[모든 계정에게 함수 실행 권한을 부여할 경우]
SQL> START 003.grant_execute_functions.sql
Grant succeeded.
□
Grant succeeded.
```

4. JAVA 버전일 경우 추가적으로 DBA 권한 계정에서 005.securej_privilege.sql 파일을 실행한다.

```
[JAVA 버전일 경우만 실행]
SQL> connect / as sysdba
Connected.
SQL> START 005.securej_privilege.sql
PL/SQL procedure successfully completed.
```

5. JAVA 버전일 경우에는 클래스 파일을 DB 서버에서 인식하기 위해서 loadjava를 실행한다. (SQL 프롬프트가 아닌, Linux 쉘에서 실행해야 한다.)

```
$>loadjava -user SCP/password ScpAgentException.class
$>loadjava -user SCP/password ScpCryptData.class
```

6. C 버전일 경우, listener 와 tnsname.ora 파일을 설정 한 후 listener를 재시작한다. JAVA 버전으로 설치 할 경우 이 과정을 생략한다.

1.15 제품 설치 확인

DB 서버에서 암/복호화 함수를 호출하여 설치를 성공했는지 확인한다.

```
SQL> SELECT ENC_STR( 'KEY1', 'abc') FROM DUAL;
ENC_STR('KEY1', 'abc')
--------------------------------------------------
```

```
5F28A50AC15E441AFE2D91545BB6EAE3


SQL> SELECT DEC_STR( 'KEY1', ENC_STR( 'KEY1', 'abc')) FROM DUAL;
DEC_STR( 'KEY1', ENC_STR( 'KEY1', 'abc'))
--------------------------------------------------
abc
```

1.16 제품 운용

1.16.1 함수 설명

DA에서 제공되는 함수와 사용하는 방법에 대해서 설명한다.

1.16.1.1 파라미터 설명

1.16.1.1.1 I_KEY

I_KEY : 암호화/복호화 때 사용하는 암호화 키
- 설정 파일(scpdb_agent.ini)의 [KEYINFO] 중 ALIAS 값
- 아래 [설정 파일 예제]의 경우 ALIAS는 'KEY1'과 'KEY2'이다.
- ALIAS는 SCPS파일로 암호화 할 것인지, SG-KMS의 서비스 ID로 암호화 할 것인지 설정 가능하다.

[설정 파일 예제]
- 설정 파일(scpdb_agent.ini)의 [KEYINFO] 중 ALIAS 값
KEY1=AES256.SCPS
KEY2=ARIA256

[함수 사용 예제]
SELECT ENC_B64('KEY1', 'abc') FROM DUAL;

1.16.1.1.2 I_DATA

I_DATA : 평문(암호화 함수일 경우), 암호문(복호화 함수일 경우)

DA에서 제공하는 암호화 함수에서 성능 향상을 위해 라이브러리에서 정책 이름(KEYINFO ALIAS)을 바탕으로 Cash하여 동작한다. 정책 이름은 같은데 실제 키(대칭 키)가 다른 정책을 사용하면 Cash에서 이전 키로 복호화를 시도하여 오류가 발생한다. 이 상황을 해결하기 위해서는 데이터베이스 서버 재시작이 필요하다.

표 1-8 DA 함수 (ORACLE)

함수 명	입력		출력
ENC_STR	I_KEY	IN 문자열,	Hex String 암호문
	I_DATA	IN 문자열 (평문)	
ENC_B64	I_KEY	IN 문자열,	Base64 Encording 암호문
	I_DATA	IN 문자열 (평문)	
DEC_STR	I_KEY	IN 문자열,	평문
	I_DATA	IN 문자열 (Hex String 암호문)	
DEC_B64	I_KEY	IN 문자열,	평문
	I_DATA	IN 문자열 (base64 String 암호문)	
INDEX_STR	I_KEY	IN 문자열,	Hex String 암호문
	I_DATA	IN 문자열 (평문),	
	I_TYPE	IN 문자열 '' or 'IX '(Plug-IN 연동 시 사용)	
DEC_INDEX_STR	I_KEY	IN 문자열,	Hex String 암호문 입력받아 OPE 데이터
	I_DATA	IN 문자열 (암호문),	
	I_TYPE	IN 문자열 '' or 'IX '(Plug-IN 연동 시 사용)	
DEC_INDEX_B64	I_KEY	IN 문자열,	Base64 Encording 암호문 입력받아 OPE 데이터
	I_DATA	IN 문자열 (암호문),	
	I_TYPE	IN 문자열 '' or 'IX '(Plug-IN 연동 시 사용)	

HASH_STR	I_ALOG	IN 숫자,	SHA1 =70	Hex String 해쉬 암호문
			SHA256 =71	
			SHA384 =72	
			SHA512 =73	
			HAS160 =74	
	I_DATA	IN 문자열		
HASH_B64	I_ALOG	IN 숫자,	SHA1 =70	Base64 String 해쉬 암호문
			SHA256 =71	
			SHA384 =72	
			SHA512 =73	
			HAS160 =74	
	I_DATA	IN 문자열		
HEXTOB64	I_DATA	IN 문자열 (Hex String 암호문)		base64 Encording 암호문
B64TOHEX	I_DATA	IN 문자열 (base64 Encording 암호문)		Hex String 암호문
CONFIG_REINIT				성공시 'SUCCESS', 그 외 에러
ENCRRN_B64	I_KEY	IN 문자열,		Base64 Encording 암호문
	I_DATA	IN 문자열 (평문)		
DECIG_B64	I_KEY	IN 문자열		평문
	I_DATA	IN 문자열 (평문)		

1.16.2 함수 호출 예제

1. ENC_STR

```
SQL> SELECT ENC_STR('KEY1', 'abc') FROM DUAL;
```

2. ENC_B64

```
SQL> SELECT ENC_B64('KEY1', 'abc') FROM DUAL;
```

3. DEC_STR

```
SQL> SELECT DEC_STR('KEY1', ENC_STR('KEY1', 'abc')) FROM DUAL;
```

4. DEC_B64

```
SQL> SELECT DEC_B64('KEY1', ENC_B64('KEY1', 'abc')) FROM DUAL;
```

5. INDEX_STR

```
DP 제품에 연동 하지 않을 경우
SQL> SELECT INDEX_STR('KEY1', 'abc', '') FROM DUAL;

DP 제품에 연동 할 경우
SQL> SELECT INDEX_STR('KEY1', 'abc', 'IX') FROM DUAL;
```

6. DEC_INDEX_STR, DEC_INDEX_B64

```
[DP 제품에 연동 하지 않을 경우]
SQL> SELECT DEC_INDEX_STR('KEY1', ENC_STR('KEY1', 'abc'), '') FROM DUAL;
SQL> SELECT DEC_INDEX_B64('KEY1', ENC_B64('KEY1', 'abc'), '') FROM DUAL;

[DP 제품에 연동 할 경우]
SQL> SELECT DEC_INDEX_STR('KEY1', ENC_STR('KEY1', 'abc'), 'IX') FROM DUAL;
SQL> SELECT DEC_INDEX_B64('KEY1', ENC_B64('KEY1', 'abc'), 'IX') FROM DUAL;
```

7. HASH_STR

```
SQL> SELECT HASH_STR( 71, 'abc' ) FROM DUAL;
```

8. HASH_B64

```
SQL> SELECT HASH_B64( 71, 'abc' ) FROM DUAL;
```

9. HEXTOB64

```
SQL> SELECT HEXTOB64('A305378D8F974F1C1537ED7CB0CB959245D1AC31') FROM DUAL;
```

10. B64TOHEX

```
SQL> SELECT B64TOHEX('owU3jY+XTxwVN+18sMuVkkXRrDE=') FROM DUAL;
```

11. CONFIG_REINIT

```
SQL> SELECT CONFIG_REINIT() FROM DUAL;
```

12. ENCRRN_B64

```
SQL> SELECT ENCRRN_B64('KEY1', '주민번호') FROM DUAL;
```

ENCRRN_B64 함수를 사용하기 위해서는 scpdb_agent.ini 설정 파일의 DiscernMode를 다음과 같이 활성화 해야 한다.

DiscernMode=1

13. DECIG_B64

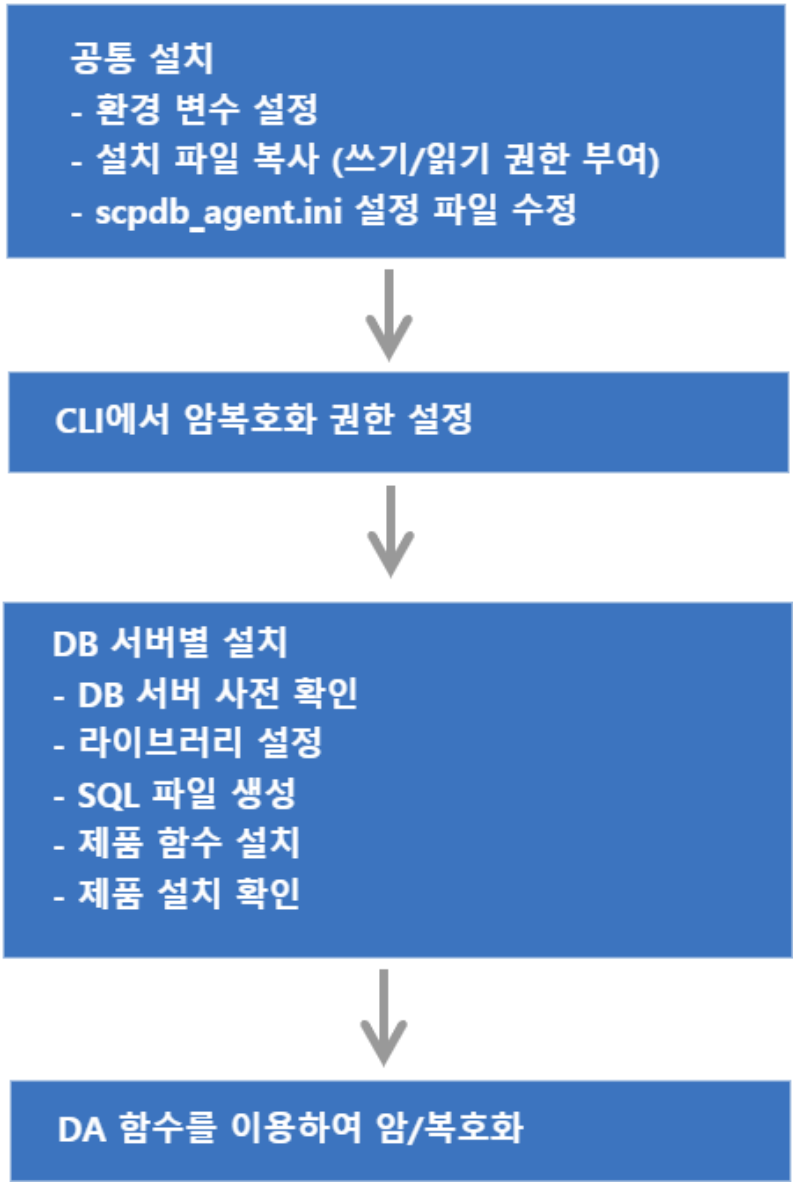
```
SQL> SELECT DECIG_B64('KEY1', '암호값') FROM DUAL;
```

2.

MySQL

DA는 DB 서버의 종류에 따라서 설치 및 운용 방법이 각각 다르다.

그림 2-1 설치 개념도



2.1 지원 운영체제 및 DB서버

DA에서 지원하는 운영체제 및 DB서버는 아래와 같다. 단, 특정 환경은 지원되지 않을 수도 있으므로, 제품 설치 전에 상세한 지원 가능 여부는 펜타시큐리티시스템으로 문의한다.

표 2-1 제품이 지원하는 운영체제 및 DB 서버 정보

구분	설명
운영체제	Windows, AIX, HP IA, HP PA-RISC, Linux, SUN, TRU64
DB 서버	ORACLE 8i ~, SQL Server 2012 ~, DB2 9.x ~, Tibero 4 SP 1 ~, MySQL 5.x ~, MariaDB 5.5, 10.0, 10.1, Cache DB 2009.1 ~, Informix IDS 9.x ~ Sybase ASE 15.7 SP61 ~, Sybase IQ 15.4 ~, CUBRID 2008 R1.3 ~ , PostgreSQL 9.4 ~

2.2 설치 파일의 구성 확인

DA의 설치 파일은 아래와 같은 명칭으로 압축 파일(zip) 형태로 제공된다.

- DA 설치파일: Install_DAmo_DA_v{버전}.zip
  - 압축을 해제 한 뒤, 설치 대상 DB 서버, OS 및 bit에 맞는 설치 파일을 준비한다.

설치 파일에 제품을 사용할 수 있는 '라이선스'는 포함되어 있지 않다.
펜타시큐리티시스템에 문의하여 '라이선스' 파일은 별도로 준비한다.

표 2-2 설치 파일(Install_DAmo_DA_v{버전}.zip)을 압축 해제 시, 디렉터리의 구성

구성	설명
_SampleScpsFiles	SG-KMS 연동 없이 암복호화를 테스트할 수 있는 테스트 키 파일
_TestAgentKeyPair	CLI에서 사용할 수 있는 테스트 키 쌍
Altibase	DA-ALT 제품의 설치 바이너리 폴더
Cache	DA-CDB 제품의 설치 바이너리 폴더
Cubrid	DA-CUB 제품의 설치 바이너리 폴더
DB2	DA-DB2 제품의 설치 바이너리 폴더
Informix	DA-IFX 제품의 설치 바이너리 폴더
MySQL	DA-MYQ 제품의 설치 바이너리 폴더
Oracle	DA-ORA 제품의 설치 바이너리 폴더
Postgres	DA-PGS 제품의 설치 바이너리 폴더
SQL Server	DA-MSQ 제품의 설치 바이너리 폴더

구성	설명
SybaseASE	DA-SYB 제품의 설치 바이너리 폴더
SybaseIQ	DA-SIQ 제품의 설치 바이너리 폴더
Tibero	DA-TIB 제품의 설치 바이너리 폴더

2.2.1 DB 서버 및 운영체제 별 SQL파일 구성

각 DB 서버 및 운영체제 별 SQL파일 구성은 다음과 같다. 다음 장에서 각 DB별로 SQL파일 설치 방법을 설명한다.

표 2-3 DB 서버 및 운영체제 별 SQL파일

DB 서버 종류	Linux 일 경우	Windows 일 경우
Oracle	000.da_user.pkg 000.da_user.sql 001.inner_function.ora(c 버전 설치 시) 002.user_interface.ora(JAVA 버전 설치 시) 002.user_interface_java.ora 003.grant_execute_functions.sql(JAVA 버전 설치 시) 005.securej_privilege.sql install_make.sh	000.da_user.pkg 000.da_user.sql 001.inner_function.ora(c 버전 설치 시) 002.user_interface.ora(JAVA 버전 설치 시) 002.user_interface_java.ora 003.grant_execute_functions.sql(JAVA 버전 설치 시) 005.securej_privilege.sql 009.da_test.sql
MYSQL	001.inner_function.mys 002.user_interface. mys 003.grant_execute_functions.sql 009.da_test.sql install_make.sh	001.inner_function.mys 002.user_interface.mys 003.grant_execute_functions.sql
TIBERO	001.inner_function.tbs(c 버전 설치 시) 001.inner_function_java.tbs(JAVA 버전 설치 시) 002.user_interface.tbs(c 버전 설치 시) 002.user_interface_java.tbs(JAVA 버전 설치 시) 003.grant_execute_functions.sql install_make.sh	001.inner_function_java.tbs 002.user_interface_java.tbs 003.grant_execute_functions.sql
INFORMIX	001.inner_function.ifx 002.user_interface. ifx 003.grant_execute_functions.sql 009.da_test.sql install_make.sh	해당 없음
POSTGRESQL	001.inner_function.post 002.user_interface.post 003.grant_execute_functions.post	해당 없음

DB 서버 종류	Linux 일 경우	Windows 일 경우
DB2	001.inner_function.db2 002.user_interface.db2 003.grant_execute_functions.sql install_make.sh	001.inner_function.db2 002.user_interface.db2 003.grant_execute_functions.sql
CUBRID	001.inner_function.sql 002.user_interface.sql	001.inner_function.sql 002.user_interface.sql
SYBASE	001.inner_function.sybase 002.user_interface.sybase 003.grant_execute_functions.sql install_make.sh	해당 없음
SYBASE IQ	001.inner_function.sybiq 002.user_interface. sybiq 003.grant_execute_functions.sql install_make.sh	해당 없음
SQL Server	해당 없음	001.inner_function.sql 002.user_interface. sql 003.grant_execute_functions.sql install_make.bat
Cache DB	SCP.xml install_make.sh	SCP.xml
Altibase	해당 없음	000.da_user.pkg 000.da_user.sql 001.inner_function.sql 002.user_interface. sql install_make.bat

2.2.2 공통 설치 파일 구성

환경에 관계 없이 공통적으로 사용하는 설치 파일은 아래와 같다.

표 2-4 공통 설치 파일 (sql파일 제외)

파일 분류	파일 명	파일 용도
Library 파일	libdamoscpdb.{so|a|sl|dll}	DA 메인 라이브러리. 주로 DBMS External Interface를 담당
	libdamocm-4.0.{so|a|sl|dll}	공통 모듈 라이브러리
	liblogw-0.2.{so|a|sl|dll}	로그를 기록하는 라이브러리
	libcis_cc-3.3.{so|a|sl|dll}	암호화, 복호화 기능을 제공하는 라이브러리
	libcis_ce-3.3.{so|a|sl|dll}	암호화, 복호화를 제외한 부가적인 기능을 제공하는 라이브러리(예: Base64, 인증서 관리, 특성 유지 암호화 등)

파일 분류	파일 명	파일 용도
설정 파일	scpdb_agent.ini	DA 구동시 실행에 필요한 설정정보를 참조
License 파일	damo_lic.cer	DA 구동 시 제품의 유효성을 검증하는데 사용
Agent key 파일	damo_agt_site.cer	SG-KMS 연동, CLI 프로그램에서 사용하는 인증서 쌍
	damo_agt.cer	
	damo_agt.key	
접근제어 파일	acl_cli 파일	DB 의 USER 별로 암·복호 권한을 설정하는데에 사용
	privilege.damo	권한 파일
JAVA class 파일 (Oracle, Tibero, Cubrid 설치 가능)	ScpAgentException.class	예외 처리 Class
	ScpCryptData.class	암호화 복호화 Class
SQL 파일	아래 새로운 표에 DB별로 표기함	

Agent Key 파일은 SG-KMS 연동에 필요한 키 발급를 참고하여 발급 받는다.

DA-PGS(PostgreSQL)의 경우, DB 서버 버전에 따라 libdamoscpdb.so 라이브러리 선택

- libdamoscpdb94.so (Postgres 9.4)
- libdamoscpdb95.so (Postgres 9.5)
- libdamoscpdb95AS.so (EDB Postgres 9.5)
- libdamoscpdb96AS.so (EDB Postgres 9.6)
- libdamoscpdb10.so (Postgres 10)

2.3 환경변수 설정

DA를 설치할 운영체제에 환경변수 DA_INST_HOME를 설정한다. 이 매뉴얼에서는 제품 설치 경로를 아래와 같이 가정하여 설명한다.

- Linux 환경일 경우: /home/dbms_api
- Windows 환경일 경우: E:₩dbms_api

주의) DA_INST_HOME 설정 시, 주의 사항

- 리눅스의 경우 "/root" 디렉토리로 설정을 권장하지 않는다.
- 윈도우의 경우 "바탕화면"으로 설정을 권장하지 않는다.

위의 경로로 설정할 경우 접근 권한 등의 이유로 문제가 발생할 가능성이 존재한다.

2.3.1 환경변수 설정 - Linux 환경일 경우

DA_INST_HOME 환경변수에 DA의 설치 디렉터리를 설정한다.

```
.profile을 사용하는 경우
export DA_INST_HOME=/home/dbms_api


.cshrc를 사용하는 경우
setenv DA_INST_HOME=/home/dbms_api

export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$DA_INST_HOME
```

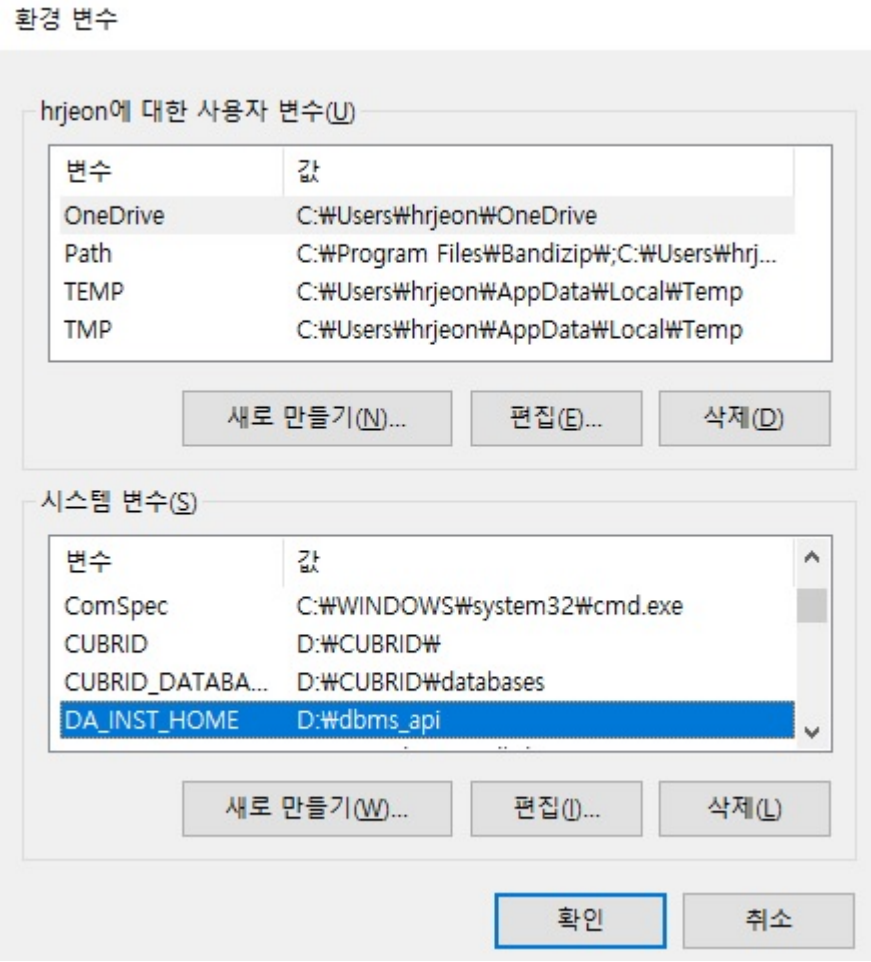
표 2-5 운영 체제별 라이브러리 PATH 명칭

운영 체제	라이브러리 PATH 명칭
HP_UX	SHLIB_PATH
AIX	LIBPATH
LINUX, SUN	LD_LIBRARY_PATH

2.3.2 환경변수 설정 - Windows 환경일 경우

[탐색기->내컴퓨터->등록정보->고급 탭->환경변수] 를 선택하여 나타나는 환경변수 설정 다이얼로그에서 시스템변수 DA_INST_HOME 변수와 값을 추가한다.

그림 2-2 DA_INST_HOME



2.4 DA의 설치 파일 복사

$DA_INST_HOME 디렉터리에 위 [설치 파일의 구성 확인]에서 나열된 파일들을 복사한다.

2.5 라이선스 파일 설정

$DA_INST_HOME 디렉토리에 라이선스 파일(damo_lic.cer)을 복사한다.

라이선스 파일명이 damo_lic.cer가 아닌 다른 이름으로 저장되어 있다면 변경해야 한다.

2.6 설치 파일의 접근 권한 부여 (Linux 설치 시)

Linux 사용자 계정에 DA 설치 파일(라이브러리, 실행 파일, 디렉토리)의 접근 권한을 부여한다. Windows에서 제품을 설치 할 경우, 이 과정은 생략한다.

```
$> cd $DA_INST_HOME
$> chmod 755 lib* acl_cli sql/install_make.sh
```

2.7 (SG-KMS 연동 시) SG-KMS에서 DA 정보 등록 및 연동에 필요한 키 내보내기

DA는 데이터를 암·복호화하기 위해 '암호화 키'가 필요하다. '암호화 키'를 얻기 위해서는 SG-KMS를 연동하거나 SCPS라는 암호화 키 파일을 이용할 수 있는데,
다음 설명은 SG-KMS 연동을 위해 SG-KMS에서 DA 정보를 등록하고 연동에 필요한 키를 내보내는 방법에 대해서 설명한다.

2.7.1 사전 준비

2.7.1.1 지원 SG-KMS 버전

DA와 연동 가능한 SG-KMS 버전은 다음과 같다.

표 2-6 연동 가능한 SG-KMS 버전

SG-KMS Major version	SG-KMS Minor version
SG-KMS v3.0	v3.0.9.0 이상 연동 가능
SG-KMS v4.0	v4.0.104.5 이상 연동 가능

SG-KMS v2.3 연동은 미지원한다.

2.7.1.2 연동 전 점검 항목

SG-KMS 연동을 위해서는 다음과 같이 사전 준비가 필요하다.

- SG-KMS 관리도구
- SG-KMS 관리도구에 접속할 수 있는 ID와 비밀번호
- SG-KMS 매뉴얼

이 매뉴얼에서는 SG-KMS의 키 종류 및 SG-KMS 관리도구 사용 방법에 대한 내용은 다루지 않으므로 **[SG-KMS 매뉴얼]**을 참고한다.

2.7.1.3 SG-KMS 연동에 필요한 키 발급

DA와 SG-KMS의 연동을 위해서는 먼저 SG-KMS에 DA를 Agent로 등록해야한다. 등록하는 절차는 SG-KMS 사용자설명서의 '대칭키를 사용하는 D'Amo Agent 등록 안내서'를 참고한다.

[D'Amo Agent 키] 파일, [Agent ID(CN)]와 [서비스 ID]정보는 DA와 SG-KMS 연동을 위한 설정 과정에서 필요하다. SG-KMS 관리자는 DA 설치 엔지니어에게 안전하게 전달한다.

SG-KMS 관리자에게 받은 D'Amo Agent의 인증서 및 키 파일 명을 다음과 같이 변경후 $DA_INST_HOME/key 디렉터리에 복사한다.

표 2-7 SG-KMS 연동에 필요한 키 목록

키 구분	생성된 키 파일명	변경할 키 파일명
사이트 키	damo-site_{발행기관/부서명}-SITE_V3.cer	damo_agt_site.cer
Agent 키	damo-scp_SITE_V3-{Agent 이름}.cer	damo_agt.cer
	damo-scp_SITE_V3-{Agent 이름}.key	damo_agt.key
	damo-scp_SITE_V3-{Agent 이름}.spin	damo_agt.spin

Agent 키 경로 지정, 키 이름 변경은 반드시 필요한 작업은 아니지만 관리를 위해 변경하는 것을 권장한다.

2.8 설정 파일(scpdb_agent.ini) 수정

DA의 운용을 위해 사용하는 scpdb_agent.ini 설정 파일 수정 방법에 대해 설명한다. DA는 SG-KMS를 이용하거나 SCPS 파일을 이용하여 암호화 키를 얻기 때문에 고객의 환경에 맞게 설정해야 한다.

$DA_INST_HOME 디렉터리에 있는 scpdb_agent.ini 설정 파일에서 아래 항목의 값을 수정한다.

1. 설정 파일의 [KEYINFO] 항목 - [KEY1]에 암호화 키 정보를 입력한다.

```
[KEYINFO]
KEY1=암복호화 하려는 암호화 키 정보를 입력한다.
//키 정보는 다음과 같은 값을 입력할 수 있다.
///ServiceID: SG-KMS에 생성한 서비스 ID를 입력한다.
///SCP_FilePath: SG-KMS에서 서비스 내보내기를 통해 발급한 SCPS 파일의 절대 경로 및 파일명, 확장자를 입력한다.

//*예제 - Windows 경우
KEY1=DA_AES256
KEY2=C:\DA\Policy\S_AES128.SCPS\DA_AES256.scps
KEY3=DA_AES256,C:\DA\Policy\S_AES128.SCPS\DA_AES256.scps

//*예제 - Linux 또는 Unix 경우
KEY1=DA_AES256
KEY2=/home/dbms_api/key/DA_AES256.scps
KEY3=DA_AES256,/home/dbms_api/key/DA_AES256.scps
```

ServiceID를 입력 할 경우 SG-KMS와 통신을 통해 암호화 정책을 가져와 데이터 암복호화를 수행한다.

SCP_FilePath를 입력 할 경우 KMS와 통신하지 않고 서버의 SCP 파일을 통해 암호화 정책을 가져와 데이터 암복호화를 수행한다.

ServiceID와 SCPS 파일명을 동시에 입력 시, SG-KMS와 네트워크 연결 실패 하면 SCPS 파일을 이용하여 암호화 한다.
ServiceID와 SCPS 파일명 사이에 공백이 있으면 SCPS 파일을 읽을 수 없다. 따라서 예제와 같이 띄어쓰기를 하지 않고 ServiceID와 SCPS 파일 경로를 붙여서 입력한다.

2. 설정 파일의 [Server], [Server2] 항목을 수정한다.

```
[Server]
ServerIP: SG-KMS의 IP를 입력한다.
ServerPort: SG-KMS의 포트번호를 입력한다. 기본값은 2525이다.

//*예제 - 모든 OS 공통
```

```
ServerIP=192.168.22.25
ServerPort=2525
```

```
[Server2]
ServerIP: 이중화를 위한 2번 SG-KMS의 IP를 입력한다.
ServerPort: 이중화를 위한 2번 SG-KMS의 포트번호를 입력한다. 기본값은 2525이다.

//*예제 - 모든 OS 공통
ServerIP=192.168.22.26
ServerPort=2525
```

ServerIP는 최소 1개 ~ 최대 10개까지 등록이 가능하다.

3. 설정 파일의 [AGENT] 항목을 설정한다.

```
[AGENT]
AgentID=SG-KMS 관리도구에서 설정한 D'Amo Agent의 Agent ID
LogDir=로그가 저장될 디렉터리 위치
LogLevel=로그가 남는 수준
SiteCertFilePath=SG-KMS 장비에서 설정한 해당 장비의 사이트 공개키(.cer)
CertFilePath=SG-KMS 관리도구에서 설정한 D'Amo Agent의 공개키(.cer)
KeyFilePath=SG-KMS 관리도구에서 설정한 D'Amo Agent의 비공개키(.key)
SPIN=SG-KMS 관리도구에서 설정한 D'Amo Agent의 SPIN으로, damo-scp_SITE_V3-[Agent이름].spin 파일의 값

//[예제]
AgentID=DA
LogDir=/home/dbms_api/log
LogLevel=4
SiteCertFilePath=/home/dbms_api/key/damo_agt_site.cer
CertFilePath=/home/dbms_api/key/damo_agt.cer
KeyFilePath=/home/dbms_api/key/damo_agt.key
SPIN=XaMh1ylXUh123XUh
```

로그가 남는 수준(LogLevel)에는 아래 5가지 숫자 입력이 가능하며, 각 값의 설정은 다음과 같다.
- 0: 아무 로그도 남기지 않을 경우
- 2: 경고 로그를 파일에 기록
- 4: 에러 로그와 경고 로그를 파일에 기록

- 6: 정보 로그, 에러 로그, 경고 로그를 파일에 기록
- 8: 디버그, 정보 로그, 에러 로그, 경고 로그를 파일에 기록

제품 운영 중 scpdb_agent.ini 파일을 수정하면 CONFIG_REINIT() 함수를 호출해야 변경된 내용이 적용된다.

2.9 CLI에서 권한 설정

$DA_INST_HOME 디렉터리에서 acl_cli 파일을 실행하여 USER 단위로 암/복호화 권한을 설정한다. USER 는 DB의 소유자명이고, KEY는 설정 파일(scpdb_agent.ini)의 [KEYINFO] 중 ALIAS 값(예: KEY1)이다.

CLI 명령어를 자세히 보려면 help 명령어를 실행한다.

```
CLI 실행 방법
$> cd $DA_INST_HOME
$> ./acl_cli  - start
Enter the PIN of CLI-key. : damo_agt.key 의 비밀번호
```

```
권한 추가할 경우
D'Amo > SET PRIV ENC [USER]"[KEY]"1"1
D'Amo > SAVE ALL
D'Amo > SHOW ALL

예제)
D'Amo > SET PRIV ENC SCOTT"KEY1"1"1
D'Amo > SET PRIV ENC SCOTT"KEY2"1"1
```

scpdb_agent.ini 설정 파일이 아래 예제와 같을 경우, CLI에서 권한 설정할 때 입력해야 하는 2번째 [KEY] 인자 값에는 KEY1을 입력해야 한다. (※ARIA256을 입력하는 것이 아님)

#scpdb_agent.ini 설정 파일 예제
[KEYINFO]
KEY1=ARIA256

```
권한 삭제할 경우
D'Amo > DEL PRIV ENC [USER]"[KEY]
D'Amo > SAVE ALL
D'Amo > SHOW ALL
```

CLI 에서 권한을 추가하거나 삭제 한 경우 반드시 SAVE ALL 명령어를 실행하며 SHOW ALL 명령어를 이용하여 적용 여부를 확인 한다.

2.10 DB 서버 사전 확인

DA를 MySQL 환경에서 설치 하기 전에 다음과 같은 사항들을 확인한다.

- DB 엔진 설치 계정에 폴더 생성
- DB 사용자로 SCP 계정 생성 (권고사항)
- DB 서버의 DBA 권한 계정
- DB 엔진 설치 계정의 . profile 파일 수정

2.11 라이브러리 설정

2.11.1 라이브러리 링크 설정 (Linux 설치 시)

4. [MYSQL 설치 디렉터리]/lib/plugin 디렉터리에 $DA_INST_HOME에 있는 libdamoscpdb.{so|a|sl} 파일을 symbolic link 로 생성한다.

```
$> cd /usr/local/mysql/lib/plugin
$> ln  -s $DA_INST_HOME/libdamoscpdb.{so|a|sl} libdamoscpdb.{so|a|sl}
```

5. /usr/lib64 디렉터리에 $DA_INST_HOME에 있는 libdamocm-0.4.{so|a|sl}, liblogw-0.2.{so|a|sl} 파일을 복사한다(root 계정 필요).

```
$> cp $DA_INST_HOME/libdamocm-0.4.{so|a|sl} /usr/lib64
$> cp $DA_INST_HOME/liblogw-0.2.{so|a|sl} /usr/lib64
```

6. /etc/ld.so.conf.d 디렉터리에서 mysql_damo.conf 파일을 생성 후 libdamoscpdb.{so|a|sl} 파일이 있는 경로를 추가하고 저장한다.

7. /etc 디렉터리에서 my.cnf 파일을 수정한다. plugin_dir 속성을 추가하고 [MYSQL 설치경로]/lib/plugin 경로를 추가한다.

8. ldconfig 명령어를 입력해서 변경 내용들을 적용한다.

ldconfig 명령어를 사용하면 '~~~.hmac' 파일 관련 오류가 발생한다. hmac 파일은 암호 모듈 무결성 값을 저장하는 TEXT 파일이기 때문에 제품 구동(LINK)과 상관없는 파일이다. hmac 파일 오류는 무시해도 된다

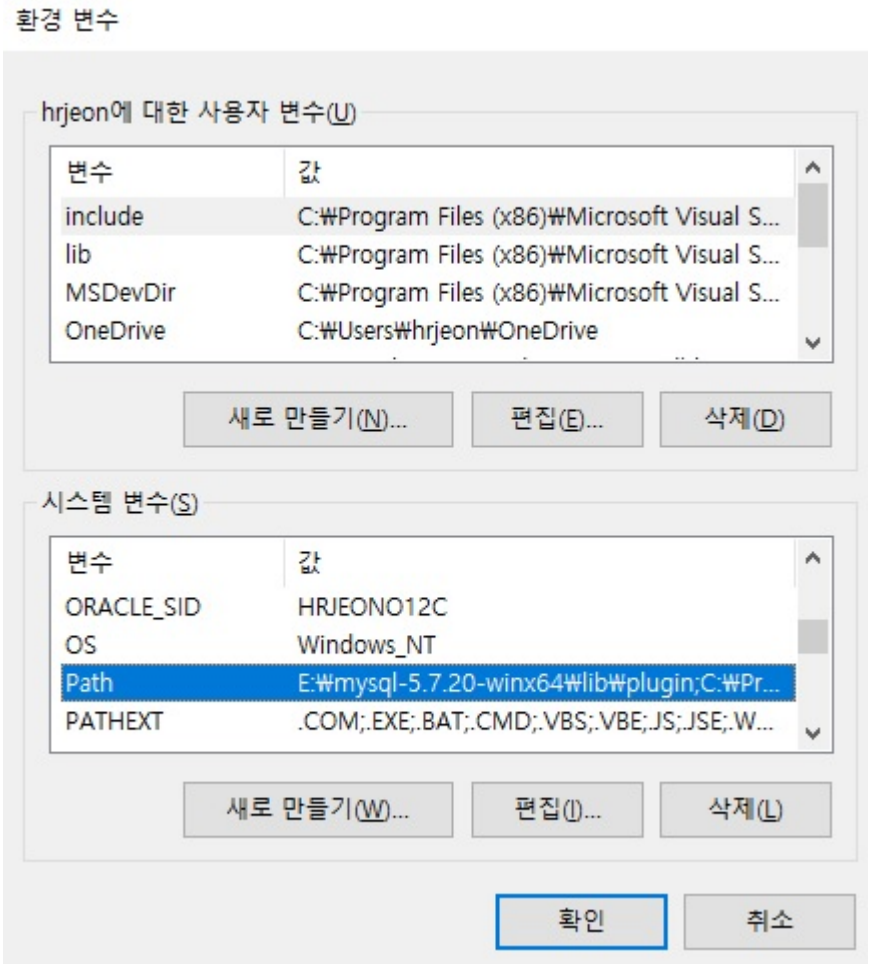
2.11.2 라이브러리 복사 및 설정 (Windows 설치 시)

1. [MYSQL 설치 경로]/lib/plugin 경로에 %DA_INST_HOME%에 있는 아래 5개 파일을 복사한다.

- cis_cc-3.3.dll
- cis_ce-3.3.dll
- logw-0.2.dll
- damocm-4.0.dll
- damoscpdb.dll

2. Windows 시스템 환경 변수 중 [PATH]에 [MYSQL 설치 경로]/lib/plugin 경로를 추가한다.

그림 2-3 환경변수 설정 다이얼로그



3. [MYSQL 설치 경로] 디렉터리에서 my.ini 파일을 생성 후 plugin_dir 속성을 추가하여 damoscpdb.dll 파일이 있는 경로를 설정한다.

```
[mysqld]
basedir=E:/mysql-5.7.20-winx64
datadir=[MYSQL 설치 경로]/data
port=3306

plugin_dir=[MYSQL 설치 경로]/lib/plugin
```

2.12 sql 파일 생성

1. $DA_INST_HOME/sql 디렉터리에서 install_make.sh을 이용하여 설치할 sql 파일을 생성한다. D_INI 는 설정

파일(scpdb_agent.ini)의 경로다.

```
$> cd $DA_INST_HOME/sql
$> ./install_make.sh D_INI

[예제]
$> ./install_make.sh /home/dbms_api
D_INI_PATH is replaced by /home/dbms_api
```

2. 아래 2개의 .sql 파일이 생성되었는지 확인한다.

- 001.inner_function.mys.sql
- 002.user_interface.mys.sql

2.13 DA의 함수 설치

1. $DA_INST_HOME/sql 위치에서 DB 접속 후 LIBRARY 와 함수를 설치한다. DB에 접속한 사용자 계정에 제품이 설치된다.

```
$>mysql [DBNAME] -u[USER] -p[PASSWORD]
Database selected.
Routine dropped.
...
Routine created.
Database closed.
$>mysql [DBNAME] -u[USER] -p[PASSWORD]
Database selected.
Routine dropped.
...
Routine created.
Database closed.
```

Linux에서 Security 기능(SELinux / Apparmor)이 활성화된 환경에서 함수 설치 중 오류(ERROR 1126)가 발생할 수 있다. MySQL(mysqld)에 대한 SELinux / Apparmor 해제(disable) 작업을 해야 한다.

2. 특정 DB 사용자에게 함수 실행 권한을 부여한다.

003.grant_execute_functions.sql 파일의 DB 사용자 default값은 'user'로 설정되어 있다.
따라서 'user'가 아닐 경우, 권한을 부여하고 싶은 사용자 계정명으로 편집 한 후 실행해야 한다.

```
$>mysql [DBNAME]  -u[USER]  -p[PASSWORD]
Database selected.
Permission granted.
Database closed.
$>
```

2.14 제품 설치 확인

DB 서버에서 암/복호화 함수를 호출하여 설치를 성공했는지 확인한다.

```
mysql> SELECT ENC_STR('KEY1', 'abc') AS RESULT;
+----------------------------------+
| RESULT                           |
+----------------------------------+
| E6878572B3287A049906A8CA57F0207C |
+----------------------------------+
1 row in set (0.00 sec)
mysql> SELECT DEC_STR('KEY1', ENC_STR('KEY1', 'abc')) AS RESULT;
+----------------+
| RESULT         |
+----------------+
| abc            |
+----------------+
1 row in set (0.00 sec)

mysql>
```

2.15 제품 운용

2.15.1 함수 설명

DA에서 제공되는 함수와 사용하는 방법에 대해서 설명한다.

2.15.1.1 파라미터 설명

2.15.1.1.1 I_KEY

I_KEY : 암호화/복호화 때 사용하는 암호화 키
- 설정 파일(scpdb_agent.ini)의 [KEYINFO] 중 ALIAS 값
- 아래 [설정 파일 예제]의 경우 ALIAS는 'KEY1'과 'KEY2'이다.
- ALIAS는 SCPS파일로 암호화 할 것인지, SG-KMS의 서비스 ID로 암호화 할 것인지 설정 가능하다.

[설정 파일 예제]
- 설정 파일(scpdb_agent.ini)의 [KEYINFO] 중 ALIAS 값
KEY1=AES256.SCPS
KEY2=ARIA256

[함수 사용 예제]
SELECT ENC_STR('KEY1', 'abc') AS RESULT;

2.15.1.1.2 I_DATA

I_DATA : 평문(암호화 함수일 경우), 암호문(복호화 함수일 경우)

DA에서 제공하는 암호화 함수에서 성능 향상을 위해 라이브러리에서 정책 이름(KEYINFO ALIAS)을 바탕으로 Cash하여 동작한다. 정책 이름은 같은데 실제 키(대칭 키)가 다른 정책을 사용하면 Cash에서 이전 키로 복호화를 시도하여 오류가 발생한다. 이 상황을 해결하기 위해서는 데이터베이스 서버 재시작이 필요하다.

표 2-8 DA 함수 (MYSQL)

함수 명	입력			출력
ENC_STR	I_KEY	IN 문자열		Hex String 암호문
	I_DATA	IN 문자열 (평문)		
ENC_B64	I_KEY	IN 문자열,		Base64 Encording 암호문
	I_DATA	IN 문자열 (평문)		
DEC_STR	I_KEY	IN 문자열,		평문
	I_DATA	IN 문자열 (Hex String 암호문)		
DEC_B64	I_KEY	IN 문자열,		평문
	I_DATA	IN 문자열 (base64 String 암호문)		
INDEX_STR	I_KEY	IN 문자열,		Hex String 암호문
	I_DATA	IN 문자열 (평문),		
	I_TYPE	IN 문자열 '' or 'IX '(Plug-IN 연동 시 사용)		
DEC_INDEX_STR	I_KEY	IN 문자열,		Hex String 암호문 입력받아 OPE 데이터
	I_DATA	IN 문자열 (암호문),		
	I_TYPE	IN 문자열 '' or 'IX '(Plug-IN 연동 시 사용)		
DEC_INDEX_B64	I_KEY	IN 문자열,		Base64 Encording 암호문 입력받아 OPE 데이터
	I_DATA	IN 문자열 (암호문),		
	I_TYPE	IN 문자열 '' or 'IX '(Plug-IN 연동 시 사용)		
HASH_STR	I_ALOG	IN 숫자,	SHA1 =70	Hex String 해쉬 암호문
			SHA256 =71	
			SHA384 =72	
			SHA512 =73	
			HAS160 =74	
	I_DATA	IN 문자열		
HASH_B64	I_ALOG	IN 숫자,	SHA1 =70	Base64 String 해쉬 암호문
			SHA256 =71	
			SHA384 =72	
			SHA512 =73	

		HAS160 =74	
	I_DATA	IN 문자열	
HEXTOB64	I_DATA	IN 문자열 (Hex String 암호문)	base64 Encording 암호문
B64TOHEX	I_DATA	IN 문자열 (base64 Encording 암호문)	Hex String 암호문
CONFIG_REINIT			성공시 'SUCCESS', 그외 에러

2.15.2 함수 호출 예제

1. ENC_STR

```
mysql> SELECT ENC_STR('KEY1', 'abc');
```

2. ENC_B64

```
mysql> SELECT ENC_B64('KEY1', 'abc');
```

3. DEC_STR

```
mysql> SELECT DEC_STR('KEY1', ENC_STR('KEY1', 'abc'));
```

4. DEC_B64

```
mysql> SELECT DEC_B64('KEY1', ENC_B64('KEY1', 'abc'));
```

5. INDEX_STR

```
[DP 제품에 연동 하지 않을 경우]
mysql> SELECT INDEX_STR('KEY1', 'abc', '');

[DP 제품에 연동 할 경우]
mysql> SELECT INDEX_STR('KEY1', 'abc', 'IX');
```

6. DEC_INDEX_STR, DEC_INDEX_B64

```
DP 제품에 연동 하지 않을 경우
mysql> SELECT DEC_INDEX_STR('KEY1', ENC_STR('KEY1', 'abc'), '');
mysql> SELECT DEC_INDEX_B64('KEY1', ENC_B64('KEY1', 'abc'), '');

DP 제품에 연동 할 경우
```

```
mysql> SELECT DEC_INDEX_STR('KEY1', ENC_STR('KEY1', 'abc'), 'IX');
mysql> SELECT DEC_INDEX_B64('KEY1', ENC_B64('KEY1', 'abc'), 'IX');
```

7. HASH_STR

```
mysql> SELECT HASH_STR( 71, 'abc' );
```

8. HASH_B64

```
mysql> SELECT HASH_B64( 71, 'abc' );
```

9. HEXTOB64

```
mysql> SELECT HEXTOB64('A305378D8F974F1C1537ED7CB0CB959245D1AC31');
```

10. B64TOHEX

```
mysql>SELECT B64TOHEX('owU3jY+XTxwVN+18sMuVkkXRrDE=');
```

11. CONFIG_REINIT

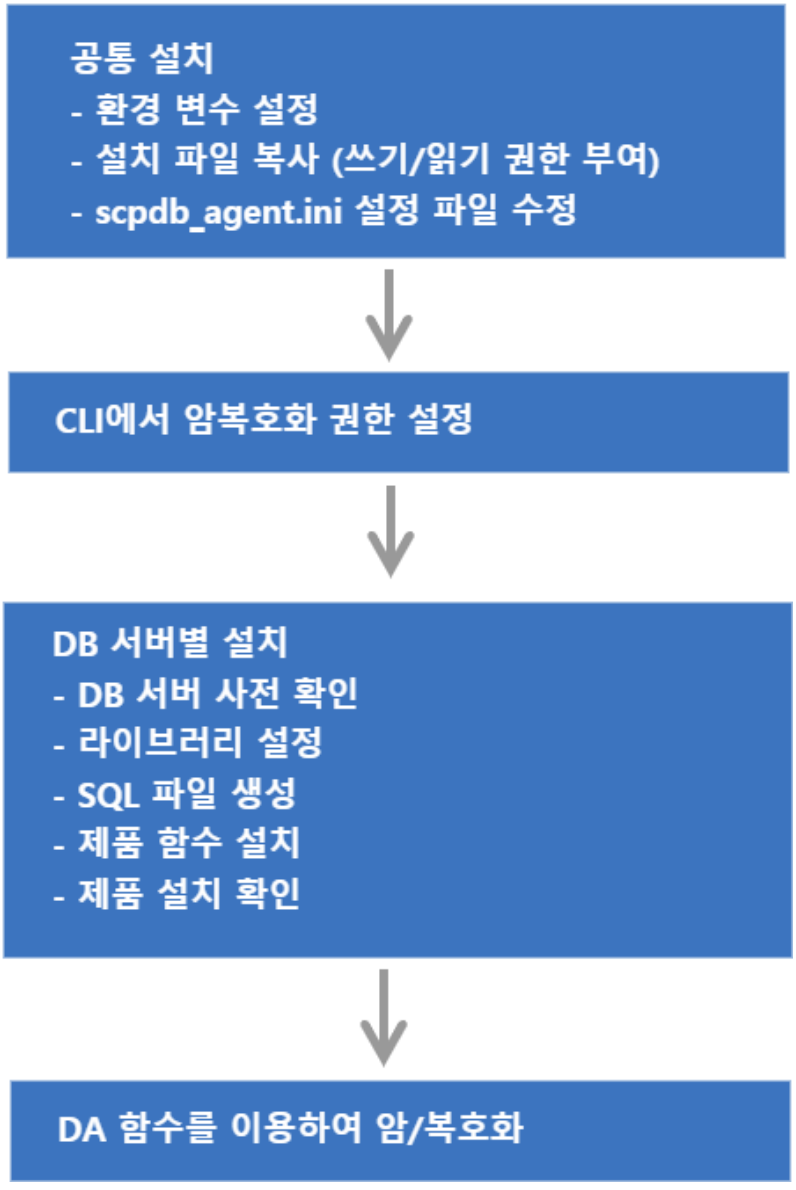
```
mysql>SELECT CONFIG_REINIT();
```

3.

TIBERO

DA는 DB 서버의 종류에 따라서 설치 및 운용 방법이 각각 다르다.

그림 3-1 설치 개념도



3.1 지원 운영체제 및 DB서버

DA에서 지원하는 운영체제 및 DB서버는 아래와 같다. 단, 특정 환경은 지원되지 않을 수도 있으므로, 제품 설치 전에 상세한 지원 가능 여부는 펜타시큐리티시스템으로 문의한다.

표 3-1 제품이 지원하는 운영체제 및 DB 서버 정보

구분	설명
운영체제	Windows, AIX, HP IA, HP PA-RISC, Linux, SUN, TRU64
DB 서버	ORACLE 8i ~, SQL Server 2012 ~, DB2 9.x ~, Tibero 4 SP 1 ~, MySQL 5.x ~, MariaDB 5.5, 10.0, 10.1, Cache DB 2009.1 ~, Informix IDS 9.x ~ Sybase ASE 15.7 SP61 ~, Sybase IQ 15.4 ~, CUBRID 2008 R1.3 ~ , PostgreSQL 9.4 ~

3.2 설치 파일의 구성 확인

DA의 설치 파일은 아래와 같은 명칭으로 압축 파일(zip) 형태로 제공된다.

- DA 설치파일: Install_DAmo_DA_v{버전}.zip
  - 압축을 해제 한 뒤, 설치 대상 DB 서버, OS 및 bit에 맞는 설치 파일을 준비한다.

설치 파일에 제품을 사용할 수 있는 '라이선스'는 포함되어 있지 않다.
펜타시큐리티시스템에 문의하여 '라이선스' 파일은 별도로 준비한다.

표 3-2 설치 파일(Install_DAmo_DA_v{버전}.zip)을 압축 해제 시, 디렉터리의 구성

구성	설명
_SampleScpsFiles	SG-KMS 연동 없이 암복호화를 테스트할 수 있는 테스트 키 파일
_TestAgentKeyPair	CLI에서 사용할 수 있는 테스트 키 쌍
Altibase	DA-ALT 제품의 설치 바이너리 폴더
Cache	DA-CDB 제품의 설치 바이너리 폴더
Cubrid	DA-CUB 제품의 설치 바이너리 폴더
DB2	DA-DB2 제품의 설치 바이너리 폴더
Informix	DA-IFX 제품의 설치 바이너리 폴더
MySQL	DA-MYQ 제품의 설치 바이너리 폴더
Oracle	DA-ORA 제품의 설치 바이너리 폴더
Postgres	DA-PGS 제품의 설치 바이너리 폴더
SQL Server	DA-MSQ 제품의 설치 바이너리 폴더

구성	설명
SybaseASE	DA-SYB 제품의 설치 바이너리 폴더
SybaseIQ	DA-SIQ 제품의 설치 바이너리 폴더
Tibero	DA-TIB 제품의 설치 바이너리 폴더

3.2.1 DB 서버 및 운영체제 별 SQL파일 구성

각 DB 서버 및 운영체제 별 SQL파일 구성은 다음과 같다. 다음 장에서 각 DB별로 SQL파일 설치 방법을 설명한다.

표 3-3 DB 서버 및 운영체제 별 SQL파일

DB 서버 종류	Linux 일 경우	Windows 일 경우
Oracle	000.da_user.pkg 000.da_user.sql 001.inner_function.ora(c 버전 설치 시) 002.user_interface.ora(JAVA 버전 설치 시) 002.user_interface_java.ora 003.grant_execute_functions.sql(JAVA 버전 설치 시) 005.securej_privilege.sql install_make.sh	000.da_user.pkg 000.da_user.sql 001.inner_function.ora(c 버전 설치 시) 002.user_interface.ora(JAVA 버전 설치 시) 002.user_interface_java.ora 003.grant_execute_functions.sql(JAVA 버전 설치 시) 005.securej_privilege.sql 009.da_test.sql
MYSQL	001.inner_function.mys 002.user_interface. mys 003.grant_execute_functions.sql 009.da_test.sql install_make.sh	001.inner_function.mys 002.user_interface.mys 003.grant_execute_functions.sql
TIBERO	001.inner_function.tbs(c 버전 설치 시) 001.inner_function_java.tbs(JAVA 버전 설치 시) 002.user_interface.tbs(c 버전 설치 시) 002.user_interface_java.tbs(JAVA 버전 설치 시) 003.grant_execute_functions.sql install_make.sh	001.inner_function_java.tbs 002.user_interface_java.tbs 003.grant_execute_functions.sql
INFORMIX	001.inner_function.ifx 002.user_interface. ifx 003.grant_execute_functions.sql 009.da_test.sql install_make.sh	해당 없음
POSTGRESQL	001.inner_function.post 002.user_interface.post 003.grant_execute_functions.post	해당 없음

DB 서버 종류	Linux 일 경우	Windows 일 경우
DB2	001.inner_function.db2 002.user_interface.db2 003.grant_execute_functions.sql install_make.sh	001.inner_function.db2 002.user_interface.db2 003.grant_execute_functions.sql
CUBRID	001.inner_function.sql 002.user_interface.sql	001.inner_function.sql 002.user_interface.sql
SYBASE	001.inner_function.sybase 002.user_interface.sybase 003.grant_execute_functions.sql install_make.sh	해당 없음
SYBASE IQ	001.inner_function.sybiq 002.user_interface. sybiq 003.grant_execute_functions.sql install_make.sh	해당 없음
SQL Server	해당 없음	001.inner_function.sql 002.user_interface. sql 003.grant_execute_functions.sql install_make.bat
Cache DB	SCP.xml install_make.sh	SCP.xml
Altibase	해당 없음	000.da_user.pkg 000.da_user.sql 001.inner_function.sql 002.user_interface. sql install_make.bat

3.2.2 공통 설치 파일 구성

환경에 관계 없이 공통적으로 사용하는 설치 파일은 아래와 같다.

표 3-4 공통 설치 파일 (sql파일 제외)

파일 분류	파일 명	파일 용도
Library 파일	libdamoscpdb.{so|a|sl|dll}	DA 메인 라이브러리. 주로 DBMS External Interface를 담당
	libdamocm-4.0.{so|a|sl|dll}	공통 모듈 라이브러리
	liblogw-0.2.{so|a|sl|dll}	로그를 기록하는 라이브러리
	libcis_cc-3.3.{so|a|sl|dll}	암호화, 복호화 기능을 제공하는 라이브러리
	libcis_ce-3.3.{so|a|sl|dll}	암호화, 복호화를 제외한 부가적인 기능을 제공하는 라이브러리(예: Base64, 인증서 관리, 특성 유지 암호화 등)

파일 분류	파일 명	파일 용도
설정 파일	scpdb_agent.ini	DA 구동시 실행에 필요한 설정정보를 참조
License 파일	damo_lic.cer	DA 구동 시 제품의 유효성을 검증하는데 사용
Agent key 파일	damo_agt_site.cer	SG-KMS 연동, CLI 프로그램에서 사용하는 인증서 쌍
	damo_agt.cer	
	damo_agt.key	
접근제어 파일	acl_cli 파일	DB 의 USER 별로 암·복호 권한을 설정하는데에 사용
	privilege.damo	권한 파일
JAVA class 파일 (Oracle, Tibero, Cubrid 설치 가능)	ScpAgentException.class	예외 처리 Class
	ScpCryptData.class	암호화 복호화 Class
SQL 파일	아래 새로운 표에 DB별로 표기함	

Agent Key 파일은 SG-KMS 연동에 필요한 키 발급를 참고하여 발급 받는다.

DA-PGS(PostgreSQL)의 경우, DB 서버 버전에 따라 libdamoscpdb.so 라이브러리 선택

- libdamoscpdb94.so (Postgres 9.4)
- libdamoscpdb95.so (Postgres 9.5)
- libdamoscpdb95AS.so (EDB Postgres 9.5)
- libdamoscpdb96AS.so (EDB Postgres 9.6)
- libdamoscpdb10.so (Postgres 10)

3.3 환경변수 설정

DA를 설치할 운영체제에 환경변수 DA_INST_HOME를 설정한다. 이 매뉴얼에서는 제품 설치 경로를 아래와 같이 가정하여 설명한다.

- Linux 환경일 경우: /home/dbms_api
- Windows 환경일 경우: E:₩dbms_api

주의) DA_INST_HOME 설정 시, 주의 사항

- 리눅스의 경우 "/root" 디렉토리로 설정을 권장하지 않는다.
- 윈도우의 경우 "바탕화면"으로 설정을 권장하지 않는다.

위의 경로로 설정할 경우 접근 권한 등의 이유로 문제가 발생할 가능성이 존재한다.

3.3.1 환경변수 설정 - Linux 환경일 경우

DA_INST_HOME 환경변수에 DA의 설치 디렉터리를 설정한다.

```
.profile을 사용하는 경우
export DA_INST_HOME=/home/dbms_api


.cshrc를 사용하는 경우
setenv DA_INST_HOME=/home/dbms_api

export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$DA_INST_HOME
```

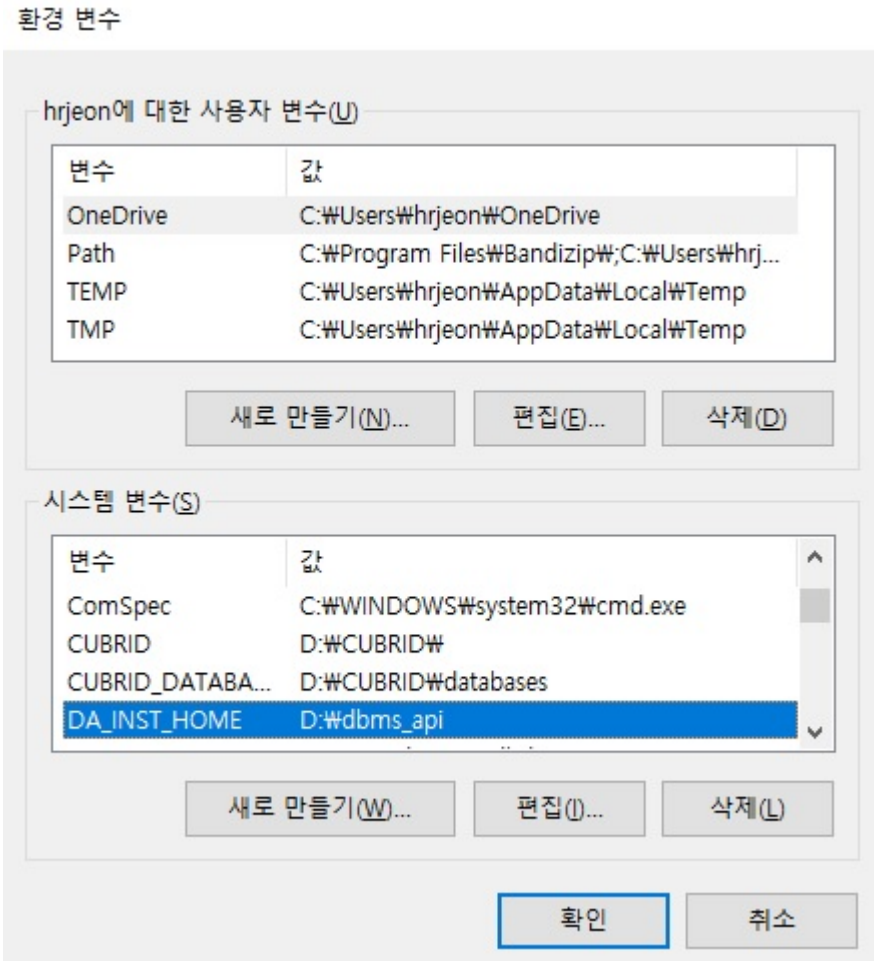
표 3-5 운영 체제별 라이브러리 PATH 명칭

운영 체제	라이브러리 PATH 명칭
HP_UX	SHLIB_PATH
AIX	LIBPATH
LINUX, SUN	LD_LIBRARY_PATH

3.3.2 환경변수 설정 - Windows 환경일 경우

[탐색기->내컴퓨터->등록정보->고급 탭->환경변수] 를 선택하여 나타나는 환경변수 설정 다이얼로그에서 시스템변수 DA_INST_HOME 변수와 값을 추가한다.

그림 3-2 DA_INST_HOME



3.4 DA의 설치 파일 복사

$DA_INST_HOME 디렉터리에 위 [설치 파일의 구성 확인]에서 나열된 파일들을 복사한다.

3.5 라이선스 파일 설정

$DA_INST_HOME 디렉토리에 라이선스 파일(damo_lic.cer)을 복사한다.

라이선스 파일명이 damo_lic.cer가 아닌 다른 이름으로 저장되어 있다면 변경해야 한다.

3.6 설치 파일의 접근 권한 부여 (Linux 설치 시)

Linux 사용자 계정에 DA 설치 파일(라이브러리, 실행 파일, 디렉토리)의 접근 권한을 부여한다. Windows에서 제품을 설치 할 경우, 이 과정은 생략한다.

```
$> cd $DA_INST_HOME
$> chmod 755 lib* acl_cli sql/install_make.sh
```

3.7 (SG-KMS 연동 시) SG-KMS에서 DA 정보 등록 및 연동에 필요한 키 내보내기

DA는 데이터를 암·복호화하기 위해 '암호화 키'가 필요하다. '암호화 키'를 얻기 위해서는 SG-KMS를 연동하거나 SCPS라는 암호화 키 파일을 이용할 수 있는데,
다음 설명은 SG-KMS 연동을 위해 SG-KMS에서 DA 정보를 등록하고 연동에 필요한 키를 내보내는 방법에 대해서 설명한다.

3.7.1 사전 준비

3.7.1.1 지원 SG-KMS 버전

DA와 연동 가능한 SG-KMS 버전은 다음과 같다.

표 3-6 연동 가능한 SG-KMS 버전

SG-KMS Major version	SG-KMS Minor version
SG-KMS v3.0	v3.0.9.0 이상 연동 가능
SG-KMS v4.0	v4.0.104.5 이상 연동 가능

SG-KMS v2.3 연동은 미지원한다.

3.7.1.2 연동 전 점검 항목

SG-KMS 연동을 위해서는 다음과 같이 사전 준비가 필요하다.

- SG-KMS 관리도구
- SG-KMS 관리도구에 접속할 수 있는 ID와 비밀번호
- SG-KMS 매뉴얼

이 매뉴얼에서는 SG-KMS의 키 종류 및 SG-KMS 관리도구 사용 방법에 대한 내용은 다루지 않으므로 **[SG-KMS 매뉴얼]**을 참고한다.

3.7.1.3 SG-KMS 연동에 필요한 키 발급

DA와 SG-KMS의 연동을 위해서는 먼저 SG-KMS에 DA를 Agent로 등록해야한다. 등록하는 절차는 SG-KMS 사용자설명서의 '대칭키를 사용하는 D'Amo Agent 등록 안내서'를 참고한다.

[D'Amo Agent 키] 파일, [Agent ID(CN)]와 [서비스 ID]정보는 DA와 SG-KMS 연동을 위한 설정 과정에서 필요하다. SG-KMS 관리자는 DA 설치 엔지니어에게 안전하게 전달한다.

SG-KMS 관리자에게 받은 D'Amo Agent의 인증서 및 키 파일 명을 다음과 같이 변경후 $DA_INST_HOME/key 디렉터리에 복사한다.

표 3-7 SG-KMS 연동에 필요한 키 목록

키 구분	생성된 키 파일명	변경할 키 파일명
사이트 키	damo-site_{발행기관/부서명}-SITE_V3.cer	damo_agt_site.cer
Agent 키	damo-scp_SITE_V3-{Agent 이름}.cer	damo_agt.cer
	damo-scp_SITE_V3-{Agent 이름}.key	damo_agt.key
	damo-scp_SITE_V3-{Agent 이름}.spin	damo_agt.spin

Agent 키 경로 지정, 키 이름 변경은 반드시 필요한 작업은 아니지만 관리를 위해 변경하는 것을 권장한다.

3.8 설정 파일(scpdb_agent.ini) 수정

DA의 운용을 위해 사용하는 scpdb_agent.ini 설정 파일 수정 방법에 대해 설명한다. DA는 SG-KMS를 이용하거나 SCPS 파일을 이용하여 암호화 키를 얻기 때문에 고객의 환경에 맞게 설정해야 한다.

$DA_INST_HOME 디렉터리에 있는 scpdb_agent.ini 설정 파일에서 아래 항목의 값을 수정한다.

1. 설정 파일의 [KEYINFO] 항목 - [KEY1]에 암호화 키 정보를 입력한다.

```
[KEYINFO]
KEY1=암복호화 하려는 암호화 키 정보를 입력한다.
//키 정보는 다음과 같은 값을 입력할 수 있다.
///ServiceID: SG-KMS에 생성한 서비스 ID를 입력한다.
///SCP_FilePath: SG-KMS에서 서비스 내보내기를 통해 발급한 SCPS 파일의 절대 경로 및 파일명, 확장자를 입력한다.

//*예제 - Windows 경우
KEY1=DA_AES256
KEY2=C:\DA\Policy\S_AES128.SCPS\DA_AES256.scps
KEY3=DA_AES256,C:\DA\Policy\S_AES128.SCPS\DA_AES256.scps

//*예제 - Linux 또는 Unix 경우
KEY1=DA_AES256
KEY2=/home/dbms_api/key/DA_AES256.scps
KEY3=DA_AES256,/home/dbms_api/key/DA_AES256.scps
```

ServiceID를 입력 할 경우 SG-KMS와 통신을 통해 암호화 정책을 가져와 데이터 암복호화를 수행한다.

SCP_FilePath를 입력 할 경우 KMS와 통신하지 않고 서버의 SCP 파일을 통해 암호화 정책을 가져와 데이터 암복호화를 수행한다.

ServiceID와 SCPS 파일명을 동시에 입력 시, SG-KMS와 네트워크 연결 실패 하면 SCPS 파일을 이용하여 암호화 한다.
ServiceID와 SCPS 파일명 사이에 공백이 있으면 SCPS 파일을 읽을 수 없다. 따라서 예제와 같이 띄어쓰기를 하지 않고 ServiceID와 SCPS 파일 경로를 붙여서 입력한다.

2. 설정 파일의 [Server], [Server2] 항목을 수정한다.

```
[Server]
ServerIP: SG-KMS의 IP를 입력한다.
ServerPort: SG-KMS의 포트번호를 입력한다. 기본값은 2525이다.

//*예제 - 모든 OS 공통
```

```
6  ServerIP=192.168.22.25
7  ServerPort=2525
```

```
1  [Server2]
2  ServerIP: 이중화를 위한 2번 SG-KMS의 IP를 입력한다.
3  ServerPort: 이중화를 위한 2번 SG-KMS의 포트번호를 입력한다. 기본값은 2525이다.
4
5  //*예제 - 모든 OS 공통
6  ServerIP=192.168.22.26
7  ServerPort=2525
```

ServerIP는 최소 1개 ~ 최대 10개까지 등록이 가능하다.

3. 설정 파일의 [AGENT] 항목을 설정한다.

```
1  [AGENT]
2  AgentID=SG-KMS 관리도구에서 설정한 D'Amo Agent의 Agent ID
3  LogDir=로그가 저장될 디렉터리 위치
4  LogLevel=로그가 남는 수준
5  SiteCertFilePath=SG-KMS 장비에서 설정한 해당 장비의 사이트 공개키(.cer)
6  CertFilePath=SG-KMS 관리도구에서 설정한 D'Amo Agent의 공개키(.cer)
7  KeyFilePath=SG-KMS 관리도구에서 설정한 D'Amo Agent의 비공개키(.key)
8  SPIN=SG-KMS 관리도구에서 설정한 D'Amo Agent의 SPIN으로, damo-scp_SITE_V3-[Agent이름].spin
파일의 값
9
10 //[예제]
11 AgentID=DA
12 LogDir=/home/dbms_api/log
13 LogLevel=4
14 SiteCertFilePath=/home/dbms_api/key/damo_agt_site.cer
15 CertFilePath=/home/dbms_api/key/damo_agt.cer
16 KeyFilePath=/home/dbms_api/key/damo_agt.key
17 SPIN=XaMh1ylXUh123XUh
```

로그가 남는 수준(LogLevel)에는 아래 5가지 숫자 입력이 가능하며, 각 값의 설정은 다음과 같다.
- 0: 아무 로그도 남기지 않을 경우
- 2: 경고 로그를 파일에 기록
- 4: 에러 로그와 경고 로그를 파일에 기록

- 6: 정보 로그, 에러 로그, 경고 로그를 파일에 기록
- 8: 디버그, 정보 로그, 에러 로그, 경고 로그를 파일에 기록

제품 운영 중 scpdb_agent.ini 파일을 수정하면 CONFIG_REINIT() 함수를 호출해야 변경된 내용이 적용된다.

3.9 CLI에서 권한 설정

$DA_INST_HOME 디렉터리에서 acl_cli 파일을 실행하여 USER 단위로 암/복호화 권한을 설정한다. USER 는 DB의 소유자명이고, KEY는 설정 파일(scpdb_agent.ini)의 [KEYINFO] 중 ALIAS 값(예: KEY1)이다.

CLI 명령어를 자세히 보려면 help 명령어를 실행한다.

```
CLI 실행 방법
$> cd $DA_INST_HOME
$> ./acl_cli  - start
Enter the PIN of CLI-key. : damo_agt.key 의 비밀번호
```

```
권한 추가할 경우
D'Amo > SET PRIV ENC [USER]"[KEY]"1"1
D'Amo > SAVE ALL
D'Amo > SHOW ALL

예제)
D'Amo > SET PRIV ENC SCOTT"KEY1"1"1
D'Amo > SET PRIV ENC SCOTT"KEY2"1"1
```

scpdb_agent.ini 설정 파일이 아래 예제와 같을 경우, CLI에서 권한 설정할 때 입력해야 하는 2번째 [KEY] 인자 값에는 KEY1을 입력해야 한다. (※ARIA256을 입력하는 것이 아님)

#scpdb_agent.ini 설정 파일 예제
[KEYINFO]
KEY1=ARIA256

```
권한 삭제할 경우
D'Amo > DEL PRIV ENC [USER]"[KEY]
D'Amo > SAVE ALL
D'Amo > SHOW ALL
```

CLI 에서 권한을 추가하거나 삭제 한 경우 반드시 SAVE ALL 명령어를 실행하며 SHOW ALL 명령어를 이용하여 적용 여부를 확인 한다.

3.10 DB 서버 사전 확인

DA를 TIBERO 환경에서 설치 하기 전에 다음과 같은 사항들을 확인한다.

- DB 엔진 설치 계정에 폴더 생성
- DB 사용자로 SCP 계정 생성 (권고사항)
- DB 서버의 DBA 권한 계정
- DB 엔진 설치 계정의 . profile 파일 수정 (Linux 설치 시)

3.11 JEPA 실행 (JAVA버전 설치 시)

DA-TIB에서는 C버전과 JAVA버전의 함수를 제공하며, 설치과정이 달라진다. JAVA 버전을 설치할 때에는 JEPA라는 프로세스가 구동된 상태여야 한다. JEPA를 실행시키기 위해서는 별도의 환경설정이 필요하다. 이 매뉴얼에서는 Tibero5 버전을 기준으로 환경설정 방법을 설명한다.

3.11.1 환경변수 값 확인

- 다음 명령어를 통해 $TB_HOME의 경로를 확인한다. (기본 설치 시, 보통 /home/tibero4, /home/tibero5)

```
[Linux의 경우]
echo $TB_HOME

[Windows의 경우]
```

```
set TB_HOME
```

- 다음 명령어를 통해 $TB_SID의 경로를 확인한다. (기본 설치 시, 보통 tibero)

```
[Linux의 경우]
echo $TB_SID

[Windows의 경우]
set TB_SID
```

3.11.2 초기화 파라미터 설정

JEPA를 실행시키기 위해서는 $TB_HOME/config 디렉터리의 $TB_SID.tip 파일의 _PSM_BOOT_JEPA 값을 'Y'로 변경해야한다. 또한 JAVA 객체 생성시 .class 파일이 생성될 위치를 지정해줘야 하는데 그 경로는 JAVA_CLASS_PATH에서 지정이 가능하다.

```
_PSM_BOOT_JEPA=Y
[Linux의 경우]
JAVA_CLASS_PATH=/home/tibero5/instance/tibero/java

[Windows의 경우]
JAVA_CLASS_PATH=C:\TmaxData\tibero6\database\tibero\java
```

3.11.3 JEPA 정보 설정

$TB_HOME/client/config/tbdsn.tbr 파일에 JEPA 프로세서의 접속 정보를 설정한다. 다음은 tbdsn.tbr 파일에 설정된 접속 정보의 예이다.

```
epa=(
 (EXTPROC=(LANG=JAVA)
 (LISTENER=(HOST=localhost)
 (PORT=9390)
 )
 )
 )
```

3.11.4 JEPA 환경 설정

$TB_HOME/client/epa/java/config 디렉터리에 위치한 epa.cfg 파일을 사용자의 시스템 환경에 맞게 수정한다. ENCODING 속성은 데이터베이스 서버와 동기화를 위해 반드시 서로 같은 인코딩을 사용해야 한다. Tibero 데이터베이스는 기본적으로 MSWIN949 인코딩으로 설정이 되어있다.

```
# listener port
LISTENER_PORT=9390

# initial thread pool size
INIT_POOL_SIZE=10

# max thread pool size
MAX_POOL_SIZE=1000

# tbjavaepa encoding "ASCII", "EUC-KR", "MSWIN949", "UTF-8", "UTF-16", "SHIFT-JIS"
ENCODING=MSWIN949

#STATIC_LOADING_CLASSES=ex.StaticClass1, ex.st.Class2

STATIC_LOADING_CLASSES=ScpCryptData, ScpAgentException
```

tbdsn.tbr 파일과 epa.cfg 파일에 있는 포트 번호는 반드시 일치해야 한다.

ENCODING 타입은 데이터베이스(DATABASE) ENCODING 타입과 동일하게 설정해야 한다.

3.12 라이브러리 설정

3.12.1 라이브러리 링크 설정 (Linux에 설치 시)

$TB_HOME/lib 디렉터리에 library 파일의 symbolic link 파일을 생성한다.

```
$> cd $TB_HOME/lib
$> ln -s $DA_INST_HOME/libcis_cc-3.3.{so|a|sl} libcis_cc-3.3.{so|a|sl}
$> ln -s $DA_INST_HOME/libcis_ce-3.3.{so|a|sl} libcis_ce-3.3.{so|a|sl}
$> ln -s $DA_INST_HOME/liblogw-0.2.{so|a|sl} liblogw-0.2.{so|a|sl}
$> ln -s $DA_INST_HOME/libdamocm-4.0.{so|a|sl} libdamocm-4.0.{so|a|sl}
$> ln -s $DA_INST_HOME/libdamoscpdb.{so|a|sl} libdamoscpdb.{so|a|sl}
```

3.12.2 라이브러리 복사 및 설정 (Windows에 설치 시)

1. 아래 경로에 제품 라이브러리를 복사하여 저장한다.

표 3-8 복사할 파일명 (TIBERO - Windows 환경 설치 시)

라이브러리가 저장되어야 하는 경로	복사할 라이브러리명
$TB_HOME₩bin	cis_cc-3.2.dll cis_ce-3.2.dll damoscpdb.dll damocm-4.0.dll, logw-0.2.dll
$TB_HOME/client/epa/java/lib	ScpCryptData.jar

2. $TB_HOME/client/bin/tbjavaepa.bat 파일의 내용을 수정한다.

- set config 아래 줄에 다음과 같이 set scpdbms 경로를 추가한다.

```
set config=%javaepahome%₩config
set scpdbms=%javaepahome%₩lib₩ScpCryptData.jar
```

- java -Xms128m -Xmx512m ~~ %tbepa%;%config%;%scpdbms% %mainclass% CONFIG=%configfile% 행의 뒤쪽에 ;%scpdbms%를 아래처럼 수정한다.

```
[원본] java -Xms128m -Xmx512m ~~ %tbepa%;%config% %mainclass% CONFIG=%configfile%
```

[수정] java -Xms128m -Xmx512m ~~ %tbepa%;%config%;%scpdbms% %mainclass% CONFIG=%configfile%

3. %TB_HOME%₩bin 디렉터리에 있는 psmjavac.bat를 수정한다. JAVA의 CLASSPATH에 다음과 같이 ScpCryptData.jar 파일이 있는 경로를 추가하면 된다.

if not exist "%1" (echo ERROR: Cannot open file '%1') else (javac -classpath %2;%3;%4;%5;%TB_HOME%₩client₩epa₩java₩lib₩ScpCryptData.jar %1)

3.13 sql 파일 생성

1. $DA_INST_HOME/sql 디렉터리에서 install_make.{sh|bat}을 이용하여 설치할 sql 파일을 생성한다. D_INI 는 설정 파일(scpdb_agent.ini)의 경로다.

```
$> cd $DA_INST_HOME/sql

[Linux의 경우]
$> ./install_make.sh D_INI

[Windows의 경우]
$> install_make.bat D_INI

[예제]
$> ./install_make.sh /home/dbms_api
D_INI_PATH is replaced by /home/dbms_api
```

2. 아래 4개의 파일이 생성되었는지 확인한다.

- 001.inner_function.tbs.sql
- 001.inner_function_java.tbs.sql
- 002.user_interface.tbs.sql
- 002.user_interface_java.tbs.sql

3.14 DB에서 DA를 사용할 계정 생성 (권장사항)

DB에서 DA를 사용할 'SCP' 계정를 생성하는 것을 권고한다. 'SCP' 계정 생성 시, CONNECT, RESOUCE, CREATE LIBRARY 권한을 부여한다.

```
SQL> CREATE USER SCP IDENTIFIED BY [password];
User created.
SQL> GRANT CONNECT, RESOURCE, CREATE LIBRARY TO SCP;
Grant succeeded.
```

3.15 제품 함수 설치

3.15.1 C 버전 설치 방법

1. $DA_INST_HOME/sql 위치에서 DB 접속 후 설치를 원하는 DB계정에 LIBRARY 와 함수를 설치한다. 예를 들어, SCP 계정에 설치를 하고자 원한다면 SCP 계정에 접속을 하여 설치를 진행한다.
2. 'SECURE_SCP_LIB'이라는 이름의 라이브러리를 설치하고자 하는 계정에 생성한다. SECURE_SCP_LIB 라이브러리의 위치는 $DA_INST_HOME 이다.

```
SQL> CREATE LIBRARY SECURE_SCP_LIB AS '/[fullpath_of_DA_INST_HOME]/libdamoscpdb.{so¦a¦sl}';
/
```

경로내에 공백이나 기타 쓰레기 값이 들어가면 라이브러리를 인식하지 못하므로 주의한다.

3. 001.inner_function.tbs.sql 파일을 실행한다.
4. 002.user_interface.tbs.sql 파일을 실행한다.

```
SQL> START 001.inner_function.tbs.sql
Function created.

SQL> START 002.user_interface.tbs.sql
Function created.
```

5. 특정 DB 사용자에게 함수 실행 권한을 부여한다. 모든 사용자에게 함수 실행 권한을 부여할 때는 003.grant_execute_functions.sql 파일을 실행한다.

```
SQL> START 003.grant_execute_functions.sql
Granted.
```

C버전 tbepa는 Linux, AIX 64bits, HP_IA 64bits, SOLALIS 5.9 64bits, SOLALIS 5.10 64bits OS만 지원한다.

3.15.2 JAVA 버전 설치 방법

다음은 JAVA 버전의 설치 방법이다. JAVA버전으로 설치할 경우, JEPA가 기동이 된 상태여야한다. JEPA를 실행시키기 위해서는 [02.IV.1.102Tibero DB 설정] 부분을 참고한다.

1. ScpCryptData.jar 및 라이브러리를 Tibero epa 경로로 복사한다.

```
cp $DA_INST_HOME/ScpCryptData.jar $TB_HOME/client/epa/java/lib
```

2. $TB_HOME/bin 디렉터리에 있는 psmjavac 스크립트를 수정한다. JAVA의 CLASSPATH에 다음과 같이 ScpCryptData.jar 파일이 있는 경로를 추가하면 된다.

```
[Linux의 경우]
javac -classpath ${TB_HOME}/client/epa/java/lib/ScpCryptData.jar:${classpath} ${src}
```

```
[Windows의 경우]
if not exist "%1" (echo ERROR: Cannot open file '%1') else (javac -classpath
%2;%3;%4;%5;%TB_HOME%\client\epa\java\lib\ScpCryptData.jar %1)
```

3. $TB_HOME/client/bin 디렉터리에 있는 tbjavaepa 스크립트를 수정한다. CLASSPATH 변수에 추가하려는 라이브러리 경로를 설정한 다음 exec java 명령 문장의 CLASSPATH 옵션에 해당 변수를 추가한다.

```
[Linux의 경우]
scpdbms=${TB_HOME}/client/epa/java/lib/ScpCryptData.jar
exec java -verbose:gc -Xms128m -Xmx512m -Djepa.home=$javaepahome
-Dlog4j.configuration=$log4jfile -Dlog4j.configuration.fullname=$log4jfile_fullname
-Dlog4j.refresh.time=$log4j_refresh_time -classpath
$pool:$collections:$activation:$mail:$logger:$log4j:$jlexer:$jdbc:$epa:$config:$scpdbms
$mainclass CONFIG=$configfile >> ${epa_console_log} 2>&1
```

```
[Windows의 경우]
scpdbms=%TB_HOME%\client\epa\java\lib\ScpCryptData.jar
java -Xms128m -Xmx512m -Djepa.home=%javaepahome% -Dlog4j.configuration=%log4jfile%
```

```
-Dlog4j.configuration.fullname=%log4jfile_fullname% -Dlog4j.refresh.time=%log4j_refresh_time%
-classpath %pool%;%collections%;%activation%;%mail%;%logger%;%log4j%;%jlexer%;%jdbc%;%epa%;%co
nfig%;%scpdbms% %mainclass% CONFIG=%configfile% >> %epa_console_log% 2>&1
```

4. $TB_HOME/client/bin 디렉터리에 있는 tbjavaepa 스크립트를 수정한다. CLASSPATH 변수에 추가하려는 라이브러리 경로를 설정한 다음 exec java 명령 문장의 CLASSPATH 옵션에 해당 변수를 추가한다.

epa 폴더 경로는 tibero 버전마다 다를 수 있다.
위의 변경사항을 적용하기 위해서 JEPA 프로세스를 재기동시킨다.

5. 001.inner_function_java.tbs.sql 파일을 실행한다.
6. 002.user_interface_java.tbs.sql 파일을 실행한다.

```
SQL> START 001.inner_function_java.tbs.sql
Function created.

Function created.

SQL> START 002.user_interface_java.tbs.sql
Function created.
```

7. 특정 DB 사용자에게 함수 실행 권한을 부여한다. 모든 사용자에게 함수 실행 권한을 부여할 때는 003.grant_execute_functions.sql 파일을 실행한다.

```
SQL> START 003.grant_execute_functions.sql
Granted.
```

3.16 제품 설치 확인

DB 서버에서 암/복호화 함수를 호출하여 설치를 성공했는지 확인한다.

```
SQL> SELECT ENC_STR('KEY1', 'abc') FROM DUAL;
ENC_STR('KEY1','abc')
--------------------------------------------------
5E41ACD673653158D7AE8C30CDA9627D3556E173

SQL> SELECT DEC_STR('KEY1', ENC_STR( 'KEY1', 'abc')) FROM DUAL;
```

```
DEC_STR('KEY1', ENC_STR('KEY1', 'abc'))
--------------------------------------------------
abc
```

만약 함수 실행시 '잘못된 ELF 클래스'라는 에러가 발생하면 JEPA가 설치하려는 OS와 맞지 않는 BIT로 실행된 것이다. 만약 OS가 64비트인데 다음 에러가 난다면 $TB_HOME/client/bin 디렉터리에 있는 tbjavaepa를 다음과 같이 수정하여서 JEPA를 64비트로 실행시킨다.

```
exec java -d64 -verbose:gc -Xms128m -Xmx512m (...) -classpath (...):$scpdbms:(...) $mainclass
CONFIG=$configfile >> ${epa_console_log} 2>&1
```

3.17 제품 운용

3.17.1 함수 설명

DA에서 제공되는 함수와 사용하는 방법에 대해서 설명한다.

3.17.1.1 파라미터 설명

3.17.1.1.1 I_KEY

I_KEY : 암호화/복호화 때 사용하는 암호화 키
- 설정 파일(scpdb_agent.ini)의 [KEYINFO] 중 ALIAS 값
- 아래 [설정 파일 예제]의 경우 ALIAS는 'KEY1'과 'KEY2'이다.
- ALIAS는 SCPS파일로 암호화 할 것인지, SG-KMS의 서비스 ID로 암호화 할 것인지 설정 가능하다.

[설정 파일 예제]
- 설정 파일(scpdb_agent.ini)의 [KEYINFO] 중 ALIAS 값
KEY1=AES256.SCPS
KEY2=ARIA256

[함수 사용 예제]
SELECT ENC_B64('KEY1', 'abc') FROM DUAL;

3.17.1.1.2 I_DATA

I_DATA : 평문(암호화 함수일 경우), 암호문(복호화 함수일 경우)

DA에서 제공하는 암호화 함수에서 성능 향상을 위해 라이브러리에서 정책 이름(KEYINFO ALIAS)을 바탕으로 Cash하여 동작한다. 정책 이름은 같은데 실제 키(대칭 키)가 다른 정책을 사용하면 Cash에서 이전 키로 복호화를 시도하여 오류가 발생한다. 이 상황을 해결하기 위해서는 데이터베이스 서버 재시작이 필요하다.

표 3-9 DA 함수 (TIBERO)

함수 명	입력		출력	비고
ENC_STR	I_KEY	IN 문자열,	Hex String 암호문	
	I_DATA	IN 문자열 (평문)		
ENC_B64	I_KEY	IN 문자열,	Base64 Encording 암호문	
	I_DATA	IN 문자열 (평문)		
DEC_STR	I_KEY	IN 문자열,	평문	
	I_DATA	IN 문자열 (Hex String 암호문)		
DEC_B64	I_KEY	IN 문자열,	평문	
	I_DATA	IN 문자열 (base64 String 암호문)		
INDEX_STR	I_KEY	IN 문자열,	Hex String 암호문	
	I_DATA	IN 문자열 (평문),		
	I_TYPE	IN 문자열 '' or 'IX '(Plug-IN 연동 시 사용)		
DEC_INDEX_STR	I_KEY	IN 문자열,	Hex String 암호문 입력받아 OPE 데이터	
	I_DATA	IN 문자열 (암호문),		
	I_TYPE	IN 문자열 '' or 'IX '(Plug-IN 연동 시 사용)		
DEC_INDEX_B64	I_KEY	IN 문자열,	Base64 Encording 암호문 입력받아 OPE 데이터	
	I_DATA	IN 문자열 (암호문),		
	I_TYPE	IN 문자열		

		'' or 'IX '(Plug-IN 연동 시 사용)			
HASH_STR	I_ALOG	IN 숫자,	SHA1 =70	Hex String 해쉬 암호문	
			SHA256 =71		
			SHA384 =72		
			SHA512 =73		
			HAS160 =74		
	I_DATA	IN 문자열			
HASH_B64	I_ALOG	IN 숫자,	SHA1 =70	Base64 String 해쉬 암호문	
			SHA256 =71		
			SHA384 =72		
			SHA512 =73		
			HAS160 =74		
	I_DATA	IN 문자열			
HEXTOB64	I_DATA	IN 문자열 (Hex String 암호문)		base64 Encording 암호문	
B64TOHEX	I_DATA	IN 문자열 (base64 Encording 암호문)		Hex String 암호문	
CONFIG_REINIT				성공시 'SUCCESS', 그 외 에러	
ENC_BLOB	I_KEY	IN 문자열,		BLOB 타입	C버전만 제공
	I_DATA	IN 문자열 (평문)			
DEC_BLOB	I_KEY	IN 문자열,		BLOB 타입	C버전만 제공
	I_DATA	IN 문자열 (암호문)			

3.17.2 함수 호출 예제

1. ENC_STR

```
SQL> SELECT ENC_STR('KEY1', 'abc') FROM DUAL;
```

2. ENC_B64

```
SQL> SELECT ENC_B64('KEY1', 'abc') FROM DUAL;
```

3. DEC_STR

```
SQL> SELECT DEC_STR('KEY1', ENC_STR('KEY1', 'abc')) FROM DUAL;
```

4. DEC_B64

```
SQL> SELECT DEC_B64('KEY1', ENC_B64('KEY1', 'abc')) FROM DUAL;
```

5. INDEX_STR

```
DP 제품에 연동 하지 않을 경우
SQL> SELECT INDEX_STR('KEY1', 'abc', '') FROM DUAL;

DP 제품에 연동 할 경우
SQL> SELECT INDEX_STR('KEY1', 'abc', 'IX') FROM DUAL;
```

6. DEC_INDEX_STR, DEC_INDEX_B64

```
DP 제품에 연동 하지 않을 경우
SQL> SELECT DEC_INDEX_STR('KEY1', ENC_STR('KEY1', 'abc'), '') FROM DUAL;
SQL> SELECT DEC_INDEX_B64('KEY1', ENC_B64('KEY1', 'abc'), '') FROM DUAL;

DP 제품에 연동 할 경우
SQL> SELECT DEC_INDEX_STR('KEY1', ENC_STR('KEY1', 'abc'), 'IX') FROM DUAL;
SQL> SELECT DEC_INDEX_B64('KEY1', ENC_B64('KEY1', 'abc'), 'IX') FROM DUAL;
```

7. HASH_STR

```
SQL> SELECT HASH_STR( 71, 'abc' ) FROM DUAL;
```

8. HASH_B64

```
SQL> SELECT HASH_B64( 71, 'abc' ) FROM DUAL;
```

9. HEXTOB64

```
SQL> SELECT HEXTOB64('A305378D8F974F1C1537ED7CB0CB959245D1AC31') FROM DUAL;
```

10. B64TOHEX

```
SQL> SELECT B64TOHEX('owU3jY+XTxwVN+18sMuVkkXRrDE=') FROM DUAL;
```

11. CONFIG_REINIT

```
SQL> SELECT CONFIG_REINIT() FROM DUAL;
```

12. ENC_BLOB

```
SQL> SELECT ENC_BLOB('KEY1', UTL_RAW.CAST_TO_RAW('ABCDE')) FROM DUAL;
```

ENC_BLOB 함수는 C버전을 설치했을 경우에서만 제공된다.

13. DEC_BLOB

```
SQL> SELECT DEC_BLOB('KEY1', ENC_BLOB('KEY1',UTL_RAW.CAST_TO_RAW('ABCDE'))) FROM DUAL;
```

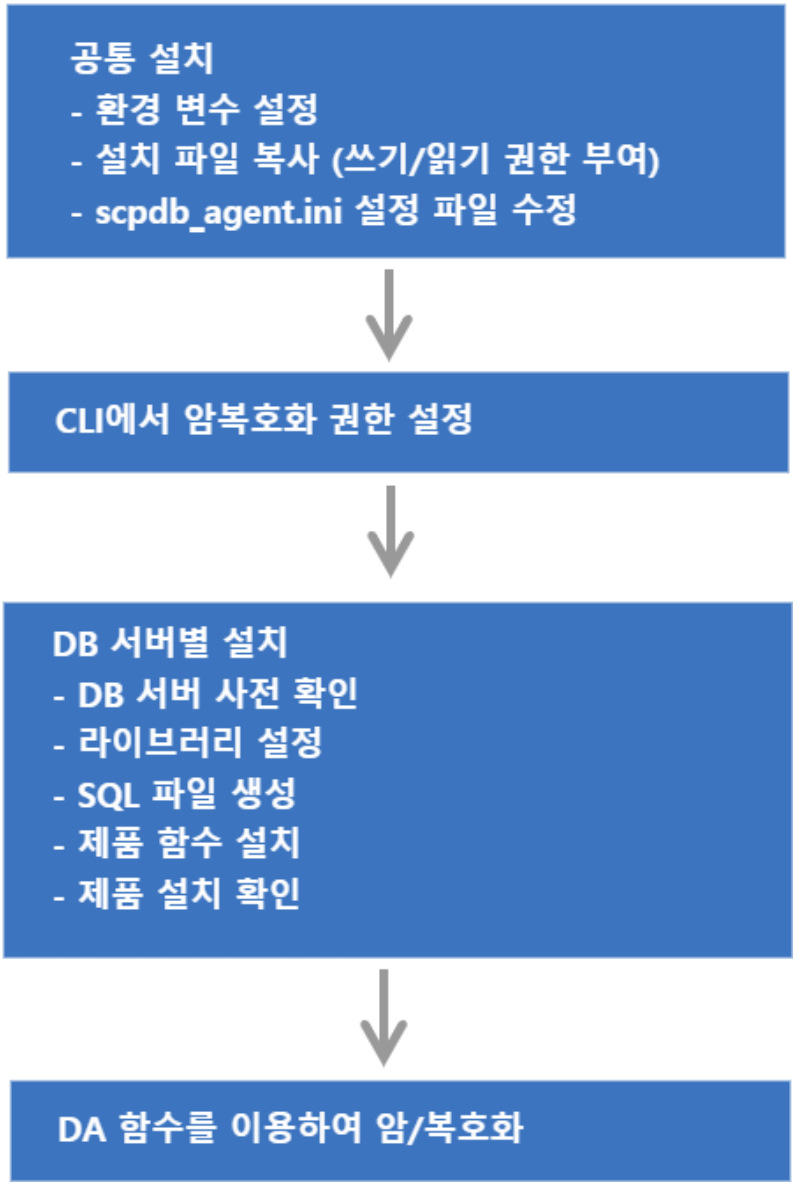
DEC_BLOB 함수는 C버전을 설치했을 경우에서만 제공된다.

4.

INFORMIX

DA는 DB 서버의 종류에 따라서 설치 및 운용 방법이 각각 다르다.

그림 4-1 설치 개념도



4.1 지원 운영체제 및 DB서버

DA에서 지원하는 운영체제 및 DB서버는 아래와 같다. 단, 특정 환경은 지원되지 않을 수도 있으므로, 제품 설치 전에 상세한 지원 가능 여부는 펜타시큐리티시스템으로 문의한다.

표 4-1 제품이 지원하는 운영체제 및 DB 서버 정보

구분	설명
운영체제	Windows, AIX, HP IA, HP PA-RISC, Linux, SUN, TRU64
DB 서버	ORACLE 8i ~, SQL Server 2012 ~, DB2 9.x ~, Tibero 4 SP 1 ~, MySQL 5.x ~, MariaDB 5.5, 10.0, 10.1, Cache DB 2009.1 ~, Informix IDS 9.x ~ Sybase ASE 15.7 SP61 ~, Sybase IQ 15.4 ~, CUBRID 2008 R1.3 ~ , PostgreSQL 9.4 ~

4.2 설치 파일의 구성 확인

DA의 설치 파일은 아래와 같은 명칭으로 압축 파일(zip) 형태로 제공된다.

- DA 설치파일: Install_DAmo_DA_v{버전}.zip
  - 압축을 해제 한 뒤, 설치 대상 DB 서버, OS 및 bit에 맞는 설치 파일을 준비한다.

설치 파일에 제품을 사용할 수 있는 '라이선스'는 포함되어 있지 않다.
펜타시큐리티시스템에 문의하여 '라이선스' 파일은 별도로 준비한다.

표 4-2 설치 파일(Install_DAmo_DA_v{버전}.zip)을 압축 해제 시, 디렉터리의 구성

구성	설명
_SampleScpsFiles	SG-KMS 연동 없이 암복호화를 테스트할 수 있는 테스트 키 파일
_TestAgentKeyPair	CLI에서 사용할 수 있는 테스트 키 쌍
Altibase	DA-ALT 제품의 설치 바이너리 폴더
Cache	DA-CDB 제품의 설치 바이너리 폴더
Cubrid	DA-CUB 제품의 설치 바이너리 폴더
DB2	DA-DB2 제품의 설치 바이너리 폴더
Informix	DA-IFX 제품의 설치 바이너리 폴더
MySQL	DA-MYQ 제품의 설치 바이너리 폴더
Oracle	DA-ORA 제품의 설치 바이너리 폴더
Postgres	DA-PGS 제품의 설치 바이너리 폴더
SQL Server	DA-MSQ 제품의 설치 바이너리 폴더

구성	설명
SybaseASE	DA-SYB 제품의 설치 바이너리 폴더
SybaseIQ	DA-SIQ 제품의 설치 바이너리 폴더
Tibero	DA-TIB 제품의 설치 바이너리 폴더

4.2.1 DB 서버 및 운영체제 별 SQL파일 구성

각 DB 서버 및 운영체제 별 SQL파일 구성은 다음과 같다. 다음 장에서 각 DB별로 SQL파일 설치 방법을 설명한다.

표 4-3 DB 서버 및 운영체제 별 SQL파일

DB 서버 종류	Linux 일 경우	Windows 일 경우
Oracle	000.da_user.pkg 000.da_user.sql 001.inner_function.ora(c 버전 설치 시) 002.user_interface.ora(JAVA 버전 설치 시) 002.user_interface_java.ora 003.grant_execute_functions.sql(JAVA 버전 설치 시) 005.securej_privilege.sql install_make.sh	000.da_user.pkg 000.da_user.sql 001.inner_function.ora(c 버전 설치 시) 002.user_interface.ora(JAVA 버전 설치 시) 002.user_interface_java.ora 003.grant_execute_functions.sql(JAVA 버전 설치 시) 005.securej_privilege.sql 009.da_test.sql
MYSQL	001.inner_function.mys 002.user_interface. mys 003.grant_execute_functions.sql 009.da_test.sql install_make.sh	001.inner_function.mys 002.user_interface.mys 003.grant_execute_functions.sql
TIBERO	001.inner_function.tbs(c 버전 설치 시) 001.inner_function_java.tbs(JAVA 버전 설치 시) 002.user_interface.tbs(c 버전 설치 시) 002.user_interface_java.tbs(JAVA 버전 설치 시) 003.grant_execute_functions.sql install_make.sh	001.inner_function_java.tbs 002.user_interface_java.tbs 003.grant_execute_functions.sql
INFORMIX	001.inner_function.ifx 002.user_interface. ifx 003.grant_execute_functions.sql 009.da_test.sql install_make.sh	해당 없음
POSTGRESQL	001.inner_function.post 002.user_interface.post 003.grant_execute_functions.post	해당 없음

DB 서버 종류	Linux 일 경우	Windows 일 경우
DB2	001.inner_function.db2 002.user_interface.db2 003.grant_execute_functions.sql install_make.sh	001.inner_function.db2 002.user_interface.db2 003.grant_execute_functions.sql
CUBRID	001.inner_function.sql 002.user_interface.sql	001.inner_function.sql 002.user_interface.sql
SYBASE	001.inner_function.sybase 002.user_interface.sybase 003.grant_execute_functions.sql install_make.sh	해당 없음
SYBASE IQ	001.inner_function.sybiq 002.user_interface. sybiq 003.grant_execute_functions.sql install_make.sh	해당 없음
SQL Server	해당 없음	001.inner_function.sql 002.user_interface. sql 003.grant_execute_functions.sql install_make.bat
Cache DB	SCP.xml install_make.sh	SCP.xml
Altibase	해당 없음	000.da_user.pkg 000.da_user.sql 001.inner_function.sql 002.user_interface. sql install_make.bat

4.2.2 공통 설치 파일 구성

환경에 관계 없이 공통적으로 사용하는 설치 파일은 아래와 같다.

표 4-4 공통 설치 파일 (sql파일 제외)

파일 분류	파일 명	파일 용도
Library 파일	libdamoscpdb.{so|a|sl|dll}	DA 메인 라이브러리. 주로 DBMS External Interface를 담당
	libdamocm-4.0.{so|a|sl|dll}	공통 모듈 라이브러리
	liblogw-0.2.{so|a|sl|dll}	로그를 기록하는 라이브러리
	libcis_cc-3.3.{so|a|sl|dll}	암호화, 복호화 기능을 제공하는 라이브러리
	libcis_ce-3.3.{so|a|sl|dll}	암호화, 복호화를 제외한 부가적인 기능을 제공하는 라이브러리(예: Base64, 인증서 관리, 특성 유지 암호화 등)

파일 분류	파일 명	파일 용도
설정 파일	scpdb_agent.ini	DA 구동시 실행에 필요한 설정정보를 참조
License 파일	damo_lic.cer	DA 구동 시 제품의 유효성을 검증하는데 사용
Agent key 파일	damo_agt_site.cer	SG-KMS 연동, CLI 프로그램에서 사용하는 인증서 쌍
	damo_agt.cer	
	damo_agt.key	
접근제어 파일	acl_cli 파일	DB 의 USER 별로 암·복호 권한을 설정하는데에 사용
	privilege.damo	권한 파일
JAVA class 파일 (Oracle, Tibero, Cubrid 설치 가능)	ScpAgentException.class	예외 처리 Class
	ScpCryptData.class	암호화 복호화 Class
SQL 파일	아래 새로운 표에 DB별로 표기함	

Agent Key 파일은 SG-KMS 연동에 필요한 키 발급를 참고하여 발급 받는다.

DA-PGS(PostgreSQL)의 경우, DB 서버 버전에 따라 libdamoscpdb.so 라이브러리 선택

- libdamoscpdb94.so (Postgres 9.4)
- libdamoscpdb95.so (Postgres 9.5)
- libdamoscpdb95AS.so (EDB Postgres 9.5)
- libdamoscpdb96AS.so (EDB Postgres 9.6)
- libdamoscpdb10.so (Postgres 10)

4.3 환경변수 설정

DA를 설치할 운영체제에 환경변수 DA_INST_HOME를 설정한다. 이 매뉴얼에서는 제품 설치 경로를 아래와 같이 가정하여 설명한다.

- Linux 환경일 경우: /home/dbms_api
- Windows 환경일 경우: E:₩dbms_api

주의) DA_INST_HOME 설정 시, 주의 사항

- 리눅스의 경우 "/root" 디렉토리로 설정을 권장하지 않는다.
- 윈도우의 경우 "바탕화면"으로 설정을 권장하지 않는다.

위의 경로로 설정할 경우 접근 권한 등의 이유로 문제가 발생할 가능성이 존재한다.

4.3.1 환경변수 설정 - Linux 환경일 경우

DA_INST_HOME 환경변수에 DA의 설치 디렉터리를 설정한다.

```
.profile을 사용하는 경우
export DA_INST_HOME=/home/dbms_api


.cshrc를 사용하는 경우
setenv DA_INST_HOME=/home/dbms_api

export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$DA_INST_HOME
```

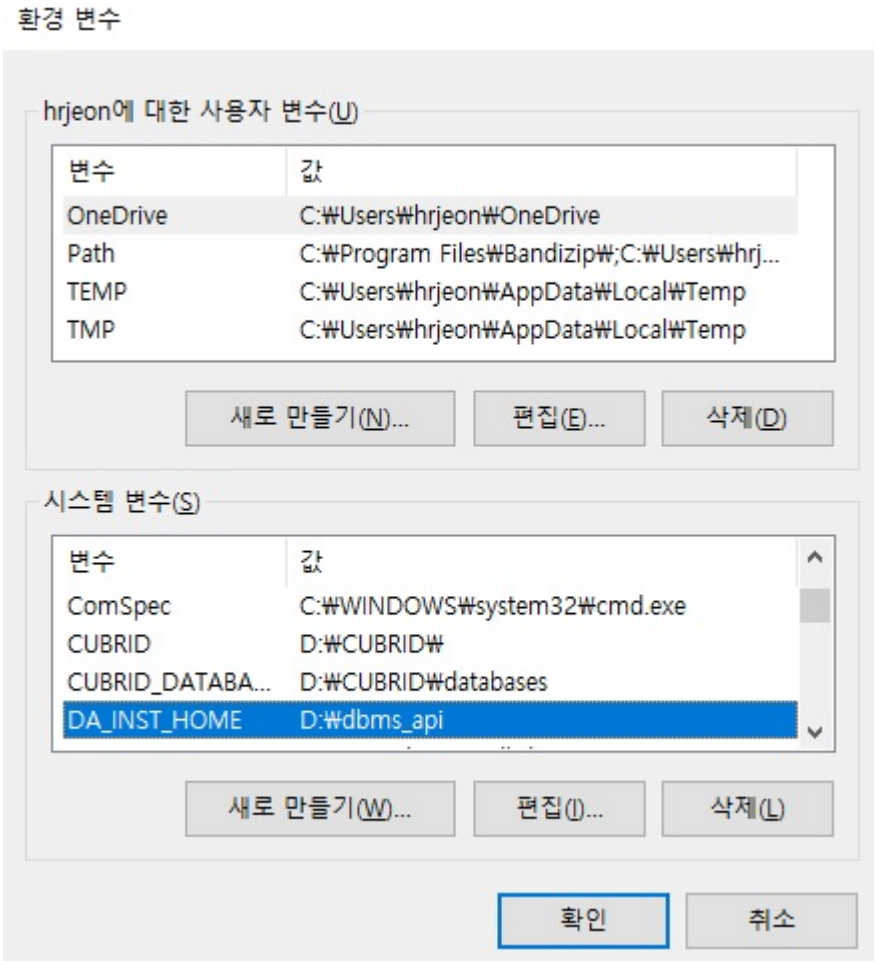
표 4-5 운영 체제별 라이브러리 PATH 명칭

운영 체제	라이브러리 PATH 명칭
HP_UX	SHLIB_PATH
AIX	LIBPATH
LINUX, SUN	LD_LIBRARY_PATH

4.3.2 환경변수 설정 - Windows 환경일 경우

[탐색기->내컴퓨터->등록정보->고급 탭->환경변수] 를 선택하여 나타나는 환경변수 설정 다이얼로그에서 시스템변수 DA_INST_HOME 변수와 값을 추가한다.

그림 4-2 DA_INST_HOME



4.4 DA의 설치 파일 복사

$DA_INST_HOME 디렉터리에 위 [설치 파일의 구성 확인]에서 나열된 파일들을 복사한다.

4.5 라이선스 파일 설정

$DA_INST_HOME 디렉토리에 라이선스 파일(damo_lic.cer)을 복사한다.

라이선스 파일명이 damo_lic.cer가 아닌 다른 이름으로 저장되어 있다면 변경해야 한다.

4.6 설치 파일의 접근 권한 부여 (Linux 설치 시)

Linux 사용자 계정에 DA 설치 파일(라이브러리, 실행 파일, 디렉토리)의 접근 권한을 부여한다. Windows에서 제품을 설치 할 경우, 이 과정은 생략한다.

```
$> cd $DA_INST_HOME
$> chmod 755 lib* acl_cli sql/install_make.sh
```

4.7 (SG-KMS 연동 시) SG-KMS에서 DA 정보 등록 및 연동에 필요한 키 내보내기

DA는 데이터를 암·복호화하기 위해 '암호화 키'가 필요하다. '암호화 키'를 얻기 위해서는 SG-KMS를 연동하거나 SCPS라는 암호화 키 파일을 이용할 수 있는데,
다음 설명은 SG-KMS 연동을 위해 SG-KMS에서 DA 정보를 등록하고 연동에 필요한 키를 내보내는 방법에 대해서 설명한다.

4.7.1 사전 준비

4.7.1.1 지원 SG-KMS 버전

DA와 연동 가능한 SG-KMS 버전은 다음과 같다.

표 4-6 연동 가능한 SG-KMS 버전

SG-KMS Major version	SG-KMS Minor version
SG-KMS v3.0	v3.0.9.0 이상 연동 가능
SG-KMS v4.0	v4.0.104.5 이상 연동 가능

SG-KMS v2.3 연동은 미지원한다.

4.7.1.2 연동 전 점검 항목

SG-KMS 연동을 위해서는 다음과 같이 사전 준비가 필요하다.

- SG-KMS 관리도구
- SG-KMS 관리도구에 접속할 수 있는 ID와 비밀번호
- SG-KMS 매뉴얼

이 매뉴얼에서는 SG-KMS의 키 종류 및 SG-KMS 관리도구 사용 방법에 대한 내용은 다루지 않으므로 **[SG-KMS 매뉴얼]**을 참고한다.

4.7.1.3 SG-KMS 연동에 필요한 키 발급

DA와 SG-KMS의 연동을 위해서는 먼저 SG-KMS에 DA를 Agent로 등록해야한다. 등록하는 절차는 SG-KMS 사용자설명서의 '대칭키를 사용하는 D'Amo Agent 등록 안내서'를 참고한다.

[D'Amo Agent 키] 파일, [Agent ID(CN)]와 [서비스 ID]정보는 DA와 SG-KMS 연동을 위한 설정 과정에서 필요하다. SG-KMS 관리자는 DA 설치 엔지니어에게 안전하게 전달한다.

SG-KMS 관리자에게 받은 D'Amo Agent의 인증서 및 키 파일 명을 다음과 같이 변경후 $DA_INST_HOME/key 디렉터리에 복사한다.

표 4-7 SG-KMS 연동에 필요한 키 목록

키 구분	생성된 키 파일명	변경할 키 파일명
사이트 키	damo-site_{발행기관/부서명}-SITE_V3.cer	damo_agt_site.cer
Agent 키	damo-scp_SITE_V3-{Agent 이름}.cer	damo_agt.cer
	damo-scp_SITE_V3-{Agent 이름}.key	damo_agt.key
	damo-scp_SITE_V3-{Agent 이름}.spin	damo_agt.spin

Agent 키 경로 지정, 키 이름 변경은 반드시 필요한 작업은 아니지만 관리를 위해 변경하는 것을 권장한다.

4.8 설정 파일(scpdb_agent.ini) 수정

DA의 운용을 위해 사용하는 scpdb_agent.ini 설정 파일 수정 방법에 대해 설명한다. DA는 SG-KMS를 이용하거나 SCPS 파일을 이용하여 암호화 키를 얻기 때문에 고객의 환경에 맞게 설정해야 한다.

$DA_INST_HOME 디렉터리에 있는 scpdb_agent.ini 설정 파일에서 아래 항목의 값을 수정한다.

1. 설정 파일의 [KEYINFO] 항목 - [KEY1]에 암호화 키 정보를 입력한다.

```
[KEYINFO]
KEY1=암복호화 하려는 암호화 키 정보를 입력한다.
//키 정보는 다음과 같은 값을 입력할 수 있다.
///ServiceID: SG-KMS에 생성한 서비스 ID를 입력한다.
///SCP_FilePath: SG-KMS에서 서비스 내보내기를 통해 발급한 SCPS 파일의 절대 경로 및 파일명, 확장자를 입력한다.

//*예제 - Windows 경우
KEY1=DA_AES256
KEY2=C:\DA\Policy\S_AES128.SCPS\DA_AES256.scps
KEY3=DA_AES256,C:\DA\Policy\S_AES128.SCPS\DA_AES256.scps

//*예제 - Linux 또는 Unix 경우
KEY1=DA_AES256
KEY2=/home/dbms_api/key/DA_AES256.scps
KEY3=DA_AES256,/home/dbms_api/key/DA_AES256.scps
```

ServiceID를 입력 할 경우 SG-KMS와 통신을 통해 암호화 정책을 가져와 데이터 암복호화를 수행한다.

SCP_FilePath를 입력 할 경우 KMS와 통신하지 않고 서버의 SCP 파일을 통해 암호화 정책을 가져와 데이터 암복호화를 수행한다.

ServiceID와 SCPS 파일명을 동시에 입력 시, SG-KMS와 네트워크 연결 실패 하면 SCPS 파일을 이용하여 암호화 한다.
ServiceID와 SCPS 파일명 사이에 공백이 있으면 SCPS 파일을 읽을 수 없다. 따라서 예제와 같이 띄어쓰기를 하지 않고 ServiceID와 SCPS 파일 경로를 붙여서 입력한다.

2. 설정 파일의 [Server], [Server2] 항목을 수정한다.

```
[Server]
ServerIP: SG-KMS의 IP를 입력한다.
ServerPort: SG-KMS의 포트번호를 입력한다. 기본값은 2525이다.

//*예제 - 모든 OS 공통
```

```
ServerIP=192.168.22.25
ServerPort=2525
```

```
[Server2]
ServerIP: 이중화를 위한 2번 SG-KMS의 IP를 입력한다.
ServerPort: 이중화를 위한 2번 SG-KMS의 포트번호를 입력한다. 기본값은 2525이다.

//*예제 - 모든 OS 공통
ServerIP=192.168.22.26
ServerPort=2525
```

ServerIP는 최소 1개 ~ 최대 10개까지 등록이 가능하다.

3. 설정 파일의 [AGENT] 항목을 설정한다.

```
[AGENT]
AgentID=SG-KMS 관리도구에서 설정한 D'Amo Agent의 Agent ID
LogDir=로그가 저장될 디렉터리 위치
LogLevel=로그가 남는 수준
SiteCertFilePath=SG-KMS 장비에서 설정한 해당 장비의 사이트 공개키(.cer)
CertFilePath=SG-KMS 관리도구에서 설정한 D'Amo Agent의 공개키(.cer)
KeyFilePath=SG-KMS 관리도구에서 설정한 D'Amo Agent의 비공개키(.key)
SPIN=SG-KMS 관리도구에서 설정한 D'Amo Agent의 SPIN으로, damo-scp_SITE_V3-[Agent이름].spin
파일의 값

//[예제]
AgentID=DA
LogDir=/home/dbms_api/log
LogLevel=4
SiteCertFilePath=/home/dbms_api/key/damo_agt_site.cer
CertFilePath=/home/dbms_api/key/damo_agt.cer
KeyFilePath=/home/dbms_api/key/damo_agt.key
SPIN=XaMh1ylXUh123XUh
```

로그가 남는 수준(LogLevel)에는 아래 5가지 숫자 입력이 가능하며, 각 값의 설정은 다음과 같다.
- 0: 아무 로그도 남기지 않을 경우
- 2: 경고 로그를 파일에 기록
- 4: 에러 로그와 경고 로그를 파일에 기록

- 6: 정보 로그, 에러 로그, 경고 로그를 파일에 기록
- 8: 디버그, 정보 로그, 에러 로그, 경고 로그를 파일에 기록

제품 운영 중 scpdb_agent.ini 파일을 수정하면 CONFIG_REINIT() 함수를 호출해야 변경된 내용이 적용된다.

4.9 CLI에서 권한 설정

$DA_INST_HOME 디렉터리에서 acl_cli 파일을 실행하여 USER 단위로 암/복호화 권한을 설정한다. USER 는 DB의 소유자명이고, KEY는 설정 파일(scpdb_agent.ini)의 [KEYINFO] 중 ALIAS 값(예: KEY1)이다.

CLI 명령어를 자세히 보려면 help 명령어를 실행한다.

```
CLI 실행 방법
$> cd $DA_INST_HOME
$> ./acl_cli  -start
Enter the PIN of CLI-key. : damo_agt.key 의 비밀번호
```

```
권한 추가할 경우
D'Amo > SET PRIV ENC [USER]"[KEY]"1"1
D'Amo > SAVE ALL
D'Amo > SHOW ALL

예제)
D'Amo > SET PRIV ENC SCOTT"KEY1"1"1
D'Amo > SET PRIV ENC SCOTT"KEY2"1"1
```

scpdb_agent.ini 설정 파일이 아래 예제와 같을 경우, CLI에서 권한 설정할 때 입력해야 하는 2번째 [KEY] 인자 값에는 KEY1을 입력해야 한다. (※ARIA256을 입력하는 것이 아님)

#scpdb_agent.ini 설정 파일 예제
[KEYINFO]
KEY1=ARIA256

```
권한 삭제할 경우
D'Amo > DEL PRIV ENC [USER]"[KEY]
D'Amo > SAVE ALL
D'Amo > SHOW ALL
```

CLI 에서 권한을 추가하거나 삭제 한 경우 반드시 SAVE ALL 명령어를 실행하며 SHOW ALL 명령어를 이용하여 적용 여부를 확인 한다.

4.10 DB 서버 사전 확인

DA를 INFORMIX 환경에서 설치 하기 전에 다음과 같은 사항들을 확인한다.

- DB 엔진 설치 계정에 폴더 생성
- DB 사용자로 SCP 계정 생성 (권고사항)
- DB 서버의 DBA 권한 계정
- DB 엔진 설치 계정의 . profile 파일 수정

4.11 라이브러리 링크 설정

또한 INFORMIX에서 인식할 수 있는 lib 위치에 library 파일의 symbolic link 로 생성한다.

libdamoscpdb.{so|a|sl} 파일의 symbolic link 파일을 생성할 때는 항상 so 파일로 생성한다.
고객의 환경에 따라 /usr/lib, /usr/lib/hpux64 (HP-UX IA64), /usr/lib/pa20_64 (HP-UX_PA RISC 64bit) 에 symbolic link 파일을 생성한다.

```
$> ln -s $DA_INST_HOME/libcis_cc-3.3.{so|a|sl} libcis_cc-3.3.{so|a|sl}
$> ln -s $DA_INST_HOME/libcis_ce-3.3.{so|a|sl} libcis_ce-3.3.{so|a|sl}
$> ln -s $DA_INST_HOME/liblogw-0.2.{so|a|sl} liblogw-0.2.{so|a|sl}
$> ln -s $DA_INST_HOME/libdamoscpdb.{so|a|sl} libdamoscpdb.so
$> ln -s $DA_INST_HOME/libdamocm-4.0.{so|a|sl} libdamocm-4.0.{so|a|sl}
```

4.12 sql 파일 생성

1. $DA_INST_HOME/sql 디렉터리에서 install_make.sh을 이용하여 설치할 sql 파일을 생성한다. D_INI 는 설정 파일(scpdb_agent.ini), D_LIB_FIX는 INFORMIX 라이브러리 경로다.

```
$> cd $DA_INST_HOME/sql
$> ./install_make.sh D_INI D_LIB_IFX

[예제]
※ D_INI D_LIB_IFX 경로를 /home/dbms_api, /home/Informix/ids121/lib으로 가정한다.
$> ./install_make.sh /home/dbms_api /home/informix/ids121/lib
D_INI_PATH are replaced by /home/dbms_api
D_LIB_IFX are replaced by /home/informix/ids121/lib
$>
```

2. 아래 2개의 파일이 생성되었는지 확인한다.

- 001.inner_function.ifx.sql
- 002.user_interface.ifx.sql

4.13 제품 함수 설치

1. * $DA_INST_HOME/sql 위치에서 DB 접속 후 설치를 원하는 DB계정에 함수를 설치한다. SCP 계정을 생성한 경우 SCP 계정에 설치한다.

```
$>dbaccess dbname 001.inner_function.ifx.sql
Database selected.
Routine dropped.
...
Routine created.
Database closed.
$>dbaccess dbname 002.user_interface.ifx.sql
Database selected.
Routine dropped.
...
Routine created.
```

```
Database closed.
$>
```

2. 특정 DB 사용자에게 함수 실행 권한을 부여한다. 모든 사용자에게 함수 실행 권한을 부여할 때는 003.grant_execute_functions.sql 파일을 실행한다.

```
$>dbaccess dbname 003.grant_execute_functions.sql
Database selected.
Permission granted.
Database closed.
$>
```

4.14 제품 설치 확인

DB 서버에서 암/복호화 함수를 호출하여 설치를 성공했는지 확인한다.

```
dbaccess>SELECT ENC_STR('KEY1', 'abc')
(expression)E6878572B3287A049906A8CA57F0207C
1 row(s) retrieved.

dbaccess> SELECT DEC_STR('KEY1', ENC_STR('KEY1', 'abc'))
(expression)abc

1 row(s) retrieved.
```

4.15 제품 운용

4.15.1 함수 설명

DA에서 제공되는 함수와 사용하는 방법에 대해서 설명한다.

4.15.1.1 파라미터 설명

4.15.1.1.1 I_KEY

I_KEY : 암호화/복호화 때 사용하는 암호화 키
- 설정 파일(scpdb_agent.ini)의 [KEYINFO] 중 ALIAS 값
- 아래 [설정 파일 예제]의 경우 ALIAS는 'KEY1'과 'KEY2'이다.
- ALIAS는 SCPS파일로 암호화 할 것인지, SG-KMS의 서비스 ID로 암호화 할 것인지 설정 가능하다.

[설정 파일 예제]
- 설정 파일(scpdb_agent.ini)의 [KEYINFO] 중 ALIAS 값
KEY1=AES256.SCPS
KEY2=ARIA256

[함수 사용 예제]
SELECT ENC_B64('KEY1', 'abc') FROM DUAL;

4.15.1.1.2 I_DATA

I_DATA : 평문(암호화 함수일 경우), 암호문(복호화 함수일 경우)

DA에서 제공하는 암호화 함수에서 성능 향상을 위해 라이브러리에서 정책 이름(KEYINFO ALIAS)을 바탕으로 Cash하여 동작한다. 정책 이름은 같은데 실제 키(대칭 키)가 다른 정책을 사용하면 Cash에서 이전 키로 복호화를 시도하여 오류가 발생한다. 이 상황을 해결하기 위해서는 데이터베이스 서버 재시작이 필요하다.

표 4-8 DA 함수 (INFORMIX)

함수 명	입력		출력
ENC_STR	I_KEY	IN 문자열,	Hex String 암호문
	I_DATA	IN 문자열 (평문)	
ENC_B64	I_KEY	IN 문자열,	Base64 Encording 암호문
	I_DATA	IN 문자열 (평문)	
DEC_STR	I_KEY	IN 문자열,	평문
	I_DATA	IN 문자열 (Hex String 암호문)	
DEC_B64	I_KEY	IN 문자열,	평문

	I_DATA	IN 문자열 (base64 String 암호문)		
INDEX_STR	I_KEY	IN 문자열,		Hex String 암호문
	I_DATA	IN 문자열 (평문),		
	I_TYPE	IN 문자열 '' or 'IX '(Plug-IN 연동 시 사용)		
DEC_INDEX_STR	I_KEY	IN 문자열,		Hex String 암호문 입력받아 OPE 데이터
	I_DATA	IN 문자열 (암호문),		
	I_TYPE	IN 문자열 '' or 'IX '(Plug-IN 연동 시 사용)		
DEC_INDEX_B64	I_KEY	IN 문자열,		Base64 Encording 암호문 입력받아 OPE 데이터
	I_DATA	IN 문자열 (암호문),		
	I_TYPE	IN 문자열 '' or 'IX '(Plug-IN 연동 시 사용)		
HASH_STR	I_ALOG	IN 숫자,	SHA1 =70	Hex String 해쉬 암호문
			SHA256 =71	
			SHA384 =72	
			SHA512 =73	
			HAS160 =74	
	I_DATA	IN 문자열		
HASH_B64	I_ALOG	IN 숫자,	SHA1 =70	Base64 String 해쉬 암호문
			SHA256 =71	
			SHA384 =72	
			SHA512 =73	
			HAS160 =74	
	I_DATA	IN 문자열		
HEXTOB64	I_DATA	IN 문자열 (Hex String 암호문)		base64 Encording 암호문
B64TOHEX	I_DATA	IN 문자열 (base64 Encording 암호문)		Hex String 암호문
CONFIG_REINIT				성공시 'SUCCESS', 그 외 에러

4.15.2 함수 호출 예제

1. ENC_STR

```
SQL> SELECT ENC_STR('KEY1', 'abc') FROM DUAL;
```

2. ENC_B64

```
SQL> SELECT ENC_B64('KEY1', 'abc') FROM DUAL;
```

3. DEC_STR

```
SQL> SELECT DEC_STR('KEY1', ENC_STR('KEY1', 'abc')) FROM DUAL;
```

4. DEC_B64

```
SQL> SELECT DEC_B64('KEY1', ENC_B64('KEY1', 'abc')) FROM DUAL;
```

5. INDEX_STR

```
DP 제품에 연동 하지 않을 경우
SQL> SELECT INDEX_STR('KEY1', 'abc', '') FROM DUAL;

DP 제품에 연동 할 경우
SQL> SELECT INDEX_STR('KEY1', 'abc', 'IX') FROM DUAL;
```

6. DEC_INDEX_STR, DEC_INDEX_B64

```
DP 제품에 연동 하지 않을 경우
SQL> SELECT DEC_INDEX_STR('KEY1', ENC_STR('KEY1', 'abc'), '') FROM DUAL;
SQL> SELECT DEC_INDEX_B64('KEY1', ENC_B64('KEY1', 'abc'), '') FROM DUAL;

DP 제품에 연동 할 경우
SQL> SELECT DEC_INDEX_STR('KEY1', ENC_STR('KEY1', 'abc'), 'IX') FROM DUAL;
SQL> SELECT DEC_INDEX_B64('KEY1', ENC_B64('KEY1', 'abc'), 'IX') FROM DUAL;
```

7. HASH_STR

```
SQL> SELECT HASH_STR( 71, 'abc' ) FROM DUAL;
```

8. HASH_B64

```
SQL> SELECT HASH_B64( 71, 'abc' ) FROM DUAL;
```

9. HEXTOB64

```
SQL> SELECT HEXTOB64('A305378D8F974F1C1537ED7CB0CB959245D1AC31') FROM DUAL;
```

10. B64TOHEX

```
SQL> SELECT B64TOHEX('owU3jY+XTxwVN+18sMuVkkXRrDE=') FROM DUAL;
```

11. CONFIG_REINIT

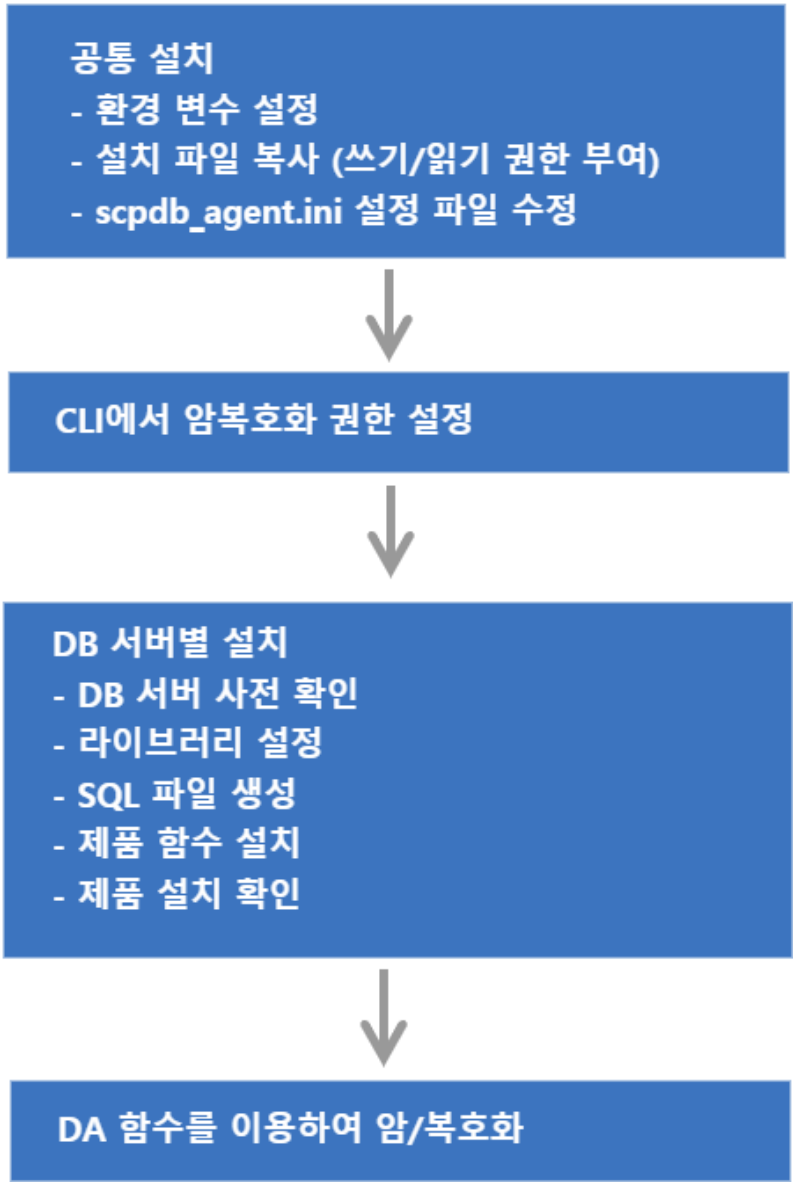
```
SQL> SELECT CONFIG_REINIT() FROM DUAL;
```

5.

PostgreSQL

DA는 DB 서버의 종류에 따라서 설치 및 운용 방법이 각각 다르다.

그림 5-1 설치 개념도



5.1 지원 운영체제 및 DB서버

DA에서 지원하는 운영체제 및 DB서버는 아래와 같다. 단, 특정 환경은 지원되지 않을 수도 있으므로, 제품 설치 전에 상세한 지원 가능 여부는 펜타시큐리티시스템으로 문의한다.

표 5-1 제품이 지원하는 운영체제 및 DB 서버 정보

구분	설명
운영체제	Windows, AIX, HP IA, HP PA-RISC, Linux, SUN, TRU64
DB 서버	ORACLE 8i ~, SQL Server 2012 ~, DB2 9.x ~, Tibero 4 SP 1 ~, MySQL 5.x ~, MariaDB 5.5, 10.0, 10.1, Cache DB 2009.1 ~, Informix IDS 9.x ~ Sybase ASE 15.7 SP61 ~, Sybase IQ 15.4 ~, CUBRID 2008 R1.3 ~ , PostgreSQL 9.4 ~

5.2 설치 파일의 구성 확인

DA의 설치 파일은 아래와 같은 명칭으로 압축 파일(zip) 형태로 제공된다.

- DA 설치파일: Install_DAmo_DA_v{버전}.zip
  - 압축을 해제 한 뒤, 설치 대상 DB 서버, OS 및 bit에 맞는 설치 파일을 준비한다.

설치 파일에 제품을 사용할 수 있는 '라이선스'는 포함되어 있지 않다.
펜타시큐리티시스템에 문의하여 '라이선스' 파일은 별도로 준비한다.

표 5-2 설치 파일(Install_DAmo_DA_v{버전}.zip)을 압축 해제 시, 디렉터리의 구성

구성	설명
_SampleScpsFiles	SG-KMS 연동 없이 암복호화를 테스트할 수 있는 테스트 키 파일
_TestAgentKeyPair	CLI에서 사용할 수 있는 테스트 키 쌍
Altibase	DA-ALT 제품의 설치 바이너리 폴더
Cache	DA-CDB 제품의 설치 바이너리 폴더
Cubrid	DA-CUB 제품의 설치 바이너리 폴더
DB2	DA-DB2 제품의 설치 바이너리 폴더
Informix	DA-IFX 제품의 설치 바이너리 폴더
MySQL	DA-MYQ 제품의 설치 바이너리 폴더
Oracle	DA-ORA 제품의 설치 바이너리 폴더
Postgres	DA-PGS 제품의 설치 바이너리 폴더
SQL Server	DA-MSQ 제품의 설치 바이너리 폴더

구성	설명
SybaseASE	DA-SYB 제품의 설치 바이너리 폴더
SybaseIQ	DA-SIQ 제품의 설치 바이너리 폴더
Tibero	DA-TIB 제품의 설치 바이너리 폴더

5.2.1 DB 서버 및 운영체제 별 SQL파일 구성

각 DB 서버 및 운영체제 별 SQL파일 구성은 다음과 같다. 다음 장에서 각 DB별로 SQL파일 설치 방법을 설명한다.

표 5-3 DB 서버 및 운영체제 별 SQL파일

DB 서버 종류	Linux 일 경우	Windows 일 경우
Oracle	000.da_user.pkg 000.da_user.sql 001.inner_function.ora(c 버전 설치 시) 002.user_interface.ora(JAVA 버전 설치 시) 002.user_interface_java.ora 003.grant_execute_functions.sql(JAVA 버전 설치 시) 005.securej_privilege.sql install_make.sh	000.da_user.pkg 000.da_user.sql 001.inner_function.ora(c 버전 설치 시) 002.user_interface.ora(JAVA 버전 설치 시) 002.user_interface_java.ora 003.grant_execute_functions.sql(JAVA 버전 설치 시) 005.securej_privilege.sql 009.da_test.sql
MYSQL	001.inner_function.mys 002.user_interface. mys 003.grant_execute_functions.sql 009.da_test.sql install_make.sh	001.inner_function.mys 002.user_interface.mys 003.grant_execute_functions.sql
TIBERO	001.inner_function.tbs(c 버전 설치 시) 001.inner_function_java.tbs(JAVA 버전 설치 시) 002.user_interface.tbs(c 버전 설치 시) 002.user_interface_java.tbs(JAVA 버전 설치 시) 003.grant_execute_functions.sql install_make.sh	001.inner_function_java.tbs 002.user_interface_java.tbs 003.grant_execute_functions.sql
INFORMIX	001.inner_function.ifx 002.user_interface. ifx 003.grant_execute_functions.sql 009.da_test.sql install_make.sh	해당 없음
POSTGRESQL	001.inner_function.post 002.user_interface.post 003.grant_execute_functions.post	해당 없음

DB 서버 종류	Linux 일 경우	Windows 일 경우
DB2	001.inner_function.db2 002.user_interface.db2 003.grant_execute_functions.sql install_make.sh	001.inner_function.db2 002.user_interface.db2 003.grant_execute_functions.sql
CUBRID	001.inner_function.sql 002.user_interface.sql	001.inner_function.sql 002.user_interface.sql
SYBASE	001.inner_function.sybase 002.user_interface.sybase 003.grant_execute_functions.sql install_make.sh	해당 없음
SYBASE IQ	001.inner_function.sybiq 002.user_interface. sybiq 003.grant_execute_functions.sql install_make.sh	해당 없음
SQL Server	해당 없음	001.inner_function.sql 002.user_interface. sql 003.grant_execute_functions.sql install_make.bat
Cache DB	SCP.xml install_make.sh	SCP.xml
Altibase	해당 없음	000.da_user.pkg 000.da_user.sql 001.inner_function.sql 002.user_interface. sql install_make.bat

5.2.2 공통 설치 파일 구성

환경에 관계 없이 공통적으로 사용하는 설치 파일은 아래와 같다.

표 5-4 공통 설치 파일 (sql파일 제외)

파일 분류	파일 명	파일 용도
Library 파일	libdamoscpdb.{so|a|sl|dll}	DA 메인 라이브러리. 주로 DBMS External Interface를 담당
	libdamocm-4.0.{so|a|sl|dll}	공통 모듈 라이브러리
	liblogw-0.2.{so|a|sl|dll}	로그를 기록하는 라이브러리
	libcis_cc-3.3.{so|a|sl|dll}	암호화, 복호화 기능을 제공하는 라이브러리
	libcis_ce-3.3.{so|a|sl|dll}	암호화, 복호화를 제외한 부가적인 기능을 제공하는 라이브러리(예: Base64, 인증서 관리, 특성 유지 암호화 등)

파일 분류	파일 명	파일 용도
설정 파일	scpdb_agent.ini	DA 구동시 실행에 필요한 설정정보를 참조
License 파일	damo_lic.cer	DA 구동 시 제품의 유효성을 검증하는데 사용
Agent key 파일	damo_agt_site.cer	SG-KMS 연동, CLI 프로그램에서 사용하는 인증서 쌍
	damo_agt.cer	
	damo_agt.key	
접근제어 파일	acl_cli 파일	DB 의 USER 별로 암·복호 권한을 설정하는데에 사용
	privilege.damo	권한 파일
JAVA class 파일 (Oracle, Tibero, Cubrid 설치 가능)	ScpAgentException.class	예외 처리 Class
	ScpCryptData.class	암호화 복호화 Class
SQL 파일	아래 새로운 표에 DB별로 표기함	

Agent Key 파일은 SG-KMS 연동에 필요한 키 발급를 참고하여 발급 받는다.

DA-PGS(PostgreSQL)의 경우, DB 서버 버전에 따라 libdamoscpdb.so 라이브러리 선택

- libdamoscpdb94.so (Postgres 9.4)
- libdamoscpdb95.so (Postgres 9.5)
- libdamoscpdb95AS.so (EDB Postgres 9.5)
- libdamoscpdb96AS.so (EDB Postgres 9.6)
- libdamoscpdb10.so (Postgres 10)

5.3 환경변수 설정

DA를 설치할 운영체제에 환경변수 DA_INST_HOME를 설정한다. 이 매뉴얼에서는 제품 설치 경로를 아래와 같이 가정하여 설명한다.

- Linux 환경일 경우: /home/dbms_api
- Windows 환경일 경우: E:₩dbms_api

주의) DA_INST_HOME 설정 시, 주의 사항

- 리눅스의 경우 "/root" 디렉토리로 설정을 권장하지 않는다.
- 윈도우의 경우 "바탕화면"으로 설정을 권장하지 않는다.

위의 경로로 설정할 경우 접근 권한 등의 이유로 문제가 발생할 가능성이 존재한다.

5.3.1 환경변수 설정 - Linux 환경일 경우

DA_INST_HOME 환경변수에 DA의 설치 디렉터리를 설정한다.

```
.profile을 사용하는 경우
export DA_INST_HOME=/home/dbms_api


.cshrc를 사용하는 경우
setenv DA_INST_HOME=/home/dbms_api

export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$DA_INST_HOME
```

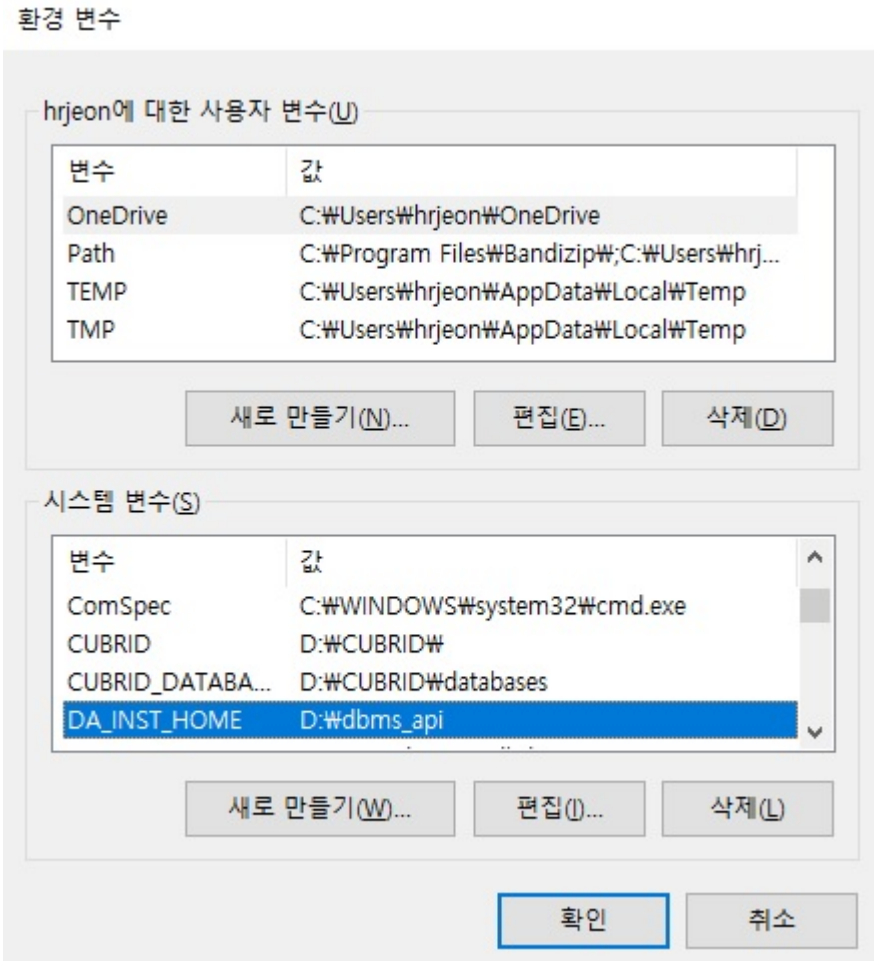
표 5-5 운영 체제별 라이브러리 PATH 명칭

운영 체제	라이브러리 PATH 명칭
HP_UX	SHLIB_PATH
AIX	LIBPATH
LINUX, SUN	LD_LIBRARY_PATH

5.3.2 환경변수 설정 - Windows 환경일 경우

[탐색기->내컴퓨터->등록정보->고급 탭->환경변수] 를 선택하여 나타나는 환경변수 설정 다이얼로그에서 시스템변수 DA_INST_HOME 변수와 값을 추가한다.

그림 5-2 DA_INST_HOME



5.4 DA의 설치 파일 복사

$DA_INST_HOME 디렉터리에 위 [설치 파일의 구성 확인]에서 나열된 파일들을 복사한다.

5.5 라이선스 파일 설정

$DA_INST_HOME 디렉토리에 라이선스 파일(damo_lic.cer)을 복사한다.

라이선스 파일명이 damo_lic.cer가 아닌 다른 이름으로 저장되어 있다면 변경해야 한다.

5.6 설치 파일의 접근 권한 부여 (Linux 설치 시)

Linux 사용자 계정에 DA 설치 파일(라이브러리, 실행 파일, 디렉토리)의 접근 권한을 부여한다. Windows에서 제품을 설치 할 경우, 이 과정은 생략한다.

```
$> cd $DA_INST_HOME
$> chmod 755 lib* acl_cli sql/install_make.sh
```

5.7 (SG-KMS 연동 시) SG-KMS에서 DA 정보 등록 및 연동에 필요한 키 내보내기

DA는 데이터를 암·복호화하기 위해 '암호화 키'가 필요하다. '암호화 키'를 얻기 위해서는 SG-KMS를 연동하거나 SCPS라는 암호화 키 파일을 이용할 수 있는데,
다음 설명은 SG-KMS 연동을 위해 SG-KMS에서 DA 정보를 등록하고 연동에 필요한 키를 내보내는 방법에 대해서 설명한다.

5.7.1 사전 준비

5.7.1.1 지원 SG-KMS 버전

DA와 연동 가능한 SG-KMS 버전은 다음과 같다.

표 5-6 연동 가능한 SG-KMS 버전

SG-KMS Major version	SG-KMS Minor version
SG-KMS v3.0	v3.0.9.0 이상 연동 가능
SG-KMS v4.0	v4.0.104.5 이상 연동 가능

SG-KMS v2.3 연동은 미지원한다.

5.7.1.2 연동 전 점검 항목

SG-KMS 연동을 위해서는 다음과 같이 사전 준비가 필요하다.

- SG-KMS 관리도구
- SG-KMS 관리도구에 접속할 수 있는 ID와 비밀번호
- SG-KMS 매뉴얼

이 매뉴얼에서는 SG-KMS의 키 종류 및 SG-KMS 관리도구 사용 방법에 대한 내용은 다루지 않으므로 **[SG-KMS 매뉴얼]**을 참고한다.

5.7.1.3 SG-KMS 연동에 필요한 키 발급

DA와 SG-KMS의 연동을 위해서는 먼저 SG-KMS에 DA를 Agent로 등록해야한다. 등록하는 절차는 SG-KMS 사용자설명서의 '대칭키를 사용하는 D'Amo Agent 등록 안내서'를 참고한다.

[D'Amo Agent 키] 파일, [Agent ID(CN)]와 [서비스 ID]정보는 DA와 SG-KMS 연동을 위한 설정 과정에서 필요하다. SG-KMS 관리자는 DA 설치 엔지니어에게 안전하게 전달한다.

SG-KMS 관리자에게 받은 D'Amo Agent의 인증서 및 키 파일 명을 다음과 같이 변경후 $DA_INST_HOME/key 디렉터리에 복사한다.

표 5-7 SG-KMS 연동에 필요한 키 목록

키 구분	생성된 키 파일명	변경할 키 파일명
사이트 키	damo-site_{발행기관/부서명}-SITE_V3.cer	damo_agt_site.cer
Agent 키	damo-scp_SITE_V3-{Agent 이름}.cer	damo_agt.cer
	damo-scp_SITE_V3-{Agent 이름}.key	damo_agt.key
	damo-scp_SITE_V3-{Agent 이름}.spin	damo_agt.spin

Agent 키 경로 지정, 키 이름 변경은 반드시 필요한 작업은 아니지만 관리를 위해 변경하는 것을 권장한다.

5.8 설정 파일(scpdb_agent.ini) 수정

DA의 운용을 위해 사용하는 scpdb_agent.ini 설정 파일 수정 방법에 대해 설명한다. DA는 SG-KMS를 이용하거나 SCPS 파일을 이용하여 암호화 키를 얻기 때문에 고객의 환경에 맞게 설정해야 한다.

$DA_INST_HOME 디렉터리에 있는 scpdb_agent.ini 설정 파일에서 아래 항목의 값을 수정한다.

1. 설정 파일의 [KEYINFO] 항목 - [KEY1]에 암호화 키 정보를 입력한다.

```
[KEYINFO]
KEY1=암복호화 하려는 암호화 키 정보를 입력한다.
//키 정보는 다음과 같은 값을 입력할 수 있다.
///ServiceID: SG-KMS에 생성한 서비스 ID를 입력한다.
///SCP_FilePath: SG-KMS에서 서비스 내보내기를 통해 발급한 SCPS 파일의 절대 경로 및 파일명, 확장자를 입력한다.

//*예제 - Windows 경우
KEY1=DA_AES256
KEY2=C:\DA\Policy\S_AES128.SCPS\DA_AES256.scps
KEY3=DA_AES256,C:\DA\Policy\S_AES128.SCPS\DA_AES256.scps

//*예제 - Linux 또는 Unix 경우
KEY1=DA_AES256
KEY2=/home/dbms_api/key/DA_AES256.scps
KEY3=DA_AES256,/home/dbms_api/key/DA_AES256.scps
```

ServiceID를 입력 할 경우 SG-KMS와 통신을 통해 암호화 정책을 가져와 데이터 암복호화를 수행한다.

SCP_FilePath를 입력 할 경우 KMS와 통신하지 않고 서버의 SCP 파일을 통해 암호화 정책을 가져와 데이터 암복호화를 수행한다.

ServiceID와 SCPS 파일명을 동시에 입력 시, SG-KMS와 네트워크 연결 실패 하면 SCPS 파일을 이용하여 암호화 한다.
ServiceID와 SCPS 파일명 사이에 공백이 있으면 SCPS 파일을 읽을 수 없다. 따라서 예제와 같이 띄어쓰기를 하지 않고 ServiceID와 SCPS 파일 경로를 붙여서 입력한다.

2. 설정 파일의 [Server], [Server2] 항목을 수정한다.

```
[Server]
ServerIP: SG-KMS의 IP를 입력한다.
ServerPort: SG-KMS의 포트번호를 입력한다. 기본값은 2525이다.

//*예제 - 모든 OS 공통
```

```
ServerIP=192.168.22.25
ServerPort=2525
```

```
[Server2]
ServerIP: 이중화를 위한 2번 SG-KMS의 IP를 입력한다.
ServerPort: 이중화를 위한 2번 SG-KMS의 포트번호를 입력한다. 기본값은 2525이다.

//*예제 - 모든 OS 공통
ServerIP=192.168.22.26
ServerPort=2525
```

ServerIP는 최소 1개 ~ 최대 10개까지 등록이 가능하다.

3. 설정 파일의 [AGENT] 항목을 설정한다.

```
[AGENT]
AgentID=SG-KMS 관리도구에서 설정한 D'Amo Agent의 Agent ID
LogDir=로그가 저장될 디렉터리 위치
LogLevel=로그가 남는 수준
SiteCertFilePath=SG-KMS 장비에서 설정한 해당 장비의 사이트 공개키(.cer)
CertFilePath=SG-KMS 관리도구에서 설정한 D'Amo Agent의 공개키(.cer)
KeyFilePath=SG-KMS 관리도구에서 설정한 D'Amo Agent의 비공개키(.key)
SPIN=SG-KMS 관리도구에서 설정한 D'Amo Agent의 SPIN으로, damo-scp_SITE_V3-[Agent이름].spin
파일의 값

//[예제]
AgentID=DA
LogDir=/home/dbms_api/log
LogLevel=4
SiteCertFilePath=/home/dbms_api/key/damo_agt_site.cer
CertFilePath=/home/dbms_api/key/damo_agt.cer
KeyFilePath=/home/dbms_api/key/damo_agt.key
SPIN=XaMh1ylXUh123XUh
```

로그가 남는 수준(LogLevel)에는 아래 5가지 숫자 입력이 가능하며, 각 값의 설정은 다음과 같다.
- 0: 아무 로그도 남기지 않을 경우
- 2: 경고 로그를 파일에 기록
- 4: 에러 로그와 경고 로그를 파일에 기록

- 6: 정보 로그, 에러 로그, 경고 로그를 파일에 기록
- 8: 디버그, 정보 로그, 에러 로그, 경고 로그를 파일에 기록

제품 운영 중 scpdb_agent.ini 파일을 수정하면 CONFIG_REINIT() 함수를 호출해야 변경된 내용이 적용된다.

5.9 CLI에서 권한 설정

$DA_INST_HOME 디렉터리에서 acl_cli 파일을 실행하여 USER 단위로 암/복호화 권한을 설정한다. USER 는 DB의 소유자명이고, KEY는 설정 파일(scpdb_agent.ini)의 [KEYINFO] 중 ALIAS 값(예: KEY1)이다.

CLI 명령어를 자세히 보려면 help 명령어를 실행한다.

```
CLI 실행 방법
$> cd $DA_INST_HOME
$> ./acl_cli  -start
Enter the PIN of CLI-key. : damo_agt.key 의 비밀번호
```

```
권한 추가할 경우
D'Amo > SET PRIV ENC [USER]"[KEY]"1"1
D'Amo > SAVE ALL
D'Amo > SHOW ALL

예제)
D'Amo > SET PRIV ENC SCOTT"KEY1"1"1
D'Amo > SET PRIV ENC SCOTT"KEY2"1"1
```

scpdb_agent.ini 설정 파일이 아래 예제와 같을 경우, CLI에서 권한 설정할 때 입력해야 하는 2번째 [KEY] 인자 값에는 KEY1을 입력해야 한다. (※ARIA256을 입력하는 것이 아님)

#scpdb_agent.ini 설정 파일 예제
[KEYINFO]
KEY1=ARIA256

```
권한 삭제할 경우
D'Amo > DEL PRIV ENC [USER]"[KEY]
D'Amo > SAVE ALL
D'Amo > SHOW ALL
```

CLI 에서 권한을 추가하거나 삭제 한 경우 반드시 SAVE ALL 명령어를 실행하며 SHOW ALL 명령어를 이용하여 적용 여부를 확인 한다.

5.10 DB 서버 사전 확인

DA를 PostgreSQL 환경에서 설치 하기 전에 다음과 같은 사항들을 확인한다.

- DB 서버 재시작
- DB 엔진 설치 계정에 폴더 생성
- DB 사용자로 SCP 계정 생성 (권고사항)
- DB 서버의 DBA 권한 계정
- DB 엔진 설치 계정의 . profile 파일 수정

5.11 .lock 파일 생성

4. 환경변수 $DA_INST_HOME과 동일한 위치에 디렉터리명을 “sql”, “key”, “log” 로 하여 빈 디렉터리를 생성한다. $DA_INST_HOME/log 디렉터리에 touch 명령어로 파일명이 “.lock” 인 빈 파일을 생성한다.

```
$> cd $DA_INST_HOME
$DA_INST_HOME> mkdir key
$DA_INST_HOME> mkdir log
$DA_INST_HOME> mkdir sql
$DA_INST_HOME> cd log
$DA_INST_HOME/log> touch .lock
```

5.12 라이브러리 설정

1. [PostgreSQL 설치 경로]/lib 경로에 $DA_INST_HOME에 있는 libdamoscpdb.{so|a|sl} 파일의 심볼릭 링크를 걸어준다. 여기서 PostgreSQL 설치 경로는 /usr/local/pgsql/로 가정하고, 환경은 Linux로 가정한다.

```
$> cd /usr/local/pgsql/lib
$> ln  -s $DA_INST_HOME/libcis_cc-3.3.so libcis_cc-3.3.so
$> ln  -s $DA_INST_HOME/libcis_ce-3.3.so libcis_ce-3.3.so
$> ln  -s $DA_INST_HOME/libdamocm-4.0.so libdamocm-4.0.so
$> ln  -s $DA_INST_HOME/liblogw-0.2.so liblogw-0.2.so
$> ln  -s $DA_INST_HOME/libdamoscpdb.so libdamoscpdb.so
```

라이브러리 경로 권한이 root라서 심볼링 링크를 root 계정으로 걸어준다.

2. ldconfig를 통해 새로 등록된 라이브러리를 활성화시켜준다.

   a. /etc/ld.so.conf 파일에 /usr/local/pgsql/lib 경로를 추가한다.
   b. ldconfig명령어를 실행한다.

5.13 sql 파일 생성

1. $DA_INST_HOME/sql 디렉터리에서 install_make.sh을 이용하여 설치할 sql 파일을 생성한다. D_INI는 설정 파일(scpdb_agent.ini)의 경로다.

```
$> cd $DA_INST_HOME/sql
$> ./install_make.sh D_INI

예) D_INI 경로를 /home/dbms_api로 가정한다.
$> ./install_make.sh /home/dbms_api
D_INI PATH is replaced by /home/dbms_api
```

2. 다음 2개의 파일이 생성되는지 확인한다.

- 001.inner_function.post.sql
- 002.user_interface.post.sql

5.14 제품 함수 설치

- $DA_INST_HOME/sql 위치에서 DB 접속 후 설치를 원하는 DB계정에 함수를 설치한다. SCP 계정을 생성한 경우 SCP 계정에 설치한다.

```
$>psql  -f 001.inner_function.post.sql postgres
$>psql  -f 002.user_interface.post.sql postgres
$>psql  -f 003.grant_execute_functions.post postgres
```

PostgreSQL 버전이 10이면 001.inner_function.post10.sql 파일을 실행한다.

5.15 제품 설치 확인

DB 서버에서 암/복호화 함수를 호출하여 설치를 성공했는지 확인한다.

```
postgres=# SELECT ENC_STR( 'DAMO', 'abc');
ENC_STR('DAMO','abc')
--------------------------------------------------
5E41ACD673653158D7AE8C30CDA9627D3556E173

postgres=# SELECT DEC_STR( 'DAMO', ENC_STR( 'DAMO', 'abc'));
DEC_STR( 'DAMO', ENC_STR( 'DAMO', 'abc'))
--------------------------------------------------
abc
```

5.16 제품 운용

5.16.1 함수 설명

DA에서 제공되는 함수와 사용하는 방법에 대해서 설명한다.

5.16.1.1 파라미터 설명

5.16.1.1.1 I_KEY

I_KEY : 암호화/복호화 때 사용하는 암호화 키
- 설정 파일(scpdb_agent.ini)의 [KEYINFO] 중 ALIAS 값
- 아래 [설정 파일 예제]의 경우 ALIAS는 'KEY1'과 'KEY2'이다.
- ALIAS는 SCPS파일로 암호화 할 것인지, SG-KMS의 서비스 ID로 암호화 할 것인지 설정 가능하다.

[설정 파일 예제]
- 설정 파일(scpdb_agent.ini)의 [KEYINFO] 중 ALIAS 값
KEY1=AES256.SCPS
KEY2=ARIA256

[함수 사용 예제]
SELECT ENC_B64('KEY1', 'abc') FROM DUAL;

5.16.1.1.2 I_DATA

I_DATA : 평문(암호화 함수일 경우), 암호문(복호화 함수일 경우)

DA에서 제공하는 암호화 함수에서 성능 향상을 위해 라이브러리에서 정책 이름(KEYINFO ALIAS)을 바탕으로 Cash하여 동작한다. 정책 이름은 같은데 실제 키(대칭 키)가 다른 정책을 사용하면 Cash에서 이전 키로 복호화를 시도하여 오류가 발생한다. 이 상황을 해결하기 위해서는 데이터베이스 서버 재시작이 필요하다.

표 5-8 DA 함수 (PostgreSQL)

함수 명	입력		출력
ENC_STR	I_KEY	IN 문자열,	Hex String 암호문
	I_DATA	IN 문자열 (평문)	
ENC_B64	I_KEY	IN 문자열,	Base64 Encording

				암호문
	I_DATA	IN 문자열 (평문)		
DEC_STR	I_KEY	IN 문자열,		평문
	I_DATA	IN 문자열 (Hex String 암호문)		
DEC_B64	I_KEY	IN 문자열,		평문
	I_DATA	IN 문자열 (base64 String 암호문)		
INDEX_STR	I_KEY	IN 문자열,		Hex String 암호문
	I_DATA	IN 문자열 (평문),		
	I_TYPE	IN 문자열 '' or 'IX '(Plug-IN 연동 시 사용)		
DEC_INDEX_STR	I_KEY	IN 문자열,		Hex String 암호문 입력받아 OPE 데이터
	I_DATA	IN 문자열 (암호문),		
	I_TYPE	IN 문자열 '' or 'IX '(Plug-IN 연동 시 사용)		
DEC_INDEX_B64	I_KEY	IN 문자열,		Base64 Encording 암호문 입력받아 OPE 데이터
	I_DATA	IN 문자열 (암호문),		
	I_TYPE	IN 문자열 '' or 'IX '(Plug-IN 연동 시 사용)		
HASH_STR	I_ALOG	IN 숫자,	SHA1 =70	Hex String 해쉬 암호문
			SHA256 =71	
			SHA384 =72	
			SHA512 =73	
			HAS160 =74	
	I_DATA	IN 문자열		
HASH_B64	I_ALOG	IN 숫자,	SHA1 =70	Base64 String 해쉬 암호문
			SHA256 =71	
			SHA384 =72	
			SHA512 =73	
			HAS160 =74	
	I_DATA	IN 문자열		
HEXTOB64	I_DATA	IN 문자열 (Hex String 암호문)		base64 Encording 암호문
B64TOHEX	I_DATA	IN 문자열		Hex String 암호문

		(base64 Encording 암호문)	
CONFIG_REINIT			성공시 'SUCCESS', 그 외 에러

5.16.2 함수 호출 예제

1. ENC_STR

```
postgres=# SELECT ENC_STR('KEY1', 'abc');
```

2. ENC_B64

```
postgres=# SELECT ENC_B64('KEY1', 'abc');
```

3. DEC_STR

```
postgres=# SELECT DEC_STR('KEY1', ENC_STR('KEY1', 'abc'));
```

4. DEC_B64

```
postgres=# SELECT DEC_B64('KEY1', ENC_B64('KEY1', 'abc'));
```

5. INDEX_STR

```
DP 제품에 연동 하지 않을 경우
postgres=# SELECT INDEX_STR('KEY1', 'abc', '');

DP 제품에 연동 할 경우
postgres=# SELECT INDEX_STR('KEY1', 'abc', 'IX');
```

6. DEC_INDEX_STR, DEC_INDEX_B64

```
DP 제품에 연동 하지 않을 경우
postgres=# SELECT DEC_INDEX_STR('KEY1', ENC_STR('KEY1', 'abc'), '');
postgres=# SELECT DEC_INDEX_B64('KEY1', ENC_B64('KEY1', 'abc'), '');

DP 제품에 연동 할 경우
postgres=# SELECT DEC_INDEX_STR('KEY1', ENC_STR('KEY1', 'abc'), 'IX');
postgres=# SELECT DEC_INDEX_B64('KEY1', ENC_B64('KEY1', 'abc'), 'IX');
```

7. HASH_STR

```
postgres=# SELECT HASH_STR( 71, 'abc' );
```

8. HASH_B64

```
postgres=# SELECT HASH_B64( 71, 'abc' );
```

9. HEXTOB64

```
postgres=# SELECT HEXTOB64('A305378D8F974F1C1537ED7CB0CB959245D1AC31');
```

10. B64TOHEX

```
postgres=# SELECT B64TOHEX('owU3jY+XTxwVN+18sMuVkkXRrDE=');
```

11. CONFIG_REINIT

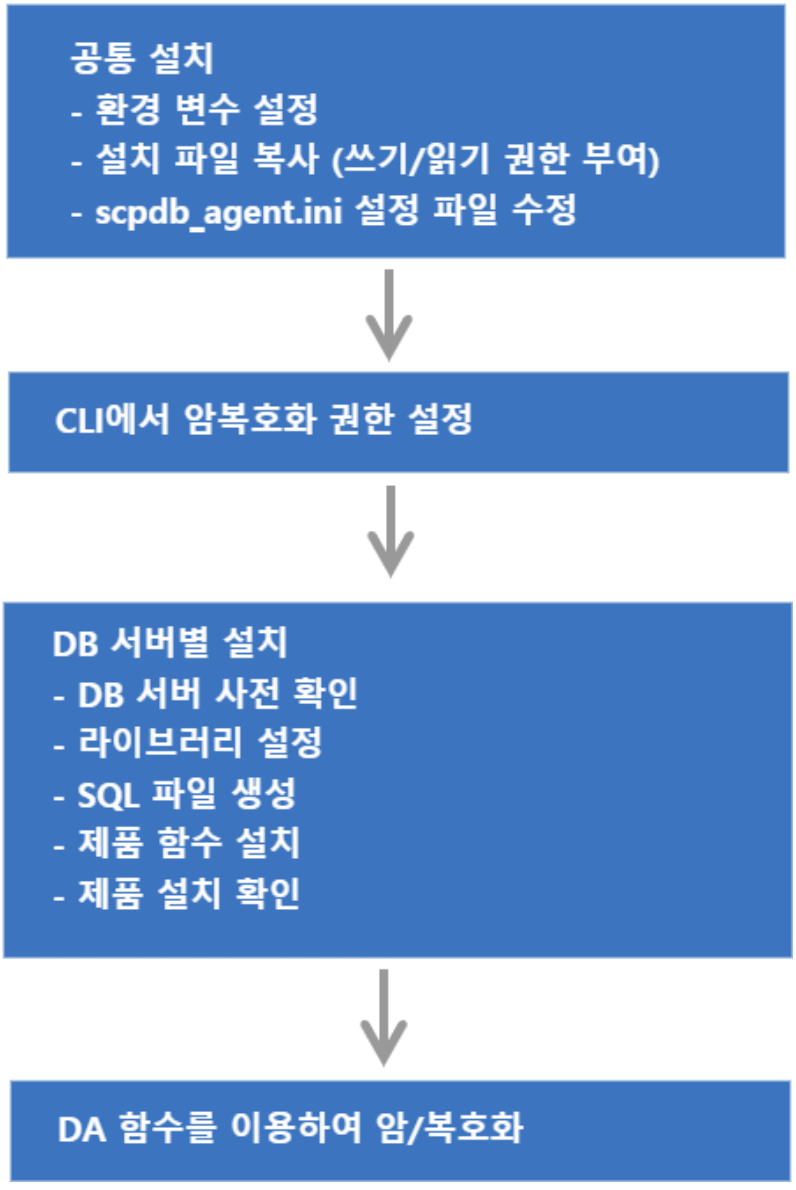
```
postgres=# SELECT CONFIG_REINIT();
```

6.

DB2

DA는 DB 서버의 종류에 따라서 설치 및 운용 방법이 각각 다르다.

그림 6-1 설치 개념도



6.1 지원 운영체제 및 DB서버

DA에서 지원하는 운영체제 및 DB서버는 아래와 같다. 단, 특정 환경은 지원되지 않을 수도 있으므로, 제품 설치 전에 상세한 지원 가능 여부는 펜타시큐리티시스템으로 문의한다.

표 6-1 제품이 지원하는 운영체제 및 DB 서버 정보

구분	설명
운영체제	Windows, AIX, HP IA, HP PA-RISC, Linux, SUN, TRU64
DB 서버	ORACLE 8i ~, SQL Server 2012 ~, DB2 9.x ~, Tibero 4 SP 1 ~, MySQL 5.x ~, MariaDB 5.5, 10.0, 10.1, Cache DB 2009.1 ~, Informix IDS 9.x ~ Sybase ASE 15.7 SP61 ~, Sybase IQ 15.4 ~, CUBRID 2008 R1.3 ~ , PostgreSQL 9.4 ~

6.2 설치 파일의 구성 확인

DA의 설치 파일은 아래와 같은 명칭으로 압축 파일(zip) 형태로 제공된다.

- DA 설치파일: Install_DAmo_DA_v{버전}.zip
  - 압축을 해제 한 뒤, 설치 대상 DB 서버, OS 및 bit에 맞는 설치 파일을 준비한다.

설치 파일에 제품을 사용할 수 있는 '라이선스'는 포함되어 있지 않다.
펜타시큐리티시스템에 문의하여 '라이선스' 파일은 별도로 준비한다.

표 6-2 설치 파일(Install_DAmo_DA_v{버전}.zip)을 압축 해제 시, 디렉터리의 구성

구성	설명
_SampleScpsFiles	SG-KMS 연동 없이 암복호화를 테스트할 수 있는 테스트 키 파일
_TestAgentKeyPair	CLI에서 사용할 수 있는 테스트 키 쌍
Altibase	DA-ALT 제품의 설치 바이너리 폴더
Cache	DA-CDB 제품의 설치 바이너리 폴더
Cubrid	DA-CUB 제품의 설치 바이너리 폴더
DB2	DA-DB2 제품의 설치 바이너리 폴더
Informix	DA-IFX 제품의 설치 바이너리 폴더
MySQL	DA-MYQ 제품의 설치 바이너리 폴더
Oracle	DA-ORA 제품의 설치 바이너리 폴더
Postgres	DA-PGS 제품의 설치 바이너리 폴더
SQL Server	DA-MSQ 제품의 설치 바이너리 폴더

구성	설명
SybaseASE	DA-SYB 제품의 설치 바이너리 폴더
SybaseIQ	DA-SIQ 제품의 설치 바이너리 폴더
Tibero	DA-TIB 제품의 설치 바이너리 폴더

6.2.1 DB 서버 및 운영체제 별 SQL파일 구성

각 DB 서버 및 운영체제 별 SQL파일 구성은 다음과 같다. 다음 장에서 각 DB별로 SQL파일 설치 방법을 설명한다.

표 6-3 DB 서버 및 운영체제 별 SQL파일

DB 서버 종류	Linux 일 경우	Windows 일 경우
Oracle	000.da_user.pkg 000.da_user.sql 001.inner_function.ora(c 버전 설치 시) 002.user_interface.ora(JAVA 버전 설치 시) 002.user_interface_java.ora 003.grant_execute_functions.sql(JAVA 버전 설치 시) 005.securej_privilege.sql install_make.sh	000.da_user.pkg 000.da_user.sql 001.inner_function.ora(c 버전 설치 시) 002.user_interface.ora(JAVA 버전 설치 시) 002.user_interface_java.ora 003.grant_execute_functions.sql(JAVA 버전 설치 시) 005.securej_privilege.sql 009.da_test.sql
MYSQL	001.inner_function.mys 002.user_interface. mys 003.grant_execute_functions.sql 009.da_test.sql install_make.sh	001.inner_function.mys 002.user_interface.mys 003.grant_execute_functions.sql
TIBERO	001.inner_function.tbs(c 버전 설치 시) 001.inner_function_java.tbs(JAVA 버전 설치 시) 002.user_interface.tbs(c 버전 설치 시) 002.user_interface_java.tbs(JAVA 버전 설치 시) 003.grant_execute_functions.sql install_make.sh	001.inner_function_java.tbs 002.user_interface_java.tbs 003.grant_execute_functions.sql
INFORMIX	001.inner_function.ifx 002.user_interface. ifx 003.grant_execute_functions.sql 009.da_test.sql install_make.sh	해당 없음
POSTGRESQL	001.inner_function.post 002.user_interface.post 003.grant_execute_functions.post	해당 없음

DB 서버 종류	Linux 일 경우	Windows 일 경우
DB2	001.inner_function.db2 002.user_interface.db2 003.grant_execute_functions.sql install_make.sh	001.inner_function.db2 002.user_interface.db2 003.grant_execute_functions.sql
CUBRID	001.inner_function.sql 002.user_interface.sql	001.inner_function.sql 002.user_interface.sql
SYBASE	001.inner_function.sybase 002.user_interface.sybase 003.grant_execute_functions.sql install_make.sh	해당 없음
SYBASE IQ	001.inner_function.sybiq 002.user_interface. sybiq 003.grant_execute_functions.sql install_make.sh	해당 없음
SQL Server	해당 없음	001.inner_function.sql 002.user_interface. sql 003.grant_execute_functions.sql install_make.bat
Cache DB	SCP.xml install_make.sh	SCP.xml
Altibase	해당 없음	000.da_user.pkg 000.da_user.sql 001.inner_function.sql 002.user_interface. sql install_make.bat

6.2.2 공통 설치 파일 구성

환경에 관계 없이 공통적으로 사용하는 설치 파일은 아래와 같다.

표 6-4 공통 설치 파일 (sql파일 제외)

파일 분류	파일 명	파일 용도
Library 파일	libdamoscpdb.{so|a|sl|dll}	DA 메인 라이브러리. 주로 DBMS External Interface를 담당
	libdamocm-4.0.{so|a|sl|dll}	공통 모듈 라이브러리
	liblogw-0.2.{so|a|sl|dll}	로그를 기록하는 라이브러리
	libcis_cc-3.3.{so|a|sl|dll}	암호화, 복호화 기능을 제공하는 라이브러리
	libcis_ce-3.3.{so|a|sl|dll}	암호화, 복호화를 제외한 부가적인 기능을 제공하는 라이브러리(예: Base64, 인증서 관리, 특성 유지 암호화 등)

파일 분류	파일 명	파일 용도
설정 파일	scpdb_agent.ini	DA 구동시 실행에 필요한 설정정보를 참조
License 파일	damo_lic.cer	DA 구동 시 제품의 유효성을 검증하는데 사용
Agent key 파일	damo_agt_site.cer	SG-KMS 연동, CLI 프로그램에서 사용하는 인증서 쌍
	damo_agt.cer	
	damo_agt.key	
접근제어 파일	acl_cli 파일	DB 의 USER 별로 암·복호 권한을 설정하는데에 사용
	privilege.damo	권한 파일
JAVA class 파일 (Oracle, Tibero, Cubrid 설치 가능)	ScpAgentException.class	예외 처리 Class
	ScpCryptData.class	암호화 복호화 Class
SQL 파일	아래 새로운 표에 DB별로 표기함	

Agent Key 파일은 SG-KMS 연동에 필요한 키 발급를 참고하여 발급 받는다.

DA-PGS(PostgreSQL)의 경우, DB 서버 버전에 따라 libdamoscpdb.so 라이브러리 선택

- libdamoscpdb94.so (Postgres 9.4)
- libdamoscpdb95.so (Postgres 9.5)
- libdamoscpdb95AS.so (EDB Postgres 9.5)
- libdamoscpdb96AS.so (EDB Postgres 9.6)
- libdamoscpdb10.so (Postgres 10)

6.3 환경변수 설정

DA를 설치할 운영체제에 환경변수 DA_INST_HOME를 설정한다. 이 매뉴얼에서는 제품 설치 경로를 아래와 같이 가정하여 설명한다.

- Linux 환경일 경우: /home/dbms_api
- Windows 환경일 경우: E:₩dbms_api

주의) DA_INST_HOME 설정 시, 주의 사항

- 리눅스의 경우 "/root" 디렉토리로 설정을 권장하지 않는다.
- 윈도우의 경우 "바탕화면"으로 설정을 권장하지 않는다.

위의 경로로 설정할 경우 접근 권한 등의 이유로 문제가 발생할 가능성이 존재한다.

6.3.1 환경변수 설정 - Linux 환경일 경우

DA_INST_HOME 환경변수에 DA의 설치 디렉터리를 설정한다.

```
.profile을 사용하는 경우
export DA_INST_HOME=/home/dbms_api


.cshrc를 사용하는 경우
setenv DA_INST_HOME=/home/dbms_api

export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$DA_INST_HOME
```

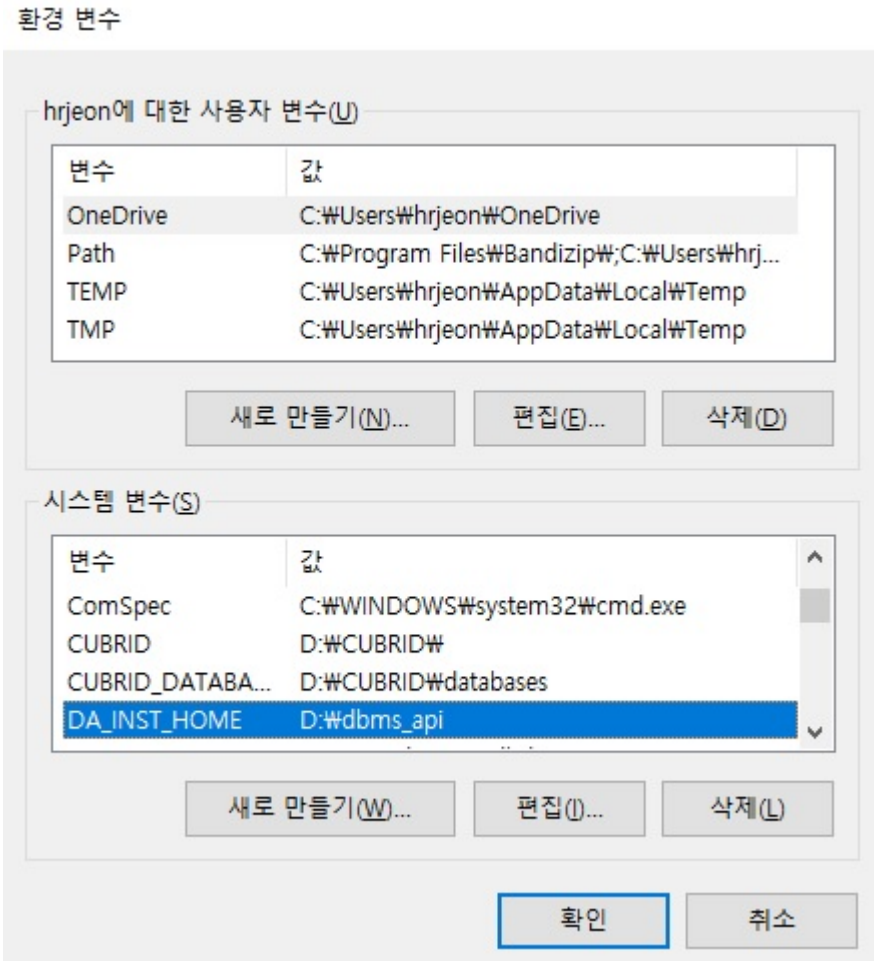
표 6-5 운영 체제별 라이브러리 PATH 명칭

운영 체제	라이브러리 PATH 명칭
HP_UX	SHLIB_PATH
AIX	LIBPATH
LINUX, SUN	LD_LIBRARY_PATH

6.3.2 환경변수 설정 - Windows 환경일 경우

[탐색기->내컴퓨터->등록정보->고급 탭->환경변수] 를 선택하여 나타나는 환경변수 설정 다이얼로그에서 시스템변수 DA_INST_HOME 변수와 값을 추가한다.

그림 6-2 DA_INST_HOME



6.4 DA의 설치 파일 복사

$DA_INST_HOME 디렉터리에 위 [설치 파일의 구성 확인]에서 나열된 파일들을 복사한다.

6.5 라이선스 파일 설정

$DA_INST_HOME 디렉토리에 라이선스 파일(damo_lic.cer)을 복사한다.

라이선스 파일명이 damo_lic.cer가 아닌 다른 이름으로 저장되어 있다면 변경해야 한다.

6.6 설치 파일의 접근 권한 부여 (Linux 설치 시)

Linux 사용자 계정에 DA 설치 파일(라이브러리, 실행 파일, 디렉토리)의 접근 권한을 부여한다. Windows에서 제품을 설치 할 경우, 이 과정은 생략한다.

```
$> cd $DA_INST_HOME
$> chmod 755 lib* acl_cli sql/install_make.sh
```

6.7 (SG-KMS 연동 시) SG-KMS에서 DA 정보 등록 및 연동에 필요한 키 내보내기

DA는 데이터를 암·복호화하기 위해 '암호화 키'가 필요하다. '암호화 키'를 얻기 위해서는 SG-KMS를 연동하거나 SCPS라는 암호화 키 파일을 이용할 수 있는데,
다음 설명은 SG-KMS 연동을 위해 SG-KMS에서 DA 정보를 등록하고 연동에 필요한 키를 내보내는 방법에 대해서 설명한다.

6.7.1 사전 준비

6.7.1.1 지원 SG-KMS 버전

DA와 연동 가능한 SG-KMS 버전은 다음과 같다.

표 6-6 연동 가능한 SG-KMS 버전

SG-KMS Major version	SG-KMS Minor version
SG-KMS v3.0	v3.0.9.0 이상 연동 가능
SG-KMS v4.0	v4.0.104.5 이상 연동 가능

SG-KMS v2.3 연동은 미지원한다.

6.7.1.2 연동 전 점검 항목

SG-KMS 연동을 위해서는 다음과 같이 사전 준비가 필요하다.

- SG-KMS 관리도구
- SG-KMS 관리도구에 접속할 수 있는 ID와 비밀번호
- SG-KMS 매뉴얼

이 매뉴얼에서는 SG-KMS의 키 종류 및 SG-KMS 관리도구 사용 방법에 대한 내용은 다루지 않으므로 **[SG-KMS 매뉴얼]**을 참고한다.

6.7.1.3 SG-KMS 연동에 필요한 키 발급

DA와 SG-KMS의 연동을 위해서는 먼저 SG-KMS에 DA를 Agent로 등록해야한다. 등록하는 절차는 SG-KMS 사용자설명서의 '대칭키를 사용하는 D'Amo Agent 등록 안내서'를 참고한다.

[D'Amo Agent 키] 파일, [Agent ID(CN)]와 [서비스 ID]정보는 DA와 SG-KMS 연동을 위한 설정 과정에서 필요하다. SG-KMS 관리자는 DA 설치 엔지니어에게 안전하게 전달한다.

SG-KMS 관리자에게 받은 D'Amo Agent의 인증서 및 키 파일 명을 다음과 같이 변경후 $DA_INST_HOME/key 디렉터리에 복사한다.

표 6-7 SG-KMS 연동에 필요한 키 목록

키 구분	생성된 키 파일명	변경할 키 파일명
사이트 키	damo-site_{발행기관/부서명}-SITE_V3.cer	damo_agt_site.cer
Agent 키	damo-scp_SITE_V3-{Agent 이름}.cer	damo_agt.cer
	damo-scp_SITE_V3-{Agent 이름}.key	damo_agt.key
	damo-scp_SITE_V3-{Agent 이름}.spin	damo_agt.spin

Agent 키 경로 지정, 키 이름 변경은 반드시 필요한 작업은 아니지만 관리를 위해 변경하는 것을 권장한다.

6.8 설정 파일(scpdb_agent.ini) 수정

DA의 운용을 위해 사용하는 scpdb_agent.ini 설정 파일 수정 방법에 대해 설명한다. DA는 SG-KMS를 이용하거나 SCPS 파일을 이용하여 암호화 키를 얻기 때문에 고객의 환경에 맞게 설정해야 한다.

$DA_INST_HOME 디렉터리에 있는 scpdb_agent.ini 설정 파일에서 아래 항목의 값을 수정한다.

1. 설정 파일의 [KEYINFO] 항목 - [KEY1]에 암호화 키 정보를 입력한다.

```
[KEYINFO]
KEY1=암복호화 하려는 암호화 키 정보를 입력한다.
//키 정보는 다음과 같은 값을 입력할 수 있다.
///ServiceID: SG-KMS에 생성한 서비스 ID를 입력한다.
///SCP_FilePath: SG-KMS에서 서비스 내보내기를 통해 발급한 SCPS 파일의 절대 경로 및 파일명, 확장자를 입력한다.

//*예제 - Windows 경우
KEY1=DA_AES256
KEY2=C:\DA\Policy\S_AES128.SCPS\DA_AES256.scps
KEY3=DA_AES256,C:\DA\Policy\S_AES128.SCPS\DA_AES256.scps

//*예제 - Linux 또는 Unix 경우
KEY1=DA_AES256
KEY2=/home/dbms_api/key/DA_AES256.scps
KEY3=DA_AES256,/home/dbms_api/key/DA_AES256.scps
```

ServiceID를 입력 할 경우 SG-KMS와 통신을 통해 암호화 정책을 가져와 데이터 암복호화를 수행한다.

SCP_FilePath를 입력 할 경우 KMS와 통신하지 않고 서버의 SCP 파일을 통해 암호화 정책을 가져와 데이터 암복호화를 수행한다.

ServiceID와 SCPS 파일명을 동시에 입력 시, SG-KMS와 네트워크 연결 실패 하면 SCPS 파일을 이용하여 암호화 한다.
ServiceID와 SCPS 파일명 사이에 공백이 있으면 SCPS 파일을 읽을 수 없다. 따라서 예제와 같이 띄어쓰기를 하지 않고 ServiceID와 SCPS 파일 경로를 붙여서 입력한다.

2. 설정 파일의 [Server], [Server2] 항목을 수정한다.

```
[Server]
ServerIP: SG-KMS의 IP를 입력한다.
ServerPort: SG-KMS의 포트번호를 입력한다. 기본값은 2525이다.

//*예제 - 모든 OS 공통
```

```
ServerIP=192.168.22.25
ServerPort=2525
```

```
[Server2]
ServerIP: 이중화를 위한 2번 SG-KMS의 IP를 입력한다.
ServerPort: 이중화를 위한 2번 SG-KMS의 포트번호를 입력한다. 기본값은 2525이다.

//*예제 - 모든 OS 공통
ServerIP=192.168.22.26
ServerPort=2525
```

ServerIP는 최소 1개 ~ 최대 10개까지 등록이 가능하다.

3. 설정 파일의 [AGENT] 항목을 설정한다.

```
[AGENT]
AgentID=SG-KMS 관리도구에서 설정한 D'Amo Agent의 Agent ID
LogDir=로그가 저장될 디렉터리 위치
LogLevel=로그가 남는 수준
SiteCertFilePath=SG-KMS 장비에서 설정한 해당 장비의 사이트 공개키(.cer)
CertFilePath=SG-KMS 관리도구에서 설정한 D'Amo Agent의 공개키(.cer)
KeyFilePath=SG-KMS 관리도구에서 설정한 D'Amo Agent의 비공개키(.key)
SPIN=SG-KMS 관리도구에서 설정한 D'Amo Agent의 SPIN으로, damo-scp_SITE_V3-[Agent이름].spin
파일의 값

//[예제]
AgentID=DA
LogDir=/home/dbms_api/log
LogLevel=4
SiteCertFilePath=/home/dbms_api/key/damo_agt_site.cer
CertFilePath=/home/dbms_api/key/damo_agt.cer
KeyFilePath=/home/dbms_api/key/damo_agt.key
SPIN=XaMh1ylXUh123XUh
```

로그가 남는 수준(LogLevel)에는 아래 5가지 숫자 입력이 가능하며, 각 값의 설정은 다음과 같다.
- 0: 아무 로그도 남기지 않을 경우
- 2: 경고 로그를 파일에 기록
- 4: 에러 로그와 경고 로그를 파일에 기록

- 6: 정보 로그, 에러 로그, 경고 로그를 파일에 기록
- 8: 디버그, 정보 로그, 에러 로그, 경고 로그를 파일에 기록

제품 운영 중 scpdb_agent.ini 파일을 수정하면 CONFIG_REINIT() 함수를 호출해야 변경된 내용이 적용된다.

6.9 CLI에서 권한 설정

$DA_INST_HOME 디렉터리에서 acl_cli 파일을 실행하여 USER 단위로 암/복호화 권한을 설정한다. USER 는 DB의 소유자명이고, KEY는 설정 파일(scpdb_agent.ini)의 [KEYINFO] 중 ALIAS 값(예: KEY1)이다.

CLI 명령어를 자세히 보려면 help 명령어를 실행한다.

```
CLI 실행 방법
$> cd $DA_INST_HOME
$> ./acl_cli  -start
Enter the PIN of CLI-key. : damo_agt.key 의 비밀번호
```

```
권한 추가할 경우
D'Amo > SET PRIV ENC [USER]"[KEY]"1"1
D'Amo > SAVE ALL
D'Amo > SHOW ALL

예제)
D'Amo > SET PRIV ENC SCOTT"KEY1"1"1
D'Amo > SET PRIV ENC SCOTT"KEY2"1"1
```

scpdb_agent.ini 설정 파일이 아래 예제와 같을 경우, CLI에서 권한 설정할 때 입력해야 하는 2번째 [KEY] 인자 값에는 KEY1을 입력해야 한다. (※ARIA256을 입력하는 것이 아님)

#scpdb_agent.ini 설정 파일 예제
[KEYINFO]
KEY1=ARIA256

```
권한 삭제할 경우
D'Amo > DEL PRIV ENC [USER]"[KEY]
D'Amo > SAVE ALL
D'Amo > SHOW ALL
```

CLI 에서 권한을 추가하거나 삭제 한 경우 반드시 SAVE ALL 명령어를 실행하며 SHOW ALL 명령어를 이용하여 적용 여부를 확인 한다.

6.10 DB 서버 사전 확인

DA를 DB2 환경에서 설치 하기 전에 다음과 같은 사항들을 확인한다.

- DB 엔진 설치 계정에 폴더 생성
- DB 사용자로 SCP 계정 생성 (권고사항)
- DB 서버의 DBA 권한 계정
- DB 엔진 설치 계정의 . profile 파일 수정 (Linux 설치 시)

6.11 라이브러리 설정

{DB설치폴더}/sqllib/lib64, {DB설치폴더}/sqllib/function 디렉터리에 library 파일의 symbolic link 파일을 생성한다.

```
$> cd {DB설치폴더}/sqllib/lib64
$> ln  -s $DA_INST_HOME/libcis_cc-3.3.{so|a|sl} libcis_cc-3.3.{so|a|sl}
$> ln  -s $DA_INST_HOME/libcis_ce-3.3.{so|a|sl} libcis_ce-3.3.{so|a|sl}
$> ln  -s $DA_INST_HOME/liblogw-0.2.{so|a|sl} liblogw-0.2.{so|a|sl}
$> ln  -s $DA_INST_HOME/libdamocm-4.0.{so|a|sl} libdamocm-4.0.{so|a|sl}
$> cd {DB설치폴더}/sqllib/function
$> ln  -s $DA_INST_HOME/libdamoscpdb.{so|a|sl} libdamoscpdb.so
```

6.12 sql 파일 생성

1. $DA_INST_HOME/sql 디렉터리에서 install_make.sh을 이용하여 설치할 sql 파일을 생성한다. D_INI 는 설정 파일(scpdb_agent.ini)의 경로다.

```
$> cd $DA_INST_HOME/sql
$> ./install_make.sh D_INI

예) D_INI 경로를 /home/dbms_api로 가정한다.
$> ./install_make.sh /home/dbms_api
D_INI_PATH is replaced by /home/dbms_api
```

```
[Windows 환경일 경우 - cmd창에서]
cmd> cd $DA_INST_HOME/sql
cmd> install_make.bat D_INI

[예제]
cmd> install_make.bat C:\da
D_INI_PATH is replaced by C:\da
```

2. 아래 파일 2개가 생성되었는지 확인한다.

- 001.inner_function.db2.sql
- 002.user_interface.db2.sql

6.13 제품 함수 설치

1. $DA_INST_HOME/sql 위치에서 DB 접속 후 설치를 원하는 DB계정에 함수를 설치한다.

```
$> db2 connect to sample
$> db2 -td@ -vf 001.inner_function.db2.sql
$> db2 -td@ -vf 002.user_interface.db2.sql
```

2. 특정 DB 사용자에게 함수 실행 권한을 부여한다.

```
$> db2 connect to sample
$> db2 -td@ -vf 003.grant_execute_functions.sql
```

6.14 제품 설치 확인

DB 서버에서 암/복호화 함수를 호출하여 설치를 성공했는지 확인한다.

```
DB2 => SELECT ENC_STR('DAMO', 'abc') FROM sysibm.sysdummy1
ENC_STR('DAMO','abc')
--------------------------------------------------
5E41ACD673653158D7AE8C30CDA9627D3556E173

DB2 => SELECT DEC_STR( 'DAMO', ENC_STR( 'DAMO', 'abc')) FROM sysibm.sysdummy1
DEC_STR( 'DAMO', ENC_STR( 'DAMO', 'abc'))
--------------------------------------------------
abc
```

6.15 제품 운용

6.15.1 함수 설명

DA에서 제공되는 함수와 사용하는 방법에 대해서 설명한다.

6.15.1.1 파라미터 설명

6.15.1.1.1 I_KEY

I_KEY : 암호화/복호화 때 사용하는 암호화 키
- 설정 파일(scpdb_agent.ini)의 [KEYINFO] 중 ALIAS 값
- 아래 [설정 파일 예제]의 경우 ALIAS는 'KEY1'과 'KEY2'이다.
- ALIAS는 SCPS파일로 암호화 할 것인지, SG-KMS의 서비스 ID로 암호화 할 것인지 설정 가능하다.

[설정 파일 예제]
- 설정 파일(scpdb_agent.ini)의 [KEYINFO] 중 ALIAS 값
KEY1=AES256.SCPS
KEY2=ARIA256

[함수 사용 예제]
SELECT ENC_B64('KEY1', 'abc') FROM DUAL;

6.15.1.1.2 I_DATA

I_DATA : 평문(암호화 함수일 경우), 암호문(복호화 함수일 경우)

DA에서 제공하는 암호화 함수에서 성능 향상을 위해 라이브러리에서 정책 이름(KEYINFO ALIAS)을 바탕으로 Cash하여 동작한다. 정책 이름은 같은데 실제 키(대칭 키)가 다른 정책을 사용하면 Cash에서 이전 키로 복호화를 시도하여 오류가 발생한다. 이 상황을 해결하기 위해서는 데이터베이스 서버 재시작이 필요하다.

표 6-8 DA 함수 (DB2)

함수 명	입력		출력
ENC_STR	I_KEY	IN 문자열,	Hex String 암호문
	I_DATA	IN 문자열 (평문)	
ENC_B64	I_KEY	IN 문자열,	Base64 Encording 암호문
	I_DATA	IN 문자열 (평문)	
DEC_STR	I_KEY	IN 문자열,	평문
	I_DATA	IN 문자열 (Hex String 암호문)	
DEC_B64	I_KEY	IN 문자열,	평문
	I_DATA	IN 문자열 (base64 String 암호문)	
INDEX_STR	I_KEY	IN 문자열,	Hex String 암호문
	I_DATA	IN 문자열 (평문),	
	I_TYPE	IN 문자열 '' or 'IX '(Plug-IN 연동 시 사용)	

DEC_INDEX_STR	I_KEY	IN 문자열,		Hex String 암호문 입력받아 OPE 데이터
	I_DATA	IN 문자열 (암호문),		
	I_TYPE	IN 문자열 '' or 'IX '(Plug-IN 연동 시 사용)		
DEC_INDEX_B64	I_KEY	IN 문자열,		Base64 Encording 암호문 입력받아 OPE 데이터
	I_DATA	IN 문자열 (암호문),		
	I_TYPE	IN 문자열 '' or 'IX '(Plug-IN 연동 시 사용)		
HASH_STR	I_ALOG	IN 숫자,	SHA1 =70	Hex String 해쉬 암호문
			SHA256 =71	
			SHA384 =72	
			SHA512 =73	
			HAS160 =74	
	I_DATA	IN 문자열		
HASH_B64	I_ALOG	IN 숫자,	SHA1 =70	Base64 String 해쉬 암호문
			SHA256 =71	
			SHA384 =72	
			SHA512 =73	
			HAS160 =74	
	I_DATA	IN 문자열		
HEXTOB64	I_DATA	IN 문자열 (Hex String 암호문)		base64 Encording 암호문
B64TOHEX	I_DATA	IN 문자열 (base64 Encording 암호문)		Hex String 암호문
CONFIG_REINIT				성공시 'SUCCESS', 그 외 에러

6.15.2 함수 호출 예제

1. ENC_STR

```
DB2 => SELECT ENC_STR('KEY1', 'abc') FROM sysibm.sysdummy1;
```

2. ENC_B64

```
DB2 => SELECT ENC_B64('KEY1', 'abc') FROM sysibm.sysdummy1;
```

3. DEC_STR

```
DB2 => SELECT DEC_STR('KEY1', ENC_STR('KEY1', 'abc')) FROM sysibm.sysdummy1;
```

4. DEC_B64

```
DB2 => SELECT DEC_B64('KEY1', ENC_B64('KEY1', 'abc')) FROM sysibm.sysdummy1;
```

5. INDEX_STR, INDEX_STR_LIKE

```
DP 제품에 연동 하지 않을 경우
DB2 => SELECT INDEX_STR('KEY1', 'abc', '') FROM sysibm.sysdummy1;

DP 제품에 연동 할 경우
DB2 => SELECT INDEX_STR('KEY1', 'abc', 'IX') FROM sysibm.sysdummy1;
```

6. DEC_INDEX_STR, DEC_INDEX_STR_LIKE, DEC_INDEX_B64, DEC_INDEX_B64_LIKE

```
DP 제품에 연동 하지 않을 경우
DB2 => SELECT DEC_INDEX_STR('KEY1', ENC_STR('KEY1', 'abc'), '') FROM sysibm.sysdummy1;
DB2 => SELECT DEC_INDEX_B64('KEY1', ENC_B64('KEY1', 'abc'), '') FROM sysibm.sysdummy1;

DP 제품에 연동 할 경우
DB2 => SELECT DEC_INDEX_STR('KEY1', ENC_STR('KEY1', 'abc'), 'IX') FROM sysibm.sysdummy1;
DB2 => SELECT DEC_INDEX_B64('KEY1', ENC_B64('KEY1', 'abc'), 'IX') FROM sysibm.sysdummy1;
```

7. HASH_STR

```
DB2 => SELECT HASH_STR( 71, 'abc' ) FROM sysibm.sysdummy1;
```

8. HASH_B64

```
DB2 => SELECT HASH_B64( 71, 'abc' ) FROM sysibm.sysdummy1;
```

9. HEXTOB64

```
DB2 => SELECT HEXTOB64('A305378D8F974F1C1537ED7CB0CB959245D1AC31') FROM sysibm.sysdummy1;
```

10. B64TOHEX

```
DB2 => SELECT B64TOHEX('owU3jY+XTxwVN+18sMuVkkXRrDE=') FROM sysibm.sysdummy1;
```

11. CONFIG_REINIT

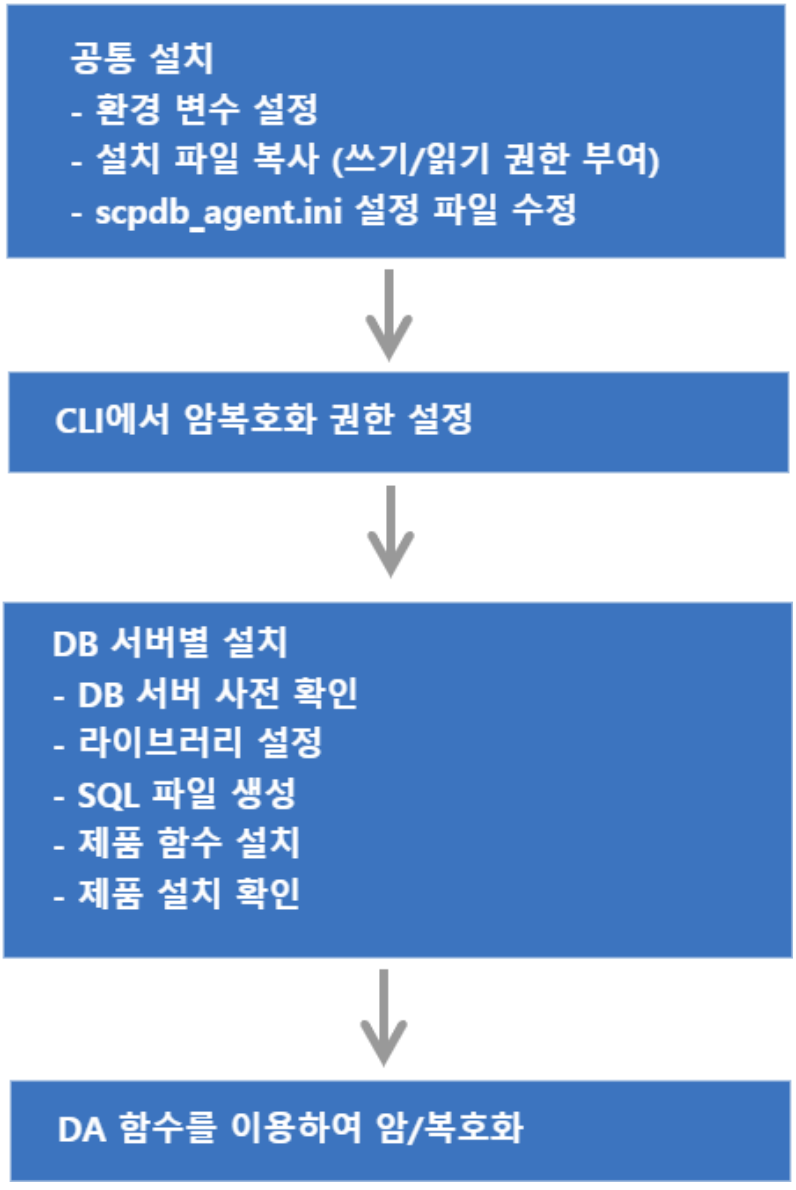
```
DB2 => SELECT CONFIG_REINIT() FROM sysibm.sysdummy1;
```

7.

CUBRID

DA는 DB 서버의 종류에 따라서 설치 및 운용 방법이 각각 다르다.

그림 7-1 설치 개념도



7.1 지원 운영체제 및 DB서버

DA에서 지원하는 운영체제 및 DB서버는 아래와 같다. 단, 특정 환경은 지원되지 않을 수도 있으므로, 제품 설치 전에 상세한 지원 가능 여부는 펜타시큐리티시스템으로 문의한다.

표 7-1 제품이 지원하는 운영체제 및 DB 서버 정보

구분	설명
운영체제	Windows, AIX, HP IA, HP PA-RISC, Linux, SUN, TRU64
DB 서버	ORACLE 8i ~, SQL Server 2012 ~, DB2 9.x ~, Tibero 4 SP 1 ~, MySQL 5.x ~, MariaDB 5.5, 10.0, 10.1, Cache DB 2009.1 ~, Informix IDS 9.x ~ Sybase ASE 15.7 SP61 ~, Sybase IQ 15.4 ~, CUBRID 2008 R1.3 ~ , PostgreSQL 9.4 ~

7.2 설치 파일의 구성 확인

DA의 설치 파일은 아래와 같은 명칭으로 압축 파일(zip) 형태로 제공된다.

- DA 설치파일: Install_DAmo_DA_v{버전}.zip
  - 압축을 해제 한 뒤, 설치 대상 DB 서버, OS 및 bit에 맞는 설치 파일을 준비한다.

설치 파일에 제품을 사용할 수 있는 '라이선스'는 포함되어 있지 않다.
펜타시큐리티시스템에 문의하여 '라이선스' 파일은 별도로 준비한다.

표 7-2 설치 파일(Install_DAmo_DA_v{버전}.zip)을 압축 해제 시, 디렉터리의 구성

구성	설명
_SampleScpsFiles	SG-KMS 연동 없이 암복호화를 테스트할 수 있는 테스트 키 파일
_TestAgentKeyPair	CLI에서 사용할 수 있는 테스트 키 쌍
Altibase	DA-ALT 제품의 설치 바이너리 폴더
Cache	DA-CDB 제품의 설치 바이너리 폴더
Cubrid	DA-CUB 제품의 설치 바이너리 폴더
DB2	DA-DB2 제품의 설치 바이너리 폴더
Informix	DA-IFX 제품의 설치 바이너리 폴더
MySQL	DA-MYQ 제품의 설치 바이너리 폴더
Oracle	DA-ORA 제품의 설치 바이너리 폴더
Postgres	DA-PGS 제품의 설치 바이너리 폴더
SQL Server	DA-MSQ 제품의 설치 바이너리 폴더

구성	설명
SybaseASE	DA-SYB 제품의 설치 바이너리 폴더
SybaseIQ	DA-SIQ 제품의 설치 바이너리 폴더
Tibero	DA-TIB 제품의 설치 바이너리 폴더

7.2.1 DB 서버 및 운영체제 별 SQL파일 구성

각 DB 서버 및 운영체제 별 SQL파일 구성은 다음과 같다. 다음 장에서 각 DB별로 SQL파일 설치 방법을 설명한다.

표 7-3 DB 서버 및 운영체제 별 SQL파일

DB 서버 종류	Linux 일 경우	Windows 일 경우
Oracle	000.da_user.pkg 000.da_user.sql 001.inner_function.ora(c 버전 설치 시) 002.user_interface.ora(JAVA 버전 설치 시) 002.user_interface_java.ora 003.grant_execute_functions.sql(JAVA 버전 설치 시) 005.securej_privilege.sql install_make.sh	000.da_user.pkg 000.da_user.sql 001.inner_function.ora(c 버전 설치 시) 002.user_interface.ora(JAVA 버전 설치 시) 002.user_interface_java.ora 003.grant_execute_functions.sql(JAVA 버전 설치 시) 005.securej_privilege.sql 009.da_test.sql
MYSQL	001.inner_function.mys 002.user_interface. mys 003.grant_execute_functions.sql 009.da_test.sql install_make.sh	001.inner_function.mys 002.user_interface.mys 003.grant_execute_functions.sql
TIBERO	001.inner_function.tbs(c 버전 설치 시) 001.inner_function_java.tbs(JAVA 버전 설치 시) 002.user_interface.tbs(c 버전 설치 시) 002.user_interface_java.tbs(JAVA 버전 설치 시) 003.grant_execute_functions.sql install_make.sh	001.inner_function_java.tbs 002.user_interface_java.tbs 003.grant_execute_functions.sql
INFORMIX	001.inner_function.ifx 002.user_interface. ifx 003.grant_execute_functions.sql 009.da_test.sql install_make.sh	해당 없음
POSTGRESQL	001.inner_function.post 002.user_interface.post 003.grant_execute_functions.post	해당 없음

DB 서버 종류	Linux 일 경우	Windows 일 경우
DB2	001.inner_function.db2 002.user_interface.db2 003.grant_execute_functions.sql install_make.sh	001.inner_function.db2 002.user_interface.db2 003.grant_execute_functions.sql
CUBRID	001.inner_function.sql 002.user_interface.sql	001.inner_function.sql 002.user_interface.sql
SYBASE	001.inner_function.sybase 002.user_interface.sybase 003.grant_execute_functions.sql install_make.sh	해당 없음
SYBASE IQ	001.inner_function.sybiq 002.user_interface. sybiq 003.grant_execute_functions.sql install_make.sh	해당 없음
SQL Server	해당 없음	001.inner_function.sql 002.user_interface. sql 003.grant_execute_functions.sql install_make.bat
Cache DB	SCP.xml install_make.sh	SCP.xml
Altibase	해당 없음	000.da_user.pkg 000.da_user.sql 001.inner_function.sql 002.user_interface. sql install_make.bat

7.2.2 공통 설치 파일 구성

환경에 관계 없이 공통적으로 사용하는 설치 파일은 아래와 같다.

표 7-4 공통 설치 파일 (sql파일 제외)

파일 분류	파일 명	파일 용도
Library 파일	libdamoscpdb.{so|a|sl|dll}	DA 메인 라이브러리. 주로 DBMS External Interface를 담당
	libdamocm-4.0.{so|a|sl|dll}	공통 모듈 라이브러리
	liblogw-0.2.{so|a|sl|dll}	로그를 기록하는 라이브러리
	libcis_cc-3.3.{so|a|sl|dll}	암호화, 복호화 기능을 제공하는 라이브러리
	libcis_ce-3.3.{so|a|sl|dll}	암호화, 복호화를 제외한 부가적인 기능을 제공하는 라이브러리(예: Base64, 인증서 관리, 특성 유지 암호화 등)

파일 분류	파일 명	파일 용도
설정 파일	scpdb_agent.ini	DA 구동시 실행에 필요한 설정정보를 참조
License 파일	damo_lic.cer	DA 구동 시 제품의 유효성을 검증하는데 사용
Agent key 파일	damo_agt_site.cer	SG-KMS 연동, CLI 프로그램에서 사용하는 인증서 쌍
	damo_agt.cer	
	damo_agt.key	
접근제어 파일	acl_cli 파일	DB 의 USER 별로 암·복호 권한을 설정하는데에 사용
	privilege.damo	권한 파일
JAVA class 파일 (Oracle, Tibero, Cubrid 설치 가능)	ScpAgentException.class	예외 처리 Class
	ScpCryptData.class	암호화 복호화 Class
SQL 파일	아래 새로운 표에 DB별로 표기함	

Agent Key 파일은 SG-KMS 연동에 필요한 키 발급를 참고하여 발급 받는다.

DA-PGS(PostgreSQL)의 경우, DB 서버 버전에 따라 libdamoscpdb.so 라이브러리 선택

- libdamoscpdb94.so (Postgres 9.4)
- libdamoscpdb95.so (Postgres 9.5)
- libdamoscpdb95AS.so (EDB Postgres 9.5)
- libdamoscpdb96AS.so (EDB Postgres 9.6)
- libdamoscpdb10.so (Postgres 10)

7.3 환경변수 설정

DA를 설치할 운영체제에 환경변수 DA_INST_HOME를 설정한다. 이 매뉴얼에서는 제품 설치 경로를 아래와 같이 가정하여 설명한다.

- Linux 환경일 경우: /home/dbms_api
- Windows 환경일 경우: E:₩dbms_api

주의) DA_INST_HOME 설정 시, 주의 사항

- 리눅스의 경우 "/root" 디렉토리로 설정을 권장하지 않는다.
- 윈도우의 경우 "바탕화면"으로 설정을 권장하지 않는다.

위의 경로로 설정할 경우 접근 권한 등의 이유로 문제가 발생할 가능성이 존재한다.

7.3.1 환경변수 설정 - Linux 환경일 경우

DA_INST_HOME 환경변수에 DA의 설치 디렉터리를 설정한다.

```
.profile을 사용하는 경우
export DA_INST_HOME=/home/dbms_api


.cshrc를 사용하는 경우
setenv DA_INST_HOME=/home/dbms_api

export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$DA_INST_HOME
```

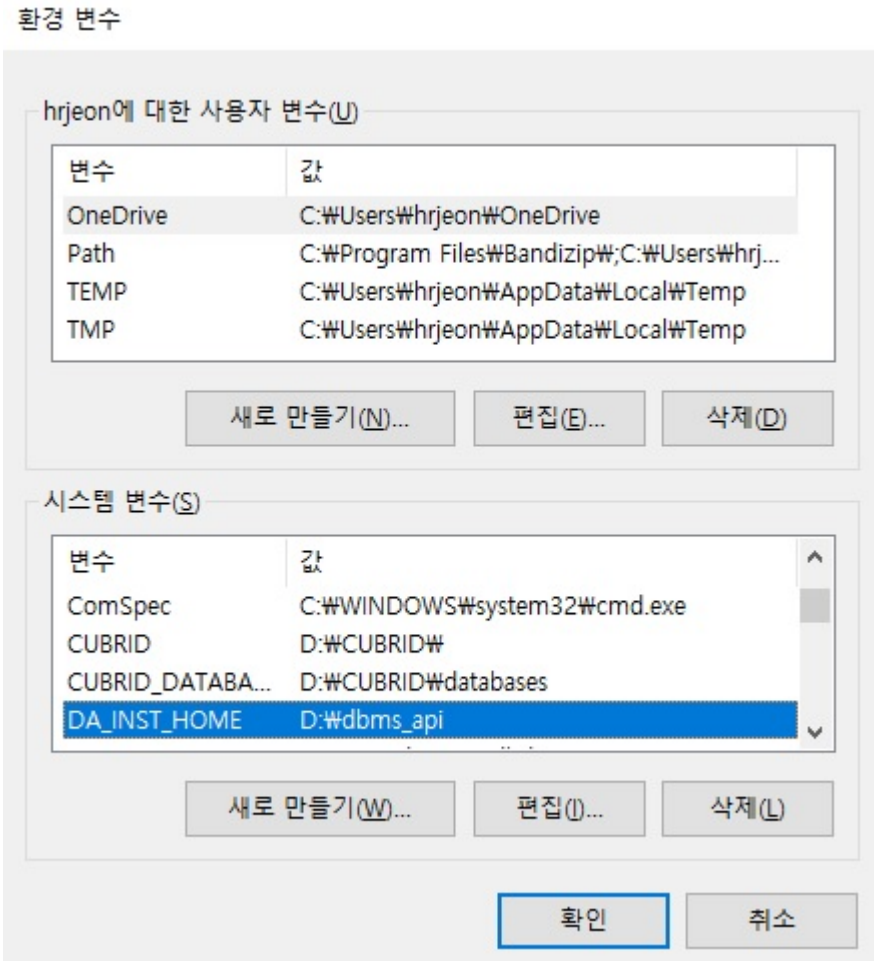
표 7-5 운영 체제별 라이브러리 PATH 명칭

운영 체제	라이브러리 PATH 명칭
HP_UX	SHLIB_PATH
AIX	LIBPATH
LINUX, SUN	LD_LIBRARY_PATH

7.3.2 환경변수 설정 - Windows 환경일 경우

[탐색기->내컴퓨터->등록정보->고급 탭->환경변수] 를 선택하여 나타나는 환경변수 설정 다이얼로그에서 시스템변수 DA_INST_HOME 변수와 값을 추가한다.

그림 7-2 DA_INST_HOME



7.4 DA의 설치 파일 복사

$DA_INST_HOME 디렉터리에 위 [설치 파일의 구성 확인]에서 나열된 파일들을 복사한다.

7.5 라이선스 파일 설정

$DA_INST_HOME 디렉토리에 라이선스 파일(damo_lic.cer)을 복사한다.

라이선스 파일명이 damo_lic.cer가 아닌 다른 이름으로 저장되어 있다면 변경해야 한다.

7.6 설치 파일의 접근 권한 부여 (Linux 설치 시)

Linux 사용자 계정에 DA 설치 파일(라이브러리, 실행 파일, 디렉토리)의 접근 권한을 부여한다. Windows에서 제품을 설치 할 경우, 이 과정은 생략한다.

```
$> cd $DA_INST_HOME
$> chmod 755 lib* acl_cli sql/install_make.sh
```

7.7 (SG-KMS 연동 시) SG-KMS에서 DA 정보 등록 및 연동에 필요한 키 내보내기

DA는 데이터를 암·복호화하기 위해 '암호화 키'가 필요하다. '암호화 키'를 얻기 위해서는 SG-KMS를 연동하거나 SCPS라는 암호화 키 파일을 이용할 수 있는데,
다음 설명은 SG-KMS 연동을 위해 SG-KMS에서 DA 정보를 등록하고 연동에 필요한 키를 내보내는 방법에 대해서 설명한다.

7.7.1 사전 준비

7.7.1.1 지원 SG-KMS 버전

DA와 연동 가능한 SG-KMS 버전은 다음과 같다.

표 7-6 연동 가능한 SG-KMS 버전

SG-KMS Major version	SG-KMS Minor version
SG-KMS v3.0	v3.0.9.0 이상 연동 가능
SG-KMS v4.0	v4.0.104.5 이상 연동 가능

SG-KMS v2.3 연동은 미지원한다.

7.7.1.2 연동 전 점검 항목

SG-KMS 연동을 위해서는 다음과 같이 사전 준비가 필요하다.

- SG-KMS 관리도구
- SG-KMS 관리도구에 접속할 수 있는 ID와 비밀번호
- SG-KMS 매뉴얼

이 매뉴얼에서는 SG-KMS의 키 종류 및 SG-KMS 관리도구 사용 방법에 대한 내용은 다루지 않으므로 **[SG-KMS 매뉴얼]**을 참고한다.

7.7.1.3 SG-KMS 연동에 필요한 키 발급

DA와 SG-KMS의 연동을 위해서는 먼저 SG-KMS에 DA를 Agent로 등록해야한다. 등록하는 절차는 SG-KMS 사용자설명서의 '대칭키를 사용하는 D'Amo Agent 등록 안내서'를 참고한다.

[D'Amo Agent 키] 파일, [Agent ID(CN)]와 [서비스 ID]정보는 DA와 SG-KMS 연동을 위한 설정 과정에서 필요하다. SG-KMS 관리자는 DA 설치 엔지니어에게 안전하게 전달한다.

SG-KMS 관리자에게 받은 D'Amo Agent의 인증서 및 키 파일 명을 다음과 같이 변경후 $DA_INST_HOME/key 디렉터리에 복사한다.

표 7-7 SG-KMS 연동에 필요한 키 목록

키 구분	생성된 키 파일명	변경할 키 파일명
사이트 키	damo-site_{발행기관/부서명}-SITE_V3.cer	damo_agt_site.cer
Agent 키	damo-scp_SITE_V3-{Agent 이름}.cer	damo_agt.cer
	damo-scp_SITE_V3-{Agent 이름}.key	damo_agt.key
	damo-scp_SITE_V3-{Agent 이름}.spin	damo_agt.spin

Agent 키 경로 지정, 키 이름 변경은 반드시 필요한 작업은 아니지만 관리를 위해 변경하는 것을 권장한다.

7.8 설정 파일(scpdb_agent.ini) 수정

DA의 운용을 위해 사용하는 scpdb_agent.ini 설정 파일 수정 방법에 대해 설명한다. DA는 SG-KMS를 이용하거나 SCPS 파일을 이용하여 암호화 키를 얻기 때문에 고객의 환경에 맞게 설정해야 한다.

$DA_INST_HOME 디렉터리에 있는 scpdb_agent.ini 설정 파일에서 아래 항목의 값을 수정한다.

1. 설정 파일의 [KEYINFO] 항목 - [KEY1]에 암호화 키 정보를 입력한다.

```
[KEYINFO]
KEY1=암복호화 하려는 암호화 키 정보를 입력한다.
//키 정보는 다음과 같은 값을 입력할 수 있다.
///ServiceID: SG-KMS에 생성한 서비스 ID를 입력한다.
///SCP_FilePath: SG-KMS에서 서비스 내보내기를 통해 발급한 SCPS 파일의 절대 경로 및 파일명, 확장자를 입력한다.

//*예제 - Windows 경우
KEY1=DA_AES256
KEY2=C:\DA\Policy\S_AES128.SCPS\DA_AES256.scps
KEY3=DA_AES256,C:\DA\Policy\S_AES128.SCPS\DA_AES256.scps

//*예제 - Linux 또는 Unix 경우
KEY1=DA_AES256
KEY2=/home/dbms_api/key/DA_AES256.scps
KEY3=DA_AES256,/home/dbms_api/key/DA_AES256.scps
```

ServiceID를 입력 할 경우 SG-KMS와 통신을 통해 암호화 정책을 가져와 데이터 암복호화를 수행한다.

SCP_FilePath를 입력 할 경우 KMS와 통신하지 않고 서버의 SCP 파일을 통해 암호화 정책을 가져와 데이터 암복호화를 수행한다.

ServiceID와 SCPS 파일명을 동시에 입력 시, SG-KMS와 네트워크 연결 실패 하면 SCPS 파일을 이용하여 암호화 한다.
ServiceID와 SCPS 파일명 사이에 공백이 있으면 SCPS 파일을 읽을 수 없다. 따라서 예제와 같이 띄어쓰기를 하지 않고 ServiceID와 SCPS 파일 경로를 붙여서 입력한다.

2. 설정 파일의 [Server], [Server2] 항목을 수정한다.

```
[Server]
ServerIP: SG-KMS의 IP를 입력한다.
ServerPort: SG-KMS의 포트번호를 입력한다. 기본값은 2525이다.

//*예제 - 모든 OS 공통
```

```
6  ServerIP=192.168.22.25
7  ServerPort=2525
```

```
1  [Server2]
2  ServerIP: 이중화를 위한 2번 SG-KMS의 IP를 입력한다.
3  ServerPort: 이중화를 위한 2번 SG-KMS의 포트번호를 입력한다. 기본값은 2525이다.
4
5  //*예제 - 모든 OS 공통
6  ServerIP=192.168.22.26
7  ServerPort=2525
```

ServerIP는 최소 1개 ~ 최대 10개까지 등록이 가능하다.

3. 설정 파일의 [AGENT] 항목을 설정한다.

```
1  [AGENT]
2  AgentID=SG-KMS 관리도구에서 설정한 D'Amo Agent의 Agent ID
3  LogDir=로그가 저장될 디렉터리 위치
4  LogLevel=로그가 남는 수준
5  SiteCertFilePath=SG-KMS 장비에서 설정한 해당 장비의 사이트 공개키(.cer)
6  CertFilePath=SG-KMS 관리도구에서 설정한 D'Amo Agent의 공개키(.cer)
7  KeyFilePath=SG-KMS 관리도구에서 설정한 D'Amo Agent의 비공개키(.key)
8  SPIN=SG-KMS 관리도구에서 설정한 D'Amo Agent의 SPIN으로, damo-scp_SITE_V3-[Agent이름].spin
파일의 값
9
10 //[예제]
11 AgentID=DA
12 LogDir=/home/dbms_api/log
13 LogLevel=4
14 SiteCertFilePath=/home/dbms_api/key/damo_agt_site.cer
15 CertFilePath=/home/dbms_api/key/damo_agt.cer
16 KeyFilePath=/home/dbms_api/key/damo_agt.key
17 SPIN=XaMh1ylXUh123XUh
```

로그가 남는 수준(LogLevel)에는 아래 5가지 숫자 입력이 가능하며, 각 값의 설정은 다음과 같다.
- 0: 아무 로그도 남기지 않을 경우
- 2: 경고 로그를 파일에 기록
- 4: 에러 로그와 경고 로그를 파일에 기록

- 6: 정보 로그, 에러 로그, 경고 로그를 파일에 기록
- 8: 디버그, 정보 로그, 에러 로그, 경고 로그를 파일에 기록

제품 운영 중 scpdb_agent.ini 파일을 수정하면 CONFIG_REINIT() 함수를 호출해야 변경된 내용이 적용된다.

7.9 CLI에서 권한 설정

$DA_INST_HOME 디렉터리에서 acl_cli 파일을 실행하여 USER 단위로 암/복호화 권한을 설정한다. USER 는 DB의 소유자명이고, KEY는 설정 파일(scpdb_agent.ini)의 [KEYINFO] 중 ALIAS 값(예: KEY1)이다.

CLI 명령어를 자세히 보려면 help 명령어를 실행한다.

```
CLI 실행 방법
$> cd $DA_INST_HOME
$> ./acl_cli  - start
Enter the PIN of CLI-key. : damo_agt.key 의 비밀번호
```

```
권한 추가할 경우
D'Amo > SET PRIV ENC [USER]"[KEY]"1"1
D'Amo > SAVE ALL
D'Amo > SHOW ALL

예제)
D'Amo > SET PRIV ENC SCOTT"KEY1"1"1
D'Amo > SET PRIV ENC SCOTT"KEY2"1"1
```

scpdb_agent.ini 설정 파일이 아래 예제와 같을 경우, CLI에서 권한 설정할 때 입력해야 하는 2번째 [KEY] 인자 값에는 KEY1을 입력해야 한다. (※ARIA256을 입력하는 것이 아님)

#scpdb_agent.ini 설정 파일 예제
[KEYINFO]
KEY1=ARIA256

```
권한 삭제할 경우
D'Amo > DEL PRIV ENC [USER]"[KEY]
D'Amo > SAVE ALL
D'Amo > SHOW ALL
```

CLI 에서 권한을 추가하거나 삭제 한 경우 반드시 SAVE ALL 명령어를 실행하며 SHOW ALL 명령어를 이용하여 적용 여부를 확인 한다.

7.10 DB 서버 사전 확인

DA를 CUBRID 환경에서 설치 하기 전에 다음과 같은 사항들을 확인한다.

- DB 서버 재시작 또는 DB 서버의 listener 재시작
- DB 엔진 설치 계정에 폴더 생성
- DB 사용자로 SCP 계정 생성 (권고사항)
- DB 서버의 DBA 권한 계정
- DB 엔진 설치 계정의 . profile 파일 수정
- JAVA_HOME의 읽기, 쓰기 권한

CUBRID 에서 외부 라이브러리를 연동하기 위해서는 다음과 CUBRID 설정파일($CUBRID/conf/cubrid.conf)에 java_stored_procedure 부분을 yes 로 변경하고 재시작한다.

7.11 “.lock” 파일 생성

4. $DA_INST_HOME 위치에 디렉터리명을 “sql”, “key”, “log” 로 하여 빈 디렉터리를 생성한다.
5. $DA_INST_HOME/log 디렉터리에 touch 명령어로 파일명이 “.lock” 인 빈 파일을 생성한다.

Windows 에서는 파일명 앞에 점(.)을 허용하지 않기 때문에 "lock"이라는 파일을 생성 한 후 CMD창에서 DOS 명령어로 rename 해야한다.

```
$> cd $DA_INST_HOME
$DA_INST_HOME> mkdir key
$DA_INST_HOME> mkdir log
$DA_INST_HOME> mkdir sql
$DA_INST_HOME> cd log
$DA_INST_HOME/log> touch .lock
```

7.12 라이브러리 설정

7.12.1 라이브러리 링크 설정 (Linux 설치 시)

Linux 설치 시, $CUBRID/lib 경로에 $DA_INST_HOME에 있는 라이브러리를 심볼릭 링크로 걸어준다.

```
$> cd $CUBRID/lib
$> ln -s $DA_INST_HOME/libcis_cc-3.3.so libcis_cc-3.3.so
$> ln -s $DA_INST_HOME/libcis_ce-3.3.so libcis_ce-3.3.so
$> ln -s $DA_INST_HOME/liblogw-0.2.so liblogw-0.2.so
$> ln -s $DA_INST_HOME/libdamocm-4.0.so libdamocm-4.0.so
$> ln -s $DA_INST_HOME/libdamoscpdb.so libdamoscpdb.so
```

7.12.2 라이브러리 복사 및 설정 (Windows 설치 시)

Windows에 설치 시, 다음과 같이 라이브러리 복사가 필요하다.

- CUBRID설치홈/lib 디렉터리: libdamoscpdb.dll을 복사
- CUBRID설치홈/bin 디렉터리: cis_cc-3.3.dll 파일, cis_ce-3.3.dll 파일, damocm-4.0.dll, logw-0.2.dll 파일을 복사

7.13 제품 함수 설치

1. $DA_INST_HOME/sql 위치에서 DB 접속 후 설치를 원하는 DB계정에 함수를 설치한다. SCP 계정을 생성한 경

우 SCP 계정에 설치한다.

```
$>csql -S -i001.inner_function.sql -udba demodb > 1.txt
$>csql -S -i002.user_interface.sql -udba demodb > 2.txt
```

2. class 파일을 DBMS에서 인식하기 위해서 loadjava를 실행한다.

```
$>loadjava -y  demodb ScpAgentException.class
$>loadjava -y  demodb ScpCryptData.class
```

7.14 함수 초기화 설정

함수를 사용하기 위한 정보를 초기화해주어야 한다. CUBRID에 접속하여 다음을 실행한다.

```
$>csql -udba demodb
csql>select DAMO_DA_INIT('INI');

[예제] ※DAMO_DA_INIT 경로를 /home/dbms_api라고 가정한다.
csql>select DAMO_DA_INIT('/home/dbms_api/scpdb_agent.ini');
DAMO_DA_INIT('/home/dbms_api/scpdb_agent.ini')
======================
  'SUCCESS'
$>
```

7.15 제품 설치 확인

DB 서버에서 암/복호화 함수를 호출하여 설치를 성공했는지 확인한다.

```
csql> SELECT ENC_STR( USER, 'DAMO', 'abc');
ENC_STR( USER, 'DAMO','abc')
------------------------------------------------
5E41ACD673653158D7AE8C30CDA9627D3556E173

csql> SELECT DEC_STR( USER, 'DAMO', ENC_STR( USER, 'DAMO', 'abc'));
```

```
DEC_STR( USER, 'DAMO', ENC_STR( USER, 'DAMO', 'abc'))
------------------------------------------------
abc
```

함수 실행 중
“ERROR: Stored procedure execute error: java.lang.reflect.InvocationTargetException”
예외가 발생했을 때, log 폴더에 damo_scp_error 로그 파일을 확인한다.

7.16 제품 운용

7.16.1 함수 설명

DA에서 제공되는 함수와 사용하는 방법에 대해서 설명한다.

7.16.1.1 파라미터 설명

7.16.1.1.1 I_USER

I_USER: USER. USER는 CUBRID에서 제공하는 DB USER명 획득 함수

'USER'외에 다른 텍스트를 입력할 경우, 제품이 오동작할 수있다. 주의 할 것은 DB 소유자 이름을 쓰는 것이 아니고 'USER'라고 입력해야 한다.

7.16.1.1.2 I_KEY

I_KEY : 암호화/복호화 때 사용하는 암호화 키
- 설정 파일(scpdb_agent.ini)의 [KEYINFO] 중 ALIAS 값
- 아래 [설정 파일 예제]의 경우 ALIAS는 'KEY1'과 'KEY2'이다.
- ALIAS는 SCPS파일로 암호화 할 것인지, SG-KMS의 서비스 ID로 암호화 할 것인지 설정 가능하다.

[설정 파일 예제]

- 설정 파일(scpdb_agent.ini)의 [KEYINFO] 중 ALIAS 값
KEY1=AES256.SCPS
KEY2=ARIA256

[함수 사용 예제]
SELECT ENC_B64('KEY1', 'abc') FROM DUAL;

7.16.1.1.3 I_DATA

I_DATA : 평문(암호화 함수일 경우), 암호문(복호화 함수일 경우)

DA에서 제공하는 암호화 함수에서 성능 향상을 위해 라이브러리에서 정책 이름(KEYINFO ALIAS)을 바탕으로 Cash하여 동작한다. 정책 이름은 같은데 실제 키(대칭 키)가 다른 정책을 사용하면 Cash에서 이전 키로 복호화를 시도하여 오류가 발생한다. 이 상황을 해결하기 위해서는 데이터베이스 서버 재시작이 필요하다.

표 7-8 DA 함수 (CUBRID)

함수 명	입력		출력
ENC_STR	I_USER	USER	Hex String 암호문
	I_KEY	IN 문자열,	
	I_DATA	IN 문자열 (평문)	
ENC_B64	I_USER	USER	Base64 Encording 암호문
	I_KEY	IN 문자열,	
	I_DATA	IN 문자열 (평문)	
DEC_STR	I_USER	USER	평문
	I_KEY	IN 문자열,	
	I_DATA	IN 문자열 (Hex String 암호문)	
DEC_B64	I_USER	USER	평문
	I_KEY	IN 문자열,	
	I_DATA	IN 문자열 (base64 String 암호문)	
INDEX_STR	I_USER	USER	Hex String 암호문
	I_KEY	IN 문자열,	
	I_DATA	IN 문자열 (평문),	
	I_TYPE	IN 문자열 '' or	

		'IX '(Plug-IN 연동 시 사용)		
DEC_INDEX_STR	I_USER	USER		Hex String 암호문 입력받아 OPE 데이터
	I_KEY	IN 문자열,		
	I_DATA	IN 문자열 (암호문),		
	I_TYPE	IN 문자열 '' or 'IX '(Plug-IN 연동 시 사용)		
DEC_INDEX_B64	I_USER	USER		Base64 Encording 암호문 입력받아 OPE 데이터
	I_KEY	IN 문자열,		
	I_DATA	IN 문자열 (암호문),		
	I_TYPE	IN 문자열 '' or 'IX '(Plug-IN 연동 시 사용)		
HASH_STR	I_ALOG	IN 숫자,	SHA1 =70	Hex String 해쉬 암호문
			SHA256 =71	
			SHA384 =72	
			SHA512 =73	
			HAS160 =74	
	I_DATA	IN 문자열		
HASH_B64	I_ALOG	IN 숫자,	SHA1 =70	Base64 String 해쉬 암호문
			SHA256 =71	
			SHA384 =72	
			SHA512 =73	
			HAS160 =74	
	I_DATA	IN 문자열		
HEXTOB64	I_DATA	IN 문자열 (Hex String 암호문)		base64 Encording 암호문
B64TOHEX	I_DATA	IN 문자열 (base64 Encording 암호문)		Hex String 암호문
CONFIG_REINIT				성공시 'SUCCESS', 그 외 에러

7.16.2 함수 호출 예제

1. ENC_STR

```
csql> SELECT ENC_STR(USER, 'KEY1', 'abc');
```

2. ENC_B64

```
csql> SELECT ENC_B64(USER, 'KEY1', 'abc');
```

3. DEC_STR

```
csql> SELECT DEC_STR(USER, 'KEY1', ENC_STR(USER, 'KEY1', 'abc'));
```

4. DEC_B64

```
csql> SELECT DEC_B64(USER, 'KEY1', ENC_B64(USER, 'KEY1', 'abc'));
```

5. INDEX_STR

```
DP 제품에 연동 하지 않을 경우
csql> SELECT INDEX_STR(USER, 'KEY1', 'abc', '');

DP 제품에 연동 할 경우
csql> SELECT INDEX_STR(USER, 'KEY1', 'abc', 'IX');
```

6. DEC_INDEX_STR, DEC_INDEX_B64

```
DP 제품에 연동 하지 않을 경우
csql> SELECT DEC_INDEX_STR(USER, 'KEY1', ENC_STR(USER, 'KEY1', 'abc'), '');
csql> SELECT DEC_INDEX_B64(USER, 'KEY1', ENC_B64(USER, 'KEY1', 'abc'), '');

DP 제품에 연동 할 경우
csql> SELECT DEC_INDEX_STR(USER, 'KEY1', ENC_STR(USER, 'KEY1', 'abc'), 'IX');
csql> SELECT DEC_INDEX_B64(USER, 'KEY1', ENC_B64(USER, 'KEY1', 'abc'), 'IX');
```

7. HASH_STR

```
csql> SELECT HASH_STR( 71, 'abc' );
```

8. HASH_B64

```
csql> SELECT HASH_B64( 71, 'abc' );
```

9. HEXTOB64

```
csql> SELECT HEXTOB64('A305378D8F974F1C1537ED7CB0CB959245D1AC31');
```

10. B64TOHEX

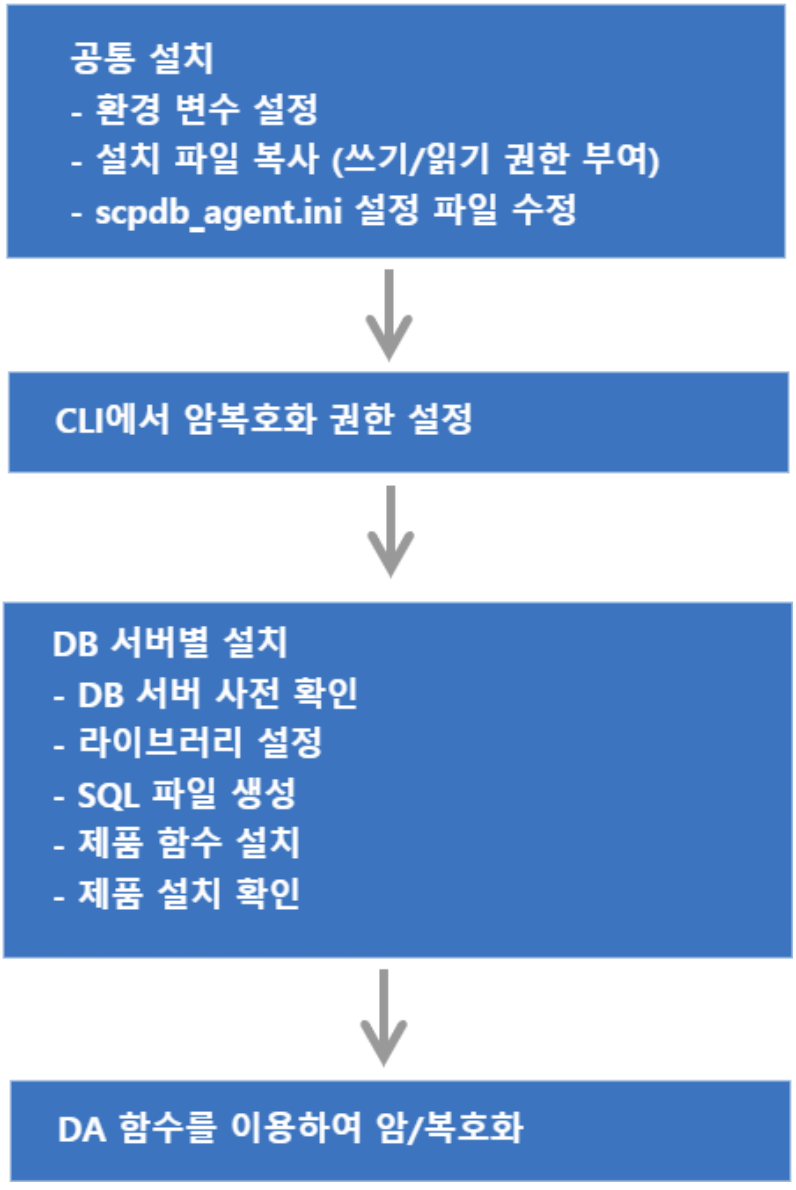
```
csql> SELECT B64TOHEX('owU3jY+XTxwVN+18sMuVkkXRrDE=');
```

8.

SYBASE

DA는 DB 서버의 종류에 따라서 설치 및 운용 방법이 각각 다르다.

그림 8-1 설치 개념도



8.1 지원 운영체제 및 DB서버

DA에서 지원하는 운영체제 및 DB서버는 아래와 같다. 단, 특정 환경은 지원되지 않을 수도 있으므로, 제품 설치 전에 상세한 지원 가능 여부는 펜타시큐리티시스템으로 문의한다.

표 8-1 제품이 지원하는 운영체제 및 DB 서버 정보

구분	설명
운영체제	Windows, AIX, HP IA, HP PA-RISC, Linux, SUN, TRU64
DB 서버	ORACLE 8i ~, SQL Server 2012 ~, DB2 9.x ~, Tibero 4 SP 1 ~, MySQL 5.x ~, MariaDB 5.5, 10.0, 10.1, Cache DB 2009.1 ~, Informix IDS 9.x ~ Sybase ASE 15.7 SP61 ~, Sybase IQ 15.4 ~, CUBRID 2008 R1.3 ~ , PostgreSQL 9.4 ~

8.2 설치 파일의 구성 확인

DA의 설치 파일은 아래와 같은 명칭으로 압축 파일(zip) 형태로 제공된다.

- DA 설치파일: Install_DAmo_DA_v{버전}.zip
  - 압축을 해제 한 뒤, 설치 대상 DB 서버, OS 및 bit에 맞는 설치 파일을 준비한다.

설치 파일에 제품을 사용할 수 있는 '라이선스'는 포함되어 있지 않다.
펜타시큐리티시스템에 문의하여 '라이선스' 파일은 별도로 준비한다.

표 8-2 설치 파일(Install_DAmo_DA_v{버전}.zip)을 압축 해제 시, 디렉터리의 구성

구성	설명
_SampleScpsFiles	SG-KMS 연동 없이 암복호화를 테스트할 수 있는 테스트 키 파일
_TestAgentKeyPair	CLI에서 사용할 수 있는 테스트 키 쌍
Altibase	DA-ALT 제품의 설치 바이너리 폴더
Cache	DA-CDB 제품의 설치 바이너리 폴더
Cubrid	DA-CUB 제품의 설치 바이너리 폴더
DB2	DA-DB2 제품의 설치 바이너리 폴더
Informix	DA-IFX 제품의 설치 바이너리 폴더
MySQL	DA-MYQ 제품의 설치 바이너리 폴더
Oracle	DA-ORA 제품의 설치 바이너리 폴더
Postgres	DA-PGS 제품의 설치 바이너리 폴더
SQL Server	DA-MSQ 제품의 설치 바이너리 폴더

구성	설명
SybaseASE	DA-SYB 제품의 설치 바이너리 폴더
SybaseIQ	DA-SIQ 제품의 설치 바이너리 폴더
Tibero	DA-TIB 제품의 설치 바이너리 폴더

8.2.1 DB 서버 및 운영체제 별 SQL파일 구성

각 DB 서버 및 운영체제 별 SQL파일 구성은 다음과 같다. 다음 장에서 각 DB별로 SQL파일 설치 방법을 설명한다.

표 8-3 DB 서버 및 운영체제 별 SQL파일

DB 서버 종류	Linux 일 경우	Windows 일 경우
Oracle	000.da_user.pkg 000.da_user.sql 001.inner_function.ora(c 버전 설치 시) 002.user_interface.ora(JAVA 버전 설치 시) 002.user_interface_java.ora 003.grant_execute_functions.sql(JAVA 버전 설치 시) 005.securej_privilege.sql install_make.sh	000.da_user.pkg 000.da_user.sql 001.inner_function.ora(c 버전 설치 시) 002.user_interface.ora(JAVA 버전 설치 시) 002.user_interface_java.ora 003.grant_execute_functions.sql(JAVA 버전 설치 시) 005.securej_privilege.sql 009.da_test.sql
MYSQL	001.inner_function.mys 002.user_interface. mys 003.grant_execute_functions.sql 009.da_test.sql install_make.sh	001.inner_function.mys 002.user_interface.mys 003.grant_execute_functions.sql
TIBERO	001.inner_function.tbs(c 버전 설치 시) 001.inner_function_java.tbs(JAVA 버전 설치 시) 002.user_interface.tbs(c 버전 설치 시) 002.user_interface_java.tbs(JAVA 버전 설치 시) 003.grant_execute_functions.sql install_make.sh	001.inner_function_java.tbs 002.user_interface_java.tbs 003.grant_execute_functions.sql
INFORMIX	001.inner_function.ifx 002.user_interface. ifx 003.grant_execute_functions.sql 009.da_test.sql install_make.sh	해당 없음
POSTGRESQL	001.inner_function.post 002.user_interface.post 003.grant_execute_functions.post	해당 없음

DB 서버 종류	Linux 일 경우	Windows 일 경우
DB2	001.inner_function.db2 002.user_interface.db2 003.grant_execute_functions.sql install_make.sh	001.inner_function.db2 002.user_interface.db2 003.grant_execute_functions.sql
CUBRID	001.inner_function.sql 002.user_interface.sql	001.inner_function.sql 002.user_interface.sql
SYBASE	001.inner_function.sybase 002.user_interface.sybase 003.grant_execute_functions.sql install_make.sh	해당 없음
SYBASE IQ	001.inner_function.sybiq 002.user_interface. sybiq 003.grant_execute_functions.sql install_make.sh	해당 없음
SQL Server	해당 없음	001.inner_function.sql 002.user_interface. sql 003.grant_execute_functions.sql install_make.bat
Cache DB	SCP.xml install_make.sh	SCP.xml
Altibase	해당 없음	000.da_user.pkg 000.da_user.sql 001.inner_function.sql 002.user_interface. sql install_make.bat

8.2.2 공통 설치 파일 구성

환경에 관계 없이 공통적으로 사용하는 설치 파일은 아래와 같다.

표 8-4 공통 설치 파일 (sql파일 제외)

파일 분류	파일 명	파일 용도
Library 파일	libdamoscpdb.{so|a|sl|dll}	DA 메인 라이브러리. 주로 DBMS External Interface를 담당
	libdamocm-4.0.{so|a|sl|dll}	공통 모듈 라이브러리
	liblogw-0.2.{so|a|sl|dll}	로그를 기록하는 라이브러리
	libcis_cc-3.3.{so|a|sl|dll}	암호화, 복호화 기능을 제공하는 라이브러리
	libcis_ce-3.3.{so|a|sl|dll}	암호화, 복호화를 제외한 부가적인 기능을 제공하는 라이브러리(예: Base64, 인증서 관리, 특성 유지 암호화 등)

파일 분류	파일 명	파일 용도
설정 파일	scpdb_agent.ini	DA 구동시 실행에 필요한 설정정보를 참조
License 파일	damo_lic.cer	DA 구동 시 제품의 유효성을 검증하는데 사용
Agent key 파일	damo_agt_site.cer	SG-KMS 연동, CLI 프로그램에서 사용하는 인증서 쌍
	damo_agt.cer	
	damo_agt.key	
접근제어 파일	acl_cli 파일	DB 의 USER 별로 암·복호 권한을 설정하는데에 사용
	privilege.damo	권한 파일
JAVA class 파일 (Oracle, Tibero, Cubrid 설치 가능)	ScpAgentException.class	예외 처리 Class
	ScpCryptData.class	암호화 복호화 Class
SQL 파일	아래 새로운 표에 DB별로 표기함	

Agent Key 파일은 SG-KMS 연동에 필요한 키 발급를 참고하여 발급 받는다.

DA-PGS(PostgreSQL)의 경우, DB 서버 버전에 따라 libdamoscpdb.so 라이브러리 선택

- libdamoscpdb94.so (Postgres 9.4)
- libdamoscpdb95.so (Postgres 9.5)
- libdamoscpdb95AS.so (EDB Postgres 9.5)
- libdamoscpdb96AS.so (EDB Postgres 9.6)
- libdamoscpdb10.so (Postgres 10)

8.3 환경변수 설정

DA를 설치할 운영체제에 환경변수 DA_INST_HOME를 설정한다. 이 매뉴얼에서는 제품 설치 경로를 아래와 같이 가정하여 설명한다.

- Linux 환경일 경우: /home/dbms_api
- Windows 환경일 경우: E:₩dbms_api

주의) DA_INST_HOME 설정 시, 주의 사항

- 리눅스의 경우 "/root" 디렉토리로 설정을 권장하지 않는다.
- 윈도우의 경우 "바탕화면"으로 설정을 권장하지 않는다.

위의 경로로 설정할 경우 접근 권한 등의 이유로 문제가 발생할 가능성이 존재한다.

8.3.1 환경변수 설정 - Linux 환경일 경우

DA_INST_HOME 환경변수에 DA의 설치 디렉터리를 설정한다.

```
.profile을 사용하는 경우
export DA_INST_HOME=/home/dbms_api


.cshrc를 사용하는 경우
setenv DA_INST_HOME=/home/dbms_api

export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$DA_INST_HOME
```

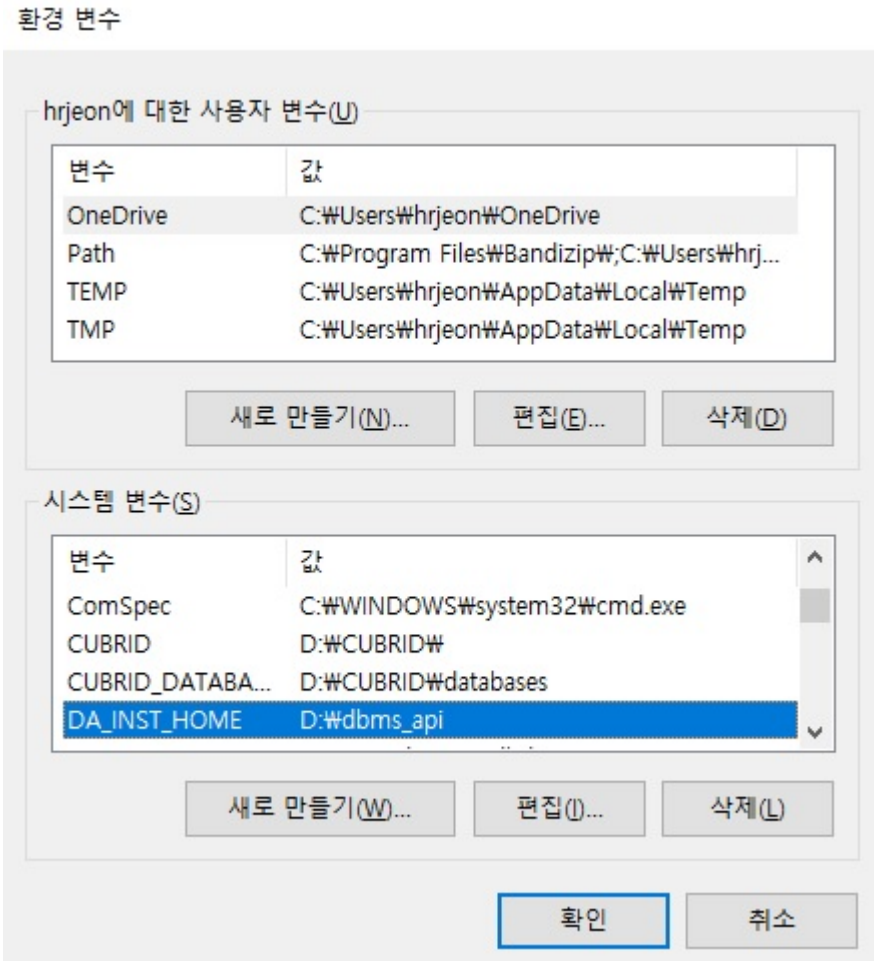
표 8-5 운영 체제별 라이브러리 PATH 명칭

운영 체제	라이브러리 PATH 명칭
HP_UX	SHLIB_PATH
AIX	LIBPATH
LINUX, SUN	LD_LIBRARY_PATH

8.3.2 환경변수 설정 - Windows 환경일 경우

[탐색기->내컴퓨터->등록정보->고급 탭->환경변수] 를 선택하여 나타나는 환경변수 설정 다이얼로그에서 시스템변수 DA_INST_HOME 변수와 값을 추가한다.

그림 8-2 DA_INST_HOME



8.4 DA의 설치 파일 복사

$DA_INST_HOME 디렉터리에 위 [설치 파일의 구성 확인]에서 나열된 파일들을 복사한다.

8.5 라이선스 파일 설정

$DA_INST_HOME 디렉토리에 라이선스 파일(damo_lic.cer)을 복사한다.

라이선스 파일명이 damo_lic.cer가 아닌 다른 이름으로 저장되어 있다면 변경해야 한다.

8.6 설치 파일의 접근 권한 부여 (Linux 설치 시)

Linux 사용자 계정에 DA 설치 파일(라이브러리, 실행 파일, 디렉토리)의 접근 권한을 부여한다. Windows에서 제품을 설치 할 경우, 이 과정은 생략한다.

```
$> cd $DA_INST_HOME
$> chmod 755 lib* acl_cli sql/install_make.sh
```

8.7 (SG-KMS 연동 시) SG-KMS에서 DA 정보 등록 및 연동에 필요한 키 내보내기

DA는 데이터를 암·복호화하기 위해 '암호화 키'가 필요하다. '암호화 키'를 얻기 위해서는 SG-KMS를 연동하거나 SCPS라는 암호화 키 파일을 이용할 수 있는데,
다음 설명은 SG-KMS 연동을 위해 SG-KMS에서 DA 정보를 등록하고 연동에 필요한 키를 내보내는 방법에 대해서 설명한다.

8.7.1 사전 준비

8.7.1.1 지원 SG-KMS 버전

DA와 연동 가능한 SG-KMS 버전은 다음과 같다.

표 8-6 연동 가능한 SG-KMS 버전

SG-KMS Major version	SG-KMS Minor version
SG-KMS v3.0	v3.0.9.0 이상 연동 가능
SG-KMS v4.0	v4.0.104.5 이상 연동 가능

SG-KMS v2.3 연동은 미지원한다.

8.7.1.2 연동 전 점검 항목

SG-KMS 연동을 위해서는 다음과 같이 사전 준비가 필요하다.

- SG-KMS 관리도구
- SG-KMS 관리도구에 접속할 수 있는 ID와 비밀번호
- SG-KMS 매뉴얼

이 매뉴얼에서는 SG-KMS의 키 종류 및 SG-KMS 관리도구 사용 방법에 대한 내용은 다루지 않으므로 **[SG-KMS 매뉴얼]**을 참고한다.

8.7.1.3 SG-KMS 연동에 필요한 키 발급

DA와 SG-KMS의 연동을 위해서는 먼저 SG-KMS에 DA를 Agent로 등록해야한다. 등록하는 절차는 SG-KMS 사용자설명서의 '대칭키를 사용하는 D'Amo Agent 등록 안내서'를 참고한다.

[D'Amo Agent 키] 파일, [Agent ID(CN)]와 [서비스 ID]정보는 DA와 SG-KMS 연동을 위한 설정 과정에서 필요하다. SG-KMS 관리자는 DA 설치 엔지니어에게 안전하게 전달한다.

SG-KMS 관리자에게 받은 D'Amo Agent의 인증서 및 키 파일 명을 다음과 같이 변경후 $DA_INST_HOME/key 디렉터리에 복사한다.

표 8-7 SG-KMS 연동에 필요한 키 목록

키 구분	생성된 키 파일명	변경할 키 파일명
사이트 키	damo-site_{발행기관/부서명}-SITE_V3.cer	damo_agt_site.cer
Agent 키	damo-scp_SITE_V3-{Agent 이름}.cer	damo_agt.cer
	damo-scp_SITE_V3-{Agent 이름}.key	damo_agt.key
	damo-scp_SITE_V3-{Agent 이름}.spin	damo_agt.spin

Agent 키 경로 지정, 키 이름 변경은 반드시 필요한 작업은 아니지만 관리를 위해 변경하는 것을 권장한다.

8.8 설정 파일(scpdb_agent.ini) 수정

DA의 운용을 위해 사용하는 scpdb_agent.ini 설정 파일 수정 방법에 대해 설명한다. DA는 SG-KMS를 이용하거나 SCPS 파일을 이용하여 암호화 키를 얻기 때문에 고객의 환경에 맞게 설정해야 한다.

$DA_INST_HOME 디렉터리에 있는 scpdb_agent.ini 설정 파일에서 아래 항목의 값을 수정한다.

1. 설정 파일의 [KEYINFO] 항목 - [KEY1]에 암호화 키 정보를 입력한다.

```
[KEYINFO]
KEY1=암복호화 하려는 암호화 키 정보를 입력한다.
//키 정보는 다음과 같은 값을 입력할 수 있다.
///ServiceID: SG-KMS에 생성한 서비스 ID를 입력한다.
///SCP_FilePath: SG-KMS에서 서비스 내보내기를 통해 발급한 SCPS 파일의 절대 경로 및 파일명, 확장자를 입력한다.

//*예제 - Windows 경우
KEY1=DA_AES256
KEY2=C:\DA\Policy\S_AES128.SCPS\DA_AES256.scps
KEY3=DA_AES256,C:\DA\Policy\S_AES128.SCPS\DA_AES256.scps

//*예제 - Linux 또는 Unix 경우
KEY1=DA_AES256
KEY2=/home/dbms_api/key/DA_AES256.scps
KEY3=DA_AES256,/home/dbms_api/key/DA_AES256.scps
```

ServiceID를 입력 할 경우 SG-KMS와 통신을 통해 암호화 정책을 가져와 데이터 암복호화를 수행한다.

SCP_FilePath를 입력 할 경우 KMS와 통신하지 않고 서버의 SCP 파일을 통해 암호화 정책을 가져와 데이터 암복호화를 수행한다.

ServiceID와 SCPS 파일명을 동시에 입력 시, SG-KMS와 네트워크 연결 실패 하면 SCPS 파일을 이용하여 암호화 한다.
ServiceID와 SCPS 파일명 사이에 공백이 있으면 SCPS 파일을 읽을 수 없다. 따라서 예제와 같이 띄어쓰기를 하지 않고 ServiceID와 SCPS 파일 경로를 붙여서 입력한다.

2. 설정 파일의 [Server], [Server2] 항목을 수정한다.

```
[Server]
ServerIP: SG-KMS의 IP를 입력한다.
ServerPort: SG-KMS의 포트번호를 입력한다. 기본값은 2525이다.

//*예제 - 모든 OS 공통
```

```
ServerIP=192.168.22.25
ServerPort=2525
```

```
[Server2]
ServerIP: 이중화를 위한 2번 SG-KMS의 IP를 입력한다.
ServerPort: 이중화를 위한 2번 SG-KMS의 포트번호를 입력한다. 기본값은 2525이다.

//*예제 - 모든 OS 공통
ServerIP=192.168.22.26
ServerPort=2525
```

ServerIP는 최소 1개 ~ 최대 10개까지 등록이 가능하다.

3. 설정 파일의 [AGENT] 항목을 설정한다.

```
[AGENT]
AgentID=SG-KMS 관리도구에서 설정한 D'Amo Agent의 Agent ID
LogDir=로그가 저장될 디렉터리 위치
LogLevel=로그가 남는 수준
SiteCertFilePath=SG-KMS 장비에서 설정한 해당 장비의 사이트 공개키(.cer)
CertFilePath=SG-KMS 관리도구에서 설정한 D'Amo Agent의 공개키(.cer)
KeyFilePath=SG-KMS 관리도구에서 설정한 D'Amo Agent의 비공개키(.key)
SPIN=SG-KMS 관리도구에서 설정한 D'Amo Agent의 SPIN으로, damo-scp_SITE_V3-[Agent이름].spin 파일의 값

//[예제]
AgentID=DA
LogDir=/home/dbms_api/log
LogLevel=4
SiteCertFilePath=/home/dbms_api/key/damo_agt_site.cer
CertFilePath=/home/dbms_api/key/damo_agt.cer
KeyFilePath=/home/dbms_api/key/damo_agt.key
SPIN=XaMh1ylXUh123XUh
```

로그가 남는 수준(LogLevel)에는 아래 5가지 숫자 입력이 가능하며, 각 값의 설정은 다음과 같다.
- 0: 아무 로그도 남기지 않을 경우
- 2: 경고 로그를 파일에 기록
- 4: 에러 로그와 경고 로그를 파일에 기록

- 6: 정보 로그, 에러 로그, 경고 로그를 파일에 기록
- 8: 디버그, 정보 로그, 에러 로그, 경고 로그를 파일에 기록

제품 운영 중 scpdb_agent.ini 파일을 수정하면 CONFIG_REINIT() 함수를 호출해야 변경된 내용이 적용된다.

8.9 CLI에서 권한 설정

$DA_INST_HOME 디렉터리에서 acl_cli 파일을 실행하여 USER 단위로 암/복호화 권한을 설정한다. USER 는 DB의 소유자명이고, KEY는 설정 파일(scpdb_agent.ini)의 [KEYINFO] 중 ALIAS 값(예: KEY1)이다.

CLI 명령어를 자세히 보려면 help 명령어를 실행한다.

```
CLI 실행 방법
$> cd $DA_INST_HOME
$> ./acl_cli  - start
Enter the PIN of CLI-key. : damo_agt.key 의 비밀번호
```

```
권한 추가할 경우
D'Amo > SET PRIV ENC [USER]"[KEY]"1"1
D'Amo > SAVE ALL
D'Amo > SHOW ALL

예제)
D'Amo > SET PRIV ENC SCOTT"KEY1"1"1
D'Amo > SET PRIV ENC SCOTT"KEY2"1"1
```

scpdb_agent.ini 설정 파일이 아래 예제와 같을 경우, CLI에서 권한 설정할 때 입력해야 하는 2번째 [KEY] 인자 값에는 KEY1을 입력해야 한다. (※ARIA256을 입력하는 것이 아님)

#scpdb_agent.ini 설정 파일 예제
[KEYINFO]
KEY1=ARIA256

```
권한 삭제할 경우
D'Amo > DEL PRIV ENC [USER]"[KEY]
D'Amo > SAVE ALL
D'Amo > SHOW ALL
```

CLI 에서 권한을 추가하거나 삭제 한 경우 반드시 SAVE ALL 명령어를 실행하며 SHOW ALL 명령어를 이용하여 적용 여부를 확인 한다.

8.10 DB 서버 사전 확인

DA를 SYBASE 환경에서 설치 하기 전에 다음과 같은 사항들을 확인한다.

- DB 서버 재시작
- DB 엔진 설치 계정에 폴더 생성
- DB 사용자로 SCP 계정 생성 (권고사항)
- DB 서버의 DBA 권한 계정
- DB 엔진 설치 계정의 . profile 파일 수정

8.11 라이브러리 설정

$SYBASE/$SYBASE_ASE/lib 디렉터리에 library 파일의 symbolic link 파일을 생성한다.

```
$> cd $SYBASE/$SYBASE_ASE/lib
$> ln -s $DA_INST_HOME/libcis_cc-3.3.{so|a|sl}  libcis_cc-3.3.{so|a|sl}
$> ln -s $DA_INST_HOME/libcis_ce-3.3.{so|a|sl}  libcis_ce-3.3.{so|a|sl}
$> ln -s $DA_INST_HOME/libdamocm-4.0.{so|a|sl} libdamocm-4.0.{so|a|sl}
$> ln -s $DA_INST_HOME/liblogw-0.2.{so|a|sl} liblogw-0.2.{so|a|sl}
$> ln -s $DA_INST_HOME/libdamoscpdb.{so|a|sl}   libdamoscpdb.{so|a|sl}
```

8.12 sql 파일 생성

1. $DA_INST_HOME/sql 디렉터리에서 install_make.sh을 이용하여 설치할 sql 파일을 생성한다. D_INI 는 설정 파일(scpdb_agent.ini)의 경로다.

```
$> cd $DA_INST_HOME/sql
$> ./install_make.sh D_INI

[예제] D_INI 경로를 /home/dbms_api라고 가정한다.
$> ./install_make.sh /home/dbms_api
D_INI_PATH are replaced by /home/dbms_api
```

2. 다음 2개의 파일이 생성되는지 확인한다.

- 001.inner_function.sybase.sql
- 002.user_interface.sybase.sql

8.13 DB에서 DA를 사용할 계정 생성 (권장사항)

DB에서 DA를 사용할 'SCP' 계정를 생성하는 것을 권고한다. 'SCP' 계정 생성 시, CONNECT, RESOUCE, CREATE LIBRARY 권한을 부여한다.

```
1> create database scp on default = 20
2> go
CREATE DATABASE: allocating 9984 logical pages (19.5 megabytes) on disk 'master'
(10240 logical pages requested).
Database 'scp' is now online.
1> sp_addlogin scp, [password], scp
2> go
Password correctly set.
Account unlocked.
New login created.
(return status = 0)
1> grant role sa_role to scp
2> go
1>
```

8.14 제품 함수 설치

1. $DA_INST_HOME/sql 위치에서 DB의 'SCP' 계정으로 접속하여, LIBRARY 와 함수를 설치한다.

```
$>isql  -U scp  -P [password]  -S [ASE_NAME] -Dscp
$>isql -Uscp -P[password] -S[ASE_NAME] -Dscp
```

2. 특정 DB 사용자에게 함수 실행 권한을 부여한다. 모든 사용자에게 함수 실행 권한을 부여할 때는 003.grant_execute_functions.sql 파일을 실행한다.

```
$>isql  -U scp  -P [password]  -S [ASE_NAME] -Dscp
```

8.15 제품 설치 확인

DB 서버에서 암/복호화 함수를 호출하여 설치를 성공했는지 확인한다.

```
1> SELECT scp.dbo.ENC_STR( 'DAMO', 'abc')
2> go
--------------------------------------------------
5E41ACD673653158D7AE8C30CDA9627D3556E173

1> SELECT scp.dbo.DEC_STR( 'DAMO', scp.dbo.ENC_STR( 'DAMO', 'abc'))
2> go
--------------------------------------------------
abc
```

8.16 제품 운용

8.16.1 함수 설명

DA에서 제공되는 함수와 사용하는 방법에 대해서 설명한다.

8.16.1.1 파라미터 설명

8.16.1.1.1 I_KEY

I_KEY : 암호화/복호화 때 사용하는 암호화 키
- 설정 파일(scpdb_agent.ini)의 [KEYINFO] 중 ALIAS 값
- 아래 [설정 파일 예제]의 경우 ALIAS는 'KEY1'과 'KEY2'이다.
- ALIAS는 SCPS파일로 암호화 할 것인지, SG-KMS의 서비스 ID로 암호화 할 것인지 설정 가능하다.

[설정 파일 예제]
- 설정 파일(scpdb_agent.ini)의 [KEYINFO] 중 ALIAS 값
KEY1=AES256.SCPS
KEY2=ARIA256

[함수 사용 예제]
SELECT ENC_B64('KEY1', 'abc') FROM DUAL;

8.16.1.1.2 I_DATA

I_DATA : 평문(암호화 함수일 경우), 암호문(복호화 함수일 경우)

DA에서 제공하는 암호화 함수에서 성능 향상을 위해 라이브러리에서 정책 이름(KEYINFO ALIAS)을 바탕으로 Cash하여 동작한다. 정책 이름은 같은데 실제 키(대칭 키)가 다른 정책을 사용하면 Cash에서 이전 키로 복호화를 시도하여 오류가 발생한다. 이 상황을 해결하기 위해서는 데이터베이스 서버 재시작이 필요하다.

표 8-8 DA 함수 (SYBASE)

함수 명	입력		출력
ENC_STR	I_KEY	IN 문자열,	Hex String 암호문
	I_DATA	IN 문자열 (평문)	
ENC_B64	I_KEY	IN 문자열,	Base64 Encording

				암호문
	I_DATA	IN 문자열 (평문)		
DEC_STR	I_KEY	IN 문자열,		평문
	I_DATA	IN 문자열 (Hex String 암호문)		
DEC_B64	I_KEY	IN 문자열,		평문
	I_DATA	IN 문자열 (base64 String 암호문)		
INDEX_STR	I_KEY	IN 문자열,		Hex String 암호문
	I_DATA	IN 문자열 (평문),		
	I_TYPE	IN 문자열 '' or 'IX '(Plug-IN 연동 시 사용)		
DEC_INDEX_STR	I_KEY	IN 문자열,		Hex String 암호문 입력받아 OPE 데이터
	I_DATA	IN 문자열 (암호문),		
	I_TYPE	IN 문자열 '' or 'IX '(Plug-IN 연동 시 사용)		
DEC_INDEX_B64	I_KEY	IN 문자열,		Base64 Encording 암호문 입력받아 OPE 데이터
	I_DATA	IN 문자열 (암호문),		
	I_TYPE	IN 문자열 '' or 'IX '(Plug-IN 연동 시 사용)		
HASH_STR	I_ALOG	IN 숫자,	SHA1 =70	Hex String 해쉬 암호문
			SHA256 =71	
			SHA384 =72	
			SHA512 =73	
			HAS160 =74	
	I_DATA	IN 문자열		
HASH_B64	I_ALOG	IN 숫자,	SHA1 =70	Base64 String 해쉬 암호문
			SHA256 =71	
			SHA384 =72	
			SHA512 =73	
			HAS160 =74	
	I_DATA	IN 문자열		
HEXTOB64	I_DATA	IN 문자열 (Hex String 암호문)		base64 Encording 암호문
B64TOHEX	I_DATA	IN 문자열		Hex String 암호문

		(base64 Encording 암호문)	
CONFIG_REINIT			성공시 'SUCCESS', 그 외 에러

8.16.2 함수 호출 예제

1. ENC_STR

```
1> SELECT dbo.ENC_STR('KEY1', 'abc')
2> go
```

2. ENC_B64

```
1> SELECT dbo.ENC_B64('KEY1', 'abc')
2> go
```

3. DEC_STR

```
1> SELECT dbo.DEC_STR('KEY1', 'abc')
2> go
```

4. DEC_B64

```
1> SELECT dbo.DEC_B64('KEY1', 'abc')
2> go
```

5. INDEX_STR

```
DP 제품에 연동 하지 않을 경우
1> SELECT dbo.INDEX_STR('KEY1', 'abc', '')
2> go

DP 제품에 연동 할 경우
1> SELECT dbo.INDEX_STR('KEY1', 'abc', 'IX')
2> go
```

6. DEC_INDEX_STR, DEC_INDEX_B64

```
DP 제품에 연동 하지 않을 경우
1> SELECT dbo.DEC_INDEX_STR('KEY1', dbo.ENC_STR('KEY1', 'abc'), '')
2> go
1> SELECT dbo.DEC_INDEX_B64('KEY1', dbo.ENC_B64('KEY1', 'abc'), '')
```

```
2> go

DP 제품에 연동 할 경우
1> SELECT dbo.DEC_INDEX_STR('KEY1', dbo.ENC_STR('KEY1', 'abc'), 'IX')
2> go
1> SELECT dbo.DEC_INDEX_B64('KEY1', dbo.ENC_B64('KEY1', 'abc'), 'IX')
2> go
```

7. HASH_STR

```
1> SELECT dbo.HASH_STR( 71, 'abc' )
2> go
```

8. HASH_B64

```
1> SELECT dbo.HASH_B64( 71, 'abc' )
2> go
```

9. HEXTOB64

```
1> SELECT dbo.HEXTOB64('A305378D8F974F1C1537ED7CB0CB959245D1AC31')
2> go
```

10. B64TOHEX

```
1> SELECT dbo.B64TOHEX('owU3jY+XTxwVN+18sMuVkkXRrDE=')
2> go
```

11. CONFIG_REINIT

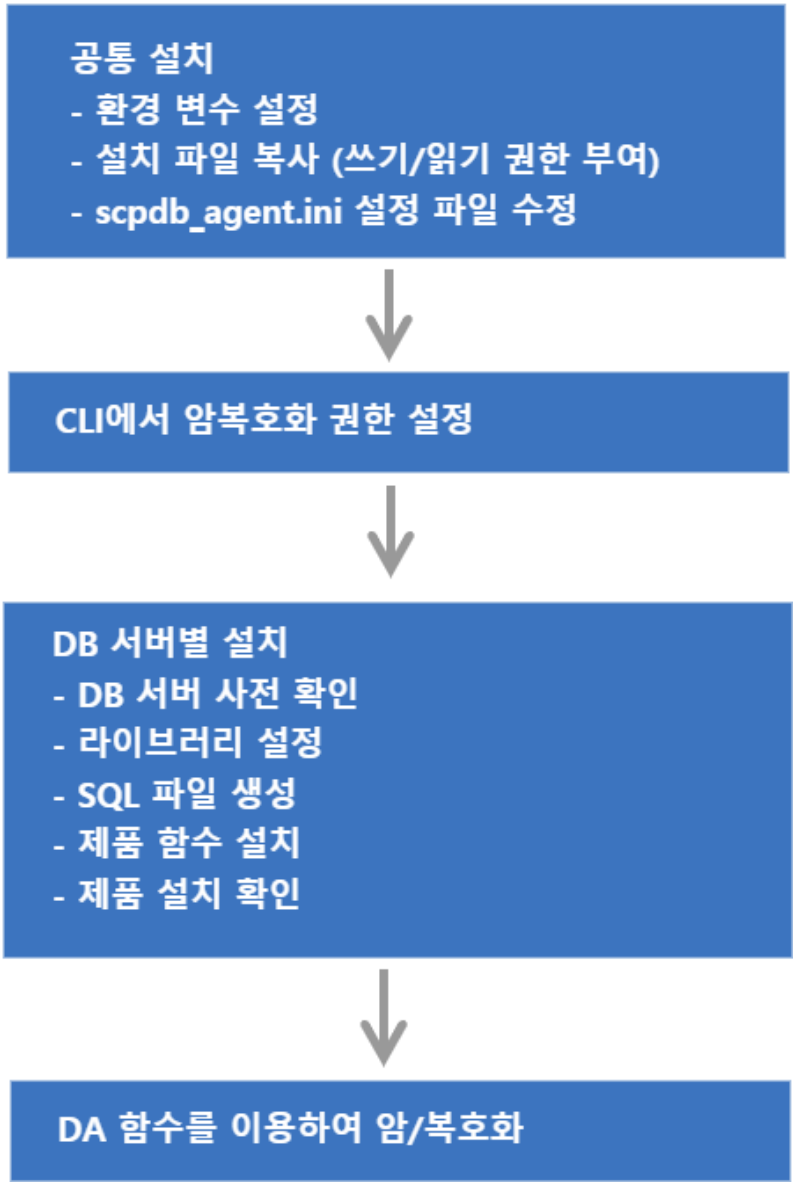
```
1> SELECT dbo.CONFIG_REINIT()
2> go
```

9.

SYBASE IQ

DA는 DB 서버의 종류에 따라서 설치 및 운용 방법이 각각 다르다.

그림 9-1 설치 개념도



9.1 지원 운영체제 및 DB서버

DA에서 지원하는 운영체제 및 DB서버는 아래와 같다. 단, 특정 환경은 지원되지 않을 수도 있으므로, 제품 설치 전에 상세한 지원 가능 여부는 펜타시큐리티시스템으로 문의한다.

표 9-1 제품이 지원하는 운영체제 및 DB 서버 정보

구분	설명
운영체제	Windows, AIX, HP IA, HP PA-RISC, Linux, SUN, TRU64
DB 서버	ORACLE 8i ~, SQL Server 2012 ~, DB2 9.x ~, Tibero 4 SP 1 ~, MySQL 5.x ~, MariaDB 5.5, 10.0, 10.1, Cache DB 2009.1 ~, Informix IDS 9.x ~ Sybase ASE 15.7 SP61 ~, Sybase IQ 15.4 ~, CUBRID 2008 R1.3 ~ , PostgreSQL 9.4 ~

9.2 설치 파일의 구성 확인

DA의 설치 파일은 아래와 같은 명칭으로 압축 파일(zip) 형태로 제공된다.

- DA 설치파일: Install_DAmo_DA_v{버전}.zip
  - 압축을 해제 한 뒤, 설치 대상 DB 서버, OS 및 bit에 맞는 설치 파일을 준비한다.

설치 파일에 제품을 사용할 수 있는 '라이선스'는 포함되어 있지 않다.
펜타시큐리티시스템에 문의하여 '라이선스' 파일은 별도로 준비한다.

표 9-2 설치 파일(Install_DAmo_DA_v{버전}.zip)을 압축 해제 시, 디렉터리의 구성

구성	설명
_SampleScpsFiles	SG-KMS 연동 없이 암복호화를 테스트할 수 있는 테스트 키 파일
_TestAgentKeyPair	CLI에서 사용할 수 있는 테스트 키 쌍
Altibase	DA-ALT 제품의 설치 바이너리 폴더
Cache	DA-CDB 제품의 설치 바이너리 폴더
Cubrid	DA-CUB 제품의 설치 바이너리 폴더
DB2	DA-DB2 제품의 설치 바이너리 폴더
Informix	DA-IFX 제품의 설치 바이너리 폴더
MySQL	DA-MYQ 제품의 설치 바이너리 폴더
Oracle	DA-ORA 제품의 설치 바이너리 폴더
Postgres	DA-PGS 제품의 설치 바이너리 폴더
SQL Server	DA-MSQ 제품의 설치 바이너리 폴더

구성	설명
SybaseASE	DA-SYB 제품의 설치 바이너리 폴더
SybaseIQ	DA-SIQ 제품의 설치 바이너리 폴더
Tibero	DA-TIB 제품의 설치 바이너리 폴더

9.2.1 DB 서버 및 운영체제 별 SQL파일 구성

각 DB 서버 및 운영체제 별 SQL파일 구성은 다음과 같다. 다음 장에서 각 DB별로 SQL파일 설치 방법을 설명한다.

표 9-3 DB 서버 및 운영체제 별 SQL파일

DB 서버 종류	Linux 일 경우	Windows 일 경우
Oracle	000.da_user.pkg 000.da_user.sql 001.inner_function.ora(c 버전 설치 시) 002.user_interface.ora(JAVA 버전 설치 시) 002.user_interface_java.ora 003.grant_execute_functions.sql(JAVA 버전 설치 시) 005.securej_privilege.sql install_make.sh	000.da_user.pkg 000.da_user.sql 001.inner_function.ora(c 버전 설치 시) 002.user_interface.ora(JAVA 버전 설치 시) 002.user_interface_java.ora 003.grant_execute_functions.sql(JAVA 버전 설치 시) 005.securej_privilege.sql 009.da_test.sql
MYSQL	001.inner_function.mys 002.user_interface. mys 003.grant_execute_functions.sql 009.da_test.sql install_make.sh	001.inner_function.mys 002.user_interface.mys 003.grant_execute_functions.sql
TIBERO	001.inner_function.tbs(c 버전 설치 시) 001.inner_function_java.tbs(JAVA 버전 설치 시) 002.user_interface.tbs(c 버전 설치 시) 002.user_interface_java.tbs(JAVA 버전 설치 시) 003.grant_execute_functions.sql install_make.sh	001.inner_function_java.tbs 002.user_interface_java.tbs 003.grant_execute_functions.sql
INFORMIX	001.inner_function.ifx 002.user_interface. ifx 003.grant_execute_functions.sql 009.da_test.sql install_make.sh	해당 없음
POSTGRESQL	001.inner_function.post 002.user_interface.post 003.grant_execute_functions.post	해당 없음

DB 서버 종류	Linux 일 경우	Windows 일 경우
DB2	001.inner_function.db2 002.user_interface.db2 003.grant_execute_functions.sql install_make.sh	001.inner_function.db2 002.user_interface.db2 003.grant_execute_functions.sql
CUBRID	001.inner_function.sql 002.user_interface.sql	001.inner_function.sql 002.user_interface.sql
SYBASE	001.inner_function.sybase 002.user_interface.sybase 003.grant_execute_functions.sql install_make.sh	해당 없음
SYBASE IQ	001.inner_function.sybiq 002.user_interface. sybiq 003.grant_execute_functions.sql install_make.sh	해당 없음
SQL Server	해당 없음	001.inner_function.sql 002.user_interface. sql 003.grant_execute_functions.sql install_make.bat
Cache DB	SCP.xml install_make.sh	SCP.xml
Altibase	해당 없음	000.da_user.pkg 000.da_user.sql 001.inner_function.sql 002.user_interface. sql install_make.bat

9.2.2 공통 설치 파일 구성

환경에 관계 없이 공통적으로 사용하는 설치 파일은 아래와 같다.

표 9-4 공통 설치 파일 (sql파일 제외)

파일 분류	파일 명	파일 용도
Library 파일	libdamoscpdb.{so|a|sl|dll}	DA 메인 라이브러리. 주로 DBMS External Interface를 담당
	libdamocm-4.0.{so|a|sl|dll}	공통 모듈 라이브러리
	liblogw-0.2.{so|a|sl|dll}	로그를 기록하는 라이브러리
	libcis_cc-3.3.{so|a|sl|dll}	암호화, 복호화 기능을 제공하는 라이브러리
	libcis_ce-3.3.{so|a|sl|dll}	암호화, 복호화를 제외한 부가적인 기능을 제공하는 라이브러리(예: Base64, 인증서 관리, 특성 유지 암호화 등)

파일 분류	파일 명	파일 용도
설정 파일	scpdb_agent.ini	DA 구동시 실행에 필요한 설정정보를 참조
License 파일	damo_lic.cer	DA 구동 시 제품의 유효성을 검증하는데 사용
Agent key 파일	damo_agt_site.cer	SG-KMS 연동, CLI 프로그램에서 사용하는 인증서 쌍
	damo_agt.cer	
	damo_agt.key	
접근제어 파일	acl_cli 파일	DB 의 USER 별로 암·복호 권한을 설정하는데에 사용
	privilege.damo	권한 파일
JAVA class 파일 (Oracle, Tibero, Cubrid 설치 가능)	ScpAgentException.class	예외 처리 Class
	ScpCryptData.class	암호화 복호화 Class
SQL 파일	아래 새로운 표에 DB별로 표기함	

Agent Key 파일은 SG-KMS 연동에 필요한 키 발급를 참고하여 발급 받는다.

DA-PGS(PostgreSQL)의 경우, DB 서버 버전에 따라 libdamoscpdb.so 라이브러리 선택

- libdamoscpdb94.so (Postgres 9.4)
- libdamoscpdb95.so (Postgres 9.5)
- libdamoscpdb95AS.so (EDB Postgres 9.5)
- libdamoscpdb96AS.so (EDB Postgres 9.6)
- libdamoscpdb10.so (Postgres 10)

9.3 환경변수 설정

DA를 설치할 운영체제에 환경변수 DA_INST_HOME를 설정한다. 이 매뉴얼에서는 제품 설치 경로를 아래와 같이 가정하여 설명한다.

- Linux 환경일 경우: /home/dbms_api
- Windows 환경일 경우: E:₩dbms_api

주의) DA_INST_HOME 설정 시, 주의 사항

- 리눅스의 경우 "/root" 디렉토리로 설정을 권장하지 않는다.
- 윈도우의 경우 "바탕화면"으로 설정을 권장하지 않는다.

위의 경로로 설정할 경우 접근 권한 등의 이유로 문제가 발생할 가능성이 존재한다.

9.3.1 환경변수 설정 - Linux 환경일 경우

DA_INST_HOME 환경변수에 DA의 설치 디렉터리를 설정한다.

```
.profile을 사용하는 경우
export DA_INST_HOME=/home/dbms_api


.cshrc를 사용하는 경우
setenv DA_INST_HOME=/home/dbms_api

export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$DA_INST_HOME
```

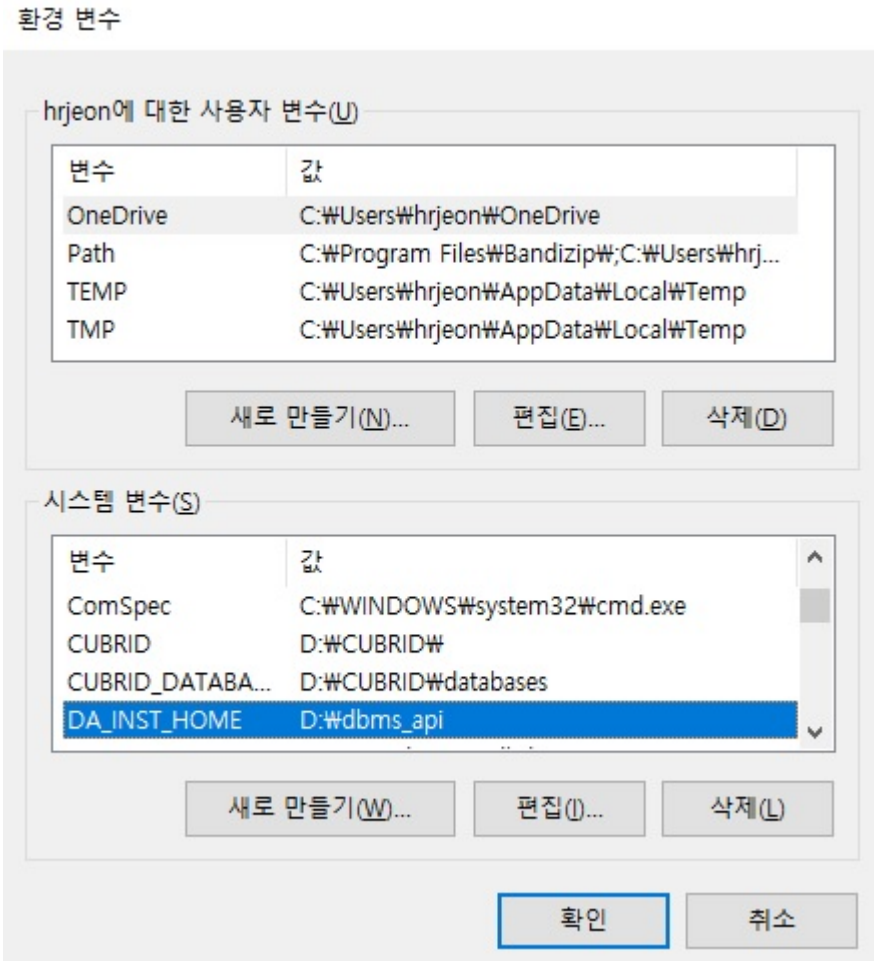
표 9-5 운영 체제별 라이브러리 PATH 명칭

운영 체제	라이브러리 PATH 명칭
HP_UX	SHLIB_PATH
AIX	LIBPATH
LINUX, SUN	LD_LIBRARY_PATH

9.3.2 환경변수 설정 - Windows 환경일 경우

[탐색기->내컴퓨터->등록정보->고급 탭->환경변수] 를 선택하여 나타나는 환경변수 설정 다이얼로그에서 시스템변수 DA_INST_HOME 변수와 값을 추가한다.

그림 9-2 DA_INST_HOME



9.4 DA의 설치 파일 복사

$DA_INST_HOME 디렉터리에 위 [설치 파일의 구성 확인]에서 나열된 파일들을 복사한다.

9.5 라이선스 파일 설정

$DA_INST_HOME 디렉토리에 라이선스 파일(damo_lic.cer)을 복사한다.


라이선스 파일명이 damo_lic.cer가 아닌 다른 이름으로 저장되어 있다면 변경해야 한다.

9.6 설치 파일의 접근 권한 부여 (Linux 설치 시)

Linux 사용자 계정에 DA 설치 파일(라이브러리, 실행 파일, 디렉토리)의 접근 권한을 부여한다. Windows에서 제품을 설치 할 경우, 이 과정은 생략한다.

```
$> cd $DA_INST_HOME
$> chmod 755 lib* acl_cli sql/install_make.sh
```

9.7 (SG-KMS 연동 시) SG-KMS에서 DA 정보 등록 및 연동에 필요한 키 내보내기

DA는 데이터를 암·복호화하기 위해 '암호화 키'가 필요하다. '암호화 키'를 얻기 위해서는 SG-KMS를 연동하거나 SCPS라는 암호화 키 파일을 이용할 수 있는데,
다음 설명은 SG-KMS 연동을 위해 SG-KMS에서 DA 정보를 등록하고 연동에 필요한 키를 내보내는 방법에 대해서 설명한다.

9.7.1 사전 준비

9.7.1.1 지원 SG-KMS 버전

DA와 연동 가능한 SG-KMS 버전은 다음과 같다.

표 9-6 연동 가능한 SG-KMS 버전

SG-KMS Major version	SG-KMS Minor version
SG-KMS v3.0	v3.0.9.0 이상 연동 가능
SG-KMS v4.0	v4.0.104.5 이상 연동 가능

SG-KMS v2.3 연동은 미지원한다.

9.7.1.2 연동 전 점검 항목

SG-KMS 연동을 위해서는 다음과 같이 사전 준비가 필요하다.

- SG-KMS 관리도구
- SG-KMS 관리도구에 접속할 수 있는 ID와 비밀번호
- SG-KMS 매뉴얼

이 매뉴얼에서는 SG-KMS의 키 종류 및 SG-KMS 관리도구 사용 방법에 대한 내용은 다루지 않으므로 **[SG-KMS 매뉴얼]**을 참고한다.

9.7.1.3 SG-KMS 연동에 필요한 키 발급

DA와 SG-KMS의 연동을 위해서는 먼저 SG-KMS에 DA를 Agent로 등록해야한다. 등록하는 절차는 SG-KMS 사용자설명서의 '대칭키를 사용하는 D'Amo Agent 등록 안내서'를 참고한다.

[D'Amo Agent 키] 파일, [Agent ID(CN)]와 [서비스 ID]정보는 DA와 SG-KMS 연동을 위한 설정 과정에서 필요하다. SG-KMS 관리자는 DA 설치 엔지니어에게 안전하게 전달한다.

SG-KMS 관리자에게 받은 D'Amo Agent의 인증서 및 키 파일 명을 다음과 같이 변경후 $DA_INST_HOME/key 디렉터리에 복사한다.

표 9-7 SG-KMS 연동에 필요한 키 목록

키 구분	생성된 키 파일명	변경할 키 파일명
사이트 키	damo-site_{발행기관/부서명}-SITE_V3.cer	damo_agt_site.cer
Agent 키	damo-scp_SITE_V3-{Agent 이름}.cer	damo_agt.cer
	damo-scp_SITE_V3-{Agent 이름}.key	damo_agt.key
	damo-scp_SITE_V3-{Agent 이름}.spin	damo_agt.spin

Agent 키 경로 지정, 키 이름 변경은 반드시 필요한 작업은 아니지만 관리를 위해 변경하는 것을 권장한다.

9.8 설정 파일(scpdb_agent.ini) 수정

DA의 운용을 위해 사용하는 scpdb_agent.ini 설정 파일 수정 방법에 대해 설명한다. DA는 SG-KMS를 이용하거나 SCPS 파일을 이용하여 암호화 키를 얻기 때문에 고객의 환경에 맞게 설정해야 한다.

$DA_INST_HOME 디렉터리에 있는 scpdb_agent.ini 설정 파일에서 아래 항목의 값을 수정한다.

1. 설정 파일의 [KEYINFO] 항목 - [KEY1]에 암호화 키 정보를 입력한다.

```
[KEYINFO]
KEY1=암복호화 하려는 암호화 키 정보를 입력한다.
//키 정보는 다음과 같은 값을 입력할 수 있다.
///ServiceID: SG-KMS에 생성한 서비스 ID를 입력한다.
///SCP_FilePath: SG-KMS에서 서비스 내보내기를 통해 발급한 SCPS 파일의 절대 경로 및 파일명, 확장자를 입력한다.

//*예제 - Windows 경우
KEY1=DA_AES256
KEY2=C:\DA\Policy\S_AES128.SCPS\DA_AES256.scps
KEY3=DA_AES256,C:\DA\Policy\S_AES128.SCPS\DA_AES256.scps

//*예제 - Linux 또는 Unix 경우
KEY1=DA_AES256
KEY2=/home/dbms_api/key/DA_AES256.scps
KEY3=DA_AES256,/home/dbms_api/key/DA_AES256.scps
```

ServiceID를 입력 할 경우 SG-KMS와 통신을 통해 암호화 정책을 가져와 데이터 암복호화를 수행한다.

SCP_FilePath를 입력 할 경우 KMS와 통신하지 않고 서버의 SCP 파일을 통해 암호화 정책을 가져와 데이터 암복호화를 수행한다.

ServiceID와 SCPS 파일명을 동시에 입력 시, SG-KMS와 네트워크 연결 실패 하면 SCPS 파일을 이용하여 암호화 한다.
ServiceID와 SCPS 파일명 사이에 공백이 있으면 SCPS 파일을 읽을 수 없다. 따라서 예제와 같이 띄어쓰기를 하지 않고 ServiceID와 SCPS 파일 경로를 붙여서 입력한다.

2. 설정 파일의 [Server], [Server2] 항목을 수정한다.

```
[Server]
ServerIP: SG-KMS의 IP를 입력한다.
ServerPort: SG-KMS의 포트번호를 입력한다. 기본값은 2525이다.

//*예제 - 모든 OS 공통
```

```
ServerIP=192.168.22.25
ServerPort=2525
```

```
[Server2]
ServerIP: 이중화를 위한 2번 SG-KMS의 IP를 입력한다.
ServerPort: 이중화를 위한 2번 SG-KMS의 포트번호를 입력한다. 기본값은 2525이다.

//*예제 - 모든 OS 공통
ServerIP=192.168.22.26
ServerPort=2525
```

ServerIP는 최소 1개 ~ 최대 10개까지 등록이 가능하다.

3. 설정 파일의 [AGENT] 항목을 설정한다.

```
[AGENT]
AgentID=SG-KMS 관리도구에서 설정한 D'Amo Agent의 Agent ID
LogDir=로그가 저장될 디렉터리 위치
LogLevel=로그가 남는 수준
SiteCertFilePath=SG-KMS 장비에서 설정한 해당 장비의 사이트 공개키(.cer)
CertFilePath=SG-KMS 관리도구에서 설정한 D'Amo Agent의 공개키(.cer)
KeyFilePath=SG-KMS 관리도구에서 설정한 D'Amo Agent의 비공개키(.key)
SPIN=SG-KMS 관리도구에서 설정한 D'Amo Agent의 SPIN으로, damo-scp_SITE_V3-[Agent이름].spin 파일의 값

//[예제]
AgentID=DA
LogDir=/home/dbms_api/log
LogLevel=4
SiteCertFilePath=/home/dbms_api/key/damo_agt_site.cer
CertFilePath=/home/dbms_api/key/damo_agt.cer
KeyFilePath=/home/dbms_api/key/damo_agt.key
SPIN=XaMh1ylXUh123XUh
```

로그가 남는 수준(LogLevel)에는 아래 5가지 숫자 입력이 가능하며, 각 값의 설정은 다음과 같다.
- 0: 아무 로그도 남기지 않을 경우
- 2: 경고 로그를 파일에 기록
- 4: 에러 로그와 경고 로그를 파일에 기록

- 6: 정보 로그, 에러 로그, 경고 로그를 파일에 기록
- 8: 디버그, 정보 로그, 에러 로그, 경고 로그를 파일에 기록

제품 운영 중 scpdb_agent.ini 파일을 수정하면 CONFIG_REINIT() 함수를 호출해야 변경된 내용이 적용된다.

9.9 CLI에서 권한 설정

$DA_INST_HOME 디렉터리에서 acl_cli 파일을 실행하여 USER 단위로 암/복호화 권한을 설정한다. USER 는 DB의 소유자명이고, KEY는 설정 파일(scpdb_agent.ini)의 [KEYINFO] 중 ALIAS 값(예: KEY1)이다.

CLI 명령어를 자세히 보려면 help 명령어를 실행한다.

```
CLI 실행 방법
$> cd $DA_INST_HOME
$> ./acl_cli  - start
Enter the PIN of CLI-key. : damo_agt.key 의 비밀번호
```

```
권한 추가할 경우
D'Amo > SET PRIV ENC [USER]"[KEY]"1"1
D'Amo > SAVE ALL
D'Amo > SHOW ALL

예제)
D'Amo > SET PRIV ENC SCOTT"KEY1"1"1
D'Amo > SET PRIV ENC SCOTT"KEY2"1"1
```

scpdb_agent.ini 설정 파일이 아래 예제와 같을 경우, CLI에서 권한 설정할 때 입력해야 하는 2번째 [KEY] 인자 값에는 KEY1을 입력해야 한다. (※ARIA256을 입력하는 것이 아님)

#scpdb_agent.ini 설정 파일 예제
[KEYINFO]
KEY1=ARIA256

```
권한 삭제할 경우
D'Amo > DEL PRIV ENC [USER]"[KEY]
D'Amo > SAVE ALL
D'Amo > SHOW ALL
```

CLI 에서 권한을 추가하거나 삭제 한 경우 반드시 SAVE ALL 명령어를 실행하며 SHOW ALL 명령어를 이용하여 적용 여부를 확인 한다.

9.10 DB 서버 사전 확인

DA를 SYBASE IQ 환경에서 설치 하기 전에 다음과 같은 사항들을 확인한다.

- DB 엔진 설치 계정에 폴더 생성
- DB 사용자로 SCP 계정 생성 (권고사항)
- DB 서버의 DBA 권한 계정
- DB 엔진 설치 계정의 . profile 파일 수정

9.11 라이브러리 설정

$IQDIR15/lib64 디렉터리에 DA의 library 파일을 symbolic link 로 생성한다.

```
$> cd $IQDIR15/lib64
$> ln -s $DA_INST_HOME/libcis_cc-3.3.{so|a|sl} libcis_cc-3.3.{so|a|sl}
$> ln -s $DA_INST_HOME/libcis_ce-3.3.{so|a|sl} libcis_ce-3.3.{so|a|sl}
$> ln -s $DA_INST_HOME/liblogw-0.2.{so|a|sl} liblogw-0.2.{so|a|sl}
$> ln -s $DA_INST_HOME/libdamocm-4.0.{so|a|sl} libdamocm-4.0.{so|a|sl}
$> ln -s $DA_INST_HOME/libdamoscpdb.{so|a|sl} libdamoscpdb.so
```

libdamoscpdb.{so|a|sl} 파일의 symbolic link 파일을 생성할 때는 항상 .so 확장자 파일로 생성한다. 나머지 4개의 파일은 기존 확장자를 유지한다.

9.12 sql 파일 생성

1. $DA_INST_HOME/sql 디렉터리에서 install_make.sh을 이용하여 설치할 sql 파일을 생성한다. D_INI 는 설정 파일(scpdb_agent.ini)의 경로다.

```
$> cd $DA_INST_HOME/sql
$> ./install_make.sh D_INI

[예제] D_INI를 /home/dbms_api로 가정한다.
$> ./install_make.sh /home/dbms_api
D_INI_PATH is replaced by /home/dbms_api
$>
```

2. 아래 2개의 .sql 파일이 생성되었는지 확인한다.

- 001.inner_function.sybiq.sql
- 002.user_interface.sybiq.sql

9.13 제품 함수 설치

1. $DA_INST_HOME/sql 위치에서 DB 접속 후 설치를 원하는 DB계정에 함수를 설치한다.

```
(DBA)> READ 001.inner_function.sybiq.sql
Execution time: 0.193 seconds

(DBA)> READ 002.user_interface.sybiq.sql
Execution time: 0.226 seconds

(DBA)>
```

2. 특정 DB 사용자에게 함수 실행 권한을 부여한다. 모든 사용자에게 함수 실행 권한을 부여할 때는 003.grant_execute_functions.sql 파일을 실행한다.

```
(DBA)> READ 003.grant_execute_functions.sql
Execution time: 0.04 seconds

(DBA)>
```

9.14 제품 설치 확인

DB 서버에서 암/복호화 함수를 호출하여 설치를 성공했는지 확인한다.

```
(DBA)> SELECT ENC_STR('KEY1', 'abc')
ENC_STR('KEY1','abc')
--------------------------------------------------
E6878572B3287A049906A8CA57F0207C
(1 rows)
Execution time: 0.01 seconds

(DBA)> SELECT DEC_STR('KEY1', ENC_STR('KEY1', 'abc'))
DEC_STR('KEY1', ENC_STR('KEY1', 'abc'))
--------------------------------------------------
abc
(1 rows)
Execution time: 0.029 seconds

(DBA)>
```

9.15 제품 운용

9.15.1 함수 설명

DA에서 제공되는 함수와 사용하는 방법에 대해서 설명한다.

9.15.1.1 파라미터 설명

9.15.1.1.1 I_KEY

I_KEY : 암호화/복호화 때 사용하는 암호화 키
- 설정 파일(scpdb_agent.ini)의 [KEYINFO] 중 ALIAS 값
- 아래 [설정 파일 예제]의 경우 ALIAS는 'KEY1'과 'KEY2'이다.
- ALIAS는 SCPS파일로 암호화 할 것인지, SG-KMS의 서비스 ID로 암호화 할 것인지 설정 가능하다.

[설정 파일 예제]
- 설정 파일(scpdb_agent.ini)의 [KEYINFO] 중 ALIAS 값
KEY1=AES256.SCPS
KEY2=ARIA256

[함수 사용 예제]
SELECT ENC_B64('KEY1', 'abc') FROM DUAL;

9.15.1.1.2 I_DATA

I_DATA : 평문(암호화 함수일 경우), 암호문(복호화 함수일 경우)

DA에서 제공하는 암호화 함수에서 성능 향상을 위해 라이브러리에서 정책 이름(KEYINFO ALIAS)을 바탕으로 Cash하여 동작한다. 정책 이름은 같은데 실제 키(대칭 키)가 다른 정책을 사용하면 Cash에서 이전 키로 복호화를 시도하여 오류가 발생한다. 이 상황을 해결하기 위해서는 데이터베이스 서버 재시작이 필요하다.

표 9-8 DA 함수 (SYBASE IQ)

함수 명	입력		출력
ENC_STR	I_KEY	IN 문자열,	Hex String 암호문
	I_DATA	IN 문자열 (평문)	
ENC_B64	I_KEY	IN 문자열,	Base64 Encording 암호문
	I_DATA	IN 문자열 (평문)	
DEC_STR	I_KEY	IN 문자열,	평문
	I_DATA	IN 문자열 (Hex String 암호문)	
DEC_B64	I_KEY	IN 문자열,	평문
	I_DATA	IN 문자열 (base64 String 암호문)	
INDEX_STR	I_KEY	IN 문자열,	Hex String 암호문
	I_DATA	IN 문자열 (평문),	
	I_TYPE	IN 문자열 '' or 'IX '(Plug-IN 연동 시 사용)	

DEC_INDEX_STR	I_KEY	IN 문자열,		Hex String 암호문 입력받아 OPE 데이터
	I_DATA	IN 문자열 (암호문),		
	I_TYPE	IN 문자열 '' or 'IX '(Plug-IN 연동 시 사용)		
DEC_INDEX_B64	I_KEY	IN 문자열,		Base64 Encording 암호문 입력받아 OPE 데이터
	I_DATA	IN 문자열 (암호문),		
	I_TYPE	IN 문자열 '' or 'IX '(Plug-IN 연동 시 사용)		
HASH_STR	I_ALOG	IN 숫자,	SHA1 =70	Hex String 해쉬 암호문
			SHA256 =71	
			SHA384 =72	
			SHA512 =73	
			HAS160 =74	
	I_DATA	IN 문자열		
HASH_B64	I_ALOG	IN 숫자,	SHA1 =70	Base64 String 해쉬 암호문
			SHA256 =71	
			SHA384 =72	
			SHA512 =73	
			HAS160 =74	
	I_DATA	IN 문자열		
HEXTOB64	I_DATA	IN 문자열 (Hex String 암호문)		base64 Encording 암호문
B64TOHEX	I_DATA	IN 문자열 (base64 Encording 암호문)		Hex String 암호문
CONFIG_REINIT				성공시 'SUCCESS', 그 외 에러

9.15.2 함수 호출 예제

1. ENC_STR

```
(DBA)> SELECT ENC_STR('KEY1', 'abc');
```

2. ENC_B64

```
(DBA)> SELECT ENC_B64('KEY1', 'abc');
```

3. DEC_STR

```
(DBA)> SELECT DEC_STR('KEY1', ENC_STR('KEY1', 'abc'));
```

4. DEC_B64

```
(DBA)> SELECT DEC_B64('KEY1', ENC_B64('KEY1', 'abc'));
```

5. INDEX_STR

```
DP 제품에 연동 하지 않을 경우
(DBA)> SELECT INDEX_STR('KEY1', 'abc', '');

DP 제품에 연동 할 경우
(DBA)> SELECT INDEX_STR('KEY1', 'abc', 'IX');
```

6. DEC_INDEX_STR, DEC_INDEX_B64

```
DP 제품에 연동 하지 않을 경우
(DBA)> SELECT DEC_INDEX_STR('KEY1', ENC_STR('KEY1', 'abc'), '');
(DBA)> SELECT DEC_INDEX_B64('KEY1', ENC_B64('KEY1', 'abc'), '');

DP 제품에 연동 할 경우
(DBA)> SELECT DEC_INDEX_STR('KEY1', ENC_STR('KEY1', 'abc'), 'IX');
(DBA)> SELECT DEC_INDEX_B64('KEY1', ENC_B64('KEY1', 'abc'), 'IX');
```

7. HASH_STR

```
(DBA)> SELECT HASH_STR( 71, 'abc' );
```

8. HASH_B64

```
(DBA)> SELECT HASH_B64( 71, 'abc' );
```

9. HEXTOB64

```
(DBA)> SELECT HEXTOB64('A305378D8F974F1C1537ED7CB0CB959245D1AC31');
```

10. B64TOHEX

```
(DBA)> SELECT B64TOHEX('owU3jY+XTxwVN+18sMuVkkXRrDE=');
```

11. CONFIG_REINIT

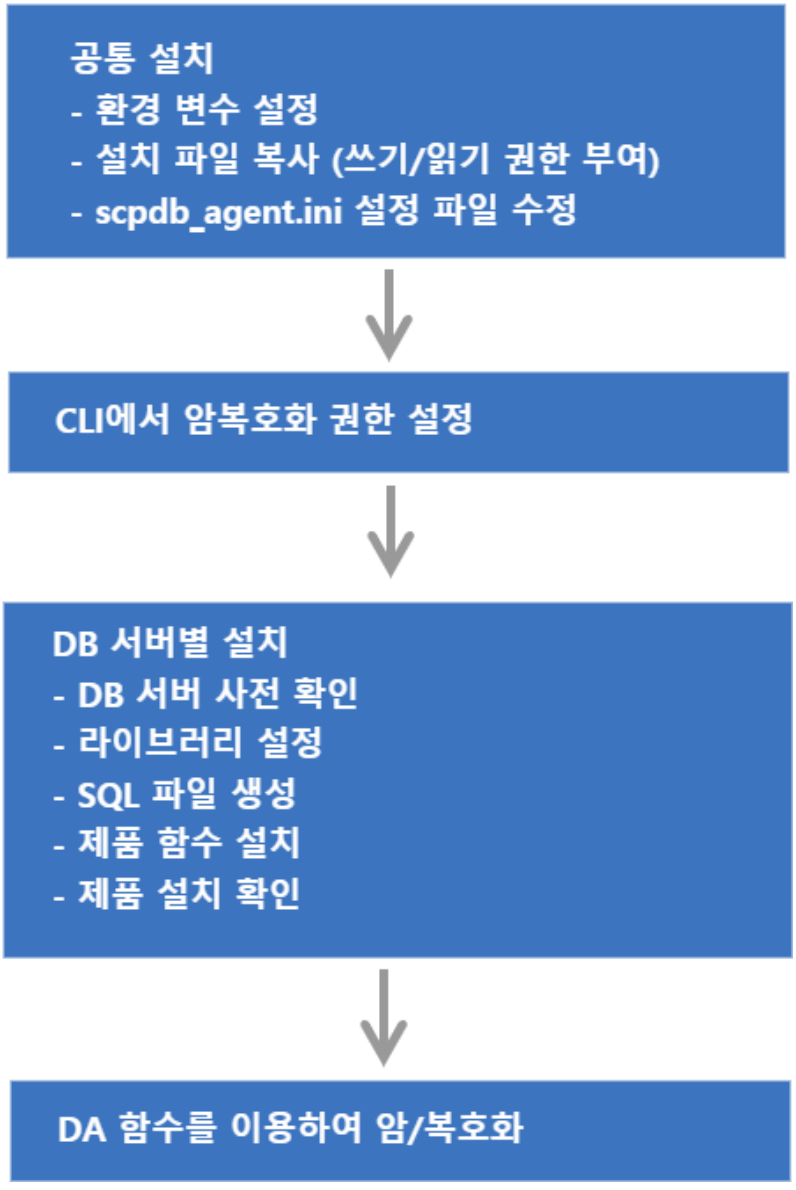
```
(DBA)> SELECT CONFIG_REINIT;
```

10.

SQL Server

DA는 DB 서버의 종류에 따라서 설치 및 운용 방법이 각각 다르다.

그림 10-1 설치 개념도



10.1 지원 운영체제 및 DB서버

DA에서 지원하는 운영체제 및 DB서버는 아래와 같다. 단, 특정 환경은 지원되지 않을 수도 있으므로, 제품 설치 전에 상세한 지원 가능 여부는 펜타시큐리티시스템으로 문의한다.

표 10-1 제품이 지원하는 운영체제 및 DB 서버 정보

구분	설명
운영체제	Windows, AIX, HP IA, HP PA-RISC, Linux, SUN, TRU64
DB 서버	ORACLE 8i ~, SQL Server 2012 ~, DB2 9.x ~, Tibero 4 SP 1 ~, MySQL 5.x ~, MariaDB 5.5, 10.0, 10.1, Cache DB 2009.1 ~, Informix IDS 9.x ~ Sybase ASE 15.7 SP61 ~, Sybase IQ 15.4 ~, CUBRID 2008 R1.3 ~ , PostgreSQL 9.4 ~

10.2 설치 파일의 구성 확인

DA의 설치 파일은 아래와 같은 명칭으로 압축 파일(zip) 형태로 제공된다.

- DA 설치파일: Install_DAmo_DA_v{버전}.zip
  - 압축을 해제 한 뒤, 설치 대상 DB 서버, OS 및 bit에 맞는 설치 파일을 준비한다.

설치 파일에 제품을 사용할 수 있는 '라이선스'는 포함되어 있지 않다.
펜타시큐리티시스템에 문의하여 '라이선스' 파일은 별도로 준비한다.

표 10-2 설치 파일(Install_DAmo_DA_v{버전}.zip)을 압축 해제 시, 디렉터리의 구성

구성	설명
_SampleScpsFiles	SG-KMS 연동 없이 암복호화를 테스트할 수 있는 테스트 키 파일
_TestAgentKeyPair	CLI에서 사용할 수 있는 테스트 키 쌍
Altibase	DA-ALT 제품의 설치 바이너리 폴더
Cache	DA-CDB 제품의 설치 바이너리 폴더
Cubrid	DA-CUB 제품의 설치 바이너리 폴더
DB2	DA-DB2 제품의 설치 바이너리 폴더
Informix	DA-IFX 제품의 설치 바이너리 폴더
MySQL	DA-MYQ 제품의 설치 바이너리 폴더
Oracle	DA-ORA 제품의 설치 바이너리 폴더
Postgres	DA-PGS 제품의 설치 바이너리 폴더
SQL Server	DA-MSQ 제품의 설치 바이너리 폴더

구성	설명
SybaseASE	DA-SYB 제품의 설치 바이너리 폴더
SybaseIQ	DA-SIQ 제품의 설치 바이너리 폴더
Tibero	DA-TIB 제품의 설치 바이너리 폴더

10.2.1 DB 서버 및 운영체제 별 SQL파일 구성

각 DB 서버 및 운영체제 별 SQL파일 구성은 다음과 같다. 다음 장에서 각 DB별로 SQL파일 설치 방법을 설명한다.

표 10-3 DB 서버 및 운영체제 별 SQL파일

DB 서버 종류	Linux 일 경우	Windows 일 경우
Oracle	000.da_user.pkg 000.da_user.sql 001.inner_function.ora(c 버전 설치 시) 002.user_interface.ora(JAVA 버전 설치 시) 002.user_interface_java.ora 003.grant_execute_functions.sql(JAVA 버전 설치 시) 005.securej_privilege.sql install_make.sh	000.da_user.pkg 000.da_user.sql 001.inner_function.ora(c 버전 설치 시) 002.user_interface.ora(JAVA 버전 설치 시) 002.user_interface_java.ora 003.grant_execute_functions.sql(JAVA 버전 설치 시) 005.securej_privilege.sql 009.da_test.sql
MYSQL	001.inner_function.mys 002.user_interface. mys 003.grant_execute_functions.sql 009.da_test.sql install_make.sh	001.inner_function.mys 002.user_interface.mys 003.grant_execute_functions.sql
TIBERO	001.inner_function.tbs(c 버전 설치 시) 001.inner_function_java.tbs(JAVA 버전 설치 시) 002.user_interface.tbs(c 버전 설치 시) 002.user_interface_java.tbs(JAVA 버전 설치 시) 003.grant_execute_functions.sql install_make.sh	001.inner_function_java.tbs 002.user_interface_java.tbs 003.grant_execute_functions.sql
INFORMIX	001.inner_function.ifx 002.user_interface. ifx 003.grant_execute_functions.sql 009.da_test.sql install_make.sh	해당 없음
POSTGRESQL	001.inner_function.post 002.user_interface.post 003.grant_execute_functions.post	해당 없음

DB 서버 종류	Linux 일 경우	Windows 일 경우
DB2	001.inner_function.db2 002.user_interface.db2 003.grant_execute_functions.sql install_make.sh	001.inner_function.db2 002.user_interface.db2 003.grant_execute_functions.sql
CUBRID	001.inner_function.sql 002.user_interface.sql	001.inner_function.sql 002.user_interface.sql
SYBASE	001.inner_function.sybase 002.user_interface.sybase 003.grant_execute_functions.sql install_make.sh	해당 없음
SYBASE IQ	001.inner_function.sybiq 002.user_interface. sybiq 003.grant_execute_functions.sql install_make.sh	해당 없음
SQL Server	해당 없음	001.inner_function.sql 002.user_interface. sql 003.grant_execute_functions.sql install_make.bat
Cache DB	SCP.xml install_make.sh	SCP.xml
Altibase	해당 없음	000.da_user.pkg 000.da_user.sql 001.inner_function.sql 002.user_interface. sql install_make.bat

10.2.2 공통 설치 파일 구성

환경에 관계 없이 공통적으로 사용하는 설치 파일은 아래와 같다.

표 10-4 공통 설치 파일 (sql파일 제외)

파일 분류	파일 명	파일 용도
Library 파일	libdamoscpdb.{so|a|sl|dll}	DA 메인 라이브러리. 주로 DBMS External Interface를 담당
	libdamocm-4.0.{so|a|sl|dll}	공통 모듈 라이브러리
	liblogw-0.2.{so|a|sl|dll}	로그를 기록하는 라이브러리
	libcis_cc-3.3.{so|a|sl|dll}	암호화, 복호화 기능을 제공하는 라이브러리
	libcis_ce-3.3.{so|a|sl|dll}	암호화, 복호화를 제외한 부가적인 기능을 제공하는 라이브러리(예: Base64, 인증서 관리, 특성 유지 암호화 등)

파일 분류	파일 명	파일 용도
설정 파일	scpdb_agent.ini	DA 구동시 실행에 필요한 설정정보를 참조
License 파일	damo_lic.cer	DA 구동 시 제품의 유효성을 검증하는데 사용
Agent key 파일	damo_agt_site.cer	SG-KMS 연동, CLI 프로그램에서 사용하는 인증서 쌍
	damo_agt.cer	
	damo_agt.key	
접근제어 파일	acl_cli 파일	DB 의 USER 별로 암·복호 권한을 설정하는데에 사용
	privilege.damo	권한 파일
JAVA class 파일 (Oracle, Tibero, Cubrid 설치 가능)	ScpAgentException.class	예외 처리 Class
	ScpCryptData.class	암호화 복호화 Class
SQL 파일	아래 새로운 표에 DB별로 표기함	

Agent Key 파일은 SG-KMS 연동에 필요한 키 발급를 참고하여 발급 받는다.

DA-PGS(PostgreSQL)의 경우, DB 서버 버전에 따라 libdamoscpdb.so 라이브러리 선택

- libdamoscpdb94.so (Postgres 9.4)
- libdamoscpdb95.so (Postgres 9.5)
- libdamoscpdb95AS.so (EDB Postgres 9.5)
- libdamoscpdb96AS.so (EDB Postgres 9.6)
- libdamoscpdb10.so (Postgres 10)

10.3 환경변수 설정

DA를 설치할 운영체제에 환경변수 DA_INST_HOME를 설정한다. 이 매뉴얼에서는 제품 설치 경로를 아래와 같이 가정하여 설명한다.

- Linux 환경일 경우: /home/dbms_api
- Windows 환경일 경우: E:₩dbms_api

주의) DA_INST_HOME 설정 시, 주의 사항

- 리눅스의 경우 "/root" 디렉토리로 설정을 권장하지 않는다.
- 윈도우의 경우 "바탕화면"으로 설정을 권장하지 않는다.

위의 경로로 설정할 경우 접근 권한 등의 이유로 문제가 발생할 가능성이 존재한다.

10.3.1 환경변수 설정 - Linux 환경일 경우

DA_INST_HOME 환경변수에 DA의 설치 디렉터리를 설정한다.

```
.profile을 사용하는 경우
export DA_INST_HOME=/home/dbms_api


.cshrc를 사용하는 경우
setenv DA_INST_HOME=/home/dbms_api

export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$DA_INST_HOME
```

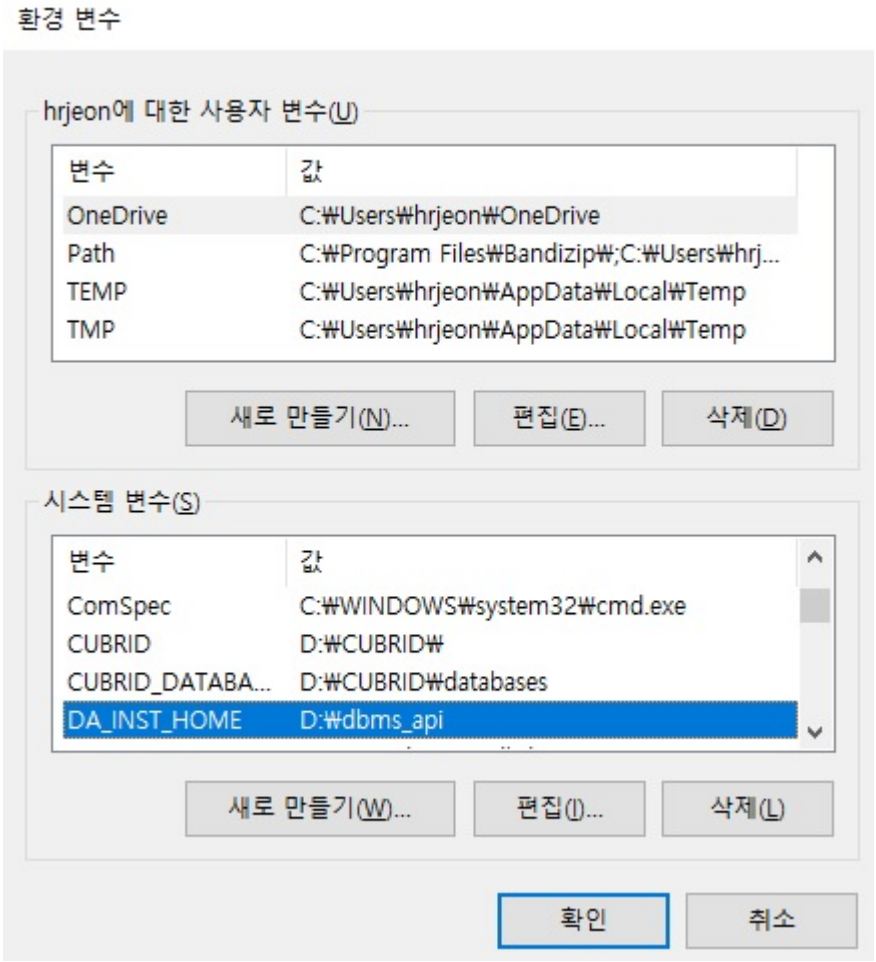
표 10-5 운영 체제별 라이브러리 PATH 명칭

운영 체제	라이브러리 PATH 명칭
HP_UX	SHLIB_PATH
AIX	LIBPATH
LINUX, SUN	LD_LIBRARY_PATH

10.3.2 환경변수 설정 - Windows 환경일 경우

[탐색기->내컴퓨터->등록정보->고급 탭->환경변수] 를 선택하여 나타나는 환경변수 설정 다이얼로그에서 시스템변수 DA_INST_HOME 변수와 값을 추가한다.

그림 10-2 DA_INST_HOME



10.4 DA의 설치 파일 복사

$DA_INST_HOME 디렉터리에 위 [설치 파일의 구성 확인]에서 나열된 파일들을 복사한다.

10.5 라이선스 파일 설정

$DA_INST_HOME 디렉토리에 라이선스 파일(damo_lic.cer)을 복사한다.

라이선스 파일명이 damo_lic.cer가 아닌 다른 이름으로 저장되어 있다면 변경해야 한다.

10.6 설치 파일의 접근 권한 부여 (Linux 설치 시)

Linux 사용자 계정에 DA 설치 파일(라이브러리, 실행 파일, 디렉토리)의 접근 권한을 부여한다. Windows에서 제품을 설치 할 경우, 이 과정은 생략한다.

```
$> cd $DA_INST_HOME
$> chmod 755 lib* acl_cli sql/install_make.sh
```

10.7 (SG-KMS 연동 시) SG-KMS에서 DA 정보 등록 및 연동에 필요한 키 내보내기

DA는 데이터를 암·복호화하기 위해 '암호화 키'가 필요하다. '암호화 키'를 얻기 위해서는 SG-KMS를 연동하거나 SCPS라는 암호화 키 파일을 이용할 수 있는데,
다음 설명은 SG-KMS 연동을 위해 SG-KMS에서 DA 정보를 등록하고 연동에 필요한 키를 내보내는 방법에 대해서 설명한다.

10.7.1 사전 준비

10.7.1.1 지원 SG-KMS 버전

DA와 연동 가능한 SG-KMS 버전은 다음과 같다.

표 10-6 연동 가능한 SG-KMS 버전

SG-KMS Major version	SG-KMS Minor version
SG-KMS v3.0	v3.0.9.0 이상 연동 가능
SG-KMS v4.0	v4.0.104.5 이상 연동 가능

SG-KMS v2.3 연동은 미지원한다.

10.7.1.2 연동 전 점검 항목

SG-KMS 연동을 위해서는 다음과 같이 사전 준비가 필요하다.

- SG-KMS 관리도구
- SG-KMS 관리도구에 접속할 수 있는 ID와 비밀번호
- SG-KMS 매뉴얼

이 매뉴얼에서는 SG-KMS의 키 종류 및 SG-KMS 관리도구 사용 방법에 대한 내용은 다루지 않으므로 **[SG-KMS 매뉴얼]**을 참고한다.

10.7.1.3 SG-KMS 연동에 필요한 키 발급

DA와 SG-KMS의 연동을 위해서는 먼저 SG-KMS에 DA를 Agent로 등록해야한다. 등록하는 절차는 SG-KMS 사용자설명서의 '대칭키를 사용하는 D'Amo Agent 등록 안내서'를 참고한다.

[D'Amo Agent 키] 파일, [Agent ID(CN)]와 [서비스 ID]정보는 DA와 SG-KMS 연동을 위한 설정 과정에서 필요하다. SG-KMS 관리자는 DA 설치 엔지니어에게 안전하게 전달한다.

SG-KMS 관리자에게 받은 D'Amo Agent의 인증서 및 키 파일 명을 다음과 같이 변경후 $DA_INST_HOME/key 디렉터리에 복사한다.

표 10-7 SG-KMS 연동에 필요한 키 목록

키 구분	생성된 키 파일명	변경할 키 파일명
사이트 키	damo-site_{발행기관/부서명}-SITE_V3.cer	damo_agt_site.cer
Agent 키	damo-scp_SITE_V3-{Agent 이름}.cer	damo_agt.cer
	damo-scp_SITE_V3-{Agent 이름}.key	damo_agt.key
	damo-scp_SITE_V3-{Agent 이름}.spin	damo_agt.spin

Agent 키 경로 지정, 키 이름 변경은 반드시 필요한 작업은 아니지만 관리를 위해 변경하는 것을 권장한다.

10.8 설정 파일(scpdb_agent.ini) 수정

DA의 운용을 위해 사용하는 scpdb_agent.ini 설정 파일 수정 방법에 대해 설명한다. DA는 SG-KMS를 이용하거나 SCPS 파일을 이용하여 암호화 키를 얻기 때문에 고객의 환경에 맞게 설정해야 한다.

$DA_INST_HOME 디렉터리에 있는 scpdb_agent.ini 설정 파일에서 아래 항목의 값을 수정한다.

1. 설정 파일의 [KEYINFO] 항목 - [KEY1]에 암호화 키 정보를 입력한다.

```
[KEYINFO]
KEY1=암복호화 하려는 암호화 키 정보를 입력한다.
//키 정보는 다음과 같은 값을 입력할 수 있다.
///ServiceID: SG-KMS에 생성한 서비스 ID를 입력한다.
///SCP_FilePath: SG-KMS에서 서비스 내보내기를 통해 발급한 SCPS 파일의 절대 경로 및 파일명, 확장자를 입력한다.

//*예제 - Windows 경우
KEY1=DA_AES256
KEY2=C:\DA\Policy\S_AES128.SCPS\DA_AES256.scps
KEY3=DA_AES256,C:\DA\Policy\S_AES128.SCPS\DA_AES256.scps

//*예제 - Linux 또는 Unix 경우
KEY1=DA_AES256
KEY2=/home/dbms_api/key/DA_AES256.scps
KEY3=DA_AES256,/home/dbms_api/key/DA_AES256.scps
```

ServiceID를 입력 할 경우 SG-KMS와 통신을 통해 암호화 정책을 가져와 데이터 암복호화를 수행한다.

SCP_FilePath를 입력 할 경우 KMS와 통신하지 않고 서버의 SCP 파일을 통해 암호화 정책을 가져와 데이터 암복호화를 수행한다.

ServiceID와 SCPS 파일명을 동시에 입력 시, SG-KMS와 네트워크 연결 실패 하면 SCPS 파일을 이용하여 암호화 한다.
ServiceID와 SCPS 파일명 사이에 공백이 있으면 SCPS 파일을 읽을 수 없다. 따라서 예제와 같이 띄어쓰기를 하지 않고 ServiceID와 SCPS 파일 경로를 붙여서 입력한다.

2. 설정 파일의 [Server], [Server2] 항목을 수정한다.

```
[Server]
ServerIP: SG-KMS의 IP를 입력한다.
ServerPort: SG-KMS의 포트번호를 입력한다. 기본값은 2525이다.

//*예제 - 모든 OS 공통
```

```
6  ServerIP=192.168.22.25
7  ServerPort=2525
```

```
1  [Server2]
2  ServerIP: 이중화를 위한 2번 SG-KMS의 IP를 입력한다.
3  ServerPort: 이중화를 위한 2번 SG-KMS의 포트번호를 입력한다. 기본값은 2525이다.
4
5  //*예제 - 모든 OS 공통
6  ServerIP=192.168.22.26
7  ServerPort=2525
```

ServerIP는 최소 1개 ~ 최대 10개까지 등록이 가능하다.

3. 설정 파일의 [AGENT] 항목을 설정한다.

```
1  [AGENT]
2  AgentID=SG-KMS 관리도구에서 설정한 D'Amo Agent의 Agent ID
3  LogDir=로그가 저장될 디렉터리 위치
4  LogLevel=로그가 남는 수준
5  SiteCertFilePath=SG-KMS 장비에서 설정한 해당 장비의 사이트 공개키(.cer)
6  CertFilePath=SG-KMS 관리도구에서 설정한 D'Amo Agent의 공개키(.cer)
7  KeyFilePath=SG-KMS 관리도구에서 설정한 D'Amo Agent의 비공개키(.key)
8  SPIN=SG-KMS 관리도구에서 설정한 D'Amo Agent의 SPIN으로, damo-scp_SITE_V3-[Agent이름].spin
파일의 값
9
10 //[예제]
11 AgentID=DA
12 LogDir=/home/dbms_api/log
13 LogLevel=4
14 SiteCertFilePath=/home/dbms_api/key/damo_agt_site.cer
15 CertFilePath=/home/dbms_api/key/damo_agt.cer
16 KeyFilePath=/home/dbms_api/key/damo_agt.key
17 SPIN=XaMh1ylXUh123XUh
```

로그가 남는 수준(LogLevel)에는 아래 5가지 숫자 입력이 가능하며, 각 값의 설정은 다음과 같다.
- 0: 아무 로그도 남기지 않을 경우
- 2: 경고 로그를 파일에 기록
- 4: 에러 로그와 경고 로그를 파일에 기록

- 6: 정보 로그, 에러 로그, 경고 로그를 파일에 기록
- 8: 디버그, 정보 로그, 에러 로그, 경고 로그를 파일에 기록

제품 운영 중 scpdb_agent.ini 파일을 수정하면 CONFIG_REINIT() 함수를 호출해야 변경된 내용이 적용된다.

10.9 CLI에서 권한 설정

$DA_INST_HOME 디렉터리에서 acl_cli 파일을 실행하여 USER 단위로 암/복호화 권한을 설정한다. USER 는 DB의 소유자명이고, KEY는 설정 파일(scpdb_agent.ini)의 [KEYINFO] 중 ALIAS 값(예: KEY1)이다.

CLI 명령어를 자세히 보려면 help 명령어를 실행한다.

```
CLI 실행 방법
$> cd $DA_INST_HOME
$> ./acl_cli  -start
Enter the PIN of CLI-key. : damo_agt.key 의 비밀번호
```

```
권한 추가할 경우
D'Amo > SET PRIV ENC [USER]"[KEY]"1"1
D'Amo > SAVE ALL
D'Amo > SHOW ALL

예제)
D'Amo > SET PRIV ENC SCOTT"KEY1"1"1
D'Amo > SET PRIV ENC SCOTT"KEY2"1"1
```

scpdb_agent.ini 설정 파일이 아래 예제와 같을 경우, CLI에서 권한 설정할 때 입력해야 하는 2번째 [KEY] 인자 값에는 KEY1을 입력해야 한다. (※ARIA256을 입력하는 것이 아님)

#scpdb_agent.ini 설정 파일 예제
[KEYINFO]
KEY1=ARIA256

```
권한 삭제할 경우
D'Amo > DEL PRIV ENC [USER]"[KEY]
D'Amo > SAVE ALL
D'Amo > SHOW ALL
```

CLI 에서 권한을 추가하거나 삭제 한 경우 반드시 SAVE ALL 명령어를 실행하며 SHOW ALL 명령어를 이용하여 적용 여부를 확인 한다.

10.10 DB 서버 사전 확인

DA를 SQL Server 환경에서 설치 하기 전에 다음과 같은 사항들을 확인한다.

- DB 엔진 설치 계정으로 DA설치 폴더 생성
  - DB 엔진 설치 계정과 권한이 동일해야 함
- DA를 사용할 DB 사용자 선정 후 해당 사용자로 데이터베이스 생성
  - 본 매뉴얼에서는 데이터베이스 명을 SCP로 가정함
- 위의 DB 사용자에 DBA 권한 부여

10.11 라이브러리 설정

[SQL Server 설치 경로]₩Binn 디렉터리에 라이브러리(damoscpdb.dll, damosaclr.dll, damocm-4.0.dll, cis_cc-3.3.dll, cis_ce-3.3.dll, logw-0.2.dll)을 복사한다.

10.12 sql 파일 생성

$DA_INST_HOME/sql 디렉터리에서 install_make.bat을 이용하여 설치할 sql 파일을 생성한다. D_INI는 설정 파일(scpdb_agent.ini)의 경로다.

```
$> cd %DA_INST_HOME%\sql
$> install_make.bat D_INI

[예제] D_INI 경로를 E:\dbms_api라고 가정한다.
E:\> install_make.bat E:\dbms_api
install_make.bat Start...
002.user_interface.sql Run...
002.user_interface.sql ok
002.user_interface_clr.sql Run...
002.user_interface_clr.sql ok
install_make.bat End...
```

10.13 제품 함수 설치

1. Microsoft SQL Server Management Studio을 이용하여 DB 접속 후 원하는 DB계정에 함수를 설치한다.
2. < XSP Mode >
   a. 001.inner_function.sql, 002.user_interface.sql 파일을 실행한다.
3. < CLR Mode >
   a. 001.inner_function_clr.sql 파일 수정 필요, 파일 내 "%PATH%"를 "[SQL Server 설치 경로]₩Binn₩"로 수정한다(총 3개).
   b. 001.inner_function_clr.sql, 002.user_interface_clr.sql 파일을 실행한다.
4. 특정 USER에게 함수 실행 권한을 부여한다. 모든 사용자에게 함수 실행 권한을 부여할 때는 003.grant_execute_functions.sql 파일을 실행한다.

10.14 제품 설치 확인

DB 서버에서 암/복호화 함수를 호출하여 설치를 성공했는지 확인한다. 암/복호화함수 첫 번째 인자 값에는 KEY ID가 아닌 scpdb_agent.ini파일의 KEY1, KEY2 항목 명으로 지정한다.

```
SELECT dbo.ENC_STR('KEY1', 'abc');

(expression)E6878572B3287A049906A8CA57F0207C
1 row(s) retrieved.
```

```
SELECT dbo.DEC_STR('KEY1', dbo.ENC_STR('KEY1', 'abc'));

(expression) abc
1 row(s) retrieved.
```

10.15 제품 운용

10.15.1 함수 설명

DA에서 제공되는 함수와 사용하는 방법에 대해서 설명한다.

10.15.1.1 파라미터 설명

10.15.1.1.1 I_KEY

I_KEY : 암호화/복호화 때 사용하는 암호화 키
- 설정 파일(scpdb_agent.ini)의 [KEYINFO] 중 ALIAS 값
- 아래 [설정 파일 예제]의 경우 ALIAS는 'KEY1'과 'KEY2'이다.
- ALIAS는 SCPS파일로 암호화 할 것인지, SG-KMS의 서비스 ID로 암호화 할 것인지 설정 가능하다.

[설정 파일 예제]
- 설정 파일(scpdb_agent.ini)의 [KEYINFO] 중 ALIAS 값
KEY1=AES256.SCPS
KEY2=ARIA256

[함수 사용 예제]
SELECT ENC_B64('KEY1', 'abc') FROM DUAL;

10.15.1.1.2 I_DATA

I_DATA : 평문(암호화 함수일 경우), 암호문(복호화 함수일 경우)

DA에서 제공하는 암호화 함수에서 성능 향상을 위해 라이브러리에서 정책 이름(KEYINFO ALIAS)을 바탕으로 Cash하여 동작한다. 정책 이름은 같은데 실제 키(대칭 키)가 다른 정책을 사용하면 Cash에서 이전 키로 복호화를 시도하여 오류가 발생한다. 이 상황을 해결하기 위해서는 데이터베이스 서버 재시작이 필요하다.

표 10-8 DA 함수 (SQL Server)

함수 명	입력			출력
ENC_STR	I_KEY	IN 문자열,		Hex String 암호문
	I_DATA	IN 문자열 (평문)		
ENC_B64	I_KEY	IN 문자열,		Base64 Encording 암호문
	I_DATA	IN 문자열 (평문)		
DEC_STR	I_KEY	IN 문자열,		평문
	I_DATA	IN 문자열 (Hex String 암호문)		
DEC_B64	I_KEY	IN 문자열,		평문
	I_DATA	IN 문자열 (base64 String 암호문)		
INDEX_STR	I_KEY	IN 문자열,		Hex String 암호문
	I_DATA	IN 문자열 (평문),		
	I_TYPE	IN 문자열 '' or 'IX '(Plug-IN 연동 시 사용)		
DEC_INDEX_STR	I_KEY	IN 문자열,		Hex String 암호문 입력받아 OPE 데이터
	I_DATA	IN 문자열 (암호문),		
	I_TYPE	IN 문자열 '' or 'IX '(Plug-IN 연동 시 사용)		
DEC_INDEX_B64	I_KEY	IN 문자열,		Base64 Encording 암호문 입력받아 OPE 데이터
	I_DATA	IN 문자열 (암호문),		
	I_TYPE	IN 문자열 '' or 'IX '(Plug-IN 연동 시 사용)		
HASH_STR	I_ALOG	IN 숫자,	SHA1 =70	Hex String 해쉬 암호문
			SHA256 =71	
			SHA384 =72	

			SHA512 =73	
			HAS160 =74	
	I_DATA	IN 문자열		
HASH_B64	I_ALOG	IN 숫자,	SHA1 =70	Base64 String 해쉬 암호문
			SHA256 =71	
			SHA384 =72	
			SHA512 =73	
			HAS160 =74	
	I_DATA	IN 문자열		
HEXTOB64 • CLR모드에서 지원안함	I_DATA	IN 문자열 (Hex String 암호문)		base64 Encording 암호문
B64TOHEX • CLR모드에서 지원안함	I_DATA	IN 문자열 (base64 Encording 암호문)		Hex String 암호문
CONFIG_REINIT				성공시 'SUCCESS', 그 외 에러

10.15.2 함수 호출 예제

암호 테이블의 소유자는 SCP, 암호 테이블명은 TAB, 암호 컬럼명은 COL, 암호할 데이터는 'abc' 로 가정한다.

1. ENC_STR

```
SELECT dbo.ENC_STR('KEY1', 'abc');
```

2. ENC_B64

```
SELECT dbo.ENC_B64('KEY1', 'abc');
```

3. DEC_STR

```
SELECT dbo.DEC_STR('KEY1', dbo.ENC_STR('KEY1', 'abc'));
```

4. DEC_B64

```
SELECT dbo.DEC_B64('KEY1', dbo.ENC_B64('KEY1', 'abc'));
```

5. INDEX_STR

```
DP 제품에 연동 하지 않을 경우
SELECT dbo.INDEX_STR('KEY1', 'abc', '');
DP 제품에 연동 할 경우
SELECT dbo.INDEX_STR( 'KEY1', 'abc', 'IX');
```

6. DEC_INDEX_STR, DEC_INDEX_B64

```
DP 제품에 연동 하지 않을 경우
SELECT dbo.DEC_INDEX_STR('KEY1', dbo.ENC_STR('KEY1', 'abc'), '');
SELECT dbo.DEC_INDEX_B64('KEY1', dbo.ENC_B64('KEY1', 'abc'), '');

DP 제품에 연동 할 경우
SELECT dbo.DEC_INDEX_STR('KEY1', dbo.ENC_STR('KEY1', 'abc'), 'IX');
SELECT dbo.DEC_INDEX_B64('KEY1', dbo.ENC_B64('KEY1', 'abc'), 'IX');
```

7. HASH_STR

```
SELECT dbo.HASH_STR( 71, 'abc' );
```

8. HASH_B64

```
SELECT dbo.HASH_B64( 71, 'abc' );
```

9. HEXTOB64

```
SELECT dbo.HEXTOB64('A305378D8F974F1C1537ED7CB0CB959245D1AC31');
```

10. B64TOHEX

```
SELECT dbo.B64TOHEX('owU3jY+XTxwVN+18sMuVkkXRrDE=');
```

11. CONFIG_REINIT

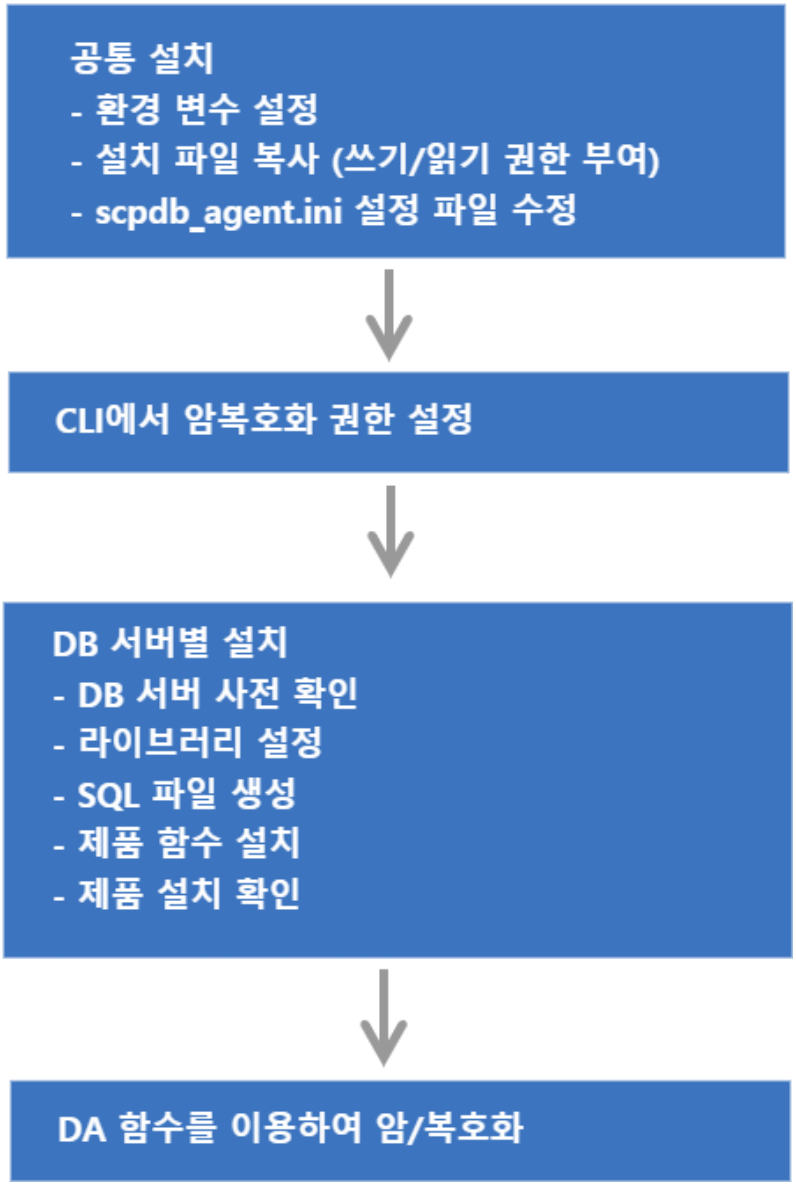
```
SELECT dbo.CONFIG_REINIT();
```

11.

Cache DB

DA는 DB 서버의 종류에 따라서 설치 및 운용 방법이 각각 다르다.

그림 11-1 설치 개념도



11.1 지원 운영체제 및 DB서버

DA에서 지원하는 운영체제 및 DB서버는 아래와 같다. 단, 특정 환경은 지원되지 않을 수도 있으므로, 제품 설치 전에 상세한 지원 가능 여부는 펜타시큐리티시스템으로 문의한다.

표 11-1 제품이 지원하는 운영체제 및 DB 서버 정보

구분	설명
운영체제	Windows, AIX, HP IA, HP PA-RISC, Linux, SUN, TRU64
DB 서버	ORACLE 8i ~, SQL Server 2012 ~, DB2 9.x ~, Tibero 4 SP 1 ~, MySQL 5.x ~, MariaDB 5.5, 10.0, 10.1, Cache DB 2009.1 ~, Informix IDS 9.x ~ Sybase ASE 15.7 SP61 ~, Sybase IQ 15.4 ~, CUBRID 2008 R1.3 ~ , PostgreSQL 9.4 ~

11.2 설치 파일의 구성 확인

DA의 설치 파일은 아래와 같은 명칭으로 압축 파일(zip) 형태로 제공된다.

- DA 설치파일: Install_DAmo_DA_v{버전}.zip
  - 압축을 해제 한 뒤, 설치 대상 DB 서버, OS 및 bit에 맞는 설치 파일을 준비한다.

설치 파일에 제품을 사용할 수 있는 '라이선스'는 포함되어 있지 않다.
펜타시큐리티시스템에 문의하여 '라이선스' 파일은 별도로 준비한다.

표 11-2 설치 파일(Install_DAmo_DA_v{버전}.zip)을 압축 해제 시, 디렉터리의 구성

구성	설명
_SampleScpsFiles	SG-KMS 연동 없이 암복호화를 테스트할 수 있는 테스트 키 파일
_TestAgentKeyPair	CLI에서 사용할 수 있는 테스트 키 쌍
Altibase	DA-ALT 제품의 설치 바이너리 폴더
Cache	DA-CDB 제품의 설치 바이너리 폴더
Cubrid	DA-CUB 제품의 설치 바이너리 폴더
DB2	DA-DB2 제품의 설치 바이너리 폴더
Informix	DA-IFX 제품의 설치 바이너리 폴더
MySQL	DA-MYQ 제품의 설치 바이너리 폴더
Oracle	DA-ORA 제품의 설치 바이너리 폴더
Postgres	DA-PGS 제품의 설치 바이너리 폴더
SQL Server	DA-MSQ 제품의 설치 바이너리 폴더

구성	설명
SybaseASE	DA-SYB 제품의 설치 바이너리 폴더
SybaseIQ	DA-SIQ 제품의 설치 바이너리 폴더
Tibero	DA-TIB 제품의 설치 바이너리 폴더

11.2.1 DB 서버 및 운영체제 별 SQL파일 구성

각 DB 서버 및 운영체제 별 SQL파일 구성은 다음과 같다. 다음 장에서 각 DB별로 SQL파일 설치 방법을 설명한다.

표 11-3 DB 서버 및 운영체제 별 SQL파일

DB 서버 종류	Linux 일 경우	Windows 일 경우
Oracle	000.da_user.pkg 000.da_user.sql 001.inner_function.ora(c 버전 설치 시) 002.user_interface.ora(JAVA 버전 설치 시) 002.user_interface_java.ora 003.grant_execute_functions.sql(JAVA 버전 설치 시) 005.securej_privilege.sql install_make.sh	000.da_user.pkg 000.da_user.sql 001.inner_function.ora(c 버전 설치 시) 002.user_interface.ora(JAVA 버전 설치 시) 002.user_interface_java.ora 003.grant_execute_functions.sql(JAVA 버전 설치 시) 005.securej_privilege.sql 009.da_test.sql
MYSQL	001.inner_function.mys 002.user_interface. mys 003.grant_execute_functions.sql 009.da_test.sql install_make.sh	001.inner_function.mys 002.user_interface.mys 003.grant_execute_functions.sql
TIBERO	001.inner_function.tbs(c 버전 설치 시) 001.inner_function_java.tbs(JAVA 버전 설치 시) 002.user_interface.tbs(c 버전 설치 시) 002.user_interface_java.tbs(JAVA 버전 설치 시) 003.grant_execute_functions.sql install_make.sh	001.inner_function_java.tbs 002.user_interface_java.tbs 003.grant_execute_functions.sql
INFORMIX	001.inner_function.ifx 002.user_interface. ifx 003.grant_execute_functions.sql 009.da_test.sql install_make.sh	해당 없음
POSTGRESQL	001.inner_function.post 002.user_interface.post 003.grant_execute_functions.post	해당 없음

DB 서버 종류	Linux 일 경우	Windows 일 경우
DB2	001.inner_function.db2 002.user_interface.db2 003.grant_execute_functions.sql install_make.sh	001.inner_function.db2 002.user_interface.db2 003.grant_execute_functions.sql
CUBRID	001.inner_function.sql 002.user_interface.sql	001.inner_function.sql 002.user_interface.sql
SYBASE	001.inner_function.sybase 002.user_interface.sybase 003.grant_execute_functions.sql install_make.sh	해당 없음
SYBASE IQ	001.inner_function.sybiq 002.user_interface. sybiq 003.grant_execute_functions.sql install_make.sh	해당 없음
SQL Server	해당 없음	001.inner_function.sql 002.user_interface. sql 003.grant_execute_functions.sql install_make.bat
Cache DB	SCP.xml install_make.sh	SCP.xml
Altibase	해당 없음	000.da_user.pkg 000.da_user.sql 001.inner_function.sql 002.user_interface. sql install_make.bat

11.2.2 공통 설치 파일 구성

환경에 관계 없이 공통적으로 사용하는 설치 파일은 아래와 같다.

표 11-4 공통 설치 파일 (sql파일 제외)

파일 분류	파일 명	파일 용도
Library 파일	libdamoscpdb.{so|a|sl|dll}	DA 메인 라이브러리. 주로 DBMS External Interface를 담당
	libdamocm-4.0.{so|a|sl|dll}	공통 모듈 라이브러리
	liblogw-0.2.{so|a|sl|dll}	로그를 기록하는 라이브러리
	libcis_cc-3.3.{so|a|sl|dll}	암호화, 복호화 기능을 제공하는 라이브러리
	libcis_ce-3.3.{so|a|sl|dll}	암호화, 복호화를 제외한 부가적인 기능을 제공하는 라이브러리(예: Base64, 인증서 관리, 특성 유지 암호화 등)

파일 분류	파일 명	파일 용도
설정 파일	scpdb_agent.ini	DA 구동시 실행에 필요한 설정정보를 참조
License 파일	damo_lic.cer	DA 구동 시 제품의 유효성을 검증하는데 사용
Agent key 파일	damo_agt_site.cer	SG-KMS 연동, CLI 프로그램에서 사용하는 인증서 쌍
	damo_agt.cer	
	damo_agt.key	
접근제어 파일	acl_cli 파일	DB 의 USER 별로 암·복호 권한을 설정하는데에 사용
	privilege.damo	권한 파일
JAVA class 파일 (Oracle, Tibero, Cubrid 설치 가능)	ScpAgentException.class	예외 처리 Class
	ScpCryptData.class	암호화 복호화 Class
SQL 파일	아래 새로운 표에 DB별로 표기함	

Agent Key 파일은 SG-KMS 연동에 필요한 키 발급를 참고하여 발급 받는다.

DA-PGS(PostgreSQL)의 경우, DB 서버 버전에 따라 libdamoscpdb.so 라이브러리 선택

- libdamoscpdb94.so (Postgres 9.4)
- libdamoscpdb95.so (Postgres 9.5)
- libdamoscpdb95AS.so (EDB Postgres 9.5)
- libdamoscpdb96AS.so (EDB Postgres 9.6)
- libdamoscpdb10.so (Postgres 10)

11.3 환경변수 설정

DA를 설치할 운영체제에 환경변수 DA_INST_HOME를 설정한다. 이 매뉴얼에서는 제품 설치 경로를 아래와 같이 가정하여 설명한다.

- Linux 환경일 경우: /home/dbms_api
- Windows 환경일 경우: E:₩dbms_api

주의) DA_INST_HOME 설정 시, 주의 사항

- 리눅스의 경우 "/root" 디렉토리로 설정을 권장하지 않는다.
- 윈도우의 경우 "바탕화면"으로 설정을 권장하지 않는다.

위의 경로로 설정할 경우 접근 권한 등의 이유로 문제가 발생할 가능성이 존재한다.

11.3.1 환경변수 설정 - Linux 환경일 경우

DA_INST_HOME 환경변수에 DA의 설치 디렉터리를 설정한다.

```
.profile을 사용하는 경우
export DA_INST_HOME=/home/dbms_api


.cshrc를 사용하는 경우
setenv DA_INST_HOME=/home/dbms_api

export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$DA_INST_HOME
```

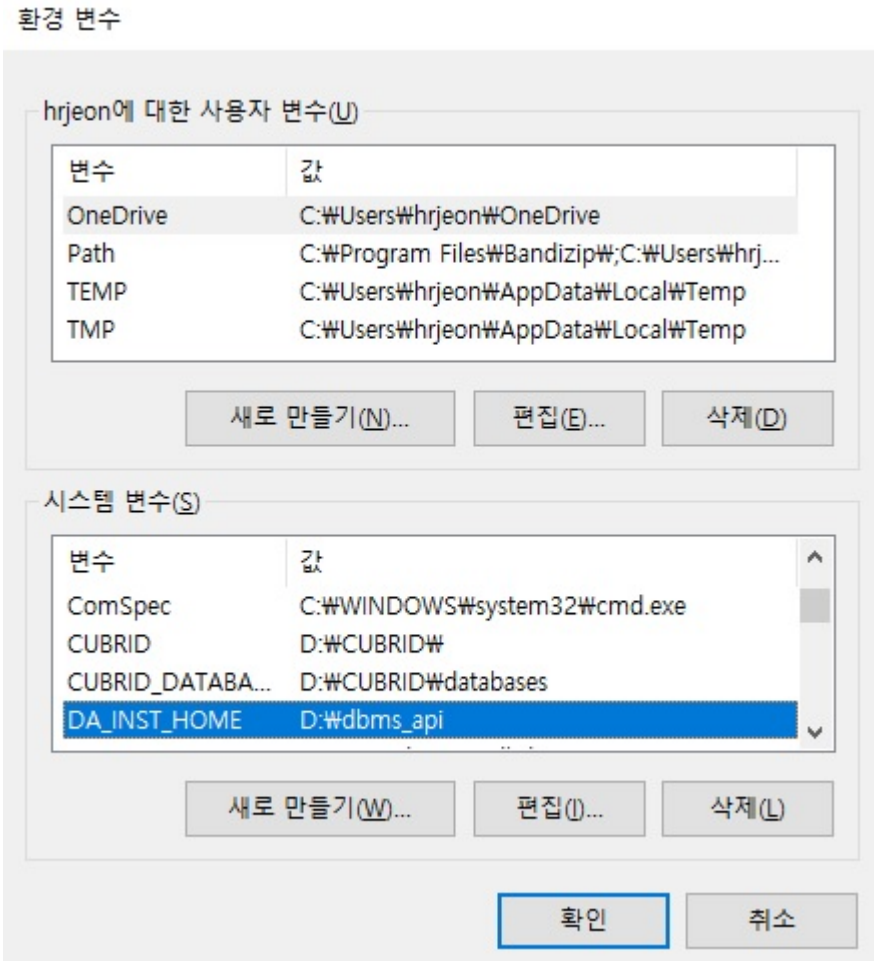
표 11-5 운영 체제별 라이브러리 PATH 명칭

운영 체제	라이브러리 PATH 명칭
HP_UX	SHLIB_PATH
AIX	LIBPATH
LINUX, SUN	LD_LIBRARY_PATH

11.3.2 환경변수 설정 - Windows 환경일 경우

[탐색기->내컴퓨터->등록정보->고급 탭->환경변수] 를 선택하여 나타나는 환경변수 설정 다이얼로그에서 시스템변수 DA_INST_HOME 변수와 값을 추가한다.

그림 11-2 DA_INST_HOME



11.4 DA의 설치 파일 복사

$DA_INST_HOME 디렉터리에 위 [설치 파일의 구성 확인]에서 나열된 파일들을 복사한다.

11.5 라이선스 파일 설정

$DA_INST_HOME 디렉토리에 라이선스 파일(damo_lic.cer)을 복사한다.


라이선스 파일명이 damo_lic.cer가 아닌 다른 이름으로 저장되어 있다면 변경해야 한다.

11.6 설치 파일의 접근 권한 부여 (Linux 설치 시)

Linux 사용자 계정에 DA 설치 파일(라이브러리, 실행 파일, 디렉토리)의 접근 권한을 부여한다. Windows에서 제품을 설치 할 경우, 이 과정은 생략한다.

```
$> cd $DA_INST_HOME
$> chmod 755 lib* acl_cli sql/install_make.sh
```

11.7 (SG-KMS 연동 시) SG-KMS에서 DA 정보 등록 및 연동에 필요한 키 내보내기

DA는 데이터를 암·복호화하기 위해 '암호화 키'가 필요하다. '암호화 키'를 얻기 위해서는 SG-KMS를 연동하거나 SCPS라는 암호화 키 파일을 이용할 수 있는데,
다음 설명은 SG-KMS 연동을 위해 SG-KMS에서 DA 정보를 등록하고 연동에 필요한 키를 내보내는 방법에 대해서 설명한다.

11.7.1 사전 준비

11.7.1.1 지원 SG-KMS 버전

DA와 연동 가능한 SG-KMS 버전은 다음과 같다.

표 11-6 연동 가능한 SG-KMS 버전

SG-KMS Major version	SG-KMS Minor version
SG-KMS v3.0	v3.0.9.0 이상 연동 가능
SG-KMS v4.0	v4.0.104.5 이상 연동 가능

SG-KMS v2.3 연동은 미지원한다.

11.7.1.2 연동 전 점검 항목

SG-KMS 연동을 위해서는 다음과 같이 사전 준비가 필요하다.

- SG-KMS 관리도구
- SG-KMS 관리도구에 접속할 수 있는 ID와 비밀번호
- SG-KMS 매뉴얼

이 매뉴얼에서는 SG-KMS의 키 종류 및 SG-KMS 관리도구 사용 방법에 대한 내용은 다루지 않으므로 **[SG-KMS 매뉴얼]**을 참고한다.

11.7.1.3 SG-KMS 연동에 필요한 키 발급

DA와 SG-KMS의 연동을 위해서는 먼저 SG-KMS에 DA를 Agent로 등록해야한다. 등록하는 절차는 SG-KMS 사용자설명서의 '대칭키를 사용하는 D'Amo Agent 등록 안내서'를 참고한다.

[D'Amo Agent 키] 파일, [Agent ID(CN)]와 [서비스 ID]정보는 DA와 SG-KMS 연동을 위한 설정 과정에서 필요하다. SG-KMS 관리자는 DA 설치 엔지니어에게 안전하게 전달한다.

SG-KMS 관리자에게 받은 D'Amo Agent의 인증서 및 키 파일 명을 다음과 같이 변경후 $DA_INST_HOME/key 디렉터리에 복사한다.

표 11-7 SG-KMS 연동에 필요한 키 목록

키 구분	생성된 키 파일명	변경할 키 파일명
사이트 키	damo-site_{발행기관/부서명}-SITE_V3.cer	damo_agt_site.cer
Agent 키	damo-scp_SITE_V3-{Agent 이름}.cer	damo_agt.cer
	damo-scp_SITE_V3-{Agent 이름}.key	damo_agt.key
	damo-scp_SITE_V3-{Agent 이름}.spin	damo_agt.spin

Agent 키 경로 지정, 키 이름 변경은 반드시 필요한 작업은 아니지만 관리를 위해 변경하는 것을 권장한다.

11.8 설정 파일(scpdb_agent.ini) 수정

DA의 운용을 위해 사용하는 scpdb_agent.ini 설정 파일 수정 방법에 대해 설명한다. DA는 SG-KMS를 이용하거나 SCPS 파일을 이용하여 암호화 키를 얻기 때문에 고객의 환경에 맞게 설정해야 한다.

$DA_INST_HOME 디렉터리에 있는 scpdb_agent.ini 설정 파일에서 아래 항목의 값을 수정한다.

1. 설정 파일의 [KEYINFO] 항목 - [KEY1]에 암호화 키 정보를 입력한다.

```
[KEYINFO]
KEY1=암복호화 하려는 암호화 키 정보를 입력한다.
//키 정보는 다음과 같은 값을 입력할 수 있다.
///ServiceID: SG-KMS에 생성한 서비스 ID를 입력한다.
///SCP_FilePath: SG-KMS에서 서비스 내보내기를 통해 발급한 SCPS 파일의 절대 경로 및 파일명, 확장자를 입력한다.

//*예제 - Windows 경우
KEY1=DA_AES256
KEY2=C:\DA\Policy\S_AES128.SCPS\DA_AES256.scps
KEY3=DA_AES256,C:\DA\Policy\S_AES128.SCPS\DA_AES256.scps

//*예제 - Linux 또는 Unix 경우
KEY1=DA_AES256
KEY2=/home/dbms_api/key/DA_AES256.scps
KEY3=DA_AES256,/home/dbms_api/key/DA_AES256.scps
```

ServiceID를 입력 할 경우 SG-KMS와 통신을 통해 암호화 정책을 가져와 데이터 암복호화를 수행한다.

SCP_FilePath를 입력 할 경우 KMS와 통신하지 않고 서버의 SCP 파일을 통해 암호화 정책을 가져와 데이터 암복호화를 수행한다.

ServiceID와 SCPS 파일명을 동시에 입력 시, SG-KMS와 네트워크 연결 실패 하면 SCPS 파일을 이용하여 암호화 한다.
ServiceID와 SCPS 파일명 사이에 공백이 있으면 SCPS 파일을 읽을 수 없다. 따라서 예제와 같이 띄어쓰기를 하지 않고 ServiceID와 SCPS 파일 경로를 붙여서 입력한다.

2. 설정 파일의 [Server], [Server2] 항목을 수정한다.

```
[Server]
ServerIP: SG-KMS의 IP를 입력한다.
ServerPort: SG-KMS의 포트번호를 입력한다. 기본값은 2525이다.

//*예제 - 모든 OS 공통
```

```
6  ServerIP=192.168.22.25
7  ServerPort=2525
```

```
1  [Server2]
2  ServerIP: 이중화를 위한 2번 SG-KMS의 IP를 입력한다.
3  ServerPort: 이중화를 위한 2번 SG-KMS의 포트번호를 입력한다. 기본값은 2525이다.
4
5  //*예제 - 모든 OS 공통
6  ServerIP=192.168.22.26
7  ServerPort=2525
```

ServerIP는 최소 1개 ~ 최대 10개까지 등록이 가능하다.

3. 설정 파일의 [AGENT] 항목을 설정한다.

```
1  [AGENT]
2  AgentID=SG-KMS 관리도구에서 설정한 D'Amo Agent의 Agent ID
3  LogDir=로그가 저장될 디렉터리 위치
4  LogLevel=로그가 남는 수준
5  SiteCertFilePath=SG-KMS 장비에서 설정한 해당 장비의 사이트 공개키(.cer)
6  CertFilePath=SG-KMS 관리도구에서 설정한 D'Amo Agent의 공개키(.cer)
7  KeyFilePath=SG-KMS 관리도구에서 설정한 D'Amo Agent의 비공개키(.key)
8  SPIN=SG-KMS 관리도구에서 설정한 D'Amo Agent의 SPIN으로, damo-scp_SITE_V3-[Agent이름].spin
파일의 값
9
10 //[예제]
11 AgentID=DA
12 LogDir=/home/dbms_api/log
13 LogLevel=4
14 SiteCertFilePath=/home/dbms_api/key/damo_agt_site.cer
15 CertFilePath=/home/dbms_api/key/damo_agt.cer
16 KeyFilePath=/home/dbms_api/key/damo_agt.key
17 SPIN=XaMh1ylXUh123XUh
```

로그가 남는 수준(LogLevel)에는 아래 5가지 숫자 입력이 가능하며, 각 값의 설정은 다음과 같다.
- 0: 아무 로그도 남기지 않을 경우
- 2: 경고 로그를 파일에 기록
- 4: 에러 로그와 경고 로그를 파일에 기록

- 6: 정보 로그, 에러 로그, 경고 로그를 파일에 기록
- 8: 디버그, 정보 로그, 에러 로그, 경고 로그를 파일에 기록

제품 운영 중 scpdb_agent.ini 파일을 수정하면 CONFIG_REINIT() 함수를 호출해야 변경된 내용이 적용된다.

11.9 CLI에서 권한 설정

$DA_INST_HOME 디렉터리에서 acl_cli 파일을 실행하여 USER 단위로 암/복호화 권한을 설정한다. USER 는 DB의 소유자명이고, KEY는 설정 파일(scpdb_agent.ini)의 [KEYINFO] 중 ALIAS 값(예: KEY1)이다.

CLI 명령어를 자세히 보려면 help 명령어를 실행한다.

```
CLI 실행 방법
$> cd $DA_INST_HOME
$> ./acl_cli  -start
Enter the PIN of CLI-key. : damo_agt.key 의 비밀번호
```

```
권한 추가할 경우
D'Amo > SET PRIV ENC [USER]"[KEY]"1"1
D'Amo > SAVE ALL
D'Amo > SHOW ALL

예제)
D'Amo > SET PRIV ENC SCOTT"KEY1"1"1
D'Amo > SET PRIV ENC SCOTT"KEY2"1"1
```

scpdb_agent.ini 설정 파일이 아래 예제와 같을 경우, CLI에서 권한 설정할 때 입력해야 하는 2번째 [KEY] 인자 값에는 KEY1을 입력해야 한다. (※ARIA256을 입력하는 것이 아님)

#scpdb_agent.ini 설정 파일 예제
[KEYINFO]
KEY1=ARIA256

```
권한 삭제할 경우
D'Amo > DEL PRIV ENC [USER]"[KEY]
D'Amo > SAVE ALL
D'Amo > SHOW ALL
```

CLI 에서 권한을 추가하거나 삭제 한 경우 반드시 SAVE ALL 명령어를 실행하며 SHOW ALL 명령어를 이용하여 적용 여부를 확인 한다.

11.10 DB 서버 사전 확인

DA를 Cache DB 환경에서 설치 하기 전에 다음과 같은 사항들을 확인한다.

- DB 엔진 설치 계정에 폴더 생성
- DB 사용자로 SCP 계정 생성 (권고사항)
- DB 서버의 DBA 권한 계정
- DB 엔진 설치 계정의 . profile 파일 수정 (Linux 설치 시)

11.11 라이브러리 설정

11.11.1 라이브러리 링크 설정(Linux 설치 시)

[CACHE DB 설치 경로]/lib 경로에 $DA_INST_HOME에 있는 libdamoscpdb.{so|a|sl} 파일의 심볼릭 링크를 건다.

```
$> cd /home/cache/lib
$> ln  -s $DA_INST_HOME/libdamoscpdb.{so|a|sl} .
```

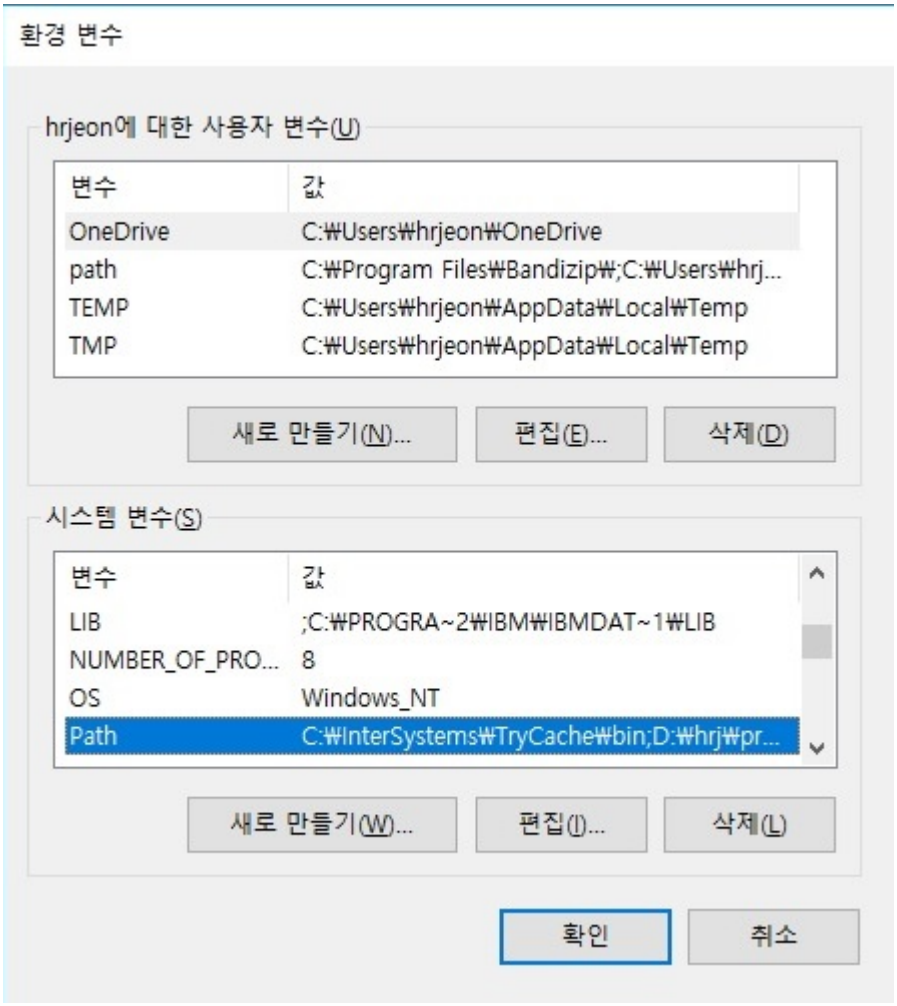
11.11.2 라이브러리 복사 및 설정 (Windows에 설치 시)

4. [CACHE DB 설치 경로]/bin 경로에 %DA_INST_HOME%에 있는 아래 5개 파일을 복사한다.

- cis_cc-3.3.dll
- cis_ce-3.3.dll
- logw-0.2.dll
- damocm-4.0.dll
- damoscpdb.dll

5. [CACHE DB 설치 경로]/bin 경로는 Windows 시스템 환경 변수 [PATH]에 추가한다.

그림 11-3 환경변수 설정 다이얼로그



11.12 DAMO_SCP.xml 파일 생성

11.12.1 DAMO_SCP.xml 파일 생성 (Linux의 경우)

1. $DA_INST_HOME/sql 디렉터리에서 install_make.sh을 이용하여 설치할 xml 파일을 생성한다. D_INI_PATH

는 설정 파일(scpdb_agent.ini)의 경로를 의미하고, D_LIBFILE은 제품 라이브러리 경로다.

```
$> cd $DA_INST_HOME/sql
$> ./install_make.sh D_INI_PATH D_LIBFILE

[예제]
$> ./install_make.sh /home/dbms_api /home/cache/lib
D_INI_PATH is replaced by /home/dbms_api
D_LIB is replaced by /home/cache/lib
$>
```

2. DAMO_SCP.xml 파일 1개가 생성되었는지 확인한다.

11.12.2 DAMO_SCP.xml 파일 생성 (Windows의 경우)

1. 텍스트 편집기를 이용하여 SCP.xml 파일의 D_INI 값을 수정한다. D_INI 는 설정 파일(scpdb_agent.ini)의 경로다.

```
수정 전 : Set initFile = "D_INI"
수정 후 : Set initFile = "E:\dbms_api\scpdb_agent.ini"
```

2. 텍스트 편집기를 이용하여 SCP.xml 파일의 D_LIBFILE 값을 수정한다. D_LIBFILE 는 암호 모듈 라이브러리(damoscpdb.dll)의 경로다.

```
수정 전 : Set libFile = "D_LIBFILE"
수정 후 : Set libFile = "C:\InterSystems\TryCache\bin\damoscpdb.dll"
```

3. 경로 수정이 끝나면 SCP.xml 파일을 DAMO_SCP.xml로 이름을 변경한다.

11.13 제품 함수 설치

1. %DA_INST_HOME%₩sql 위치에서 DB 접속 후 설치를 원하는 DB계정에 함수를 설치한다.

```
$> css cterminal [InstanceName]

$> css cterminal TRYCACHE
USER>Do $system.OBJ.Load("/home/dbms_api/sql/DAMO_SCP.xml","ck")
```

11.14 제품 설치 확인

DB 서버에서 암/복호화 함수를 호출하여 설치를 성공했는지 확인한다.

```
USER> SET ENC=##class(DAMO.SCP).ENCSTR("DAMO", "abc")
USER> WRITE ENC
E6878572B3287A049906A8CA57F0207C

USER> SET DEC=##class(DAMO.SCP).DECSTR("DAMO", "abc")
USER> WRITE DEC
abc

USER>
```

11.15 제품 운용

11.15.1 함수 설명

DA에서 제공되는 함수와 사용하는 방법에 대해서 설명한다.

11.15.1.1 파라미터 설명

11.15.1.1.1 I_KEY

I_KEY : 암호화/복호화 때 사용하는 암호화 키
- 설정 파일(scpdb_agent.ini)의 [KEYINFO] 중 ALIAS 값
- 아래 [설정 파일 예제]의 경우 ALIAS는 'KEY1'과 'KEY2'이다.
- ALIAS는 SCPS파일로 암호화 할 것인지, SG-KMS의 서비스 ID로 암호화 할 것인지 설정 가능하다.

[설정 파일 예제]
- 설정 파일(scpdb_agent.ini)의 [KEYINFO] 중 ALIAS 값
KEY1=AES256.SCPS
KEY2=ARIA256

[함수 사용 예제]
SELECT ENC_B64('KEY1', 'abc') FROM DUAL;

11.15.1.1.2 I_DATA

I_DATA : 평문(암호화 함수일 경우), 암호문(복호화 함수일 경우)

DA에서 제공하는 암호화 함수에서 성능 향상을 위해 라이브러리에서 정책 이름(KEYINFO ALIAS)을 바탕으로 Cash하여 동작한다. 정책 이름은 같은데 실제 키(대칭 키)가 다른 정책을 사용하면 Cash에서 이전 키로 복호화를 시도하여 오류가 발생한다. 이 상황을 해결하기 위해서는 데이터베이스 서버 재시작이 필요하다.

표 11-8 DA 함수 (Cache DB)

함수 명	입력		출력
ENCSTR	I_KEY	IN 문자열	Hex String 암호문
	I_DATA	IN 문자열 (평문)	
ENCB64	I_KEY	IN 문자열,	Base64 Encording 암호문
	I_DATA	IN 문자열 (평문)	
DECSTR	I_KEY	IN 문자열,	평문
	I_DATA	IN 문자열 (Hex String 암호문)	
DECB64	I_KEY	IN 문자열,	평문
	I_DATA	IN 문자열 (base64 String 암호문)	
INDEXSTR	I_KEY	IN 문자열,	Hex String 암호문
	I_DATA	IN 문자열 (평문),	
	I_TYPE	IN 문자열 '' or 'IX '(Plug-IN 연동 시 사용)	
DECINDEXSTR	I_KEY	IN 문자열,	Hex String 암호문 입력받아 OPE 데이터
	I_DATA	IN 문자열 (암호문),	
	I_TYPE	IN 문자열 '' or 'IX '(Plug-IN 연동 시 사용)	
DECINDEXB64	I_KEY	IN 문자열,	Base64 Encording 암호문 입력받아 OPE 데이터

	I_DATA	IN 문자열 (암호문),		
	I_TYPE	IN 문자열 '' or 'IX '(Plug-IN 연동 시 사용)		
HASHSTR	I_ALOG	IN 숫자,	SHA1 =70	Hex String 해쉬 암호문
			SHA256 =71	
			SHA384 =72	
			SHA512 =73	
			HAS160 =74	
	I_DATA	IN 문자열		
HASHB64	I_ALOG	IN 숫자,	SHA1 =70	Base64 String 해쉬 암호문
			SHA256 =71	
			SHA384 =72	
			SHA512 =73	
			HAS160 =74	
	I_DATA	IN 문자열		
HEXTOB64	I_DATA	IN 문자열 (Hex String 암호문)		base64 Encording 암호문
B64TOHEX	I_DATA	IN 문자열 (base64 Encording 암호문)		Hex String 암호문
CONFIG_REINIT				성공시 'SUCCESS', 그외 에러

11.15.2 함수 호출 예제

1. ENCSTR

```
USER> set a = ##class(DAMO.SCP).ENCSTR("KEY1", "abc")
```

2. ENCB64

```
USER> set a = ##class(DAMO.SCP).ENCB64("KEY1", "abc")
```

3. DECSTR

```
USER> set a = ##class(DAMO.SCP).DECSTR("KEY1", #class(DAMO.SCP).ENCSTR("KEY1", "abc"))
```

4. DECB64

```
USER> set a = ##class(DAMO.SCP).DECB64("KEY1", #class(DAMO.SCP).ENCB64("KEY1", "abc"))
```

5. INDEXSTR

```
DP 제품에 연동 하지 않을 경우
USER> set a = ##class(DAMO.SCP).INDEXSTR("KEY1", "abc", "")

DP 제품에 연동 할 경우
USER> set a = ##class(DAMO.SCP).INDEXSTR("KEY1", "abc", "IX")
```

6. DECINDEXSTR, DECINDEXB64

```
DP 제품에 연동 하지 않을 경우
USER> set a = ##class(DAMO.SCP).DECINDEXSTR("KEY1", ##class(DAMO.SCP).ENCSTR("KEY1", "abc"),
"")
USER> set a = ##class(DAMO.SCP).DECINDEXB64("KEY1", ##class(DAMO.SCP).ENCB64("KEY1", "abc"),
"")

DP 제품에 연동 할 경우
USER> set a = ##class(DAMO.SCP).DECINDEXSTR("KEY1", ##class(DAMO.SCP).ENCSTR("KEY1", "abc"),
"IX")
USER> set a = ##class(DAMO.SCP).DECINDEXB64("KEY1", ##class(DAMO.SCP).ENCB64("KEY1", "abc"),
"IX")
```

7. HASHSTR

```
USER> set a = ##class(DAMO.SCP).HASHSTR(71, "abc")
```

8. HASHB64

```
USER> set a = ##class(DAMO.SCP).HASHB64(71, "abc")
```

9. HEXTOB64

```
USER> set a = ##class(DAMO.SCP).HEXTOB64('A305378D8F974F1C1537ED7CB0CB959245D1AC31')
```

10. B64TOHEX

```
USER> set a = ##class(DAMO.SCP). B64TOHEX('owU3jY+XTxwVN+18sMuVkkXRrDE=')
```

11. CONFIGREINIT

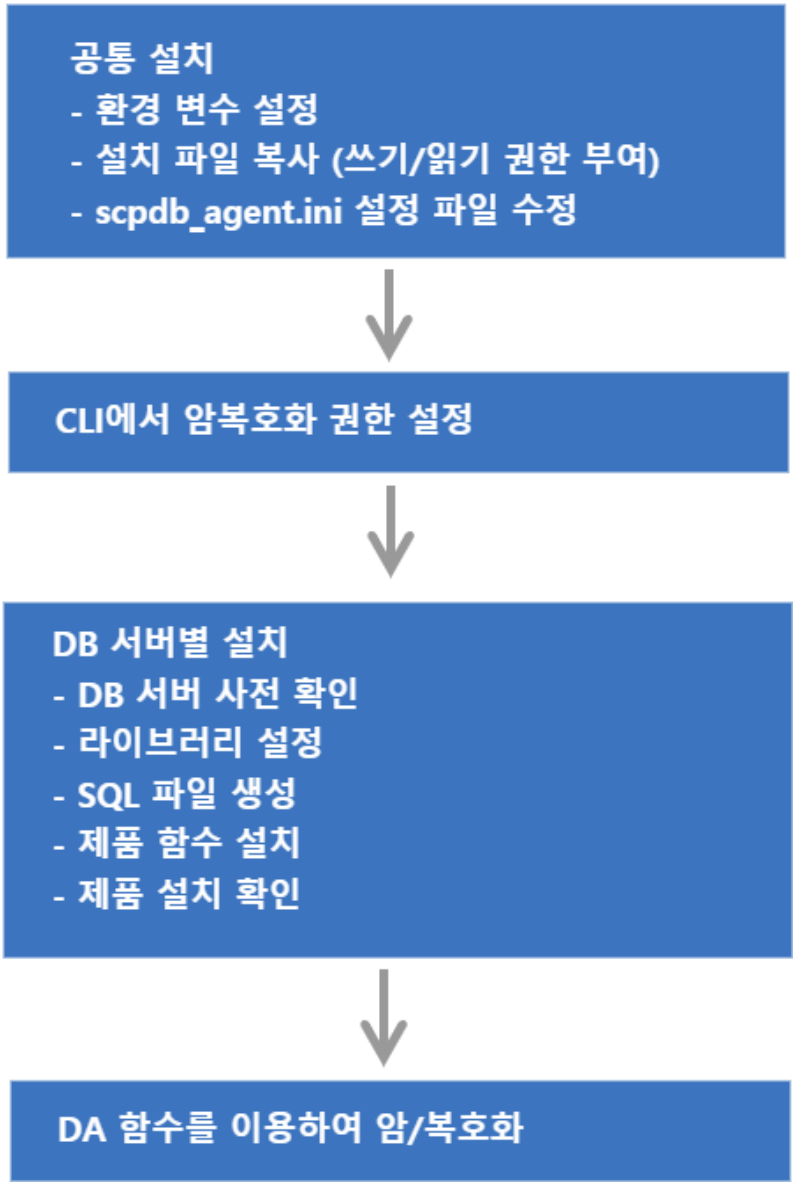
```
USER> set a = ##class(DAMO.SCP).CONFIGREINIT();
```

12.

Altibase

DA는 DB 서버의 종류에 따라서 설치 및 운용 방법이 각각 다르다.

그림 12-1 설치 개념도



12.1 지원 운영체제 및 DB서버

DA에서 지원하는 운영체제 및 DB서버는 아래와 같다. 단, 특정 환경은 지원되지 않을 수도 있으므로, 제품 설치 전에 상세한 지원 가능 여부는 펜타시큐리티시스템으로 문의한다.

표 12-1 제품이 지원하는 운영체제 및 DB 서버 정보

구분	설명
운영체제	Windows, AIX, HP IA, HP PA-RISC, Linux, SUN, TRU64
DB 서버	ORACLE 8i ~, SQL Server 2012 ~, DB2 9.x ~, Tibero 4 SP 1 ~, MySQL 5.x ~, MariaDB 5.5, 10.0, 10.1, Cache DB 2009.1 ~, Informix IDS 9.x ~ Sybase ASE 15.7 SP61 ~, Sybase IQ 15.4 ~, CUBRID 2008 R1.3 ~ , PostgreSQL 9.4 ~

12.2 설치 파일의 구성 확인

DA의 설치 파일은 아래와 같은 명칭으로 압축 파일(zip) 형태로 제공된다.

- DA 설치파일: Install_DAmo_DA_v{버전}.zip
  - 압축을 해제 한 뒤, 설치 대상 DB 서버, OS 및 bit에 맞는 설치 파일을 준비한다.

설치 파일에 제품을 사용할 수 있는 '라이선스'는 포함되어 있지 않다.
펜타시큐리티시스템에 문의하여 '라이선스' 파일은 별도로 준비한다.

표 12-2 설치 파일(Install_DAmo_DA_v{버전}.zip)을 압축 해제 시, 디렉터리의 구성

구성	설명
_SampleScpsFiles	SG-KMS 연동 없이 암복호화를 테스트할 수 있는 테스트 키 파일
_TestAgentKeyPair	CLI에서 사용할 수 있는 테스트 키 쌍
Altibase	DA-ALT 제품의 설치 바이너리 폴더
Cache	DA-CDB 제품의 설치 바이너리 폴더
Cubrid	DA-CUB 제품의 설치 바이너리 폴더
DB2	DA-DB2 제품의 설치 바이너리 폴더
Informix	DA-IFX 제품의 설치 바이너리 폴더
MySQL	DA-MYQ 제품의 설치 바이너리 폴더
Oracle	DA-ORA 제품의 설치 바이너리 폴더
Postgres	DA-PGS 제품의 설치 바이너리 폴더
SQL Server	DA-MSQ 제품의 설치 바이너리 폴더

구성	설명
SybaseASE	DA-SYB 제품의 설치 바이너리 폴더
SybaseIQ	DA-SIQ 제품의 설치 바이너리 폴더
Tibero	DA-TIB 제품의 설치 바이너리 폴더

12.2.1 DB 서버 및 운영체제 별 SQL파일 구성

각 DB 서버 및 운영체제 별 SQL파일 구성은 다음과 같다. 다음 장에서 각 DB별로 SQL파일 설치 방법을 설명한다.

표 12-3 DB 서버 및 운영체제 별 SQL파일

DB 서버 종류	Linux 일 경우	Windows 일 경우
Oracle	000.da_user.pkg 000.da_user.sql 001.inner_function.ora(c 버전 설치 시) 002.user_interface.ora(JAVA 버전 설치 시) 002.user_interface_java.ora 003.grant_execute_functions.sql(JAVA 버전 설치 시) 005.securej_privilege.sql install_make.sh	000.da_user.pkg 000.da_user.sql 001.inner_function.ora(c 버전 설치 시) 002.user_interface.ora(JAVA 버전 설치 시) 002.user_interface_java.ora 003.grant_execute_functions.sql(JAVA 버전 설치 시) 005.securej_privilege.sql 009.da_test.sql
MYSQL	001.inner_function.mys 002.user_interface. mys 003.grant_execute_functions.sql 009.da_test.sql install_make.sh	001.inner_function.mys 002.user_interface.mys 003.grant_execute_functions.sql
TIBERO	001.inner_function.tbs(c 버전 설치 시) 001.inner_function_java.tbs(JAVA 버전 설치 시) 002.user_interface.tbs(c 버전 설치 시) 002.user_interface_java.tbs(JAVA 버전 설치 시) 003.grant_execute_functions.sql install_make.sh	001.inner_function_java.tbs 002.user_interface_java.tbs 003.grant_execute_functions.sql
INFORMIX	001.inner_function.ifx 002.user_interface. ifx 003.grant_execute_functions.sql 009.da_test.sql install_make.sh	해당 없음
POSTGRESQL	001.inner_function.post 002.user_interface.post 003.grant_execute_functions.post	해당 없음

DB 서버 종류	Linux 일 경우	Windows 일 경우
DB2	001.inner_function.db2 002.user_interface.db2 003.grant_execute_functions.sql install_make.sh	001.inner_function.db2 002.user_interface.db2 003.grant_execute_functions.sql
CUBRID	001.inner_function.sql 002.user_interface.sql	001.inner_function.sql 002.user_interface.sql
SYBASE	001.inner_function.sybase 002.user_interface.sybase 003.grant_execute_functions.sql install_make.sh	해당 없음
SYBASE IQ	001.inner_function.sybiq 002.user_interface. sybiq 003.grant_execute_functions.sql install_make.sh	해당 없음
SQL Server	해당 없음	001.inner_function.sql 002.user_interface. sql 003.grant_execute_functions.sql install_make.bat
Cache DB	SCP.xml install_make.sh	SCP.xml
Altibase	해당 없음	000.da_user.pkg 000.da_user.sql 001.inner_function.sql 002.user_interface. sql install_make.bat

12.2.2 공통 설치 파일 구성

환경에 관계 없이 공통적으로 사용하는 설치 파일은 아래와 같다.

표 12-4 공통 설치 파일 (sql파일 제외)

파일 분류	파일 명	파일 용도
Library 파일	libdamoscpdb.{so|a|sl|dll}	DA 메인 라이브러리. 주로 DBMS External Interface를 담당
	libdamocm-4.0.{so|a|sl|dll}	공통 모듈 라이브러리
	liblogw-0.2.{so|a|sl|dll}	로그를 기록하는 라이브러리
	libcis_cc-3.3.{so|a|sl|dll}	암호화, 복호화 기능을 제공하는 라이브러리
	libcis_ce-3.3.{so|a|sl|dll}	암호화, 복호화를 제외한 부가적인 기능을 제공하는 라이브러리(예: Base64, 인증서 관리, 특성 유지 암호화 등)

파일 분류	파일 명	파일 용도
설정 파일	scpdb_agent.ini	DA 구동시 실행에 필요한 설정정보를 참조
License 파일	damo_lic.cer	DA 구동 시 제품의 유효성을 검증하는데 사용
Agent key 파일	damo_agt_site.cer	SG-KMS 연동, CLI 프로그램에서 사용하는 인증서 쌍
	damo_agt.cer	
	damo_agt.key	
접근제어 파일	acl_cli 파일	DB 의 USER 별로 암·복호 권한을 설정하는데에 사용
	privilege.damo	권한 파일
JAVA class 파일 (Oracle, Tibero, Cubrid 설치 가능)	ScpAgentException.class	예외 처리 Class
	ScpCryptData.class	암호화 복호화 Class
SQL 파일	아래 새로운 표에 DB별로 표기함	

Agent Key 파일은 SG-KMS 연동에 필요한 키 발급를 참고하여 발급 받는다.

DA-PGS(PostgreSQL)의 경우, DB 서버 버전에 따라 libdamoscpdb.so 라이브러리 선택

- libdamoscpdb94.so (Postgres 9.4)
- libdamoscpdb95.so (Postgres 9.5)
- libdamoscpdb95AS.so (EDB Postgres 9.5)
- libdamoscpdb96AS.so (EDB Postgres 9.6)
- libdamoscpdb10.so (Postgres 10)

12.3 환경변수 설정

DA를 설치할 운영체제에 환경변수 DA_INST_HOME를 설정한다. 이 매뉴얼에서는 제품 설치 경로를 아래와 같이 가정하여 설명한다.

- Linux 환경일 경우: /home/dbms_api
- Windows 환경일 경우: E:₩dbms_api

주의) DA_INST_HOME 설정 시, 주의 사항

- 리눅스의 경우 "/root" 디렉토리로 설정을 권장하지 않는다.
- 윈도우의 경우 "바탕화면"으로 설정을 권장하지 않는다.

위의 경로로 설정할 경우 접근 권한 등의 이유로 문제가 발생할 가능성이 존재한다.

12.3.1 환경변수 설정 - Linux 환경일 경우

DA_INST_HOME 환경변수에 DA의 설치 디렉터리를 설정한다.

```
.profile을 사용하는 경우
export DA_INST_HOME=/home/dbms_api


.cshrc를 사용하는 경우
setenv DA_INST_HOME=/home/dbms_api

export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$DA_INST_HOME
```

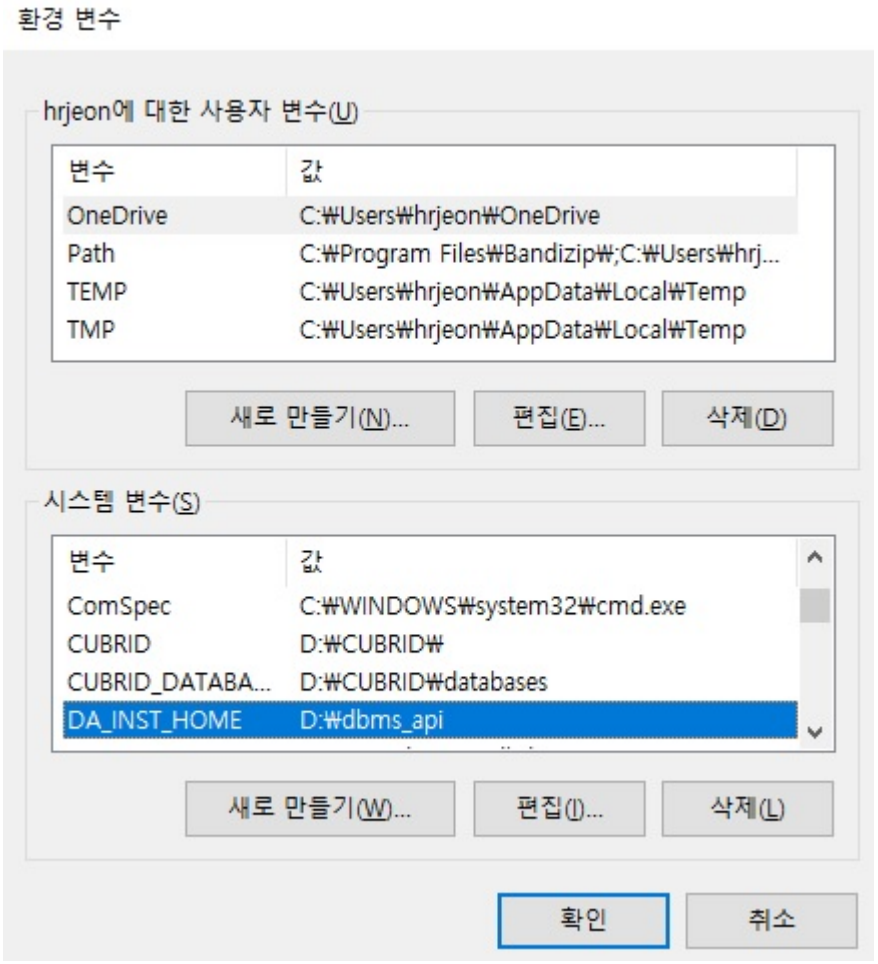
표 12-5 운영 체제별 라이브러리 PATH 명칭

운영 체제	라이브러리 PATH 명칭
HP_UX	SHLIB_PATH
AIX	LIBPATH
LINUX, SUN	LD_LIBRARY_PATH

12.3.2 환경변수 설정 - Windows 환경일 경우

[탐색기->내컴퓨터->등록정보->고급 탭->환경변수] 를 선택하여 나타나는 환경변수 설정 다이얼로그에서 시스템변수 DA_INST_HOME 변수와 값을 추가한다.

그림 12-2 DA_INST_HOME



12.4 DA의 설치 파일 복사

$DA_INST_HOME 디렉터리에 위 [설치 파일의 구성 확인]에서 나열된 파일들을 복사한다.

12.5 라이선스 파일 설정

$DA_INST_HOME 디렉토리에 라이선스 파일(damo_lic.cer)을 복사한다.


라이선스 파일명이 damo_lic.cer가 아닌 다른 이름으로 저장되어 있다면 변경해야 한다.

12.6 설치 파일의 접근 권한 부여 (Linux 설치 시)

Linux 사용자 계정에 DA 설치 파일(라이브러리, 실행 파일, 디렉토리)의 접근 권한을 부여한다. Windows에서 제품을 설치 할 경우, 이 과정은 생략한다.

```
$> cd $DA_INST_HOME
$> chmod 755 lib* acl_cli sql/install_make.sh
```

12.7 (SG-KMS 연동 시) SG-KMS에서 DA 정보 등록 및 연동에 필요한 키 내보내기

DA는 데이터를 암·복호화하기 위해 '암호화 키'가 필요하다. '암호화 키'를 얻기 위해서는 SG-KMS를 연동하거나 SCPS라는 암호화 키 파일을 이용할 수 있는데,
다음 설명은 SG-KMS 연동을 위해 SG-KMS에서 DA 정보를 등록하고 연동에 필요한 키를 내보내는 방법에 대해서 설명한다.

12.7.1 사전 준비

12.7.1.1 지원 SG-KMS 버전

DA와 연동 가능한 SG-KMS 버전은 다음과 같다.

표 12-6 연동 가능한 SG-KMS 버전

SG-KMS Major version	SG-KMS Minor version
SG-KMS v3.0	v3.0.9.0 이상 연동 가능
SG-KMS v4.0	v4.0.104.5 이상 연동 가능

SG-KMS v2.3 연동은 미지원한다.

12.7.1.2 연동 전 점검 항목

SG-KMS 연동을 위해서는 다음과 같이 사전 준비가 필요하다.

- SG-KMS 관리도구
- SG-KMS 관리도구에 접속할 수 있는 ID와 비밀번호
- SG-KMS 매뉴얼

이 매뉴얼에서는 SG-KMS의 키 종류 및 SG-KMS 관리도구 사용 방법에 대한 내용은 다루지 않으므로 **[SG-KMS 매뉴얼]**을 참고한다.

12.7.1.3 SG-KMS 연동에 필요한 키 발급

DA와 SG-KMS의 연동을 위해서는 먼저 SG-KMS에 DA를 Agent로 등록해야한다. 등록하는 절차는 SG-KMS 사용자설명서의 '대칭키를 사용하는 D'Amo Agent 등록 안내서'를 참고한다.

[D'Amo Agent 키] 파일, [Agent ID(CN)]와 [서비스 ID]정보는 DA와 SG-KMS 연동을 위한 설정 과정에서 필요하다. SG-KMS 관리자는 DA 설치 엔지니어에게 안전하게 전달한다.

SG-KMS 관리자에게 받은 D'Amo Agent의 인증서 및 키 파일 명을 다음과 같이 변경후 $DA_INST_HOME/key 디렉터리에 복사한다.

표 12-7 SG-KMS 연동에 필요한 키 목록

키 구분	생성된 키 파일명	변경할 키 파일명
사이트 키	damo-site_{발행기관/부서명}-SITE_V3.cer	damo_agt_site.cer
Agent 키	damo-scp_SITE_V3-{Agent 이름}.cer	damo_agt.cer
	damo-scp_SITE_V3-{Agent 이름}.key	damo_agt.key
	damo-scp_SITE_V3-{Agent 이름}.spin	damo_agt.spin

Agent 키 경로 지정, 키 이름 변경은 반드시 필요한 작업은 아니지만 관리를 위해 변경하는 것을 권장한다.

12.8 설정 파일(scpdb_agent.ini) 수정

DA의 운용을 위해 사용하는 scpdb_agent.ini 설정 파일 수정 방법에 대해 설명한다. DA는 SG-KMS를 이용하거나 SCPS 파일을 이용하여 암호화 키를 얻기 때문에 고객의 환경에 맞게 설정해야 한다.

$DA_INST_HOME 디렉터리에 있는 scpdb_agent.ini 설정 파일에서 아래 항목의 값을 수정한다.

1. 설정 파일의 [KEYINFO] 항목 - [KEY1]에 암호화 키 정보를 입력한다.

```
[KEYINFO]
KEY1=암복호화 하려는 암호화 키 정보를 입력한다.
//키 정보는 다음과 같은 값을 입력할 수 있다.
///ServiceID: SG-KMS에 생성한 서비스 ID를 입력한다.
///SCP_FilePath: SG-KMS에서 서비스 내보내기를 통해 발급한 SCPS 파일의 절대 경로 및 파일명, 확장자를 입력한다.

//*예제 - Windows 경우
KEY1=DA_AES256
KEY2=C:\DA\Policy\S_AES128.SCPS\DA_AES256.scps
KEY3=DA_AES256,C:\DA\Policy\S_AES128.SCPS\DA_AES256.scps

//*예제 - Linux 또는 Unix 경우
KEY1=DA_AES256
KEY2=/home/dbms_api/key/DA_AES256.scps
KEY3=DA_AES256,/home/dbms_api/key/DA_AES256.scps
```

ServiceID를 입력 할 경우 SG-KMS와 통신을 통해 암호화 정책을 가져와 데이터 암복호화를 수행한다.

SCP_FilePath를 입력 할 경우 KMS와 통신하지 않고 서버의 SCP 파일을 통해 암호화 정책을 가져와 데이터 암복호화를 수행한다.

ServiceID와 SCPS 파일명을 동시에 입력 시, SG-KMS와 네트워크 연결 실패 하면 SCPS 파일을 이용하여 암호화 한다.
ServiceID와 SCPS 파일명 사이에 공백이 있으면 SCPS 파일을 읽을 수 없다. 따라서 예제와 같이 띄어쓰기를 하지 않고 ServiceID와 SCPS 파일 경로를 붙여서 입력한다.

2. 설정 파일의 [Server], [Server2] 항목을 수정한다.

```
[Server]
ServerIP: SG-KMS의 IP를 입력한다.
ServerPort: SG-KMS의 포트번호를 입력한다. 기본값은 2525이다.

//*예제 - 모든 OS 공통
```

```
ServerIP=192.168.22.25
ServerPort=2525
```

```
[Server2]
ServerIP: 이중화를 위한 2번 SG-KMS의 IP를 입력한다.
ServerPort: 이중화를 위한 2번 SG-KMS의 포트번호를 입력한다. 기본값은 2525이다.

//*예제 - 모든 OS 공통
ServerIP=192.168.22.26
ServerPort=2525
```

ServerIP는 최소 1개 ~ 최대 10개까지 등록이 가능하다.

3. 설정 파일의 [AGENT] 항목을 설정한다.

```
[AGENT]
AgentID=SG-KMS 관리도구에서 설정한 D'Amo Agent의 Agent ID
LogDir=로그가 저장될 디렉터리 위치
LogLevel=로그가 남는 수준
SiteCertFilePath=SG-KMS 장비에서 설정한 해당 장비의 사이트 공개키(.cer)
CertFilePath=SG-KMS 관리도구에서 설정한 D'Amo Agent의 공개키(.cer)
KeyFilePath=SG-KMS 관리도구에서 설정한 D'Amo Agent의 비공개키(.key)
SPIN=SG-KMS 관리도구에서 설정한 D'Amo Agent의 SPIN으로, damo-scp_SITE_V3-[Agent이름].spin 파일의 값

//[예제]
AgentID=DA
LogDir=/home/dbms_api/log
LogLevel=4
SiteCertFilePath=/home/dbms_api/key/damo_agt_site.cer
CertFilePath=/home/dbms_api/key/damo_agt.cer
KeyFilePath=/home/dbms_api/key/damo_agt.key
SPIN=XaMh1ylXUh123XUh
```

로그가 남는 수준(LogLevel)에는 아래 5가지 숫자 입력이 가능하며, 각 값의 설정은 다음과 같다.
- 0: 아무 로그도 남기지 않을 경우
- 2: 경고 로그를 파일에 기록
- 4: 에러 로그와 경고 로그를 파일에 기록

- 6: 정보 로그, 에러 로그, 경고 로그를 파일에 기록
- 8: 디버그, 정보 로그, 에러 로그, 경고 로그를 파일에 기록

제품 운영 중 scpdb_agent.ini 파일을 수정하면 CONFIG_REINIT() 함수를 호출해야 변경된 내용이 적용된다.

12.9 CLI에서 권한 설정

$DA_INST_HOME 디렉터리에서 acl_cli 파일을 실행하여 USER 단위로 암/복호화 권한을 설정한다. USER 는 DB의 소유자명이고, KEY는 설정 파일(scpdb_agent.ini)의 [KEYINFO] 중 ALIAS 값(예: KEY1)이다.

CLI 명령어를 자세히 보려면 help 명령어를 실행한다.

```
CLI 실행 방법
$> cd $DA_INST_HOME
$> ./acl_cli  - start
Enter the PIN of CLI-key. : damo_agt.key 의 비밀번호
```

```
권한 추가할 경우
D'Amo > SET PRIV ENC [USER]"[KEY]"1"1
D'Amo > SAVE ALL
D'Amo > SHOW ALL

예제)
D'Amo > SET PRIV ENC SCOTT"KEY1"1"1
D'Amo > SET PRIV ENC SCOTT"KEY2"1"1
```

scpdb_agent.ini 설정 파일이 아래 예제와 같을 경우, CLI에서 권한 설정할 때 입력해야 하는 2번째 [KEY] 인자 값에는 KEY1을 입력해야 한다. (※ARIA256을 입력하는 것이 아님)

#scpdb_agent.ini 설정 파일 예제
[KEYINFO]
KEY1=ARIA256

```
권한 삭제할 경우
D'Amo > DEL PRIV ENC [USER]"[KEY]
D'Amo > SAVE ALL
D'Amo > SHOW ALL
```

CLI 에서 권한을 추가하거나 삭제 한 경우 반드시 SAVE ALL 명령어를 실행하며 SHOW ALL 명령어를 이용하여 적용 여부를 확인 한다.

12.10 DB 서버 사전 확인

DA를 Altibase 환경에서 설치 하기 전에 다음과 같은 사항들을 확인한다.

- DB 엔진 설치 계정에 폴더 생성
- DB 사용자로 SCP 계정 생성 (권고사항)
- DB 서버의 DBA 권한 계정

12.11 라이브러리 설정

[Altibase 설치 경로]₩lib 경로에 라이브러리(damoscpdb.dll, damocm-4.0.dll, cis_cc-3.3.dll, cis_ce-3.3.dll, logw-0.2.dll)을 복사한다.

12.12 sql 파일 생성

1. $DA_INST_HOME/sql 디렉터리에서 install_make.sh을 이용하여 설치할 sql 파일을 생성한다. D_INI 는 설정 파일(scpdb_agent.ini)의 경로다.

```
$> cd %DA_INST_HOME%\sql
$> install_make.bat D_INI
```

```
[예제]
$> install_make.bat C:\dbms_api
D_INI_PATH is replaced by C:\dbms_api
```

12.13 제품 함수 설치

1. [DA_INST_HOME]₩sql 폴더에서 install_make.bat 파일을 이용하여 sql 파일을 생성한다. 생성된 파일 001.inner_function.alt.sql, 002.user_interface.alt.sql을 DB 접속 후 실행한다.

```
E:\dbms_api\sql\install_make.bat "DA_INST_HOME 경로"
iSQL> @ 000.da_user.pkg
iSQL> @ 000.da_user.sql
iSQL> @ 001.inner_function.alt.sql
iSQL> @ 002.user_interface.alt.sql
```

함수 사용이 필요한 DB 계정에서 위 설치 절차를 모두 수행해야 한다. Altibase는 함수별로 실행 권한(GRANT EXECUTE)을 부여할 수 없기 때문이다.

12.14 제품 설치 확인

DB 서버에서 암/복호화 함수를 호출하여 설치를 성공했는지 확인한다.

```
SELECT ENC_STR('DAMO', 'abc');
(expression)E6878572B3287A049906A8CA57F0207C

1 row(s) retrieved.

SELECT DEC_STR('DAMO', ENC_STR('DAMO', 'abc'));
(expression) abc

1 row(s) retrieved.
```

12.15 제품 운용

12.15.1 함수 설명

DA에서 제공되는 함수와 사용하는 방법에 대해서 설명한다.

12.15.1.1 파라미터 설명

12.15.1.1.1 I_KEY

I_KEY : 암호화/복호화 때 사용하는 암호화 키
- 설정 파일(scpdb_agent.ini)의 [KEYINFO] 중 ALIAS 값
- 아래 [설정 파일 예제]의 경우 ALIAS는 'KEY1'과 'KEY2'이다.
- ALIAS는 SCPS파일로 암호화 할 것인지, SG-KMS의 서비스 ID로 암호화 할 것인지 설정 가능하다.

[설정 파일 예제]
- 설정 파일(scpdb_agent.ini)의 [KEYINFO] 중 ALIAS 값
KEY1=AES256.SCPS
KEY2=ARIA256

[함수 사용 예제]
```
SELECT ENC_B64('KEY1', 'abc') FROM DUAL;
```

12.15.1.1.2 I_DATA

I_DATA : 평문(암호화 함수일 경우), 암호문(복호화 함수일 경우)

DA에서 제공하는 암호화 함수에서 성능 향상을 위해 라이브러리에서 정책 이름(KEYINFO ALIAS)을 바탕으로 Cash하여 동작한다. 정책 이름은 같은데 실제 키(대칭 키)가 다른 정책을 사용하면 Cash에서 이전 키로 복호화를 시도하여 오류가 발생한다. 이 상황을 해결하기 위해서는 데이터베이스 서버 재시작이 필요하다.

표 12-8 DA 함수 (Altibase)

함수 명	입력			출력
ENCSTR	I_KEY	IN 문자열		Hex String 암호문
	I_DATA	IN 문자열 (평문)		
ENCB64	I_KEY	IN 문자열,		Base64 Encording 암호문
	I_DATA	IN 문자열 (평문)		
DECSTR	I_KEY	IN 문자열,		평문
	I_DATA	IN 문자열 (Hex String 암호문)		
DECB64	I_KEY	IN 문자열,		평문
	I_DATA	IN 문자열 (base64 String 암호문)		
INDEXSTR	I_KEY	IN 문자열,		Hex String 암호문
	I_DATA	IN 문자열 (평문),		
	I_TYPE	IN 문자열 '' or 'IX '(Plug-IN 연동 시 사용)		
DECINDEXSTR	I_KEY	IN 문자열,		Hex String 암호문 입력받아 OPE 데이터
	I_DATA	IN 문자열 (암호문),		
	I_TYPE	IN 문자열 '' or 'IX '(Plug-IN 연동 시 사용)		
DECINDEXB64	I_KEY	IN 문자열,		Base64 Encording 암호문 입력받아 OPE 데이터
	I_DATA	IN 문자열 (암호문),		
	I_TYPE	IN 문자열 '' or 'IX '(Plug-IN 연동 시 사용)		
HASHSTR	I_ALOG	IN 숫자,	SHA1 =70	Hex String 해쉬 암호문
			SHA256 =71	
			SHA384 =72	
			SHA512 =73	
			HAS160 =74	
	I_DATA	IN 문자열		
HASHB64	I_ALOG	IN 숫자,	SHA1 =70	Base64 String 해쉬 암호문
			SHA256 =71	
			SHA384 =72	
			SHA512 =73	

		HAS160 =74	
	I_DATA	IN 문자열	
HEXTOB64	I_DATA	IN 문자열 (Hex String 암호문)	base64 Encording 암호문
B64TOHEX	I_DATA	IN 문자열 (base64 Encording 암호문)	Hex String 암호문
CONFIG_REINIT			성공시 'SUCCESS', 그외 에러

12.15.2 함수 호출 예제

1. ENC_STR

```
SELECT ENC_STR('KEY1', 'abc');
```

2. ENC_B64

```
SELECT ENC_B64('KEY1', 'abc');
```

3. DEC_STR

```
SELECT DEC_STR('KEY1', ENC_STR('KEY1', 'abc'));
```

4. DEC_B64

```
SELECT DEC_B64('KEY1', ENC_B64('KEY1', 'abc'));
```

5. INDEX_STR

```
DP 제품에 연동 하지 않을 경우
SELECT INDEX_STR('KEY1', 'abc', '');

DP 제품에 연동 할 경우
SELECT INDEX_STR('KEY1', 'abc', 'IX');
```

6. DEC_INDEX_STR, DEC_INDEX_B64

```
DP 제품에 연동 하지 않을 경우
SELECT DEC_INDEX_STR('KEY1', dbo.ENC_STR('KEY1', 'abc'), '');
SELECT DEC_INDEX_B64('KEY1', dbo.ENC_B64('KEY1', 'abc'), '');

DP 제품에 연동 할 경우
```

```
SELECT DEC_INDEX_STR('KEY1', dbo.ENC_STR('KEY1', 'abc'), 'IX');
SELECT DEC_INDEX_B64(KEY1', dbo.ENC_B64('KEY1', 'abc'), 'IX');
```

7. HASH_STR

```
SELECT HASH_STR( 71, 'abc' );
```

8. HASH_B64

```
SELECT HASH_B64( 71, 'abc' );
```

9. HEXTOB64

```
SELECT HEXTOB64('A305378D8F974F1C1537ED7CB0CB959245D1AC31');
```

10. B64TOHEX

```
SELECT B64TOHEX('owU3jY+XTxwVN+18sMuVkkXRrDE=');
```

11. CONFIG_REINIT

```
SELECT CONFIG_REINIT();
```

파트 II.

부록

1.

함수 지원표

DA의 함수마다 지원하는 기능이 다르므로 아래의 표를 참고한다.

1.1 양방향 암호화

양방향 암호화는 평문을 암호화하여 복호화가 가능한 암호화 방식이다. 양방향 암호화 중 일반 암호화는 평문 전체를 암호화하는 방식이며, 부분 암호화는 평문 중 일부를 암호화하는 방식이고, FPE는 평문과 동일한 형식으로 암호화하는 방식이다.

표 2 DA의 함수별 양방향 암호화 지원 여부

함수 이름	양방향 암호화		
	일반	부분	FPE
(ENC)DEC_STR	O	O	O
(ENC)DEC_B64	O	O	O(Str과 동일)

1.2 단방향 암호화

단방향 암호화는 평문을 암호화만 가능하며, 복호화가 불가능한 암호화 방식이다. 단방향 암호화 중 HASH는 일반적인 단방향 암호화 방식이며, HMAC은 암호화 키를 사용한 HASH 암호화 방식이다.

표 3 DA의 함수별 단방향 암호화 지원 여부

함수 이름	단방향 암호화	
	HASH	HMAC
ENC_STR		O
ENC_B64		O
HASH_STR	O	
HASH_B64	O	

2.

D'Amo 용어 정의

여기에서는 IT 일반 용어를 포함하여 D'Amo 제품에서 사용되는 용어에 대해 정의한다.

2.1 D'Amo 공통

D'Amo 제품 공통으로 사용되는 용어 중, 크게 일반적인 개념과 특정 키로 분류하여 정의한다.

2.1.1 개념 및 일반

표 2-1 D'Amo 공통_개념 및 일반

용어	정의
CLI	Command Line Interface의 약자로, 제품의 운용을 위한 설정 작업을 할 수 있는 명령어 입력 기반의 관리도구이다.
관리도구	제품의 운용을 위한 설정 작업을 할 수 있는 GUI(Graphical User Interface) 입력 기반의 관리도구이다.
DCA	D'Amo Contorl Agent의 약자로, 관리도구와 각 제품을 연결하는 서버이다.
D'Amo Agent	D'Amo SG-KMS와 연동하여 데이터 암/복호화를 수행하는 다음의 제품들을 통칭한다. ◦ DP-ORA /DP-MSQ / DE-MYQ / DE-PGS / BA-SCP / KE-LNX 등
Security Library	DBMS에 탑재되는 제품의 구성 요소 중 하나로, 실제 암/복호화를 수행한다.
암호화 컬럼	데이터가 암호화되어 있는 DB의 컬럼이다.
정책(Policy)	암/복호화에 필요한 키와 알고리즘의 정보(암호 알고리즘 ID, IV, 부분 암호화 범위)로 구성되며, 보안 정책이라고도 말한다.
보안관리자	• 회사/조직 내의 DB 보안 책임자, 또는 제품의 운용을 책임지는 담당자를 의미한다. • 사이트 키, 관리자 키를 생성/관리하고, 하위 운용자에게 관리자 키를 배포한다.

2.1.2 키

D'Amo 제품 공통으로 사용되는 각종 키에 대해 정의한다.

표 2-2 D'Amo 공통_키

용어	정의
컬럼 키	테이블의 컬럼을 암호화 할 때 사용하는 대칭키이다.
DB 키	• 비공개키와 공개키로 구성된 PKI 기반의 암호화 키이다. • 컬럼 키를 안전하게 보관하기 위한 암호화에 사용한다.
관리자 키	• CLI를 통한 로그인 시에 사용된다. • '*공개키 쌍*'으로 구성되어 있다. ◦ PKI 인증서 형식(x.509)의 '*공개키*' ◦ 암호화된 '*개인키*' • 보안 관리자가 관리도구에서 '생성/관리' 한다.
사이트 키	• CLI를 통해 모든 공개키 기반의 키 생성 시에 사용된다. • '*공개키 쌍*'으로 구성되어 있다. ◦ PKI 인증서 형식(x509)의 '*공개키*' ◦ 암호화 된 '*개인 키*' • 보안 관리자가 CLI를 통해 생성(1회에 한함)하고 사용한다.
라이선스 키	• 제품에서 사용할 수 있는 기능과 기간을 설정하는데 사용한다. • 당사에서 운영하는 '라이선스 서버'를 통해 해당 키 발급을 신청하여 사용한다. ◦ 라이선스 서버는 별도 문의
사이트 공개키	사이트 공개키(사이트 인증서) 쌍의 일부로 암호화 주체끼리 공유하는 키이다.
사이트 비공개키	사이트 공개키 쌍의 일부로 암호화 시에 비공개로 사용된다.
DB 공개키	DB 키 쌍의 일부로 암호화 주체끼리 공유하는 키이다.
DB 비공개키	DB 키 쌍의 일부로 암호화 시에 비공개로 사용된다.

2.2 D'Amo 제품별

여기에서는 펜타시큐리티에서 정의한 D'Amo 제품 고유의 용어에 대해 정의한다.

2.2.1 DCC

D'Amo Control Center에서 정의한 용어는 다음과 같다.

표 2-3 DCC

용어	정의
DCC 관리도구	D'Amo 제품 운영을 위한 D'Amo Control Center 관리도구이다.
DCC 관리도구 공개키	DCC 관리도구 공개키 쌍의 일부로 암호화 주체끼리 공유한다.
DCC 관리도구 비공개키	DCC 관리도구 공개키 쌍의 일부로 암호화 시 비공개로 사용한다.
DCC 관리도구 키 쌍	• 보안관리자 인증서로 DCC 관리도구를 사용하기 위한 키이다. • 공개키와 비공개키가 한 쌍을 이룬다.
DB 보안관리자	보안관리자 인증서(DCC 관리도구 키)를 이용하여 DCC 관리도구를 사용하는 관리자이다.

2.2.2 SG-KMS_권한

SG-KMS에서 정의한 용어는 다음과 같다.

표 2-4 SG-KMS_권한 관련 용어 정의

용어	정의
보안 관리자	• 회사/조직 내의 DB 보안 책임자, 또는 제품의 운용을 책임지는 담당자를 의미한다. • 사이트 키, 관리자 키를 생성/관리하고, 하위 운용자에게 관리자 키를 배포한다.
로컬 보안 관리자	CLI를 통해, ◦ 제품의 내부 설정을 변경할 수 있는 권한을 가진다. ◦ 원격 보안 관리자를 생성한다.
원격 보안 관리자	• CLI에서 로컬 보안 관리자에 의해 생성된다. • 관리도구를 통해, ◦ 각종 키 및 암호화 정책 등에 대한 '추가/변경/삭제'가 가능하다. ◦ 원격 보조 보안 관리자를 생성한다.
원격 보조 보안 관리자	관리도구를 통해, ◦ 암호화 정책 및 관리자 목록, 로그 등에 대한 조회 권한만 가진다. ◦ 통해 원격 보안 관리자에 의해 생성된다.

2.2.3 SG-KMS_키

SG-KMS 중 각종 키에 대해 정의한 용어는 다음과 같다.

표 2-5 SG-KMS_키 관련 용어 정의

용어	정의
데이터 암호화 키	• 데이터 암호화 및 데이터 HMAC에 사용한다. • 대칭키 형태로 구성되어 있다. • 관리도구에서 '생성/관리' 한다.
비대칭키	• 데이터를 암호화 한다.

용어	정의
	• 관리도구에서 '생성/관리' 한다.
D'Amo Agent 키	• SG-KMS와 D'Amo Agent 연동 시에 사용된다. • 관리도구에서 '생성/관리' 한다.
관리자 키	• CLI를 통한 로그인 시에 사용된다. • '*공개키 쌍*'으로 구성되어 있다. ◦ PKI 인증서 형식(x.509)의 '*공개키*' ◦ 암호화된 '*개인키*' • 보안 관리자가 관리도구에서 '생성/관리' 한다.
사이트 키	• CLI를 통해 모든 공개키 기반의 키 생성 시에 사용된다. • '*공개키 쌍*'으로 구성되어 있다. ◦ PKI 인증서 형식(x509)의 '*공개키*' ◦ 암호화 된 '*개인 키*' • 보안 관리자가 CLI를 통해 생성(1회에 한함)하고 사용한다.
라이선스 키	• 제품에서 사용할 수 있는 기능과 기간을 설정하는데 사용한다. • 당사에서 운영하는 '라이선스 서버'를 통해 해당 키 발급을 신청하여 사용한다. ◦ 라이선스 서버는 별도 문의
SPIN	• Secure Personal Identification Number의 약자로, 보안 개인 식별 번호를 의미한다. • SG-KMS에서는 인증서의 비밀번호를 암호화 한다.

2.3 DB 암호 제품

DB 암호화 제품 내에서 공통으로 사용되는 용어를 정의한다.

2.3.1 암호화 알고리즘의 종류

본 제품에서 사용하는 암호화 알고리즘은 한국 표준(KS)을 포함하여, 아래의 표준 규격에 근거하여 사용한다.

표 2-6 암호화 알고리즘의 종류

표준 규격	알고리즘	설명
NIST[a]	DES (Data Encryption Standard)	• 평문을 64bit로 나눠 56bit의 키를 이용해 다시 64bit의 암호문을 만들어 내는 알고리즘 • 블록 암호(Block Chipher)의 일종으로 1977년 미국 NBS[b]에서 국가 표준으로 정한 대칭키 암호이다. • 현재, DES 보다 Triple-DES가 더 안전한 것으로 알려져 있으며 AES가 새 표준으로 정해져 있다.

표준 규격	알고리즘	설명
	TDES (Triple Data Encryption Standard)	3DES라고도 하며, DES의 키 길이 단점을 보완하기 위해 각 데이터 블록에 DES를 3번 반복하는 암호화 알고리즘
	AES (Advanced Encryption Standard)	• 고급 암호화 표준이라고 불리며, 암복호화 과정에서 동일한 키를 사용하는 대칭키 알고리즘 • DES를 대체하기 위해 2001년 NIST에 채택되었다.
ISO[c] KS[d]	SEED	• 미국에서 수출되는 웹 브러우저 보안 수준이 40bit로 제한됨에 따라 128bit 보안을 위해 별도로 개발된 알고리즘 • 1999년 TTA에서 발표한 한국 표준(KS)으로 지정되었다.
	ARIA	• 경량 환경 및 하드웨어 구현을 위해 최적화 된 Involutional SPN 구조 • ISO 표준인 SEED와 함께 사용되는 국가 표준 128bit의 범용 블록 암호 알고리즘

a NIST(National Institute of Standards and Technology): 미국표준기술연구소
b NBS(National Bureau of Standards): 미국표준규격국, 현재의 NIST
c ISO(International Organization for Standardization): 국제표준화기구
d KS(Korean (Industrial) Standards): 한국공업표준규격

2.3.2 블록 암호(Block Cipher) 운영 모드의 종류

블록 암호는 TDES, SEED, AES 등 대칭 키 블록 암호 알고리즘을 사용할 때 암호화 할 정보가 블록 길이와 다를 경우에 사용하며, 운영 모드의 종류는 아래와 같다.

표 2-7 블록 암호(Block Cipher) 운영 모드의 종류

운영 모드	설명
초기화 벡터 (IV: initialization vector)	블록 암호화 시 첫 블록(128bit)을 암호화할 때 사용하는 데이터를 의미한다.
ECB Mode (electric codebook)	• 전자 코드북 방식으로 블록 암호 운영 모드 중 가장 간단한 구조를 가진다. • 각 블록을 독립적으로 암호화 하며 초기화 벡터가 필요 없다.
CBC Mode (cipher-block chaining)	• 암호 블록 체인 방식으로 1976년 IBM에 의해 개발되었다. • 블록의 평문과 앞 블록 암호문과의 배타적 논리합으로 암호화 한다. 블록 간의 의존 관계를 갖는 모드로서 보안적 특성이 뛰어남
CFB Mode (cipher feedback)	• 암호 피드백 방식으로 CBC를 변형한 모드이다. • 고정 길이의 평문을 고정 길이의 암호문으로 변환하는 비밀 키 암호 방식을 사용하여 송수신에서 같은 길이의 비트 수 단위로 암호화 한다. • 암호화 데이터 길이가 늘어나지 않는 것이 장점이다. • 단, 짧은 길이의 데이터에서 고정된 초기화 벡터(IV)를 사용할 경우 보안적 특성이 떨어질 수 있다.
OFB Mode (output feedback)	평문이 직접 암호화 되지 않는 것은 CFB와 동일하지만 평문 블록과 암호 알고리즘의 출력을 XOR하여 암호문을 생성한다.
CTR	

운영 모드	설명
(Counter)	블록 암호를 스트림 암호로 바꾸는 구조를 가진다.
CFB_BYTE	CFB를 블록 길이 단위가 아닌, 바이트 단위로 처리한 운영 모드이다.

2.3.3 기타 DB 암호 제품의 용어 정의

표 2-8 기타 DB 암호 제품의 용어 정의

용어	설명
BASE64	• 바이너리 데이터(2진수)를 문자 코드의 영향을 받지 않는 공통 아스키(ASCII) 문자로 표현하기 위해 만들어진 인코딩 방식 중 하나이다. • 8bit짜리 바이트 3개를 6bit씩 4개로 쪼개서 Base64 코드 4개로 바꾸어 표현한다. • 메일에서 이미지, 오디오 파일을 보낼 때 이용하는 코딩으로 모든 플랫폼에서 안 보이거나 해독할 수 없는 현상이 발생하지 않도록 공통으로 64개 아스키 코드를 이용하여 2진 데이터를 변환하기 위해 베이스 64를 이용한다. • 따라서 BASE64로 인코딩하면 크기가 원문에서 대략 33% 커진다.
HEXSTRING	• 바이너리 데이터(2진수)를 문자 코드에 영향을 받지 않는 공통 아스키(ASCII) 문자로 표현하기 위해 만들어진 인코딩 방식 중 하나이다. • 1byte 데이터를 16진수 2byte의 아스키 코드로 표현하는 방식으로, HEXSTRING으로 인코딩하면 크기가 2배로 커진다.
Hash	• 임의의 데이터로부터 일종의 짧은 "전자 지문"을 만들어 내는 방법이다. • 해시 함수는 데이터를 자르고 치환하거나 위치를 바꾸는 등의 방법을 사용해 결과를 만들어 내며, 이 결과를 흔히 해시 값(hash value)이라 한다. • 두 개의 데이터를 각각 해시한 후, 그 결과값이 다르다면 그 해시 값에 대한 원본 데이터도 다름을 보장한다. • 대표적인 Hash 함수로는 MD5, SHA1, SHA256, HAS-160 등이 있다.
HMAC	• MAC[a] 을 연산하는 특정 구조로 Hash 함수와 비밀 키의 조합으로 연산 된다. • 송신자와 수신자만 공유하고 있는 키와 데이터(메시지)를 혼합하여 해시값을 만든다. • MAC과 마찬가지로 메시지의 무결성 확인 뿐만 아니라 인증 기능으로도 사용된다.
SHA	• 미국 국가 안전 보장국(NSA)이 1993년에 처음으로 설계했으며 미국 표준 기술 연구소(NIST)에 의해 미국 국가 표준으로 지정된 Hash 함수이다. • SHA 함수군에 속하는 최초의 함수는 공식적으로 SHA라고 불리지만, 나중에 설계된 함수들과 구별하기 위하여 SHA-0이라고도 불린다.
BLOWFISH	• 1993년 브루스 슈나이어가 설계한 키(key)방식의 대칭형 블록 암호이다. • DES의 대안으로서 다른 알고리즘에 관련된 제약과 문제를 해결하기 위해 고안되었다. • BLOWFISH는 64비트 블록 크기, 또 32비트에서 최대 448비트에 이르는 가변 키 길이를 갖추고 있다. • 16라운드 파이스텔 암호로서 대형 키 의존 S박스를 이용한다. 구조는 수정된 S박스를 사용하는 CAST-128과 비슷하다.
RC4	로널드 라이베스트(Ron Rivest)가 만든 스트림 암호로, 전송 계층 보안(TLS)이나 WEP등의 여러 프로토콜에 사용되어 왔다. 하지만 이후 여러 연구를 통해 취약한 것으로 밝혀졌으며, RC4를 사용한 WEP의 경우 해당 프로토콜의 사용을 권장하지 않는다.

용어	설명
PENTA_FFX	• NIST_FFX를 보완하여 펜타시큐리티의 자체 기술로 개발된 알고리즘 • 평문과 암호문에서 제외될 문자열의 타입을 사용자가 직접 지정한다.
PADDING	• 블록의 고정된 길이를 사용하기 위해 블록의 길이를 고정된 값으로 맞춤 • CBC 모드인 경우만 선택 가능 • NON-PADDING: 블록의 고정된 길이를 사용하지 않음

a MAC: Message Authentication Code

2.3.4 FPE 알고리즘의 암호화 규칙

FPE(format preserving encryption)란, 암호화 알고리즘 중 암호문과 평문의 형태(format)를 그대로 보존하기 위한 암호 기술 중 하나이다.

아래의 문자 정의를 토대로, 각 알고리즘의 암호화 규칙을 설명한다.

- **대/소문자**: 영 대문자, 영 소문자
- **2Byte 문자열**: 한글, 일본어, 한자 등

표 2-9 FPE 알고리즘의 암호화 규칙

알고리즘	설명
FPE_NUM2SYMBOL	• 숫자가 포함된 문자열을 숫자가 없는 문자열로 암호화 한다. • 숫자를 지정된 특수 문자의 형태로 암호화 하고, 그 외의 나머지 문자는 암호화에서 제외된다. ◦ 단, 입력값에 지정된 특수 문자가 있는 경우에는 오류가 발생한다. ◦ 지정된 특수 문자: # $ % ^ = 〈 〉 ? ~ ;
FPE_NUM2SYMBOL2	• 숫자가 포함된 문자열을 숫자가 없는 문자열로 암호화 한다. • 숫자를 지정된 특수 문자의 형태로 암호화 하고, 그 외의 나머지 문자들은 암호화 제외된다. ◦ 단, 입력값에 지정된 특수 문자가 있는 경우에는 오류가 발생한다. ◦ 지정된 특수 문자: # $; ? [ ] ^ { } 〈
FPE_CHAR2CHAR	특수 문자를 포함하는 문자를, 특수 문자를 포함하는 문자로 암호화 한다. ◦ 숫자, 영 대/소문자, 특수 문자 - 숫자, 영 대/소문자, 특수 문자
FPE_CHAR2CHAR_S	• FPE_CHAR2CHAR 운영 모드와 동일하지만 공백은 암호화 하지 않는다. • 암호화 대상에 * | 과 같은 특수 문자는 포함할 수 없다. ◦ 숫자, 영 대/소문자, 특수 문자 - 숫자, 영 대/소문자, 특수 문자
FPE_UCHAR2UCHAR	• 특수 문자를 포함하는 영 대문자를, 특수 문자를 포함하는 영 대문자로 암호화 한다. ◦ 숫자, 영 대문자, 특수 문자 - 숫자, 영 대문자, 특수 문자
FPE_UCHAR2UCHAR_S	• FPE_UCHAR2UCHAR 운영 모드와 동일하지만 공백은 암호화 하지 않는다. • 암호화 대상에 * | 과 같은 특수 문자는 포함할 수 없다. ◦ 숫자, 영 대문자, 특수 문자 - 숫자, 영 대문자, 특수 문자

알고리즘	설명
FPE_NUM2CHAR	• 숫자와 지정된 특수 문자(- 공백)를, 특수 문자를 포함하는 문자로 암호화 한다. ◦ 숫자, 지정된 특수 문자(- 공백) - 숫자, 영 대/소문자, 특수 문자
FPE_NUM2RRN	• 주민등록번호 형태의 숫자를 주민등록번호 형태가 아닌 숫자로 암호화 한다. ◦ 숫자 - 숫자, 암호화 대상은 숫자만 입력 가능 • 주의 사항 ◦ 암호화 조건: 길이가 3이고 세 번째 자릿수가 0 또는 1인 경우만 ◦ 그 외는 입력된 평문값을 그대로 결과값으로 내보내고 성공을 리턴
FPE_KOR2KOR	• 특수 문자를 포함하는 한글을, 특수 문자를 포함하는 한글로 암호화 한다. ◦ 숫자, 영 대/소문자, 특수 문자, 2Byte 문자열 - 숫자, 영 대/소문자, 특수 문자, 2Byte 문자열
FPE_KOR2KOR_S	• FPE_KOR2KOR 운영 모드와 동일하지만 공백은 암호화 하지 않는다. • 암호화 대상에 * | 과 같은 특수 문자는 포함할 수 없다. ◦ 숫자, 영 대/소문자, 특수 문자, 2Byte 문자열 - 숫자, 영 대/소문자, 특수 문자, 2Byte 문자열
FPE_PASSPORTID	• 여권 번호 형식의 데이터를 암호화 한다. • 여권 번호는 아래의 두 형식 모두 지원한다. ◦ 숫자, 영 대문자 - 숫자, 영 대문자, 지정된 특수 문자(! @ # $ % - ? = +)) ◦ 구 여권 번호: 공관 부호(2자리) + 여권 번호(7자리) ◦ 신 여권 번호: 여권의 종류(1자리) + 여권 번호(8자리)
FPE_NUM2CHAR_SYMBOL	• FPE_NUM2CHAR 운영 모드와 동일하며, 숫자를 문자로 암호화 한다. • 공백, 특수 문자는 암호화 하지 않는다. ◦ 숫자 - 숫자, 영 대/소문자
FPE_CHAR2CHAR_SYMBOL	FPE_CHAR2CHAR 운영 모드와 동일하며 공백, 특수 문자는 암호화 하지 않는다. ◦ 숫자, 영 대/소문자 - 숫자, 영 대/소문자
FPE_UCHAR2UCHAR_SYMBOL	FPE_UCHAR2UCHAR 운영 모드와 동일하며, 공백, 특수 문자는 암호화 하지 않는다. ◦ 숫자, 영 대문자 - 숫자, 영 대문자
FPE_KOR2KOR_SYMBOL	FPE_KOR2KOR 운영 모드와 동일하며 공백, 특수 문자는 암호화 하지 않는다. ◦ 숫자, 영 대/소문자, 2Byte 문자열 - 숫자, 영 대/소문자, 2Byte 문자열
FPE_KOR2UCHARKOR_SYMBOL	한글을 영 대문자, 한글로 암호화하며 공백, 특수 문자는 암호화하지 않는다. ◦ 숫자, 영 대문자, 2Byte 문자열 - 숫자, 영 대문자, 2Byte 문자열
FPE_ACCOUNT_SYMBOL	• 계좌번호 형식의 데이터를 암호화 한다. • 숫자, 영 대문자, 지정된 특수 문자를 제외한 모든 문자는 암호화 하지 않는다. ◦ 숫자, 영 대문자 - 영 대문자, 지정된 특수 문자 ◦ 지정된 특수 문자: # $ % ^ = < > ? ~ ;

3.

DA 오류 코드 및 메시지

3.1 오류 코드 및 메시지

3.1.1 성공인 경우

다음 값의 경우에는 정상적으로 동작하는 상태이다.

표 3-1 오류 코드 - 성공

Error Code Number	Error Code Message	DESCRIPTION	Handle
0	SCPDB_SUCCESS	성공	-
118	SCPDB_ERR_ALREADY_INITIALIZED	init 함수 호출 시 이미 초기화 되어 있을 때 발생하며 정상적인 사용이 가능한 상태임을 나타낸다.	-

3.1.2 라이선스 오류

다음 값의 경우에는 라이선스로 인하여 발생한 오류 코드이다.

표 3-2 오류 코드 - 라이선스 오류

Error Code Number	Error Code Message	DESCRIPTION	Handle
	SCPDB_ERR_LICENS	라이선스 파	scpdb_agent.ini 파일의 LicenseFilePath의 값을 확인하여 제대로

Error Code Number	Error Code Message	DESCRIPTION	Handle
121	E_FILE_NOT_FOUND	일을 찾을 수 없음	된 라이선스 파일의 경로를 입력한다.
122	SCPDB_ERR_LICENSE_INVALID_DATE	라이선스 기간 오류	라이선스의 파일의 기간을 확인한다. 기간이 종료되었거나 현재의 시간이 시작 시간보다 이전의 경우에도 발생할 수 있다.
123	SCPDB_ERR_LICENSE_INVALID_NAME	라이선스 제품명 오류	라이선스에 명시되어 있는 제품명을 확인한다. OU 값이 SCP For DB Encryption Ver 2.0로 시작하는지 확인한다.
124	SCPDB_ERR_LICENSE_INVALID_HOST	라이선스 호스트(IP) 오류	라이선스에 발급된 IP와 BA-SCP가 설치된 서버의 IP가 동일한지 확인한다.
125	SCPDB_ERR_LICENSE_INVALID_CPU_COUNT	라이선스 코어개수 오류	라이선스에 발급된 CPU 수와 BA-SCP가 설치된 서버의 CPU 갯수보다 많은지 확인한다.
126	SCPDB_ERR_LICENSE_BUNDLE	라이선스 기능 제약	현재 발급된 번들 라이선스에 의해 기능이 제약되어 발생하거나, 라이선스에 의해 기능이 제약되어 발생한다. 라이선스를 다시 문의한다.

3.1.3 설정 파일(scpdb_agent.ini) 오류

다음 값의 경우에는 scpdb_agent.ini 파일의 문제로 발생한 오류 코드이다.

표 3-3 오류 코드 - 설정 파일(Scpdb_agent.ini) 오류

Error Code Number	Error Code Message	DESCRIPTION	Handle
104	SCPDB_ERR_CONFIG_READ	설정 파일에 필요 필드값이 없음	scpdb_agent.ini 파일에 누락된 부분이 없는지 확인한다.
105	SCPDB_ERR_CONFIG_INVALID_SERVER_IP	설정 파일에 잘못된 설정 - [AGENT] ServerIP	입력값을 확인한다.
106	SCPDB_ERR_CONFIG_INVALID_SERVER_PORT	설정 파일에 잘못된 설정 - [AGENT] ServerPort	입력값을 확인한다. 허용되는 값은 0 ~ 65535이다.
107	SCPDB_ERR_CONFIG_INVALID_LICENSE_FILE_PATH	설정 파일에 잘못된 설정 - [AGENT] LicenseFilePath	입력값을 확인한다. LicenseFilePath 입력값은 255자까지 가능하다.
108	SCPDB_ERR_CONFIG_INVALID_LOG_DIR	설정 파일에 잘못된 설정 - [AGENT] LogDir	입력값을 확인한다. LogDir 입력값은 255자까지 가능하다.
109	SCPDB_ERR_CONFIG_INVALID_LOG_LE	설정 파일에 잘못된 설정	입력값을 확인한다.

Error Code Number	Error Code Message	DESCRIPTION	Handle
	VEL	- [AGENT] LogLevel	허용되는 값은 0, 4, 6, 8 입니다.
110	SCPDB_ERR_CONFIG_INVALID_KEYCACHE_MAX	설정 파일에 잘못된 설정 - [AGENT] CacheMax	입력값을 확인한다. 허용되는 값은 1 ~ 999까지 가능하다 기본값은 50이다.
111	SCPDB_ERR_CONFIG_INVALID_AGENTID_CERT_FILE_PATH	설정 파일에 잘못된 설정 - [AGENT] SiteCertFilePath - [AGENT] CertFilePath	입력값을 확인한다.
112	SCPDB_ERR_CONFIG_INVALID_AGENTID_KEY_FILE_PATH	설정 파일에 잘못된 설정 - [AGENT] KeyFilePath	입력값을 확인한다.
113	SCPDB_ERR_CONFIG_INVALID_AGENTID_SPIN	설정 파일에 잘못된 설정 - [AGENT] SPIN	입력값을 확인한다.
114	SCPDB_ERR_CONFIG_INVALID_AGENTID	설정 파일에 잘못된 설정 - [AGENT] AgentID	입력값을 확인한다. AgentID에 입력값의 길이가 255보다 길 경우에 발생한다.
115	SCPDB_ERR_CONFIG_INVALID_PKP_DATA_VERSION	설정 파일에 잘못된 설정 - [AGENT] PKP	입력값을 확인한다. 허용되는 값은 100, 102, 200, 202이다.
116	SCPDB_ERR_CONFIG_INVALID_ACL_PKP_DATA_VERSION	설정 파일에 잘못된 설정 - [AGENT] AclPKP	입력값을 확인한다. 허용되는 값은 100, 102, 200, 202이다.
117	SCPDB_ERR_CONFIG_INVALID_PKD_DATA_VERSION	설정 파일에 잘못된 설정 - [AGENT] PKD	입력값을 확인한다. 허용되는 값은 101 ~ 109이다.

3.1.4 SG-KMS 통신 오류

다음 값의 경우에는 SG-KMS와 통신 문제로 발생한 오류 코드이다.

표 3-4 오류 코드 - SG-KMS 통신 오류

Error Code Number	Error Code Message	DESCRIPTION	Handle
	SCPDB_ERR_SOC	SOCKET 연결 실패	• SG-KMS의 IP와 Port를 확인하고, 특히 P

Error Code Number	Error Code Message	DESCRIPTION	Handle
702	K_CONNECT	- KMS는 살아 있는데, Port가 닫혀있을 경우	ort를 제대로르 입력했는지 확인한다. • SG-KMS가 잘 동작하고 있는지 확인한다.
704	SCPDB_ERR_SOCK_RECEIVE	SOCKET 받기 실패 - KMS에 연결은 되어 있지마, 응답받은 메세지가 이상한 경우 - KMS에 등록되지 않는 AgentID로 요청한 경우	• scpdb_agent.ini 파일의 AgentID가 제대로 입력되었는지 확인한다. • SG-KMS의 로그를 확인한다.
707	SCPDB_ERR_SOCK_CONNECT_TIMEOUT	SOCKET 연결 타임아웃 - KMS가 죽었거나, 방화벽 등으로 응답을 못 받는 경우 - scpdb_agent.ini의 KMS 정보가 제대로 입력되지 않는 경우	• SG-KMS의 IP와 Port를 확인하고 현재 SG-KMS가 잘 동작하고 있는지 확인한다. • 서버에 방화벽으로 응답을 받지 못하는지 확인한다.
709	SCPDB_ERR_SOCK_RECV_TIMEOUT	SOCKET 받기 타임아웃 - KMS에 연결은 되었지만, 네트워크 문제 or KMS 부하 등으로 응답을 못 받는 경우	• SG-KMS의 IP와 Port를 확인하고 현재 SG-KMS가 잘 동작하고 있는지 확인한다. • SG-KMS의 부하를 확인해 본다.
813	SCPDB_ERR_GETSYMKEY_RES__IVLD_DATA	SG-KMS와 호환되지 않는 버전	설정 파일(Scpdb_agent.ini)의 [AGENT] > PKD 버전을 SG-KMS에서 지원하는 버전으로 변경한다.
903	SCPDB_ERR_EXPIRE_PKD_PROTOCOL_MESSAGE	SG-KMS와 서버와의 시간이 차이 날 경우	SG-KMS나 SA의 시간을 확인하여 서로 시간을 맞춘다.
1201	SCPDB_ERR_NOTMATCH_ROLE	잘못된 요청 - AgentID에 포함된 암호화 서비스 ID가 아님	SG-KMS 관리도구의 암호화 서비스 ID를 확인한다.
1202	SCPDB_ERR_INVALID_AGENTID	잘못된 요청 - 등록되지 않은 AgentID	scpdb_agent.ini의 AgentID값을 제대로 입력하거나, SG-KMS에 AgentID를 등록한다.
1203	SCPDB_ERR_INVALID_AGENTIP	잘못된 요청 - 허용되지 않는 AgentIP	SG-KMS에 AgentIP를 등록한다.
1205	SCPDB_ERR_NOT_FOUND_AGENT_KEYPAIR	잘못된 요청 - SG-KMS Agent 키 쌍 오류	설정 파일(Scpdb_agent.ini)에 있는 CertFilePath, KeyFilePath의 파일를 확인한다.
1206	SCPDB_ERR_NOT_FOUND_SERVICE_AGENT	잘못된 요청 - 등록되지 않은 AgentID	scpdb_agent.ini의 AgentID값을 제대로 입력하거나, SG-KMS에 AgentID를 등록한다.
1207	SCPDB_ERR_NOT_FOUND_SERVICE_COLUMN	잘못된 요청 - 컬럼 없음	SG-KMS 관리도구의 컬럼 정보 및 BA-SCP에서 요청한 컬럼 정보를 확인한다.
1208	SCPDB_ERR_NOT_FOUND_SERVICE_ALIAS	잘못된 요청 - 암호화 서비스 ID 없음	SG-KMS 관리도구의 암호화 서비스 ID 및 BA-SCP에서 요청한 암호화 서비스 ID를 확인한다.

3.1.5 API 사용 오류

다음 값의 경우에는 BA-SCP API 사용 시 잘못된 사용으로 인하여 발생한 오류 코드이다

표 3-5 오류 코드 - BA-SCP API 사용 오류

Error Code Number	Error Code Message	DESCRIPTION	Handle
101	SCPDB_ERR_NOT_INITIALIZED	초기화되지 않음 - Init 함수 호출없이 암·복호화 함수를 호출한 경우 - Init 함수 호출 후 결과값(오류코드)을 제대로 확인 안한 경우	init 함수가 정상적으로 호출되었는지 확인한다.
130	SCPDB_ERR_CIS_INIT	CIS_CC가 초기화 오류	-
131	SCPDB_ERR_LOG_INIT	LogWriter 초기화 오류	설정 파일(Scpdb_agent.ini)의 LogLevel을 8로 변경하여 디버그 로그를 남겨서 LogWriter 초기화 오류 코드를 확인하여 처리한다.
136	SCPDB_ERR_REINIT_MULTIFUL_CALL_PREVENT_PERIOD	ReInit 함수 중복호출 오류	마지막으로 ReInit 함수가 호출된지 10초가 지나지 않았을 경우에 발생한다. 해당 오류에 대해 예외처리를 한다.
200	SCPDB_ERR_ALLOCATION	메모리 할당 오류	-
201	SCPDB_ERR_INSUFFICIENT_ALLOC_LEN	할당된 메모리 길이 부족	-
202	SCPDB_ERR_INVALID_INPUT	잘못된 입력 - 입력 값이 입력되지 않을 경우 (NULL 입력) - 입력값의 길이 0일 경우	API 함수 호출이 정상적으로 이루어져 있는지 sample code와 비교하여 확인한다.
203	SCPDB_ERR_INVALID_INPUT_LEN	잘못된 입력 길이 - 허용하는 입력 길이를 초과될 경우	API 함수 호출이 정상적으로 이루어져 있는지 sample code와 비교하여 확인한다.
204	SCPDB_ERR_NULLPOINT_INPUT	잘못된 입력(널포인트)	이 경우 API 함수 호출이 정상적으로 이루어져 있는지 sample code와 비교하여 확인한다.
300	SCPDB_ERR_INVALID_ALGORITHM	잘못된 HASH 알고리즘 입력	일방향 알고리즘 번호를 확인한다. ◦ 70 : SHA1 ◦ 71 : SHA256 ◦ 72 : SHA384 ◦ 73 : SHA512 ◦ 74 : HAS160
302	SCPDB_ERR_INVALID_IVTYPE	암호화 실패 - 허용 불가능한 IV 타입	파일 암호화 함수를 RECORD IV 정책을 통해 암호화 하려고 했는지 확인한다.
		암호화 실패	• NPAD일 경우에는 입력값의 길이가 블럭 사이

Error Code Number	Error Code Message	DESCRIPTION	Handle
305	SCPDB_ERR_ENCRYPT	- NPAD 일 경우 입력값의 길이가 블럭사이즈의 배수가 아닐 경우	즈의 배수인지 확인한다. • 블럭 사이즈의 배수가 아닐 경우 PAD 정책을 사용하거나, 블럭 사이즈의 배수가 되도록 입력값을 변경한다.
306	SCPDB_ERR_DECRYPT	복호화 실패 - 암호문이 제대로 입력되지 않는 경우 - 키 or 알고리즘 등의 정책이 잘못 입력된 경우	• 복호화 함수의 입력값을 다시 확인한다. • 암호화 정책을 제대로 입력했는지 확인한다.
308	SCPDB_ERR_B64_DECODE	복호화 실패 - Base64 디코딩 실패, 입력된 값 확인 필요	-
330	SCPDB_ERR_WRONG_PIN	잘못된 CertFilePath, KeyFilePath경로	[AGENT] 섹션의 CertFilePath, KeyFilePath 의 값이 제대로 입력되었는지, .cer 또는 .key확장자를 확인한다.
333	SCPDB_ERR_WRONG_KEY	잘못된 Agent 키쌍 입력	Agent 키쌍 파일이 SG-KMS에서 제대로 발급받았는지 확인한다.
334	SCPDB_ERR_ENCRYPT_FPE	FPE 암호화 실패	복호화 함수에 잘못 암호화된 값을 입력했거나, 평문을 입력하지 않았는지 확인한다.
335	SCPDB_ERR_DECRYPT_FPE	FPE 복호화 실패	복호화 함수에 잘못 암호화된 값을 입력했거나, 평문을 입력하지 않았는지 확인한다.
339	SCPDB_ERR_PARTIAL_ENCRYPTION_CONTEXT	부분 암호화 오류	정책 또는 API를 확인한다.
340	SCPDB_ERR_ACCESS_PROCESS_NAME	접근제어 오류 - 설치 경로	SG-KMS 관리도구에 입력된 설치 경로 및 BA-SCP의 설치 경로를 확인한다.
341	SCPDB_ERR_ACCESS_ACCOUNT	접근제어 오류 - 계정	SG-KMS 관리도구에 입력된 계정 이름 및 BA-SCP를 실행한 계정 이름을 확인한다.
342	SCPDB_ERR_ACCESS_DATE	접근제어 오류 - 기간	SG-KMS 관리도구에 입력된 기간 및 BA-SCP의 실행된 기간을 확인한다.
343	SCPDB_ERR_ACCESS_TIME	접근제어 오류 - 시간	SG-KMS 관리도구에 입력된 시간 및 BA-SCP의 실행된 시간을 확인한다.
344	SCPDB_ERR_ACCESS_DAY_OF_WEEK	접근제어 오류 - 요일	SG-KMS 관리도구에 입력된 요일 및 BA-SCP의 실행된 요일을 확인한다.
346	SCPDB_ERR_DECRYPT_SESSION_KEY	.SCP파일 복호화 실패	현재 사용하는 Agent 키쌍으로 생성된 .SCP 파일인지 확인한다.
349	SCPDB_ERR_DISCERN_MODE	SCP_EncRRNB64 함수 실행 오류	설정 파일(Scpdb_agent.ini)의 DiscernMode의 값을 1로 변경한다.
351	SCPDB_ERR_FPE_ENCRYPT	NIST FPE 암호화 실패	입력값을 확인한다.
352	SCPDB_ERR_FPE_DE	NIST FPE 복호화 실패	입력값을 확인한다.

Error Code Number	Error Code Message	DESCRIPTION	Handle
	CRYPT		
353	SCPDB_ERR_FPE_DATA_SET	NIST FPE 입력 데이터 오류	입력값을 확인한다.
354	SCPDB_ERR_FPE_EXCLUDE_SEARCH_DATA_SET	FPE 암호화 치환 - 입력값 검사 실패	입력값에 치환 대상 문자가 들어 있는지 확인한다.
355	SCPDB_ERR_FPE_INCLUDE_REPLACE_DATA_SET	FPE 암호화 치환 실패 - 입력값에 치환 결과 문자 포함 오류	입력값이 제대로 입력되었는지 확인한다.
356	SCPDB_ERR_FPE_NOT_SEQUENCE_INCLUDE_SEARCH_DATA_SET	FPE 암호화 치환 - 입력값 순서 검사 실패	입력값에 치환 대상 문자가 순서대로 입력되었는지 확인한다.
366	SCPDB_ERR_INVALID_PARTIAL_INPUT	부분암호화 입력시 입력된 값이 잘못된 경우	암호화된 데이터가 부분 암호화로 암호화되었는지 확인한다.
367	SCPDB_ERR_PARTIAL_ENCRYPTION_NEW_TYPE_AND_CHARSET	새로운 부분 암호화 방식 사용시 글자단위 중간만 암호화하는 경우에 발생	기존의 부분 암호화 방식을 사용하거나, 문자 단위 끝까지 사용하거나, 바이트 단위를 사용한다.
371	SCPDB_ERR_INVALID_IV_LENGTH	잘못된 IV 길이	SG-KMS에서 IV의 길이가 알고리즘에 맞도록 생성되었는지 확인한다.
506	SCPDB_ERR_INVALID_SYMM_KEYTYPE	허용되지 않는 대칭 키 타입 - 외부 키 연동시에 난수 생성된 키 입력시 발생	• 난수 생성된 키인지 확인한다. • 난수 생성되지 않는 키를 요청한다.
609	SCPDB_ERR_KEYCACHE_INDEX	CONTEXT 입력 실패	• ScpAgt_CreateContext와 ScpAgt_CreateContextImportFile의 결과를 다시 한번 확인한다. • ScpAgt_CreateContext와 ScpAgt_CreateContextImportFile의 성공된 결과가 제대로 입력되었는지 확인한다.

3.1.6 API 사용 오류 (FILE)

다음 값의 경우에는 BA-SCP API 사용 시 파일과 관련된 잘못된 사용으로 인하여 발생한 오류 코드이다

표 3-6 오류 코드 - BA-SCP API 사용 오류(FILE)

Error Code Number	Error Code Message	DESCRIPTION	Handle
205	SCPDB_ERR_FILE_NOT_FOUND	API에서 사용하는 파일을 찾을 수 없을 때 발생한다.	

Error Code Number	Error Code Message	DESCRIPTION	Handle
206	SCPDB_ERR_FILE_OPEN	API에서 사용하는 파일을 찾을 수 없을 때 발생한다.	
207	SCPDB_ERR_FILE_READ	파일을 읽을 수 있는 권한이 없을 경우 발생한다.	
208	SCPDB_ERR_FILE_WRITE	파일을 쓸 수 있는 권한이 없거나, 디스크 공간이 부족할 때 발생한다.	
211	SCPDB_ERR_CONFIG_FILE_NOT_FOUND	init 함수 호출 시 설정 파일(Scpdb_agent.ini)을 찾을 수 없을 때 발생한다.	설정 파일(Scpdb_agent.ini)의 경로를 다시 한번 확인한다.
212	SCPDB_ERR_AGENT_CERT_FILE_NOT_FOUND	설정 파일(Scpdb_agent.ini)의 AGENTID에 해당하는 CertFilePath의 값의 파일이 존재하지 않을 때 발생한다.	CertFilePath의 경로를 다시 한번 확인한다.
213	SCPDB_ERR_AGENT_KEY_FILE_NOT_FOUND	설정 파일(Scpdb_agent.ini)의 AGENTID에 해당하는 KeyFilePath의 값의 파일이 존재하지 않을 때 발생한다.	KeyFilePath의 경로를 다시 한번 확인한다.
215	SCPDB_ERR_SCP_FILE_NOT_FOUND	SCP, SCPS 파일을 찾을 수 없음	입력된 SCP, SCPS 파일 경로를 확인한다.
216	SCPDB_ERR_INVALID_SCP_FILE	2.3에서 사용되던 SCP 파일이거나, 손상된 SCP, SCPS일 경우 발생한다.	입력된 파일을 다시 한번 새롭게 변경한다.
222	SCPDB_ERR_FILE_INVALID_SIGNATURE	통합함수 파일 사용시 암호화된 파일이 아닐 경우	입력한 암호화된 파일의 경로를 확인합니다.
223	SCPDB_ERR_FILE_INVALID_VERSION	통합함수 파일 암호화 사용 시 암호화된 파일 형식 중 지원하지 않는 버전	암호화된 파일이 현재 버전과 동일한 버전에서 암호화되었는지 확인합니다.
226	SCPDB_ERR_FILE_OUTPUT_FAIL	파일 암·복호화 시 결과 파일을 생성하지 못했을 경우	파일의 경로에 쓸 수 있는 권한이 있는지, 디스크 공간이 부족하지는 않는지 확인한다.

3.1.7 API 사용 오류 (키 그룹)

다음 값의 경우에는 BA-SCP API 사용 시 키 그룹과 관련된 잘못된 사용으로 인하여 발생한 오류 코드이다

표 3-7 오류 코드 - BA-SCP API 사용 오류(키 그룹)

Error Code Number	Error Code Message	DESCRIPTION	Handle
250	SCPDB_ERR_GROUP_KEY_CONTEXT	파일 암호화 함수 사용 시 키 그룹을 사용했을 경우	파일 암호화 함수에서는 키 그룹을 지원하지 않는다. ServiceID를 확인한다.

Error Code Number	Error Code Message	DESCRIPTION	Handle
251	SCPDB_ERR_GROUP_KEY_CONTEXT_FILE	키 정보 갱신 시 SCP, SCPS 사용할 경우	-
252	SCPDB_ERR_GROUP_KEY_CONTEXT_UPDATE_GK	키 정보 갱신 실패할 경우	SG-KMS가 정상 동작하는지 확인한다.
253	SCPDB_ERR_GROUP_KEY_CONTEXT_UPDATE_GKMK	키 정보 갱신 실패할 경우	SG-KMS가 정상 동작하는지 확인한다.
254	SCPDB_ERR_GROUP_KEY_CONTEXT_UPDATE_KMS	SG-KMS에서 키 정보를 제대로 가져오지 못했을 경우	SG-KMS가 정상 동작하는지 확인한다.
256	SCPDB_ERR_GROUP_KEY_CONTEXT_BACKUP_WRITE	파일 백업 기능 사용 시 파일을 생성하지 못했을 경우	파일 백업 경로에 쓰기 권한이 있는지, 디스크의 사용 가능 용량이 있는지 확인한다.

3.2 DA Error Code

Error Code Number	Error Code Message	Description
7000	DAMO_DA_ERROR_PRIVILEGE	암/복호화 권한이 없음
7001	DAMO_DA_ERROR_GET_PRIVILEGE_FILE	privilege.damo 파일이 없거나 읽기 실패
7002	DAMO_DA_ERROR_GET_AGENT_PIN	SPIN 입력이 안되어 있거나, SPIN 값이 잘못됨
7003	DAMO_DA_ERROR_GET_AGENT_KEY	AGENT 키 파일이 없거나 불러오기 실패
7004	DAMO_DA_ERROR_INVALID_INI_FILE_PATH	잘못된 INI_FILE_PATH
7005	DAMO_DA_ERROR_READ_INI_FILE_PATH	INI_FILE 읽기 실패
7006	DAMO_DA_ERROR_CIS_CC_INIT	CIS 초기화 실패

INCLUDE_ERROR(SECTION_REMOVED)