

# 운영자 안내서

OpenFrame OSI 7.2

**TMAXSOFT**

## 저작권 공지

Copyright 2025. TmaxSoft Co., Ltd. All Rights Reserved.

## 회사 정보

(주)티맥스소프트

주소 : 경기도 성남시 분당구 정자일로 45, 티맥스소프트타워

기술 서비스 센터: 1544-8629

홈페이지: <https://www.tmaxsoft.com>

## 제한된 권리

이 소프트웨어(Tmax OpenFrame®) 사용설명서와 프로그램은 저작권법과 국제 조약에 의해 보호됩니다. 사용설명서와 프로그램은 TmaxSoft Co., Ltd.와의 사용권 계약 하에서만 사용할 수 있으며, 사용설명서는 사용권 계약의 범위 내에서만 배포 또는 복제할 수 있습니다. 이 사용설명서의 전부 또는 일부를 TmaxSoft의 사전 서면 동의 없이 전자, 기계, 녹음 등의 수단으로 전송, 복제, 배포하거나 2차적 저작물을 작성할 수 없습니다.

이 소프트웨어 사용설명서와 프로그램의 사용권 계약은 어떠한 경우에도 사용설명서 및 프로그램과 관련된 지적 재산권(등록 여부를 불문)을 양도하는 것으로 해석되지 않으며, 브랜드나 로고, 상표 등을 사용할 권한을 부여하지 않습니다. 사용설명서는 오로지 정보 제공만을 목적으로 하며, 이로 인한 계약상의 직접적 또는 간접적 책임을 지지 않습니다. 또한 사용설명서 상의 내용이 법적 또는 상업적인 특정 조건을 만족시킬 것을 보장하지 않습니다. 사용설명서는 제품의 업그레이드나 수정에 따라 예고 없이 변경될 수 있으며, 내용상의 오류가 없음을 보장하지 않습니다.

## 상표 공지

Tmax®와 Tmax OpenFrame®은 TmaxSoft Co., Ltd.의 등록 상표입니다. 본 사용설명서에 기재된 모든 제품과 회사 이름은 각각 해당 소유주의 상표로서 참조용으로만 사용되며 반드시 상표 표시 (™, ®)를 하지는 않습니다.

## 오픈소스 소프트웨어 공지

본 제품은 "OpenSSL", "RSA Data Security, Inc.", "Apache Foundation" 및 "Jean-loup Gailly와 Mark Adler"에 의해 개발 또는 라이선스된 오픈 소스 소프트웨어를 포함합니다. 관련 상세 정보는 제품의 다음 디렉터리에 기재된 사항을 참고하시기 바랍니다. : \${INSTALL\_PATH}/license/oss\_licenses

## 안내서 이력

제품 버전	안내서 버전	발행일	비고
OpenFrame OSI 7.2	3.2.1	2025-08-14	
OpenFrame OSI 7.2	3.1.2	2023-12-29	
OpenFrame OSI 7.2	3.1.1	2023-08-30	

# 목차

1. 소개	1
1.1. OSI 시스템 기본개념	1
1.2. OSI 시스템 구성 요소	2
1.3. OSI 시스템 전체 구조	2
1.3.1. OSI 시스템 서버(Control Region)	3
1.3.2. OSI 사용자 서버(Dependent Region)	3
1.3.3. OSIOMSVR	4
1.3.4. OSI OTMA 서버	4
1.3.5. OSI 관리자 서버	5
1.3.6. OpenFrame Gateway	5
1.3.7. OpenFrame OSC	5
1.3.8. 시스템 및 사용자 서버 DB 세션	5
2. 시스템 환경설정	8
2.1. 개요	8
2.2. 시스템 환경설정 파일	8
2.2.1. openframe_osi.conf	8
2.2.2. osi.oftsys.seq	9
2.2.3. TCache 설정	9
2.3. 라이브러리 설정	10
2.3.1. DBDLIB	11
2.3.2. DFSRESLB	12
2.3.3. FORMAT / FORMATA / FORMATB	14
2.3.4. IMSACB / IMSACBA / IMSACBB	16
2.3.5. MODBLKS / MODBLKSA / MODBLKSB	17
2.3.6. MODSTAT	20
2.3.7. PSBLIB	21
2.3.8. RESLIB	23
2.3.9. STEPLIB	23
2.4. 스토리지 설정	23
2.4.1. MQ	23
2.4.2. REGION	24
2.4.3. RTSD	25
3. 시스템 서버 설정	26
3.1. 개요	26
3.2. 기동/종료 서버(osiomsvr)	26
3.2.1. Tmax 환경설정	26
3.3. 스케줄 서버(osisschd)	27
3.3.1. Tmax 환경설정	27
3.4. 커맨드 서버(osicmdsv)	27

3.4.1. Tmax 환경설정	28
3.5. OTMA 서버(osiotmasvr)	28
3.5.1. Tmax 환경설정	28
3.6. 관리자 서버(osiofmgr)	29
3.6.1. Tmax 환경설정	29
4. 사용자 서버 설정	30
4.1. 개요	30
4.2. 서버 그룹 설정	30
4.2.1. 일반 환경설정	30
4.2.2. XA 환경설정	30
4.3. MPP 사용자 서버	31
4.4. BMP 사용자 서버	32
5. 시스템 운영	34
5.1. 기동 및 종료	34
5.1.1. 기동	34
5.1.2. 종료	38
5.1.3. 재기동	39
5.2. 로그 관리	40
5.2.1. 시스템 서버 및 MPP 사용자 서버	40
5.2.2. BMP 사용자 서버	42
Appendix A: IMSBATCH 프러시저	43
Appendix B: Printer 기능	44
B.1. 개요	44
B.2. Display HardCopy	44
B.3. SCS-DATA Printer	44
Appendix C: 리소스 테이블	46

# 1. 소개

본 장에서는 OpenFrame OSI(Online Server type I) 시스템과 그 구성 요소에 대해 소개하고 전체 구조에 대해 설명한다.

## 1.1. OSI 시스템 기본개념

OpenFrame OSI(이하 OSI)는 리호스팅 솔루션인 OpenFrame을 구성하는 여러 제품들 중 하나로 메인프레임에서 운영되는 Online 업무를 오픈 시스템(UNIX)에서 운영할 수 있게 해준다.

일반적으로 기존 메인프레임에서 운영 중이던 업무 시스템을 오픈 시스템으로의 전환을 고려할 때 다음과 같은 이슈가 있을 수 있다.

- 기존 메인프레임의 성능 및 안전성을 어떻게 제공하는가?
- 얼마나 쉽게 전환할 수 있는가?

OSI는 TP-Monitor Tmax 제품과 마이그레이션 툴을 기반으로 위 두 가지 이슈를 해결한다.

OSI는 성능 및 안전성 문제를 해결하기 위하여 오픈 시스템에서 안정성 및 성능이 검증된 TP-Monitor인 Tmax 엔진을 기반으로 한다. 그렇기 때문에 OSI는 아래와 같은 Tmax의 특징점을 포함한다.

- 편리한 프로세스 관리

OSI에서 사용자가 생성한 프로세스들은 Tmax에 의하여 기동부터 종료까지 관리되고 Tmax가 제공하는 다양한 모니터링 정보를 사용하므로 프로세스를 편리하게 관리할 수 있다.

- 대용량 트랜잭션 지원

Tmax는 대용량 트랜잭션의 처리를 위해 스케줄링 및 서비스큐 관리 기능을 내부에 포함하고 있다.

Tmax는 오픈 환경의 미션 크리티컬한 업무를 수행해야 하는 시스템에 적합하다. Tmax를 기반으로 한 OSI 역시 대용량의 트랜잭션을 안정적으로 지원하고 있다.

- 오픈 환경에서의 자유로운 연동

연동 이슈는 IBM 메인프레임 환경에서 오픈 시스템 환경으로 리호스팅 직후에 바로 직면하게 되는 중요한 이슈는 아니다. 그러나 운영 중인 시스템을 확장하거나 다른 시스템과 연동해야 하는 상황은 자주 발생한다. 이러한 경우에 Tmax는 다른 X/Open DTP 모델을 준수하는 Tuxedo와 같은 상용 TP-Monitor와 자연스럽게 연동되므로 OSI는 기타 다른 리호스팅 솔루션과 비교하여 뛰어난 장점을 갖는다.

Tmax의 이러한 장점은 OSI로 작성된 업무 시스템의 확장을 용이하게 한다. 또한 TmaxSoft의 Web Application Server인 JEUS 등의 연동을 통해 웹 환경과도 자연스럽게 연동할 수 있는 장점을 가지고 있다.

OSI는 사용자 프로그램들과 그에 필요한 각종 리소스들을 간단한 마이그레이션 작업으로 메인프레임에서 운영되던 업무를 오픈 시스템에서도 그대로 사용할 수 있도록 여러가지 툴을 제공한다. 또한 메인프레임의 IMS/DC의 MPP(Message Processing Program), BMP(Batch Message Processing)에서 운영 중인 사용자의 프로그램들 OSI에서 그대로 운영할 수 있도록 DL/I와 동일한 인터페이스를 제공한다.

OSI 시스템의 MPP, BMP에 대한 자세한 내용은 [OSI 사용자 서버\(Dependent Region\)](#)를 참고한다.

## 1.2. OSI 시스템 구성 요소

OSI는 크게 OSI 시스템의 중요한 기능을 담당하는 OSI 시스템 서버(스케줄 서버, 커맨드 서버와 같은 시스템 서버들)와 사용자 애플리케이션을 실행하는 OSI 사용자 서버(MPP, BMP 서버와 같은 사용자 서버들)로 구성되어 있다.

- OSI 시스템 서버(Control Region)

OSI 시스템을 운영하는 데 필요한 시스템 모듈이다.

OSI 시스템 서버들의 기능은 메시지 스케줄링, Message Queue 관리, MFS 를 통한 메시지 변환, DB 연동, 시스템 및 사용자 커맨드 처리 와 같은 시스템 운영에 필요한 것들이다.

- OSI 사용자 서버(Dependent Region)

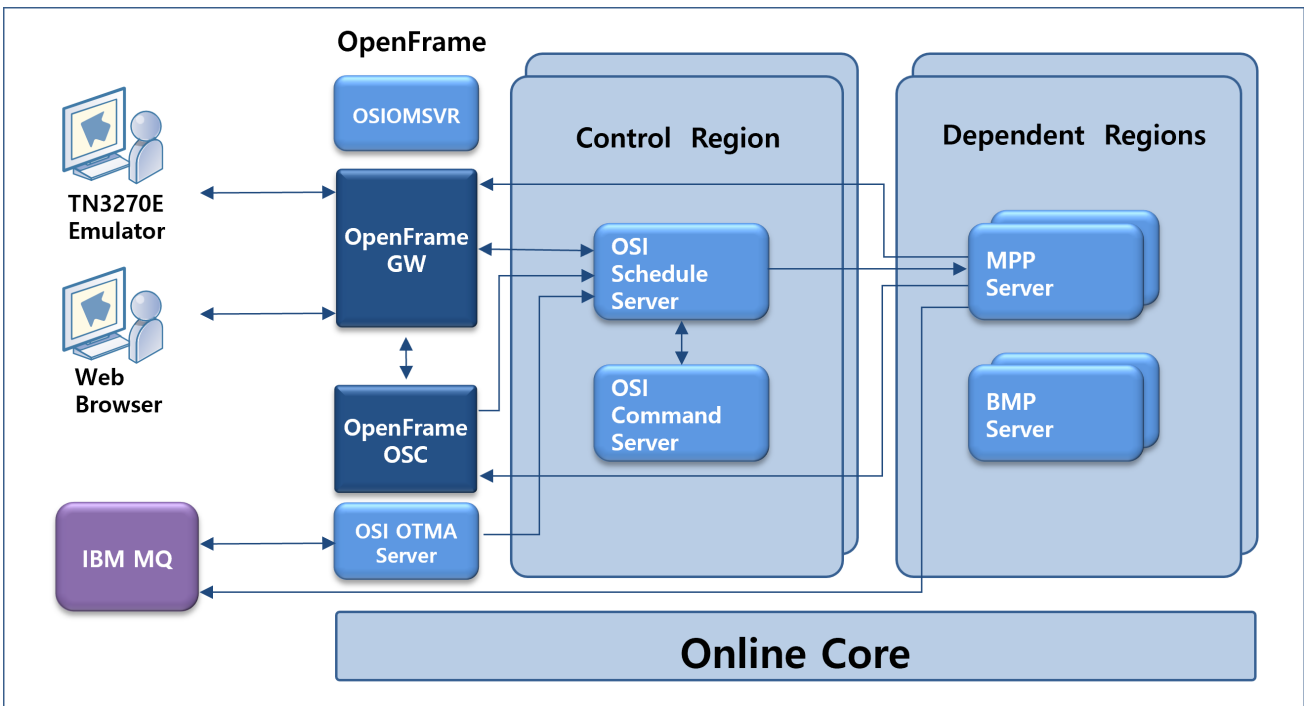
사용자 애플리케이션 프로그램을 실행하기 위해 사용자가 직접 프로세스(Tmax 서버 형태)를 JCL을 통해 기동하거나 "/START REGION" 명령을 통해 기동한다.



1. MPP 서버는 고정된 서버 이름(OSIMPPSVR)으로 제공되고 있으며, BMP 서버는 osibmpsv 라는 모듈이 제공되고 있다.
2. 현재 버전부터 TN3270 게이트웨이는 OSI에서 제공하지 않고, OpenFrame GW 제품으로 대체된다. 기존에 지원하였던 모든 기능을 동일하게 제공하고 있으며, 자세한 내용은 OpenFrame GW "운영자 안내서"를 참고한다.

## 1.3. OSI 시스템 전체 구조

일반적으로 실제적인 업무 프로그램이 운영되기 위해서 시스템 서버 차원과 사용자 서버 차원에서 업무 프로그램의 운영 기반이 제공되어야 한다. 이러한 운영 기반을 위한 OSI 구조는 다음과 같다.



OSI 시스템 구조

시스템 서버(Control Region)는 OSI 시스템 모듈에 해당하는 영역으로, OSI 시스템 자체에서 애플리케이션 서버 및 업무 프로그램을 동작시키기 위해 필요로 하는 서버를 포함한 각종 설정이 포함되어 있다. 사용자 서버(Dependent Region)는 OSI 사용자 모듈에 해당하는 영역으로 OSI 시스템에서 제공하는 애플리케이션 서버인 MPP와 BMP로 구분한다. osiomsvr(Online Manager) 서버는 시스템 서버와 사용자 서버를 활성화하기 위해 필요하며, 그 밖에 OSI 시스템에는 시스템 엔진 영역인 Online Core(Tmax)가 포함된다.

### 1.3.1. OSI 시스템 서버(Control Region)

OSI 시스템 서버(Control Region)에는 스케줄 서버(osisschd)와 커맨드 서버(osicmdsv) 두 가지 서버가 존재한다. 이 두 개의 서버는 Tmax UCS 서버 타입으로 제공되고 있다.

- 스케줄 서버(osisschd)

요청 받은 메시지의 유효성 판단(유효한 메시지인지를 판단)과 함께 메시지 저장(백업) 및 전달(포워딩)을 동시에 하고 있다. 현재 버전에서 스케줄 서버는 유효한 메시지인지를 판단한 이후에 해당 목적지로 전달만하는 역할을 수행한다. 터미널(화면)로부터 온 메시지의 경우에는 MFS를 통한 데이터 변환이 이루어진 후 MPP 서버로 보내주게 된다.

구분	설명
메시지 저장 및 전달	요청받은 메시지를 메시지 큐(Message Queue 이하 MQ) 테이블에 백업하고 목적지의 타입(커맨드, 트랜잭션 등)에 따라서 적절하게 메시지를 MPP 또는 커맨드 서버, OpenFrame GW에 접속되어 있는 에뮬레이터로 포워딩한다.
MFS 변환	MPP와 TN3270 에뮬레이터(중간에 OpenFrame GW가 있음) 사이에서 화면 정보와 애플리케이션에서 정의한 데이터 포맷을 일치시켜주고, 서로 다른 코드 페이지(ASCII와 EBCDIC)로 된 데이터를 중간에서 변환해주는 역할을 한다.

- 커맨드 서버(osicmdsv)

OSI 시스템 운영에 필요한 모든 명령어를 처리한다. 명령어는 터미널, imscmd 툴, DL/I CMD call 등을 통해 입력할 수 있다. 명령어 처리 결과는 `#{OPENFRAME_HOME}/log/cmd/imscmd_{DATE}.log`에 기록된다.



커맨드 서버에서 처리하는 명령어에 대한 자세한 내용은 OpenFrame OSI "명령어 참조 안내서"를 참고한다.

### 1.3.2. OSI 사용자 서버(Dependent Region)

OSI에서 기본적으로 제공하는 시스템 모듈 외에 사용자가 작성한 업무 프로그램을 실제로 운영하기 위해서 사용자는 시스템 설정 단계에서 이를 담당할 서버 모듈을 직접 준비해야 한다. 사용자가 준비해야 하는 서버 모듈은 개념적으로 IMS/DC의 OSI 사용자 서버(Dependent Region)에 해당하는 영역을 의미한다. 일반적으로 IBM 메인프레임에서 운영하던 Dependent Region 하나당 최소 하나씩 사용자 모듈을 준비한다.

다음은 OSI를 운영에 사용되는 사용자 서버이다.

구분	설명
MPP 사용자 서버	IMS/DC에서 MPP Region에 대응되는 부분으로, OSI에서는 메시지의 class 단위로 MPP 서버를 운영한다. 즉, 기존 IMS/DC에서 운영 중이던 MPP Region당 최대 1:4 대응되는 서버이다.
BMP 사용자 서버	IMS/DC의 BMP Region에서 동작하는 사용자 프로그램이 운영될 수 있도록 하는 서버이다.

사용자 서버를 준비하는 과정은 JCL을 사용하여 직접 기동하거나 OSI 명령어를 사용하여 기동할 수 있다. 이와 관련된 사용자 서버 환경설정에 대한 자세한 내용은 [사용자 서버 설정](#)을 참고한다.

### 1.3.3. OSIOMSVR

OSIOMSVR는 OSI 시스템 서버와 MPP 사용자 서버를 기동 및 종료를 담당하는 서버이다. JCL 혹은 "/START REGION" 명령을 수행하면, OSIOMSVR는 내부적으로 기동에 필요한 리전 정보들을 DB에 저장 후 tmboot를 이용하여 서버들을 기동한다. 반대로 "/STOP REGION" 혹은 "/CHECKPOINT FREEZE" 명령을 수행하면, OSIOMSVR는 내부적으로 DB에서 리전 정보들을 지우고 tmdown을 이용하여 서버들을 종료한다.

OSIOMSVR는 Region을 띄우기 전에 반드시 기동되어 있어야 하며, osiboot 혹은 osidown 툴을 사용하여 OpenFrame Core(Tmax/Base/Batch) 서버와 함께 기동 및 종료할 수 있다.



BMP 사용자 서버는 OSIOMSVR를 거치지 않고 직접 기동/종료한다.

### 1.3.4. OSI OTMA 서버

IBM MQ 제품과 연동 및 OTMA 기능을 제공하는 서버이다. MQ 기능은 별도의 설정으로 사용 여부를 지정할 수 있으며, 사용하지 않아도 시스템의 동작에는 영향이 없도록 하고 있다.

다음은 OTMA 서버에서 사용하고 있는 MQI(MQ Interface)이다.

이름	설명
MQCONN	IBM MQ의 큐 관리자에 연결한다.
MQOPEN	오브젝트에 대한 접근을 활성화한다.
MQGET	MQOPEN을 통해 열린 로컬 큐로부터 메시지를 가져온다.
MQPUT	MQOPEN을 통해 열린 로컬 큐로 메시지를 보낸다.
MQCLOSE	오브젝트에 대한 접근을 비활성화한다.
MQDISC	IBM MQ의 큐 관리자와의 연결을 끊는다.



OTMA 서버의 IBM MQ 기능을 사용하기 위한 설정 및 사용법에 대한 자세한 내용은 OpenFrame OSI "환경설정 안내서"를 참고한다.

### 1.3.5. OSI 관리자 서버

OpenFrame Manager 제품과 연동하여 OSI 관리(Manage) 기능을 제공하는 서버이다.



OSI 관리 기능에 대한 자세한 내용은 OpenFrame Manager "사용자 안내서"를 참고한다.

### 1.3.6. OpenFrame Gateway

OpenFrame Gateway(이하 OpenFrame GW)는 OSI 시스템 서버(Control Region)와 TN3270 / TN3270E 에뮬레이터나 웹 브라우저를 통해 접속하는 터미널 사이에 위치하며, OSI 시스템 서버(Control Region)에게 트랜잭션 요청 등을 수행하는 서버이다. 터미널의 접속 정보와 시스템 서버(Control Region)의 정보를 관리하며, VTAM(Virtual Terminal Access Method) Gateway라고도 한다.

다음은 OpenFrame GW의 주요 기능이다.

- TN3270 프로토콜 지원
- APPLID 정보 관리(APPLID(Application Identifier)는 OpenFrame GW에 접속하는 터미널 및 OSI 시스템 서버(Control Region)를 의미한다.)
- IP-LU 매핑 기능 지원



OpenFrame GW의 설정, 기능 및 사용법에 대한 자세한 내용은 OpenFrame GW "운영자 안내서"를 참고한다.

### 1.3.7. OpenFrame OSC

OpenFrame OSC(이하 OSC)는 메인프레임 리호스팅 솔루션인 OpenFrame을 구성하는 여러 제품들 중 하나로 CICS에서 운영되는 Online 업무를 간단한 전환 절차를 통해 오픈 시스템에서 운영할 수 있게 하는 제품이다.

OSC의 CICS SEND, CICS RECEIVE 커맨드를 통해 OSI와 통신이 가능하며 OSC에서 보낸 메시지는 스케줄 서버(osisschd)로 전달된다.



OpenFrame OSC의 설정, 기능 및 사용법에 대한 자세한 내용은 OpenFrame OSC "운영자 안내서"를 참고한다.

### 1.3.8. 시스템 및 사용자 서버 DB 세션

OSI의 시스템 서버 및 리전 서버들은 시스템 메타 데이터 관리 및 OpenFrame HiDB 사용을 위해 DB를 사용하고 있으며 이에 따라 각자 필요한 개수의 DB 세션 가진다.

- **osiomsvr**

osiomsvr 서버는 기본적으로 다음과 같은 DB 세션을 맺는다.

순번	세션	유형	목적
1	System connection	ODBC	OpenFrame meta

- **osiotmasvr**

osiotmasvr 서버는 기본적으로 다음과 같은 DB 세션을 맺는다.

순번	세션	유형	목적
1	System connection	ODBC	OpenFrame meta
2	OpenFrame HiDB	ESQL	HiDB connection

- **osiofmgr**

osiofmgr 서버는 기본적으로 다음과 같은 DB 세션을 맺는다.

순번	세션	유형	목적
1	System connection	ODBC	OpenFrame meta

### 1.3.8.1. 시스템 서버

- **osisschd**

osisschd 서버는 기본적으로 다음과 같은 DB 세션을 맺는다.

순번	세션	유형	목적
1	System connection	ODBC	OpenFrame meta
2	OpenFrame HiDB	ESQL	HiDB connection

- **osicmdsv**

osicmdsv 서버는 기본적으로 다음과 같은 DB 세션을 맺는다.

순번	세션	유형	목적
1	System connection	ODBC	OpenFrame meta
2	OpenFrame HiDB	ESQL	HiDB connection

### 1.3.8.2. 사용자 서버

- **OSIMPPSVR**

OSIMPPSVR 서버는 기본적으로 다음과 같은 DB 세션을 맺는다.

순번	세션	유형	목적
1	System connection	ODBC	OpenFrame meta
2	OpenFrame HiDB	ESQL	HiDB connection

- **osibmpsv**

osibmpsv 서버는 기본적으로 다음과 같은 DB 세션을 맺는다.

순번	세션	유형	목적
1	System connection	ODBC	OpenFrame meta
2	OpenFrame HiDB	ESQL	HiDB connection



리전 서버의 경우 서버 그룹 설정에 따라 DB 세션이 증가할 수 있다. 자세한 내용은 [사용자 서버 설정](#)을 참고한다.

## 2. 시스템 환경설정

본 장에서는 OSI 시스템의 운영을 위해 필요한 환경설정 방법 및 예제에 대해 기술한다.

### 2.1. 개요

OSI 시스템 설치가 완료된 후에 사용자는 먼저 OSI 시스템의 환경에 기본적으로 설정되어야 하는 항목에 대해 시스템에서 지정한 대로 설정을 완료해야 한다.

OSI 시스템 환경에 구성해야 하는 내용은 다음과 같다.

- [시스템 환경설정 파일](#)
- [라이브러리 설정](#)
- [스토리지 설정](#)

### 2.2. 시스템 환경설정 파일

OSI 시스템에 필요한 시스템 설정값 중 동적으로 반영할 수 없는 항목은 대부분 환경설정 파일에 기술되고, OSI 시스템이 기동될 때 지정된 값이 시스템에 적용된다.

#### 2.2.1. openframe\_osi.conf

OSI 시스템을 운영하기 위해서는 openframe\_osi.conf 파일에 다음의 서브젝트 항목들을 설정한 후 **ofconfig** 툴을 이용하여 OpenFrame 시스템에 적용해야 한다.

서브젝트	설명
osi	OSI 시스템의 모든 모듈에서 공통으로 참조하는 항목으로, 항목에 설정된 값은 OSI의 모든 모듈에 적용된다.
osi.[ <i>IMSID</i> ]	OSI에서 각 IMS 시스템에 해당하는 IMSID로 구분하며, 항목에 설정된 값은 해당 IMS 시스템의 모든 모듈에 적용된다. <i>IMSID</i> 는 4자리의 Alphanumeric 문자로 설정해야 한다.
osi.[ <i>osiotmasvrname</i> ]	OSI와 IBM MQ 간의 OTMA 기능을 위한 설정을 한다. 한 개의 OTMA 서버당 한 개의 서브젝트 항목이 필요하며, <i>osiotmasvrname</i> 은 MQ 사용자 서버 설정에 등록된 서버 이름으로 설정해야 한다.
ssm.[ <i>IMSID</i> ][ <i>SSM</i> ]	OSI에서 IMS 시스템별로 MultipleRM을 사용하기 위해 Tmax XA 서버에 적용하는 항목을 설정한다.



1. OpenFrame 환경설정을 위한 서브젝트와 설정방법에 대한 자세한 내용은 OpenFrame OSI "환경설정 안내서"를 참고한다.
2. ofconfig 툴과 관련된 사용법 및 자세한 설명은 OpenFrame Base "툴 참조 안내서"를 참고한다.

## 2.2.2. osi.ofsys.seq

OSI 시스템에서 리전 서버(Region Server)를 제외한 Base, Batch, TACF 서버들의 이름을 지정하여 osiboot할 때 기동할 Tmax 서버들을 선택할 수 있다. OTMA 서버도 여기에 포함된다.

다음은 osi.ofsys.seq의 예제이다.

```
#BASE
TPFMAGENT
ofrsasvr
ofrlhsrv
ofrdmsvr
ofrdsedt
ofrcmsvr
ofruisvr
ofrsmlog

#BATCH
obmjmsvr
obmjschd
obmjinit
obmjhist
obmjspb
ofrpmsvr
obmtmgr
obmjtimr

#TACF
tmsvr

#OSI
#IMSAOTMA
```

## 2.2.3. TCache 설정

OSI는 멀티 노드 클러스터링 기능 지원을 위해 RTSD를 포함하여 리전 간에 공유가 필요한 정보들을 DB 테이블에서 관리하고 있는데, TCache를 이용하여 DB 테이블로부터 Access하는 데이터를 공유 메모리에 캐싱함으로써 Read 성능을 보장하고 있다. TCache 설정은 노드별로 필요하며, 사용할 공유 메모리 키와 리전별 리소스 정보를 설정한다.



기본 설정에 대한 자세한 사항은 "Tmax TCache 안내서"를 참고한다.

다음은 pfmtcache.conf의 예제이다. LOCK\_WAITTIME 설정은 사용하는 TCache 버전에 맞는 설정을 사용한다.

```
# the configuration file of TCACHE
SHMKEY=0x70005 # the key of shared memory
IPCPerm=0600 # permission of the shared memory
SIZE_LOCAL=32 # L1 cache size in kilo-bytes
LIB_PTHREAD=libpthread.so # the pthread library file name
INVALIDATE_TYPE=0 # Invalidate type 0:multi-node 1:multi-domain
AGENT_SVC=SPFMAGENT # multi-node sync. svc SPFMAGENT|2 = SPFMAGENT01, SPFMAGENT02
```

```

# lock wait time for earlier than r11724
# LOCK_WAITTIME=1000                # read, write lock wait time (default: 1000)

# lock wait time for r11724 and later
# WRITE_LOCK_WAITTIME=1000          # write lock wait time (default: 1000, recommed larger value
than READ_LOCK_WAITTIME)
# READ_LOCK_WAITTIME=1000          # read lock wait time (default: 0)

# cache for OSI Region Status
CACHE_NAME=OFM_OSI_REGION_STATUS    # table name
SIZE_MEM=32                          # the total cache memory size in kilo-bytes
SIZE_HASH=32                          # the number of hash key (MAX=65536)
SIZE_KEY=4                            # the number of digits of the index column
SIZE_REC=8                            # the size of a single record in bytes
INV_TIMEOUT=0                         # invalidation timeout in sec

# cache for IMSA Region
CACHE_NAME=IMSA                      # IMSID
SIZE_MEM=65535                       # the total cache memory size in kilo-bytes
SIZE_HASH=32                          # the number of hash key (MAX=65536)
SIZE_KEY=13                           # the number of digits of the index column
SIZE_REC=289                          # the size of a single record in bytes
INV_TIMEOUT=0                         # invalidation timeout in sec

#cache for OFM_OSI_DBD
CACHE_NAME=OFM_OSI_DBD               # the name of cache
SIZE_MEM=65535                       # the total cache memory size in kilo-bytes
SIZE_HASH=32                          # the number of hash key (MAX=65536)
SIZE_KEY=64                           # the number of digits of the index column
SIZE_REC=2048                         # the size of a single record in bytes
INV_TIMEOUT=0                         # invalidation timeout in sec

#cache for OFM_OSI_PSB
CACHE_NAME=OFM_OSI_PSB               # the name of cache
SIZE_MEM=65535                       # the total cache memory size in kilo-bytes
SIZE_HASH=32                          # the number of hash key (MAX=65536)
SIZE_KEY=64                           # the number of digits of the index column
SIZE_REC=1024                         # the size of a single record in bytes
INV_TIMEOUT=0                         # invalidation timeout in sec

```

## 2.3. 라이브러리 설정

시스템 기동 전에 시스템 운영에 필요한 정보들이 미리 준비되어 있어야 한다. 본 절에서는 OSI 운영에 필요한 정보들의 종류와 정보별로 해당 정보를 준비하는 과정을 설명한다. 대부분의 정보들은 RDB 테이블의 레코드 형태로 저장되며, MDA, MFS 등 일부는 데이터셋 또는 UNIX 파일 형태로 저장된다.

OSI 운영에 필요한 라이브러리는 다음과 같다.

라이브러리	설명
<a href="#">DBDLIB</a>	OpenFrame HiDB의 사용을 위한 DBD 정보를 저장하는 라이브러리이다.
<a href="#">DFSRESLB</a>	데이터셋의 동적 사용에 필요한 정보를 저장하는 라이브러리이다.
<a href="#">FORMAT</a>	OSI에서 매핑 기능을 지원하기 위한 MFS 정보를 저장하는 Staging Library이다.

라이브러리	설명
FORMATA / FORMATB	OSI에서 매핑 기능을 지원하기 위한 MFS 정보를 저장하고 Online에서 실제 사용하는 라이브러리이다.
IMSACB	PSB와 DBD를 통합 관리를 위한 ACB 정보를 저장하는 Staging Library이다.
IMSACBA/IMSACB B	PSB와 DBD를 통합 관리를 위한 ACB 정보를 저장하고 Online에서 실제 사용하는 라이브러리이다.
MODBLKS	OSI 운영을 위한 시스템 정보를 저장하는 Staging Library이다.
MODBLKSA / MODBLKSB	OSI 운영을 위한 시스템 정보를 저장하고 Online에서 실제 사용하는 라이브러리이다.
MODSTAT	동적 변경을 지원하는 데이터셋의 현재 ACTIVE 데이터셋 정보를 담고 있는 데이터셋이다.
PSBLIB	애플리케이션 프로그램의 사용을 위한 PSB 정보를 저장하는 라이브러리이다.
RESLIB	데이터셋의 동적 사용에 필요한 MDA 정보를 저장하는 라이브러리이다. (DFSRESLB과 동일)
STEPLIB	애플리케이션 개발자가 작성한 프로그램의 Shared Object 형태 실행 모듈 관리를 위한 디렉터리이다.



1. 각 라이브러리 설명 중 데이터셋 사용법 및 PDS의 자세한 설명은 OpenFrame Base "데이터셋 안내서"를 참고한다.
2. 각 라이브러리 설명 중 IDCAMS 사용법의 자세한 내용은 OpenFrame Batch "유틸리티 참조 안내서"를 참고한다.
3. 각 라이브러리 설명 중 DBD, PSB, ACB, MDA 정보 등록 및 관리와 OpenFrame HiDB 생성의 자세한 내용은 OpenFrame HiDB "HiDB 안내서"를 참고한다.
4. 라이브러리 설정하는 과정에 사용되는 툴과 관련된 사용법 및 자세한 설명은 OpenFrame OSI "툴 참조 안내서"를 참고한다.

### 2.3.1. DBDLIB

DBD는 OpenFrame HiDB의 특징을 정의하는 매크로 파라미터 문장의 집합으로 데이터베이스 구조, 액세스 메소드, 데이터베이스 내의 세그먼트와 필드 및 세그먼트 타입 간의 관계를 정의한다.

#### 데이터셋 생성

DBDLIB을 사용하기 위해서는 시스템 준비 과정에서 데이터셋을 생성하는 절차가 필요하다. DBDLIB은 PDS 형태로 구성되며 개별 환경설정 파일은 DBDLIB의 멤버 형태로 저장된다. OSI에서 DBDLIB을 생성하기 위해서는 **pdsgen**이라는 툴을 사용한다.

다음은 pdsgen을 사용하여 IMS.DBDLIB을 DEFVOL이라는 볼륨에 생성하는 예제이다.

```
$ pdsgen IMS.DBDLIB DEFVOL -f LB -l 32760
pdsgen version 7.1.0(0) oframe@tmax:ofsrc/base(#1) 2020-06-29 17:30:26
```

```
PDS Dataset Generation Program
```

```
pdsgen: *** PDS IMS.DBDLIB is created.
```

## DBD 정보 등록

OSI에서 DBD를 사용하기 위해서는 **dbdgen**이라는 툴을 사용하여 DBD 정보를 등록해야 한다.

다음은 DBD 컨트롤문이 들어 있는 OIVPIDBD라는 파일을 dbdgen을 이용하여 DBD 정보를 등록하는 예제이다.

```
$ dbdgen OIVPIDBD
dbdgen version 7.2.0(0) oframe@tmax:ofsrc/ims(#3) 2020-08-08 16:21:16
Database Description Block Generation Program

dbdgen: force flag on; the tool overwrites existing data
dbdgen: 1 files are requested in total
-----
dbdgen: processing DBD script "OIVPIDBD"
-----

dbdgen: processing DBD "OIVPIDBD"
dbdgen: removing existing DBD OIVPIDBD metadata
dbdgen: successfully processed DBD "OIVPIDBD"
dbdgen: successfully processed for the requested DBDs (total 1)
```



현재 버전에서는 dbdgen을 수행하게 되면 DBD 스크립트에 기술되어 있는 데이터베이스 스키마 정보가 메타 테이블에 저장된다. 실제로 DBDLIB에는 아무것도 저장되지 않는다.

## 2.3.2. DFSRESLB

PSB 리소스를 사용하는 프로그램을 기동하려면 사용할 PSB 정보와 DBD에 정의한 데이터셋 정보를 프로그램에 전달할 수 있는 방법이 필요하다.

OpenFrame Batch는 JCL을 사용하여 애플리케이션 프로그램을 실행하는데, 이때 JCL의 DD 문을 통해 PSB 정보와 DBD의 데이터셋 정보를 프로그램에 전달할 수 있다. 하지만 OSI는 하나의 Region에서 여러 프로그램을 수행해야 하고 Region을 기동할 때 사용하는 JCL에는 모든 DD 문을 기술할 수 없기 때문에 OpenFrame Batch와는 다른 방법으로 필요한 정보를 프로그램에 전달해야 한다. 따라서 OSI에서는 DFSRESLB에 데이터셋의 동적 사용에 필요한 정보를 저장하는 방식을 이용한다.



동적 DD 할당에 대한 자세한 사항은 [IBM IMS DFSMDA macro](#) 매뉴얼을 참조한다.

## 데이터셋 생성

DFSRESLB를 위해서는 시스템을 준비하는 과정에서 데이터셋을 생성하는 절차가 필요하다. DFSRESLB는 PDS 데이터셋 형태로 구성되며 개별 설정 파일은 DFSRESLB의 멤버 형태로 저장된다. OSI에서 DFSRESLB를 생성하기

위해서는 **pdsgen** 툴을 사용한다.

다음은 pdsgen을 사용하여 IMS.RESLIB을 DEFVOL이라는 볼륨에 생성하는 예제이다.

```
$ pdsgen IMS.RESLIB DEFVOL -f LB -l 32760
pdsgen version 7.1.0(0) oframe@tmax:ofsrc/base(#1) 2020-06-29 17:30:26
PDS Dataset Generation Program

pdsgen: *** PDS IMS.RESLIB is created.
```

## MDA 정보 등록

OSI에서 MDA를 사용하기 위해서는 **imsdalloc** 툴을 사용하여 MDA 정보를 등록해야 한다.

다음은 MDA를 정의하는 스크립트가 저장되어 있는 OIVPIDBD 파일을 이용해서 DFSRESLIB에 MDA 정보를 등록하는 예제이다.

<OIVPIDBD>

```
DFSMDA TYPE=DATABASE,DBNAME=OIVPIDBD
DFSMDA TYPE=DATASET,DSNAME=IMS.TEST.OIVPI,DDNAME=OIVPI
END
```

imsdalloc으로 MDA 정보를 등록한다.

```
$ imsdalloc -l IMS.RESLIB -v DEFVOL OIVPIDBD
imsdalloc version 7.2.0(0) oframe@tmax:ofsrc/ims(#3) 2020-08-08 16:21:16
Dynamic Allocation Block Generation Program

IMSDALOC FCOUNT=1,RESLIB=IMS.RESLIB,VOLSER=DEFVOL
-----
*** processing filepath="OIVPIDBD"
-----
mdaparser: *** dfsmda_statement matched!
mdaparser: *** dfsmda_statement matched!
mdaparser: *** end_statement matched!
mdaparser: *** mda_generation finished!!!
-----
*** ims_parse_mda("OIVPIDBD") success.
-----
MDA TYPE=DATABASE,DBNAME=OIVPIDBD
DATASET TYPE=DATASET,DDNAME=OIVPI,DSNAME=IMS.TEST.OIVPI,DISP=
-----
*** ims_print_mda("OIVPIDBD") success.
-----
PROGRAM COMPLETED SUCCESSFULLY.
```

### 2.3.3. FORMAT / FORMATA / FORMATB

FORMAT은 OSI에서 매핑 기능을 지원하기 위한 정보를 저장하는 라이브러리이다.

Mapping 기능은 포맷을 사용하여 포맷 안에 정의되어 있는 필드들과 애플리케이션 데이터들 간의 관계를 맺어주는 것을 의미한다. 데이터셋은 MODBLKS, IMSACB와 같이 3개를 준비한다.

#### 데이터셋 생성

FORMAT을 사용하기 위해서는 시스템 준비 과정에서 데이터셋을 생성하는 절차가 필요하다.

FORMAT은 PDS 형태로 구성되며 개별 환경설정 파일은 MFSLIB의 멤버 형태로 저장된다. OSI에서 FORMAT을 생성하기 위해서는 **pdsgen** 툴을 사용한다.

다음은 pdsgen을 사용하여 OSI.IMSA.MFSLIB을 DEFVOL이라는 볼륨에 생성하는 예제이다.

```
$ pdsgen OSI.IMSA.MFSLIB DEFVOL -f LB -l 32760
pdsgen version 7.1.0(0) oframe@tmax:ofsrc/base(#1) 2020-06-29 17:30:26
PDS Dataset Generation Program

pdsgen: *** PDS OSI.IMSA.MFSLIB is created.
```

#### Map 정보 등록

다음은 OIVP001.TXT 파일을 이용해서 OSI.IMSA.MFSLIB에 맵 정보를 등록하는 예제이다.

<OIVP001.TXT>

```
PRINT ON,NOGEN
*****
TITLE 'FORMAT SET FOR OPENFRAME ONLINE IVP'
*****
OIVP001 FMT
DEV TYPE=(3270,2), X
FEAT=IGNORE, X
DSCA=X'00A0', X
PFK=(PFKFIELD, X
12='/FOR OIVP0060')
DIV TYPE=INOUT
DPAGE CURSOR=((7,40)), X
FILL=PT
CURDATE DFLD POS=(1,2), X
LTH=8, X
ATTR=(PROT,ALPHA,NORM,NOMOD)
DFLD '** WELCOME TO OPENFRAME ONLINE **', X
POS=(2,24), X
ATTR=(PROT,ALPHA,HI,NOMOD), X
EATTR=YELLOW
DFLD '-----X
-----', X
POS=(3,2), X
ATTR=(PROT,ALPHA,NORM,NOMOD), X
```

	EATTR=YELLOW	
	DFLD '** OSI INSTALLATION VERIFICATION PROCEDURE **',	X
	POS=(4,17),	X
	ATTR=(PROT,ALPHA,HI,NOMOD),	X
	EATTR=GREEN	
	DFLD 'CODE ',	X
	POS=(7,34),	X
	ATTR=(PROT,ALPHA,NORM,NOMOD)	
CODE	DFLD POS=(7,40),	X
	LTH=4,	X
	ATTR=(NOPROT,ALPHA,NORM,MOD),	X
	EATTR=HREV	
	DFLD ' ',	X
	POS=(7,45),	X
	ATTR=(PROT,ALPHA,HI,NOMOD)	
	DFLD 'INQR INQUIRY ACCOUNT INFORMATION',	X
	POS=(10,23),	X
	ATTR=(PROT,ALPHA,HI,NOMOD)	
	DFLD 'INSR INSERT NEW ACCOUNT',	X
	POS=(11,23),	X
	ATTR=(PROT,ALPHA,HI,NOMOD)	
	DFLD 'UPDT UPDATE ACCOUNT INFORMATION',	X
	POS=(12,23),	X
	ATTR=(PROT,ALPHA,HI,NOMOD)	
	DFLD 'DELT DELETE ACCOUNT',	X
	POS=(13,23),	X
	ATTR=(PROT,ALPHA,HI,NOMOD)	
	DFLD '-----X	
	-----',	X
	POS=(18,2),	X
	ATTR=(PROT,ALPHA,NORM,NOMOD),	X
	EATTR=YELLOW	
ERRMSG	DFLD POS=(19,2),	X
	LTH=79,	X
	ATTR=(PROT,ALPHA,NORM,NOMOD),	X
	EATTR=TURQ	
	DFLD 'ENTER APPROPRIATE CODE.',	X
	POS=(20,2),	X
	ATTR=(PROT,ALPHA,NORM,NOMOD),	X
	EATTR=PINK	
	DFLD '-----X	
	-----',	X
	POS=(21,2),	X
	ATTR=(PROT,ALPHA,NORM,NOMOD),	X
	EATTR=YELLOW	
	DFLD '[SYS INFO] - OSI IVP STARTING NOW',	X
	POS=(22,2),	X
	ATTR=(PROT,ALPHA,HI,NOMOD),	X
	EATTR=RED	
CURTIME	DFLD POS=(22,71),	X
	LTH=8,	X
	ATTR=(PROT,ALPHA,NORM,NOMOD),	X
	EATTR=RED	
	DFLD '-----X	
	-----',	X
	POS=(23,2),	X
	ATTR=(PROT,ALPHA,NORM,NOMOD),	X
	EATTR=YELLOW	
	DFLD 'CopyRight(c) 2007, TmaxSoft, All Rights Reserved.',	X

```

        POS=(24,31),
        ATTR=(PROT,ALPHA,NORM,NOMOD),
        EATTR=BLUE
    FMTEND
*****
    EJECT
*****
OIVP001I MSG  TYPE=INPUT,
              SOR=(OIVP001,IGNORE),
              NXT=OIVP0010
    SEG
    MFLD  'OIVPMPP1',
          LTH=8
    MFLD  CODE,
          LTH=4
    MSGEND
*****
OIVP0010 MSG  TYPE=OUTPUT,
              SOR=(OIVP001,IGNORE),
              NXT=OIVP001I
    SEG
    MFLD  (CURDATE,DATE2)
    MFLD  CODE,
          LTH=4
    MFLD  ERRMSG,
          LTH=79
    MFLD  (CURTIME,TIME)
    MSGEND
*****
    END

```

osimfsgen으로 맵 정보를 등록한다.

```

$ osimfsgen -m OSI.IMSA.MFSLIB OIVP001.TXT
osimfsgen 7.2.0(0) oframe@tmax:ofsrc/osi(#1) 2020-11-12 19:43:54
[/home/oframe/products/ofsrc/OpenFrame/volume_default/OSI.IMSA.MFSLIB/OIVP001.mfs] create ok.
[/home/oframe/products/ofsrc/OpenFrame/volume_default/OSI.IMSA.MFSLIB/OIVP001I.mfs] create ok.
[/home/oframe/products/ofsrc/OpenFrame/volume_default/OSI.IMSA.MFSLIB/OIVP0010.mfs] create ok.

```

### 2.3.4. IMSACB / IMSACBA / IMSACBB

OSI에서 PSB와 DBD 정보는 개별로 관리 및 운영되지 않고 PSB와 DBD를 하나의 라이브러리로 통합하여 관리된다. PSB와 DBD가 통합된 라이브러리를 **ACB**라고 한다. MODBLKS와 마찬가지로 3개의 데이터셋을 준비한다.

#### 데이터셋 생성

ACBLIB을 사용하기 위해서는 시스템 준비 과정에서 데이터셋을 생성하는 절차가 필요하다. ACBLIB은 PDS 형태로 구성되며 개별 환경설정 파일은 ACBLIB의 멤버 형태로 저장된다. OSI에서 PSBLIB을 생성하기 위해서는 **pdsgen** 툴을 사용한다.

다음은 pdsgen을 사용하여 IMS.ACBLIB을 DEFVOL이라는 볼륨에 생성하는 예제이다.

```
$ pdsgen IMS.ACBLIB DEFVOL -f LB -l 32760
pdsgen version 7.1.0(0) oframe@tmax:ofsrc/base(#1) 2020-06-29 17:30:26
PDS Dataset Generation Program

pdsgen: *** PDS IMS.ACBLIB is created.
```

## ACB 정보 등록

OSI에서 ACB를 사용하기 위해서는 **acbgen**이라는 툴을 사용하여 ACB 정보를 등록해야 한다.

다음은 PSBLIB의 OIVPI002라는 이름의 PSB에 대하여 ACBLIB에 ACB 정보를 등록하는 예제이다.

```
$ acbgen build -p IMS.PSBLIB -d IMS.DBDLIB -l IMS.ACBLIB PSB=OIVPI002
acbgen version 7.2.0(0) oframe@tmax:ofsrc/ims(#3) 2020-08-08 16:21:16
Application Control Block Generation Program

ACBGEN COMMAND=BUILD,OPERAND=(PSB=OIVPI002),ACBLIB=IMS.ACBLIB
-----
*** ACBGEN BUILD PSB=OIVPI002
-----
*** BUILDING PSB BLOCK..... PSBNAME=OIVPI002
*** BUILDING DBD BLOCK..... DBDNAME=OIVPIDBD
*** BUILDING DBD BLOCK..... DBDNAME=OIVPIX1
-----
PROGRAM COMPLETED SUCCESSFULLY.
```



현재 버전에서는 acbgen을 수행하게 되면 DBD 메타 정보 및 PSB 메타 정보를 이용해서 ACB 메타 정보를 생성한 후 각각의 메타 테이블에 저장된다. 실제로 ACBLIB에는 아무것도 저장되지 않는다.

## 2.3.5. MODBLKS / MODBLKSA / MODBLKSB

IMS/DC에서 매크로를 이용해서 등록하고 관리하는 시스템 정의(System Definition)를 OSI에서는 OSD(Online System Definition)라는 구조로 대신한다.

OSD는 OSI를 운영하기 위해 필요한 각종 시스템 설정 정보 중 동적으로 운영이 가능한 정보들을 저장하는 영역이다. OSD는 실제 데이터가 저장되는 데이터셋과 OSD 관리를 위한 기타 모듈로 구성된다. OSD는 시스템이 구동될 때 RTSD(RunTime System Definition)라는 별도의 영역이 활성화되어 시스템 정보의 동적 관리 및 OSD의 변경 작업을 용이하게 한다.

데이터셋은 Staging Library와 접미사A,B가 붙은 3개를 준비한다. 데이터 등록은 A,B에 직접 하는 것도 가능하지만, 동적 변경을 하는 경우는 Staing Library에 넣은 정보를 동적 변경 툴과 명령어로 적용하도록 한다.



현재 버전의 OSD에서 데이터는 데이터셋이 아닌 메타 테이블에 저장되어 관리된다. Staging Library를 포함한 3개의 데이터셋은 Control Region을 기동하는 시점에만 필요하고, 실제로

해당 데이터셋에는 아무것도 저장되지 않는다.

## 테이블 생성

OSD를 사용하기 위해서는 시스템 준비 과정에서 메타 테이블을 생성하는 절차가 필요하다. 메타 테이블은 **osiinit**이라는 툴을 사용한다.

다음은 **osiinit** 툴을 이용해서 OSD 중 APPLCTN 매크로와 관련된 정보를 저장할 메타 테이블을 DEFVOL이라는 테이블 스페이스에 생성하는 예제이다.

```
$ osiinit create -t OFM_OSI_SD_APPLCTN -st DEFVOL
osiinit version 7.2.0(0) oframe@tmax:ofsrc/osi(#1) 2020-11-12 19:43:54
Initialize OpenFrame OSI System Tables

Creating OFM_OSI_SD_APPLCTN...
> "OFM_OSI_SD_APPLCTN" created...
```



메타 테이블에 대한 자세한 내용은 [리소스 테이블](#)을 참고한다.

## 데이터셋 생성

IMS/DC에서 Control Region을 기동하는 JCL을 OSI에서 그대로 사용하기 위해서는 시스템 준비 과정에서 데이터셋을 생성하는 절차가 필요하다. 데이터셋은 **idcams**라는 툴을 사용하여 생성할 수 있다.

다음은 **idcams** 툴을 이용하여 OSD 데이터셋을 DEFVOL이라는 볼륨에 생성하는 예제이다.

```
$ idcams define -t CL -n OSI.IMSA.DEFLIB -o KS -k 10,0 -l 100,32760 -s 1024,128,128 -v DEFVOL
idcams version 7.1.0(0) oframe@tmax:ofsrc/base(#1) 2020-06-29 17:30:26
Access Method Services for Catalogs

IDCAMS COMMAND=DEFINE,TYPE=CL,NAME=OSI.IMSA.DEFLIB,RELATE=,CATALOG=

tbESQL Precompiler 6

TmaxData Corporation Copyright (c) 2008-. All rights reserved.

/home/oframe/products/ofsrc/OpenFrame/tsam/temp/OSI_IMSA_DEFLIB.tbc is precompiled successfully!
COMPLETED SUCCESSFULLY.
```

## 리소스 정보 등록

생성한 OSD 테이블에 시스템을 운영할 때 사용하는 리소스 정보를 설정하는 방법은 IMS/DC에서 매크로를 이용해서 리소스를 정의하는 방식과 유사하다.

OSI에서는 **osisdgen**이라는 툴이 해당 기능을 제공한다. osisdgen은 Batch JOB 환경이 아닌 UNIX 환경에서 동작하는 툴 프로그램이다. osisdgen은 설정할 리소스 정보가 저장된 입력 파일과 리소스 정보를 저장할 Region의 IMSID를 가지고 작업을 수행하며 리소스 정보는 기존 IMS/DC에서 사용하던 매크로 문법을 그대로 지원한다.

다음은 일부 리소스 정의 스크립트가 저장되어 있는 osi\_sdlib.dat 파일을 이용해서 IMSA라는 Region에 리소스 정보를 등록하는 예제이다.

<osi\_sdlib.dat>

```

TYPE
TERMINAL NAME=OIVPTRM1, X
          FEAT=(PFK,CARD,PEN)
NAME     N031E01

APPLCTN PSB=OIVPI001,PGMTYPE=(,,),SCHDTYP=PARALLEL
TRANSACT CODE=OIVPMPP1,MSGTYPE=(SNGLSEG,RESPONSE,1),PRTY=(1,5), X
          MODE=SNGL

APPLCTN PSB=OIVPI002,PGMTYPE=(TP,,1),SCHDTYP=PARALLEL
TRANSACT CODE=OIVPMPP2,MSGTYPE=(SNGLSEG,RESPONSE,1),PRTY=(1,5), X
          MODE=SNGL,MAXRGN=1

APPLCTN PSB=OIVPI003,PGMTYPE=(TP,,1),SCHDTYP=PARALLEL
TRANSACT CODE=OIVPMPP3,MSGTYPE=(SNGLSEG,RESPONSE,1),PRTY=(1,5), X
          MODE=SNGL

APPLCTN PSB=OIVPI004,PGMTYPE=(TP,,1),SCHDTYP=PARALLEL
TRANSACT CODE=OIVPMPP4,MSGTYPE=(SNGLSEG,RESPONSE,1),PRTY=(1,5), X
          SPA=(60,STRUNC),MODE=SNGL

APPLCTN PSB=OIVPI005,PGMTYPE=(TP,,1),SCHDTYP=PARALLEL
TRANSACT CODE=OIVPMPP5,MSGTYPE=(SNGLSEG,RESPONSE,1),PRTY=(1,5), X
          SPA=(60,STRUNC),MODE=SNGL

APPLCTN PSB=OIVPIL02,PGMTYPE=(BATCH,,1),SCHDTYP=PARALLEL
TRANSACT CODE=OIVPBMP2,MSGTYPE=(SNGLSEG,RESPONSE,1),PRTY=(1,5), X
          MODE=SNGL

APPLCTN PSB=OIVPIL03,PGMTYPE=BATCH,SCHDTYP=PARALLEL
TRANSACT CODE=OIVPBMP3,MSGTYPE=(SNGLSEG,RESPONSE,1),PRTY=(1,5), X
          MODE=SNGL

APPLCTN PSB=OIVPIL04,PGMTYPE=BATCH,SCHDTYP=PARALLEL
TRANSACT CODE=OIVPBMP4,MSGTYPE=(SNGLSEG,RESPONSE,1),PRTY=(1,5), X
          MODE=SNGL

APPLCTN PSB=OIVPIL05,PGMTYPE=BATCH,SCHDTYP=PARALLEL
TRANSACT CODE=OIVPBMP5,MSGTYPE=(SNGLSEG,RESPONSE,1),PRTY=(1,5), X
          MODE=SNGL

```

osisdgen로 리소스 정보를 등록한다.

```

$ osisdgen osi_sd.dat IMSA
[2020-12-21T16:59:20.510891] [osisdgen(21666) ] [M] [OSI7203M] Processing result :

```

```
Success[20], Ignore[0], Error[0]
```

데이터셋에 등록된 리소스를 기동 중인 OSI 시스템에 반영하기 위해서는 다음 명령어들을 수행한다.

```
$ dfsuocu0 IMSA MODBLKS
- ACTIVE MODBLKS is MODBLKSA
- Succeeded to copy MODBLKS to MODBLKSB

$ imscmd IMSA /MOD PREPARE MODBLKS
IMS control region : [IMSA]
Requested command : [MOD PREPARE MODBLKS]
-----
ACTIVE DD: MODBLKSA IMSACBA FORMATA
MODIFY PREPARE COMMAND COMPLETED
*20356/170207*
-----
Command '/MOD PREPARE MODBLKS' execution done

$ imscmd IMSA /MOD COMMIT
IMS control region : [IMSA]
Requested command : [MOD COMMIT]
-----
ACTIVE DD: MODBLKSB IMSACBA FORMATA
MODIFY COMMIT COMMAND COMPLETED
*20356/170211*
-----
Command '/MOD COMMIT' execution done
```

## 2.3.6. MODSTAT

OSI에서 사용하는 데이터셋 가운데 MODBLKS, IMSACB, FORMAT의 Active Library의 정보를 보관하는 라이브러리이다.

MODSTAT을 처음 생성해서 레코드가 없는 경우, OSI는 부팅할 때 위의 3가지 라이브러리의 Staging Library의 내용을 접미사 A,B가 붙은 데이터셋 모두에 복사해준 뒤, A가 붙은 데이터셋들을 Active Library로 지정한다. OSI는 시스템을 부팅할 때마다 Active Library의 리소스를 읽어와서 사용한다.



현재 버전에서 MODSTAT의 Active Library 정보는 데이터셋이 아닌 메타 테이블에 레코드 형태로 저장되어 관리된다. 데이터셋은 Control Region을 기동하는 시점에만 필요하고, 실제로 해당 데이터셋에는 아무것도 저장되지 않는다.

### 테이블 생성

MODSTAT 리소스를 저장 및 관리하기 위해서는 시스템 준비 과정에서 메타 테이블을 생성하는 절차가 필요하다. 메타 테이블은 **osiinit**이라는 툴을 사용한다.

다음은 **osiinit** 툴을 이용해서 MODSTAT 테이블을 DEFVOL이라는 테이블 스페이스에 생성하는 예제이다.

```
$ osiinit create -t OFM_OSI_MODSTAT -st DEFVOL
osiinit version 7.2.0(0) oframe@tmax:ofsrc/osi(#1) 2020-11-12 19:43:54
Initialize OpenFrame OSI System Tables

Creating OFM_OSI_MODSTAT...
> "OFM_OSI_MODSTAT" created...
```



메타 테이블에 대한 자세한 내용은 [리소스 테이블](#)을 참고한다.

## 데이터셋 생성

IMS/DC에서 Control Region을 기동하는 JCL을 OSI에서 그대로 사용하기 위해서는 시스템 준비 과정에서 데이터셋을 생성하는 절차가 필요하다. 데이터셋은 **idcams**라는 툴을 사용하여 생성할 수 있다.

다음은 idcams 툴을 이용하여 MODSTAT 데이터셋을 DEFVOL이라는 볼륨에 생성하는 예제이다.

```
$ idcams define -t CL -n OSI.IMSA.MODSTAT -o KS -k 8,0 -l 160,160 -s 1024,128,128 -v DEFVOL
idcams version 7.1.0(0) oframe@tmax:ofsrc/base(#1) 2020-06-29 17:30:26
Access Method Services for Catalogs

IDCAMS COMMAND=DEFINE,TYPE=CL,NAME=OSI.IMSA.MODSTAT,RELATE=,CATALOG=

tbESQL Precompiler 6

TmaxData Corporation Copyright (c) 2008-. All rights reserved.

/home/oframe/products/ofsrc/OpenFrame/tsam/temp/OSI_IMSA_MODSTAT.tbc is precompiled successfully!

COMPLETED SUCCESSFULLY.
```

## 2.3.7. PSBLIB

PSB는 애플리케이션에서 사용되는 데이터베이스나 메시지를 사용하기 위한 컨트롤 블록인 PCB의 집합체로, 통상 하나의 사용자 프로그램당 하나의 PSB가 구성된다.

PCB(Program Communication Block)는 애플리케이션에서 OpenFrame HiDB에 대한 VIEW나 메시지 소스 또는 메시지 대상과 통신하기 위해 시스템에서 제공하는 컨트롤 블록이다. IO PCB/ALT PCB는 PCB 중에서도 OSI에서 제공하는 메시지 큐에 데이터를 읽고 쓰기 위해서 사용되는 리소스로서 IO PCB/ALT PCB는 OSI를 기동시킨 경우에만 사용이 가능하다. IO PCB는 OSI에서 구동되는 프로그램이라면 PSB의 기술과 상관없이 프로그램의 첫 번째 파라미터로 제공되는 IOPCB-MASK를 이용해서만 사용이 가능한 반면, ALT PCB는 PSB에 필요한만큼 기술하여 사용이 가능하다.

## 데이터셋 생성

PSBLIB을 사용하기 위해서는 시스템 준비 과정에서 데이터셋을 생성하는 절차가 필요하다.

PSBLIB은 PDS 형태로 구성되며 개별 환경설정 파일은 PSBLIB의 멤버 형태로 저장된다. OSI에서 PSBLIB을 생성하기 위해서는 **pdsgen**이라는 툴을 사용한다.

다음은 pdsgen을 사용하여 IMS.PSBLIB을 DEFVOL이라는 볼륨에 생성하는 예제이다.

```
$ pdsgen IMS.PSBLIB DEFVOL -f LB -l 32760
pdsgen version 7.1.0(0) oframe@tmax:ofsrc/base(#1) 2020-06-29 17:30:26
PDS Dataset Generation Program

pdsgen: *** PDS IMS.PSBLIB is created.
```

## PSB 정보 등록

OSI에서 PSB를 사용하기 위해서는 **psbgen**이라는 툴을 사용하여 PSB 정보를 등록해야 한다.

다음은 PSB를 정의하는 스크립트가 저장되어 있는 OIVPI002 파일을 이용해서 PSB 정보를 등록하는 예제이다.

<OIVPI002>

```
PCB TYPE=TP,MODIFY=YES
PCB TYPE=DB,DBDNAME=OIVPIDBD,KEYLEN=10,PROCOPT=A
SENSEG NAME=DBSEG
PSBGEN LANG=COBOL,PSBNAME=OIVPI002
END
```

psbgen으로 PSB 정보를 등록한다.

```
$ psbgen OIVPI002
psbgen version 7.2.0(0) oframe@tmax:ofsrc/ims(#3) 2020-08-08 16:21:16
Program Specification Block Generation Program

psbgen: force flag on; the tool overwrites existing data
psbgen: 1 files are requested in total
-----
psbgen Processing PSB script "OIVPI002"
-----

psbgen: Processing PSB "OIVPI002"
psbgen: removing existing PSB OIVPI002 metadata
psbgen: successfully processed PSB "OIVPI002"

psbgen: successfully processed for the requested PSBs (total 1)
```



현재 버전에서는 psbgen을 수행하게 되면 PSB 스크립트에 기술되어 있는 메타 정보는 메타 테이블에 저장된다. 실제로 PSBLIB에는 아무것도 저장되지 않는다.

## 2.3.8. RESLIB

데이터셋의 동적 사용에 필요한 MDA 정보를 저장하는 라이브러리이다. 상세한 내용은 [DFSRESLB](#)와 동일하다.

## 2.3.9. STEPLIB

OSI 서버에서 작동하는 애플리케이션 프로그램은 COBOL 프로그래밍 언어로 개발된다.

애플리케이션 개발자는 작성한 프로그램을 컴파일한 후 생성된 바이너리를 STEPLIB에 배치해야 한다. 컴파일 전에 전처리가 필요한 경우 전처리를 수행하도록 한다. 배치된 프로그램은 OSD 테이블에 프로그램 리소스 정의를 등록한 후 OSI 시스템을 기동하면 동작시킬 수 있다.

### 데이터셋 생성

FORMAT 생성과 마찬가지로 pdsgen을 사용하여 생성한다.

다음은 pdsgen을 사용하여 OSI.IMSA.STEPLIB를 DEFVOL이라는 볼륨에 생성하는 예제이다.

```
$ pdsgen OSI.IMSA.STEPLIB DEFVOL -f LB -l 32760
pdsgen version 7.1.0(0) oframe@tmax:ofsrc/base(#1) 2020-06-29 17:30:26
PDS Dataset Generation Program

pdsgen: *** PDS OSI.IMSA.STEPLIB is created.
```

## 2.4. 스토리지 설정

OSI는 시스템 운영을 위해서 스토리지 역할을 위한 시스템 테이블의 준비가 필요하다.

### 2.4.1. MQ

OSI는 터미널과 애플리케이션 사이에 메시지 전달을 위해서 기본적으로 TP-Monitor인 Tmax의 기능을 이용하고, 동시에 모든 메시지는 MQ(Message Queue) 테이블에 저장한다. OSI 시스템을 운영하기 위해서는 Master, MPP(Terminal), BMP 테이블 그리고 MQ 데이터셋을 준비한다.



현재 버전에서는 운용 중에 전달되는 모든 메시지를 MQ 테이블에 저장한다. MQ 데이터셋은 Control Region이 기동하는 시점에만 필요하며, 실제로 해당 데이터셋에는 아무것도 저장되지 않는다.

### 테이블 생성

MQ 시스템 테이블을 생성하기 위해서는 **osiinit** 툴을 사용한다.

다음은 osiinit 툴을 이용하여 3개의 MQ 테이블을 생성하는 예제이다.

```
$ osiinit create -t OFM_OSI_MQ -st DEFVOL
osiinit version 7.2.0(0) oframe@tmax:ofsrc/osi(#1) 2020-11-12 19:43:54
Initialize OpenFrame OSI System Tables

Creating OFM_OSI_MQ...
> "OFM_OSI_MQ" created...
```



MQ 시스템 테이블에 대한 자세한 내용은 [리소스 테이블](#)을 참고한다.

## 데이터셋 생성

IMS/DC에서 Control Region을 기동하는 JCL을 OSI에서 그대로 사용하기 위해서는 시스템 준비 과정에서 데이터셋을 생성하는 절차가 필요하다. 데이터셋은 **idcams**라는 툴을 사용하여 생성할 수 있다.

다음은 idcams 툴을 이용하여 MQ 데이터셋을 DEFVOL이라는 볼륨에 생성하는 예제이다.

```
$ idcams define -t CL -n OSI.IMSA.MQLIB -o KS -k 32,0 -l 10000,32000 -s 1024,128,128 -v DEFVOL
idcams version 7.1.0(0) oframe@tmax:ofsrc/base(#1) 2020-06-29 17:30:26
Access Method Services for Catalogs

IDCAMS COMMAND=DEFINE,TYPE=CL,NAME=OSI.IMSA.MQLIB,RELATE=,CATALOG=

tbESQL Precompiler 6

TmaxData Corporation Copyright (c) 2008-. All rights reserved.

/home/oframe/products/ofsrc/OpenFrame/tsam/temp/OSI_IMSA_MQLIB.tbc is precompiled successfully!

COMPLETED SUCCESSFULLY.
```



데이터셋 사용법의 자세한 설명은 OpenFrame Base "데이터셋 안내서"를 참고한다.

## 2.4.2. REGION

OSI 시스템을 운영하는데 사용되는 사용자 서버(Dependent Region)의 Region ID, JOB 이름, PSB 이름, PROGRAM 이름 등과 같은 정보들을 저장한다.

### 테이블 생성

REGION 테이블을 생성하기 위해서는 **osiinit** 툴을 사용한다.

다음은 osiinit 툴을 이용하여 REGION 테이블을 생성하는 예제이다.

```
$ osiinit create -t OFM_OSI_REGION -st DEFVOL
```

```
osiinit version 7.2.0(0) oframe@tmax:ofsrc/osi(#1) 2020-11-12 19:43:54
Initialize OpenFrame OSI System Tables

Creating OFM_OSI_REGION...
> "OFM_OSI_REGION" created...

$ osiinit create -t OFM_OSI_REGION_SVR -st DEFVOL
osiinit version 7.2.0(0) oframe@tmax:ofsrc/osi(#1) 2020-11-12 19:43:54
Initialize OpenFrame OSI System Tables

Creating OFM_OSI_REGION_SVR...
> "OFM_OSI_REGION_SVR" created...
```



REGION 테이블에 대한 자세한 내용은 [리소스 테이블](#)을 참고한다.

### 2.4.3. RTSD

OSI에서 각 Control Region이 기동되는 시점에 시스템 정의(System Definition) 리소스를 RTSD(RunTime System Definition)라는 별도의 영역으로 복사하여 관리한다. 시스템 운용 중에는 동일한 IMS 시스템 내에 있는 모든 Region 서버들이 동시에 같은 RTSD 리소스를 공유하게 된다.

#### 테이블 생성

RTSD 테이블을 생성하기 위해서는 **osiinit** 툴을 사용한다.

다음은 osiinit 툴을 이용하여 APPLCTN, TRANSACT 테이블을 DEFVOL이라는 테이블 스페이스에 생성하는 예제이다.

```
$ osiinit create -t OFM_OSI_RTSD_APPLCTN -st DEFVOL
osiinit version 7.2.0(0) oframe@tmax:ofsrc/osi(#1) 2020-11-12 19:43:54
Initialize OpenFrame OSI System Tables

Creating OFM_OSI_RTSD_APPLCTN...
> "OFM_OSI_RTSD_APPLCTN" created...

$ osiinit create -t OFM_OSI_RTSD_TRANSACT -st DEFVOL
osiinit version 7.2.0(0) oframe@tmax:ofsrc/osi(#1) 2020-11-12 19:43:54
Initialize OpenFrame OSI System Tables

Creating OFM_OSI_RTSD_TRANSACT...
> "OFM_OSI_RTSD_TRANSACT" created...
```



RTSD 테이블에 대한 자세한 내용은 [리소스 테이블](#)을 참고한다.

# 3. 시스템 서버 설정

본 장에서는 OSI 시스템 서버(Control Region)를 설정하는 방법을 기술한다.

## 3.1. 개요

OSI 시스템 서버들은 모두 TP-Monitor인 Tmax의 관리를 받는다.

다음은 OSI에서 운영을 위해 설정해야 하는 시스템 서버이다. 그 외 서버는 별도의 설정 과정 없이 사용이 가능하다.

구분	설명
기동/종료 서버(osiomsvr)	OSI 시스템 서버와 MPP 사용자 서버를 기동 및 종료를 담당하는 서버이다.
스케줄 서버(osisschd)	하나의 IMSID로 요청되는 모든 메시지는 이 스케줄 서버를 통해서 스케줄링되어 MPP 서버들로 메시지가 전달된다.
커맨드 서버(osicmdsv)	하나의 IMSID에서 명령어(OSI Command) 처리를 담당하기 위한 서버이다.
OTMA 서버(osiotmasvr)	IBM MQ 제품과 연동 및 OTMA 기능을 제공하는 서버이다.
관리자 서버(osiofmgr)	OpenFrame Manager 기능을 담당하는 서버이다.

OSI 시스템 서버를 운영하기 위해서는 시스템 서버 정보를 Tmax에 등록하는 과정이 필요하다. 시스템 서버를 Tmax에 등록하기 위해 Tmax의 환경설정 파일을 작성해야 한다. 환경설정의 자세한 설명은 각 절에서 자세하게 설명한다.

## 3.2. 기동/종료 서버(osiomsvr)

OSI 시스템 서버와 MPP 사용자 서버를 기동 및 종료를 담당하는 서버이다. IMS 시스템별로 등록할 필요없이 공통으로 등록된다.

### 3.2.1. Tmax 환경설정

다음은 osiomsvr 서버를 OSI에서 사용하기 위해서 Tmax 환경설정 파일에 공통으로 등록해야하는 서버 및 서비스를 설정하는 예제이다.

- 공통 설정

Tmax 환경설정 파일에 [SERVER] 절 및 [SERVICE] 절을 다음과 같이 설정한다.

```
*SERVER
osiomsvr          SVGNAME = svg_domain, MIN = 1, MAX = 1, SVRTYPE = UCS
```

```
*SERVICE
OSIOMSVRBOOT      SVRNAME = osiomsvr
OSIOMSVRDOWN      SVRNAME = osiomsvr
OSIOMSVRREL        SVRNAME = osiomsvr
```

### 3.3. 스케줄 서버(osisschd)

OSI 메시지의 스케줄링과 그에 필요한 제반 기능들을 제공하는 시스템 서버이다. 스케줄 서버는 IMS시스템별로 하나씩 등록된다.

#### 3.3.1. Tmax 환경설정

다음은 스케줄 서버를 OSI에서 사용하기 위해서 Tmax 환경설정 파일에 공통으로 등록해야하는 서버 및 서비스와 각 IMS 시스템별로 등록해야하는 서버를 설정하는 예제이다.

- 공통 설정

Tmax 환경설정 파일에 [SERVER] 절 및 [SERVICE] 절을 다음과 같이 설정한다. 서비스는 IMS 시스템별로 등록할 필요없이 공통으로 한번만 설정하면 된다.

```
*SERVER
osisschd          SVGNAME = svg_domain, MIN = 0, MAX = 10, SVRTYPE = UCS

*SERVICE
OSISSCHDCTL       SVRNAME = osisschd
OSISSCHDSVC       SVRNAME = osisschd
OSISSCHDREG       SVRNAME = osisschd
OSISSCHDDTP       SVRNAME = osisschd
OSISSCHDOTMA      SVRNAME = osisschd
```

- IMS 시스템별 설정

IMSID가 IMSA인 경우에는 IMSASCHD라는 이름으로 된 서버를 등록해야 한다. 예제에서 굵은 글씨체로 표시된 부분이 IMSID가 다른 경우 변경해야 하는 부분이다. IMSASCHD 서버의 실제 바이너리 이름인 osisschd는 TARGET 옵션을 사용하여 지정하도록 한다.

```
*SERVER
IMSASCHD         SVGNAME = svg_domain, MIN = 1, MAX = 1, SVRTYPE = UCS,
                   TARGET = osisschd
```

### 3.4. 커맨드 서버(osicmdsv)

커맨드 서버는 OSI 시스템 운영에 필요한 여러 가지 기능을 명령어로 처리하는 서버로, IMS 시스템별로 하나씩 등록되어야 한다.

### 3.4.1. Tmax 환경설정

다음은 커맨드 서버를 OSI에서 사용하기 위해서 Tmax 환경설정 파일에 등록하는 예제이다.

스케줄 서버와 마찬가지로 공통으로 등록해야 하는 서버 및 서비스와 각 IMS 시스템별로 등록해야하는 서버가 있다.

- 공통 설정

Tmax 환경설정 파일에 [SERVER] 절 및 [SERVICE] 절을 다음과 같이 설정한다. 서비스는 IMS 시스템별로 등록할 필요없이 공통으로 한번만 설정하면 된다.

```
*SERVER
osicmdsv      SVGNAME = svg_domain, MIN = 0, MAX = 10, SVRTYPE = UCS

*SERVICE
OSICMDSVSVC   SVRNAME = osicmdsv
```

- IMS 시스템별 설정

IMSID가 IMSA인 경우에는 IMSACMMD라는 이름으로 된 서버를 등록해야 한다. 예제에서 굵은 글씨체로 표시된 부분이 IMSID가 다른 경우 변경해야 하는 부분이다. IMSACMMD 서버의 실제 바이너리 이름인 osicmdsv는 TARGET 옵션을 사용하여 지정하도록 한다.

```
*SERVER
IMSACMMD      SVGNAME = svg_domain, MIN = 1, MAX = 1, SVRTYPE = UCS,
TARGET = osicmdsv
```

## 3.5. OTMA 서버(osiotmasvr)

OTMA 서버는 IBM MQ 제품과 연동 및 OTMA 기능을 제공하는 서버로 IMS 시스템별로 하나씩 등록되어야 한다.

### 3.5.1. Tmax 환경설정

다음은 OSI에서 OTMA 서버를 사용하기 위해 Tmax 환경설정 파일에 등록하는 예제이다.

스케줄 서버와 마찬가지로 공통으로 등록해야 하는 서버 및 서비스와 각 IMS 시스템별로 등록해야하는 서버가 있다.

- 공통 설정

Tmax 환경설정 파일에 [SERVER] 절 및 [SERVICE] 절을 다음과 같이 설정한다. 서비스는 IMS 시스템별로 등록할 필요없이 공통으로 한번만 설정하면 된다.

```
*SERVER
osiotmasvr    SVGNAME = svg_node1, MIN = 0, MAX = 10, SVRTYPE = UCS
```

- IMS 시스템별 설정

IMSID가 IMSA인 경우에는 IMSAOTMA라는 이름으로 된 서버를 등록해야 한다. 예제에서 굵은 글씨체로 표시된 부분이 IMSID가 다른 경우 변경해야 하는 부분이다. IMSAOTMA 서버의 실제 바이너리 이름인 osiotmasvr는 TARGET 옵션을 사용하여 지정하도록 한다.

```
*SERVER
IMSAOTMA          SVGNAME = svg_node1, MIN = 1, MAX = 1, SVRTYPE = UCS,
                   TARGET = osiotmasvr
```

## 3.6. 관리자 서버(osiofmgr)

OpenFrame Manager 기능을 담당하는 서버이다. IMS 시스템별로 등록할 필요없이 공통으로 등록된다.

### 3.6.1. Tmax 환경설정

다음은 OSI에서 관리자 서버를 사용하기 위해 Tmax 환경설정 파일에 등록하는 예제이다.

- 공통 설정

Tmax 환경설정 파일에 [SERVER] 절 및 [SERVICE] 절을 다음과 같이 설정한다.

```
*SERVER
osiofmgr          SVGNAME = svg_node, MIN = 1, MAX = 10

*SERVICE
OSIOFMGRSVC      SVRNAME = osiomsvr
```

## 4. 사용자 서버 설정

본 장에서는 OSI 사용자 서버(Dependent Region)를 설정하는 방법을 기술한다.

### 4.1. 개요

OSI 시스템에서는 사용자가 작성한 업무 프로그램을 기동하기 위해 IBM 메인프레임 IMS/DC에서 사용자 서버(Dependent Region)와 대응하는 서버를 준비해야 한다.

다음은 OSI를 운영하는 데 사용되는 사용자 서버에 대한 설명이다.

구분	설명
MPP 사용자 서버	IMS/DC에서 MPP Region에 대응되는 부분으로 OSI에서는 메시지의 class 단위로 MPP 서버를 운영한다. 즉, 기존 IMS/DC에서 운영 중이던 MPP Region 당 최대 1:4 대응되는 서버이다.
BMP 사용자 서버	IMS/DC의 BMP Region에서 동작하는 사용자 프로그램이 운영될 수 있도록 하는 서버이다.

OSI 사용자 서버들은 모두 TP 모니터인 Tmax의 관리를 받으며 이를 운영하기 위해서는 사용자 서버들의 정보를 Tmax에 등록하는 과정이 필요하다. 서버를 Tmax에 등록하기 위해 Tmax의 환경설정 파일을 작성해야 한다.

사용자 서버를 준비하는 과정에서 MPP 사용자 서버와 BMP 타입의 사용자 서버를 준비하는 과정은 서로 다르다. Batch 작업을 실행시키는 BMP Region은 시스템이 제공하는 서버 모듈을 이용하기 때문에 따로 서버를 생성하지는 않지만 동시에 최대로 수행할 BMP JOB의 개수를 고려해서 설정하여야 한다. 환경설정에 대한 상세한 설명은 각 절에서 설명한다.

### 4.2. 서버 그룹 설정

다음은 서버 그룹의 환경설정에 대한 예시이다.

#### 4.2.1. 일반 환경설정

OSI 일반 환경설정 예시이다.

```
*SVRGROUP
svg_node1
  NODENAME = "NODE1"
```

#### 4.2.2. XA 환경설정

OSI가 사용하는 HiDB와 일반 Tibero를 Multiple RM으로 설정한 XA 환경설정 예시이다.

```

*SVRGROUP
svg_node1
  NODENAME = "NODE1"
  SVGTYPE = MTMAX,
  SVGLIST = "svg_hidb_xa,svg_tibero_xa"

svg_hidb_xa
  NODENAME = "NODE1"
  SVGTYPE = STMAX,
  TMSNAME = tms_tbr,
  DBNAME = TIBERO,
  OPENINFO = "TIBERO_XA:user=tibero, pwd=tmax, sestm=60,db=tb_fix3,conn_id=db1"

svg_tibero_xa
  NODENAME = "NODE1"
  SVGTYPE = STMAX,
  TMSNAME = tms_tbr,
  DBNAME = TIBERO,
  OPENINFO = "TIBERO_XA:user=tibero, pwd=tmax, sestm=60,db=tb_fix3"

```

`#{TAXMDIR}/config/RM`에 Multiple RM 설정된 서버가 동적 로드할 라이브러리를 지정한다.

```

#TIBERO
TIBERO:tbxa,tbs

# IBM MQ for 64bit
MQ:mqmx64,mqs

# IBM MQ for 32bit
MQ:mqmx32,mqs

```



1. HiDB를 위한 XA 설정이 필요한 경우는 Tmax 설정파일에 관련 설정을 추가해야 한다. OpenFrame 환경설정 중 **ssm.#{IMSID}{SSM}** 서브젝트, **GENERAL** 섹션에 **DLI\_CONN\_ID** 키의 VALUE 항목 값과 Tmax 설정파일의 SVRGROUP 설정에서 OPENINFO에 기술한 conn\_id가 동일해야 한다. 자세한 내용은 OpenFrame OSI "환경설정 안내서"의 'OpenFrame OSI 환경설정'을 참고한다.
2. XA 설정을 할 경우 사용자 서버는 해당 서버그룹 내의 모든 DB 에 대하여 세션을 가지게 된다.

### 4.3. MPP 사용자 서버

IMS/DC에서 MPP Region에 대응되는 부분으로 기존 IMS/DC에서 운영 중이던 MPP Region 당 최대 4개의 MPP 사용자 서버로 대응된다. 준비된 MPP 사용자 서버에서 사용자가 작성한 MPP 애플리케이션이 IMS/DC에서 동작했던 방식 그대로 동일하게 스케줄링되어 동작한다.

시스템 서버들과 마찬가지로 MPP 사용자 서버도 TP-Monitor인 Tmax의 관리를 받는다. Tmax의 관리를 받기 위해서는 Tmax 환경설정 파일을 작성해야 한다.

다음은 MPP 사용자 서버를 OSI에서 사용하기 위해서 Tmax 환경설정 파일에 공통으로 등록해야하는 서버 및 서비스와 IMSA라는 IMS 시스템에서 transaction class 1,2,3,4를 처리할 수 있는 서버를 설정하는 예제이다.

- 공통 설정

Tmax 환경설정 파일에 [SERVER] 절 및 [SERVICE] 절을 다음과 같이 설정한다. 서비스는 IMS 시스템별로 등록할 필요없이 공통으로 한번만 설정하면 된다.

```
*SERVER
OSIMPPSVR      SVGNAME = svg_node1, MIN = 0, MAX = 10

*SERVICE
OSIMPPSVRSVC   SVRNAME = OSIMPPSVR, SVCTIME=60
OSIMPPSVRMGR   SVRNAME = OSIMPPSVR, SVCTIME=60
```

- IMS 시스템별 설정

IMSID와 메시지의 class로 구분하며 각각 서버명에 접두어, 접미어로 추가하여 등록한다. 예제에서 굵은 글씨로 표시된 부분이 IMSID와 class가 다른 경우 변경해야 하는 부분이다. IMSAMPP\_TCL1~4 서버의 실제 바이너리 이름인 OSIMPPSVR은 TARGET 옵션을 사용하여 지정하도록 한다.

```
*SERVER
IMSAMPP_TCL1  SVGNAME = svg_node1, MIN = 1, MAX = 10,
                TARGET = OSIMPPSVR
IMSAMPP_TCL2  SVGNAME = svg_node1, MIN = 1, MAX = 10,
                TARGET = OSIMPPSVR
IMSAMPP_TCL3  SVGNAME = svg_node1, MIN = 1, MAX = 10,
                TARGET = OSIMPPSVR
IMSAMPP_TCL4  SVGNAME = svg_node1, MIN = 1, MAX = 10,
                TARGET = OSIMPPSVR
```



Tmax 환경설정에 MPP 사용자 서버를 등록하는 경우 MAX 값 설정에 주의해야 한다. MAX 값은 해당 MPP 사용자 서버에서 동시 처리가 가능한 트랜잭션의 수를 의미하는 값으로, 최대로 기동할 MPP 수를 설정하면 MPP사용자 서버는 IMS/DC의 MPP Region과 동일하게 동작한다.

## 4.4. BMP 사용자 서버

BMP 사용자 서버는 IMS/DC의 BMP Region에서 동작하는 사용자 프로그램이 운영될 수 있도록 하는 서버이다. MPP 사용자 서버는 사용자 서버(Dependent Region)별로 별도의 서버 모듈을 생성하는 방식이었으나 BMP 사용자 서버는 모든 사용자 서버(Dependent Region)에 대하여 하나의 제품에서 제공된 BMP 사용자 서버 모듈을 사용한다.

BMP 사용자 서버는 OSI의 서버 모듈을 제품에 포함하여 제공하므로 MPP 사용자 서버와는 다르게 별도의 생성 과정이 필요하지 않고 전체 OSI 시스템에 대하여 등록하는 과정만 필요하다.

시스템 서버들과 마찬가지로 BMP 사용자 서버도 TP-Monitor인 Tmax의 관리를 받는다. Tmax의 관리를 받기

위해서는 Tmax 환경설정 파일을 작성해야 한다.

다음은 Tmax 환경설정 파일에 BMP 사용자 서버를 등록하는 예제이다.

- 공통 설정

Tmax 환경설정 파일에 [SERVER] 절 및 [SERVICE] 절을 다음과 같이 설정한다.

```
*SERVER
osibmpsv          SVGNAME = svg_node1, MIN = 0, MAX = 10, SVRTYPE = UCS,
                  RESTART = NO

*SERVICE
OSIBMPSVSVC      SVRNAME = osibmpsv
OSIBMPSVSHUTDOWN SVRNAME = osibmpsv
```

- XA 설정

ssm.{IMSID}{SSM} 설정 이름으로 구분하며, 서버명에 접미어로 추가하여 등록한다. 예제에서 굵은 글씨로 표시된 부분이 SSM 설정이 다른 경우 변경해야 하는 부분이다. osibmpsv.DB2T 서버의 실제 바이너리 이름인 osibmpsv는 TARGET 옵션을 사용하여 지정하도록 한다.

```
*SERVER
osibmpsv.DB2T    SVGNAME = svg_node1, MIN = 0, MAX = 10, SVRTYPE = UCS,
                  TARGET = osibmpsv, RESTART = NO
```



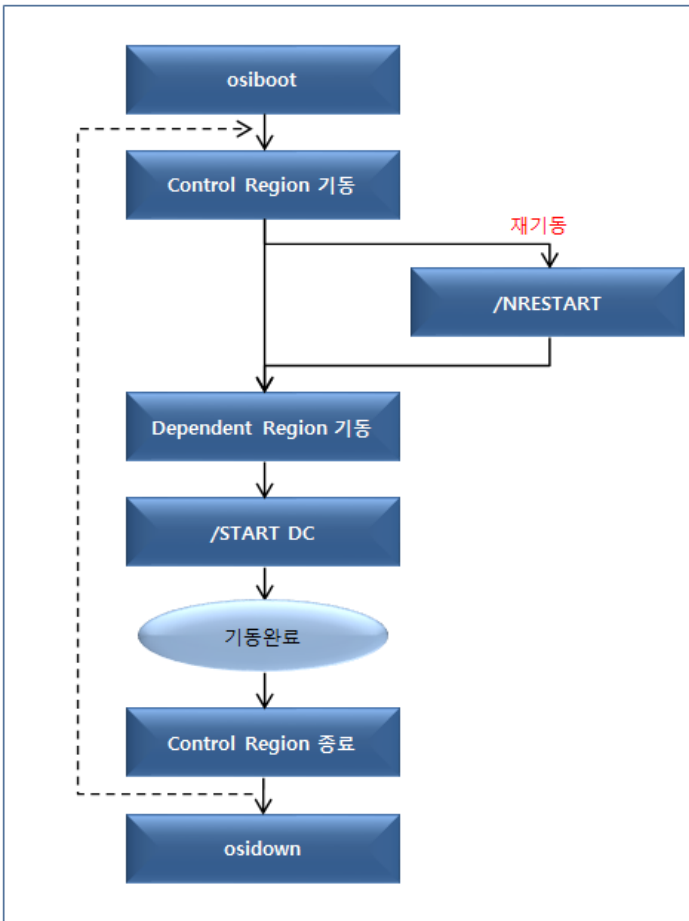
Tmax 환경설정에 BMP 사용자 서버를 등록할 경우 주의해야 할 부분은 MAX 값 설정이다. MAX 값은 OSI에서 동시에 활성화가 가능한 BMP JOB의 수를 의미한다.

# 5. 시스템 운영

본 장에서는 OSI 시스템을 기동 및 종료하는 방법 및 운영 중인 OSI 시스템의 운영정보가 기록되는 로그 파일을 사용하는 방법에 대해 기술한다.

## 5.1. 기동 및 종료

OSI 시스템이 동작하기 위해서는 TP-Monitor인 Tmax가 기동되어 있고, OpenFrame Base, OpenFrame Batch의 모듈들이 동작 가능하도록 준비되어 있어야 한다.



OSI 기동 및 종료 흐름



Control Region 기동 전에는 TCache가 반드시 생성되어 있어야 한다. TCache는 **pfmtcacheadmin** 툴을 사용하여 생성할 수 있다.

### 5.1.1. 기동

다음은 OSI의 기동 순서이다.

1. Tmax 엔진과 OSI 시스템에 필요한 base/batch 모듈 및 osiomsvr를 기동한다. (osiboot)

Tmax 기동은 내부적으로 2단계로 진행된다. 1, 2, 3단계는 **osiboot** 툴을 통하여 한꺼번에 수행된다.

- a. Tmax가 기동된다.
- b. OSI 기동에 필요한 다른 OpenFrame의 패키지들에서 사용되는 서버 모듈들이 기동된다.

2번째 단계의 수행을 위해서 OSI 이외의 패키지에서 사용되는 서버 모듈들은 미리 "osi.ofsys.seq"라는 환경설정 파일에 서버별 기동 순서를 고려하여 설정되어 있어야 한다.

OSI에서는 BMP의 동작을 위해서 통상적으로 OpenFrame Batch에서 필요한 서버 모듈들이 Tmax가 기동될 때 함께 기동되어야 한다. 아래의 예제는 이를 위한 일반적인 예제이다.

```
$ cat osi.ofsys.seq
TPFMAGENT
ofrsasvr
ofrlhsvr
ofrdmsvr
ofrdsedt
ofrcmsvr
ofruisvr
ofrsmlog
obmjmsvr
obmj schd
obmjinit
obmjhist
obmjspb k
ofrpmsvr
obmtsmgr
obmjtimr
tmsvr
#IMSAOTMA
```

- c. OSI 시스템 서버와 사용자 서버를 기동시키기 위한 osiomsvr가 기동된다.

다음과 같이 osiboot로 Tmax를 기동시킨다.

```
$ osiboot

TMBOOT for node(NODE1) is starting:
  TMBOOT: TMM is starting: Wed Mar 10 19:37:02 2021
  TMBOOT: CLL is starting: Wed Mar 10 19:37:02 2021
  TMBOOT: CLH is starting: Wed Mar 10 19:37:02 2021
  TMBOOT: TLM(tlm) is starting: Wed Mar 10 19:37:02 2021
[2021-03-10T19:37:02.848282] [osiboot(5670) ] [M] [OSI7061M] System server(ofrsasvr)
booting ok
[2021-03-10T19:37:02.851625] [osiboot(5670) ] [M] [OSI7061M] System server(ofrlhsvr)
booting ok
[2021-03-10T19:37:02.855374] [osiboot(5670) ] [M] [OSI7061M] System server(ofrdmsvr)
booting ok
[2021-03-10T19:37:02.859919] [osiboot(5670) ] [M] [OSI7061M] System server(ofrdsedt)
booting ok
[2021-03-10T19:37:02.864295] [osiboot(5670) ] [M] [OSI7061M] System server(ofrcmsvr)
booting ok
[2021-03-10T19:37:02.868285] [osiboot(5670) ] [M] [OSI7061M] System server(ofruisvr)
booting ok
[2021-03-10T19:37:02.873191] [osiboot(5670) ] [M] [OSI7061M] System server(ofrsmlog)
booting ok
```

```

[2021-03-10T19:37:02.877533] [osiboot(5670) ] [M] [OSI7061M] System server(obmjmsvr)
booting ok
[2021-03-10T19:37:02.881853] [osiboot(5670) ] [M] [OSI7061M] System server(obmjschd)
booting ok
[2021-03-10T19:37:02.886434] [osiboot(5670) ] [M] [OSI7061M] System server(obmjinit)
booting ok
[2021-03-10T19:37:02.890673] [osiboot(5670) ] [M] [OSI7061M] System server(obmjhist)
booting ok
[2021-03-10T19:37:02.897808] [osiboot(5670) ] [M] [OSI7061M] System server(obmjspb)
booting ok
[2021-03-10T19:37:02.905868] [osiboot(5670) ] [M] [OSI7061M] System server(ofrpmsvr)
booting ok
[2021-03-10T19:37:02.920575] [osiboot(5670) ] [M] [OSI7061M] System server(obmtsmgr)
booting ok
[2021-03-10T19:37:02.937001] [osiboot(5670) ] [M] [OSI7061M] System server(tmsvr)
booting ok
[2021-03-10T19:37:02.943765] [osiboot(5670) ] [M] [OSI7061M] System server(osiomsvr)
booting ok
[2021-03-10T19:37:02.943791] [osiboot(5670) ] [M] [OSI7051M] Booting process complete.

```

## 2. JCL로 IMSID별 시스템 서버(Control Region)를 기동한다.

Tmax가 기동된 이후에 OSI 시스템 서버(Control Region)를 기동한다. OSI 시스템은 IMSID별로 기동되므로 3개의 OSI 시스템을 사용하는 경우라면 IMSID별로 3번의 OSI 시스템을 기동한다. OSI 시스템 기동은 tjesmgr 툴을 사용하여 JCL로 된 JOB을 실행한다.

다음과 IMSID가 IMSA인 시스템 서버(Control Region)를 기동시키는 JCL의 예제이다.

```

//IMSACTL JOB
//STEP1 EXEC PGM=DFSMVRC0,
// PARM='CTL,IMS,IMSID=IMSA'
//MODBLKSA DD DISP=SHR,DSN=OSI.IMSA.DEFLIBA
//MODBLKSB DD DISP=SHR,DSN=OSI.IMSA.DEFLIBB
//IMSACBA DD DISP=SHR,DSN=IMS.ACBLIBA
//IMSACBB DD DISP=SHR,DSN=IMS.ACBLIBB
//DFSRESLB DD DISP=SHR,DSN=IMS.RESLIB
//FORMATA DD DISP=SHR,DSN=OSI.IMSA.MFSLIBA
//FORMATB DD DISP=SHR,DSN=OSI.IMSA.MFSLIBB
//STEPLIB DD DISP=SHR,DSN=OSI.IMSA.STEPLIB
//QBLKS DD DISP=SHR,DSN=OSI.IMSA.MQLIB
//MODSTAT DD DISP=SHR,DSN=OSI.IMSA.MODSTAT
//SYSOUT DD SYSOUT=*
//

$ tjesmgr boot
>
Command : [boot]
Node name : A L L
NODE1 is booted.

$ tjesmgr run IMSACTL
>
Command : [run IMSACTL]
Node name : A N Y
(JOB00001) /root/OF71/OHOME/volume_DEFVOL/SYS1.JCLLIB/IMSACTL is submitted as IMSACTL(JOB00001).

```

IMS 시스템 서버를 기동하는 JCL의 PARM 파라미터에는 다음 두 가지 항목의 설정이 가능하다.

항목	설명
IMSID	4bytes의 시스템 서버의 고유한 IMS ID를 지정한다.
AUTO	서버 기동 후에 "/NRESTART"나 "/ERESTART" 명령을 자동으로 수행할 것인지 여부를 설정한다. YES나 NO로 지정이 가능하다.

### 3. OSI 명령어를 사용하여 OSI 시스템을 운영 가능한 상태로 만든다.

JCL로 시스템 서버(Control Region)를 기동한 후 OSI 시스템 최초 기동이라면 "/NRE CHKPT 0", 재기동이라면 "/NRE" 또는 "/ERE" 명령어를 사용해야 한다. "/NRE" 명령은 이전에 "/CHE" 명령을 통해서 서버를 정상적으로 종료한 경우에 사용되며, 정상적으로 종료되지 않은 경우에는 "/ERE" 명령어를 사용해야 한다. 그렇지 않으면 에러가 발생하고 명령이 실행되지 않는다. 이후 단말의 로그온을 위해서 "/START DC" 명령어를 실행해야 한다.

다음은 OSI 명령어로 IMSID가 IMSA인 시스템 서버의 /NRE, /STA DC를 순서대로 입력하는 예제이다.

```
$ imscmd IMSA /NRE
IMS control region : [IMSA]
Requested command : [NRE]
-----
NRESTART COMMAND IN PROGRESS
*21069/194013*
-----
Command '/NRE' execution done

$ imscmd IMSA /STA DC
IMS control region : [IMSA]
Requested command : [STA DC]
-----
START COMMAND COMPLETED
*21069/194018*
-----
Command '/STA DC' execution done
```

### 4. JCL로 IMSID별 사용자 서버(Dependent Region)를 기동한다.

시스템 서버(Control Region)가 기동된 후에는 처리할 CLASS에 따라서 MPP 기동 JCL을 실행시킨다. 처리할 갯수만큼 JOB을 실행하여 MPP를 기동시킬 수 있다. 또한 OSI 명령어인 "/START REGION"을 사용해서 기동할 수 있다.

다음은 IMSID가 IMSA인 사용자 서버(Dependent Region)를 기동시키는 JCL의 예제이다. 클래스를 지정하지 않을 시 '000' 으로 설정 가능하며 첫번째 클래스는 필수로 지정해야 한다. (본 예제에서 처리할 클래스는 1, 2, 3, 4이다)

```
//IMSMSG JOB
//STEP1 EXEC PGM=DFSRR00,
// PARM='MSG,001002003004,W00099000,,,R,,,,IMSA,,,,'
//STEPLIB DD DISP=SHR,DSN=OSI.IMSA.STEPLIB
//SYSPRINT DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//SYSDBOU DD SYSOUT=*
```

```
//
```

다음은 JCL이름이 IMSAMSG인 사용자 서버(Dependent Region)를 기동시키는 명령어 "/START REGION"의 예제이다.

```
$ imscmd IMSA /START REGION IMSAMSG
IMS control region : [IMSA]
Requested command  : [START REGION IMSAMSG]
-----
START COMMAND COMPLETED
*21069/194050*
-----
Command '/START REGION IMSAMSG' execution done
```



"/START REGION"에 대한 자세한 사용법은 OpenFrame OSI "명령어 참조 안내서"를 참고한다.

## 5.1.2. 종료

다음은 OSI의 종료 순서이다.

### 1. 시스템 서버(Control Region)와 사용자 서버(Dependet Region)를 종료한다.

시스템 서버(Control Region)와 사용자 서버(Dependet Region)를 종료하기 위해서는 OSI 명령어를 사용한다. OSI 명령어는 CHECKPOINT가 사용되며, 사용자 서버 종료 후 시스템 서버를 종료한다.

다음은 OSI 명령어로 IMSID가 IMSA인 시스템 서버와 사용자 서버를 종료하는 예제이다.

```
$ imscmd IMSA /CHE FREEZE
IMS control region : [IMSA]
Requested command  : [CHE FREEZE]
-----
CHECKPOINT COMMAND IN PROGRESS
*21069/194108*
-----
Command '/CHE FREEZE' execution done
```

### 2. Tmax를 종료한다.

osidown 툴을 이용하면 OpenFrame의 패키지들에서 사용되는 서버 모듈들을 종료된 후 Tmax가 종료된다.

다음은 osidown로 Tmax를 종료시키는 예제이다.

```
$ osidown
[2021-03-10T19:41:27.956075] [osidown(5969)          ] [M] [OSI7141M] System server(osiomsvr)
shutdown ok
[2021-03-10T19:41:27.960864] [osidown(5969)          ] [M] [OSI7141M] System server(tmsvr)
```

```

shutdown ok
[2021-03-10T19:41:27.983167] [osidown(5969) ] [M] [OSI7141M] System server(obmtsmgr)
shutdown ok
[2021-03-10T19:41:27.995447] [osidown(5969) ] [M] [OSI7141M] System server(ofrpsvr)
shutdown ok
[2021-03-10T19:41:27.000641] [osidown(5969) ] [M] [OSI7141M] System server(obmjspb)
shutdown ok
[2021-03-10T19:41:28.003942] [osidown(5969) ] [M] [OSI7141M] System server(obmjhist)
shutdown ok
[2021-03-10T19:41:28.033647] [osidown(5969) ] [M] [OSI7141M] System server(obmjinit)
shutdown ok
[2021-03-10T19:41:28.040265] [osidown(5969) ] [M] [OSI7141M] System server(obmjschd)
shutdown ok
[2021-03-10T19:41:28.047434] [osidown(5969) ] [M] [OSI7141M] System server(obmjmsvr)
shutdown ok
[2021-03-10T19:41:28.076810] [osidown(5969) ] [M] [OSI7141M] System server(ofrsmlog)
shutdown ok
[2021-03-10T19:41:28.086320] [osidown(5969) ] [M] [OSI7141M] System server(ofruisvr)
shutdown ok
[2021-03-10T19:41:28.093782] [osidown(5969) ] [M] [OSI7141M] System server(ofrcmsvr)
shutdown ok
[2021-03-10T19:41:28.101464] [osidown(5969) ] [M] [OSI7141M] System server(ofrdsedt)
shutdown ok
[2021-03-10T19:41:28.108771] [osidown(5969) ] [M] [OSI7141M] System server(ofrdmsvr)
shutdown ok
[2021-03-10T19:41:28.116562] [osidown(5969) ] [M] [OSI7141M] System server(ofrlhsvr)
shutdown ok
[2021-03-10T19:41:28.124770] [osidown(5969) ] [M] [OSI7141M] System server(ofrsasvr)
shutdown ok

```

```

TMDOWN for node(NODE1) is starting:
TMDOWN: CLH downed: Wed Mar 10 19:41:28 2021
TMDOWN: CLL downed: Wed Mar 10 19:41:28 2021
TMDOWN: TLM downed: Wed Mar 10 19:41:28 2021
TMDOWN: TMM downed: Wed Mar 10 19:41:28 2021
TMDOWN: TMAX is down

```

### 5.1.3. 재기동

다음은 OSI의 재기동 순서이다.

1. Tmax를 기동한다. (osiboot)
2. JCL로 IMSID별 시스템 서버(Control Region)를 기동한다.
3. 명령어로 IMSID별 시스템 서버(Control Region)를 재기동한다. OSI 명령어를 통해 종료한 후 부터는 이전 시스템 사용 정보를 가져오기 위해 반드시 재기동이 필요하다.

다음은 OSI 명령어로 IMSID가 IMSA인 시스템 서버(Control Region)를 재기동하는 예제이다.

```

$ imscmd IMSA /NRE
IMS control region : [IMSA]
Requested command : [NRE]
-----
NRESTART COMMAND IN PROGRESS
*21069/195025*

```

```
Command '/NRE' execution done
```

4. "/START DC"를 사용하여 터미널이 로그인할 수 있는 상태로 만든다.
5. JCL로 IMSID별 사용자 서버(Dependent Region)를 기동하거나 "/START REGION"을 사용하여 기동한다.



"/START REGION"에 대한 자세한 사용법은 OpenFrame OSI "명령어 참조 안내서"를 참고한다.

## 5.2. 로그 관리

OpenFrame OSI 시스템에서는 서버 로그를 시스템 서버 및 MPP 사용자 서버와 BMP 사용자 서버로 나눠 관리한다. 본 절에서는 각 서버별 관리 방식을 설명한다.

### 5.2.1. 시스템 서버 및 MPP 사용자 서버

OSI 시스템 서버와 MPP 사용자 서버들이 생성하는 로그는 OUT 로그와 ERR 로그가 있다. 두 로그 모두 Tmax 설정 파일에서 NODE 절에 기술된 ULOGDIR에 저장된다. ULOGDIR은 사용자가 자유롭게 설정할 수 있으나 `${OPENFRAME_HOME}/log` 아래의 디렉터리로 설정할 것을 권장한다.

#### • OUT 로그

CLOPT 절에 [-o] 옵션과 함께 파일명을 지정하면, OSI 시스템 서버 및 MPP 사용자 서버에서 stdout으로 쓰는 모든 내용이 해당 파일에 출력된다. 출력되는 로그는 OpenFrame의 공통 출력 포맷을 따른다.

다음은 로그 출력 포맷이다.

```
[YYYY-MM-DDTHH:MI:SS.FFFFFFF] [MODULE(PID)] [LEVEL] [MSGCODE] MESSAGE-CONTENTS
```

항목	설명
[YYYY-MM-DDTHH:MI:SS.FFFFFFF]	날짜를 포함한 YYYY-MM-DDTHH:MI:SS.FFFFFFF 타입의 timestamp이다.
MODULE	메시지를 출력하는 OSI 시스템의 모듈명이다.
PID	Process ID이다.

항목	설명
LEVEL	<p>다음 중 하나가 출력된다.</p> <ul style="list-style-type: none"> <li>• M(Message) : 시스템이 사용자에게 정보를 알려주는 메시지이다. 서버의 각종 정보들이 출력된다.</li> <li>• E(Error) : 시스템에서 에러가 발생한 경우에 출력되는 메시지로, 사용자가 시스템에 잘못된 정보를 주었거나 시스템 내부적으로 에러가 있는 경우에 출력된다.</li> <li>• W(Warning) : 시스템에서 에러가 발생하였으나, 시스템을 운영하는데에는 영향을 미치지 않는 경우에 출력되는 메시지이다.</li> <li>• D(Debug) : 시스템 디버깅 용도로 출력되는 메시지이다.</li> </ul>
MSGCODE	8자리의 OSI 메시지 코드이다.
MESSAGE-CONTENTS	로그 메시지가 출력되는 필드이다.

다음은 이름이 IMSA인 IMS 시스템에서 메시지 class가 1인 MPP 사용자 서버의 OUT로그 파일에 대한 예제이다.

```
[2021-03-10T19:59:51.194767] [IMSAMPP_TCL1(6839) ] [D] [DEBUGP2D] [GENERAL]LOG_LEVEL = D
[2021-03-10T19:59:51.217509] [IMSAMPP_TCL1(6839) ] [D] [DEBUGP1D] AGN =
[2021-03-10T19:59:51.217539] [IMSAMPP_TCL1(6839) ] [D] [DEBUGP2D] [SECURITY]TYPE = TACF
[2021-03-10T19:59:51.217564] [IMSAMPP_TCL1(6839) ] [D] [DEBUGP2D] security type is TACF[2]
[2021-03-10T19:59:51.217591] [IMSAMPP_TCL1(6839) ] [D] [DEBUGP3D]
[GENERAL]SCHEDULE_RECOVER_MAXCNT = 5
[2021-03-10T19:59:51.227545] [IMSAMPP_TCL1(6839) ] [D] [DEBUGP3D] [MQ]USE_MQ = NO
[2021-03-10T19:59:51.227821] [IMSAMPP_TCL1(6839) ] [D] [DEBUGP1D] CTL's JOBID : JOB02316
[2021-03-10T19:59:51.227844] [IMSAMPP_TCL1(6839) ] [M] [OSI0291M] IMSA server boots -
resource manager initialization starts
[2021-03-10T19:59:51.229970] [IMSAMPP_TCL1(6839) ] [D] [DEBUGP2D] [CPM]REGION_CCSID =
[2021-03-10T19:59:51.229992] [IMSAMPP_TCL1(6839) ] [D] [DEBUGP2D] [CPM]ASCII_TO_EBCDIC =
ASCEBCUS.cpm
[2021-03-10T19:59:51.230114] [IMSAMPP_TCL1(6839) ] [D] [DEBUGP2D] [CPM]EBCDIC_TO_ASCII =
EBCASCUS.cpm
[2021-03-10T19:59:51.230214] [IMSAMPP_TCL1(6839) ] [D] [DEBUGP0D] cpm version
[2021-03-10T19:59:51.230228] [IMSAMPP_TCL1(6839) ] [D] [DEBUGP2D] [SCREEN]3270_TYPE = 3270-
A2
[2021-03-10T19:59:51.230239] [IMSAMPP_TCL1(6839) ] [D] [DEBUGP0D] SCREEN TYPE : A2
[2021-03-10T19:59:51.230255] [IMSAMPP_TCL1(6839) ] [D] [DEBUGP3D] [CPM]CONVERT_TO_SPACE =
X'00'
[2021-03-10T19:59:51.230270] [IMSAMPP_TCL1(6839) ] [D] [DEBUGP2D] X'00'(in ASCII) will be
converted X'20'
[2021-03-10T19:59:51.230280] [IMSAMPP_TCL1(6839) ] [D] [DEBUGP1D] null character=1a
[2021-03-10T19:59:51.230289] [IMSAMPP_TCL1(6839) ] [M] [OSI0101M] osimfs version: 7.2.0(4)
sbaek@tmax:of7_1_mvs_dev/osi(#1) 2020-11-12 19:43:54
[2021-03-10T19:59:51.231791] [IMSAMPP_TCL1(6839) ] [D] [DEBUGP2D] [GENERAL]SVCLLOG = N
[2021-03-10T19:59:51.231812] [IMSAMPP_TCL1(6839) ] [M] [OSI0291M] IMSA server boots -
resource manager initialization completed
```

## • ERR 로그

CLOPT 절에 [-e] 옵션과 함께 파일명을 지정해주면, OSI 시스템 서버 및 MPP 사용자 서버에서 stderr으로 쓰는

모든 내용이 해당 파일에 출력된다. 출력되는 로그는 OUT 로그와 동일하게 OpenFrame의 공통 출력 포맷을 따른다.

## 5.2.2. BMP 사용자 서버

BMP 사용자 서버는 JCL을 사용하여 Batch JOB 형태로 애플리케이션 프로그램을 수행하기 때문에 SPOOL 로그 디렉터리에 생성된다. BMP 사용자 서버는 MPP 사용자 서버와 달리 하나의 IMS 시스템에서 동일한 서버 모듈을 사용하므로, JOBID로 구분된 디렉터리를 통해 로그를 관리하고 있다. 마찬가지로 OpenFrame의 공통 출력 포맷을 따른다.



JOB 관리 및 운영에 대한 내용은 OpenFrame Batch "Batch 안내서"를 참고한다.

# Appendix A: IMSBATCH 프러시저

본 부록에서는 IMS/DC에서 BMP를 사용하기 위해 제공하는 IMSBATCH 프러시저를 OSI에서 사용하기 위한 방법을 기술한다.

IMS/DC에서 BMP는 DFSRRC00라는 유틸리티를 이용하여 기동되며, 보통 IMSBATCH 프러시저를 이용하게 된다. OSI에서는 IMSBATCH 프러시저를 제공하지는 않지만 IMS/DC와 같이 DFSRRC00라는 HiDB 패키지에서 제공하는 유틸리티를 이용하여 BMP를 기동한다.

다음은 IMSBATCH 프러시저에 대한 예제이다. 예제에서 굵은 글씨체로 표시된 부분이 OSI에서 지원하는 파라미터 항목이다.

```
//      PROC  MBR=TEMPNAME,PSB=,IN=,OUT=,
//          OPT=N,SPIE=0,TEST=0,DIRCA=000,
//          PRLD=,STIMER=,CKPTID=,PARDLI=,
//          CPUTIME=,NBA=,OBA=,IMSID=,AGN=,
//          SSM=,PREINIT=,RGN=56K,SOUT=A,
//          SYS2=,ALTID=,APARM=,LOCKMAX=
// *
//G      EXEC  PGM=DFSRRC00,REGION=&RGN,
//          PARM=(BMP,&MBR,&PSB,&IN,&OUT,
//          &OPT&SPIE&TEST&DIRCA,&PRLD,
//          &STIMER,&CKPTID,&PARDLI,&CPUTIME,
//          &NBA,&OBA,&IMSID,&AGN,&SSM,
//          &PREINIT,&ALTID,
//          '&APARM',&LOCKMAX)
//STEPLIB DD DSN=IMS.&SYS2.SDFSRESL,DISP=SHR
//          DD DSN=IMS.&SYS2.PGMLIB,DISP=SHR
//PROCLIB DD DSN=IMS.&SYS2.PROCLIB,DISP=SHR
//SYSUDUMP DD SYSOUT=&SOUT,
//          DCB=(LRECL=121,RECFM=VBA,BLKSIZE=3129),
//          SPACE=(125,(2500,100),RLSE,,ROUND)
```

OSI에서 BMP를 기동하기 위해 DFSRRC00에 제공되는 PARM 중 다음과 같은 파라미터를 지원한다.

파라미터	설명
MBR	애플리케이션 프로그램의 이름을 지정한다.
PSB	PSB 이름이 애플리케이션 프로그램 이름과 다른 경우에 PSB 이름을 지정하는 옵션 파라미터이다.
IN	입력 트랜잭션 코드를 지정한다. 파라미터를 생략한 경우에는 batch-oriented BMP로 처리된다.
OUT	출력 트랜잭션 코드를 지정한다. IN 파라미터가 기술되어 있으면 무시된다.
IMSID	운영체제에서 사용되는 IMS 시스템의 식별자이다.

# Appendix B: Printer 기능

본 부록에서는 OpenFrame OSI 시스템에서 지원하는 Printer 기능에 대해 설명한다.

## B.1. 개요

OpenFrame OSI 시스템에서는 프린터 사용을 위해 두 가지 방식을 제공한다.

- Display HardCopy
- SCS-DATA Printer



Printer 기능 사용을 위해 OpenFrame GW에 접속이 필요하다. 해당 제품에 대한 자세한 내용은 OpenFrame GW "웹 터미널 안내서"를 참고한다.

## B.2. Display HardCopy

OpenFrame GW에 접속 중인 터미널이 Display 모드(에뮬레이터의 Device Type을 IBM-3278-2-E로 설정했을 경우)이고 OSI 시스템 정의에 이 터미널의 LU가 프린터로 정의되어 있으면, 이 터미널로 보내는 모든 화면은 해당 PC의 기본 프린터로 출력된다.

1. 터미널을 Display 모드(에뮬레이터마다 설정할 수 있는 옵션이 있다)로 설정하고 OpenFrame GW에 접속한다.
2. 로그온할 시스템 서버(Control Region)에 이 터미널의 LU(Netname)가 프린터로 설정되어 있는지 확인한다.
3. 해당 터미널로 다른 터미널이나 프로그램에서 메시지를 전달하여 HardCopy가 되는지 확인한다.



OSI 시스템은 불필요한 프린팅을 방지하기 위해서 프린터 터미널로 사용자가 데이터 외 다른 데이터를 보내지 않기 때문에 프린터로 설정된 터미널은 기본적으로 X SYSTEM인 상태로 되어 있다.

## B.3. SCS-DATA Printer

OpenFrame GW에 접속 중인 터미널이 Printer 모드(에뮬레이터의 Device Type을 IBM-3287-1 로 설정했을 경우를 칭함)이고 OSI 시스템 정의에 이 터미널의 LU가 프린터로 정의되어 있으면, 이 터미널로 보내는 모든 데이터는 SCS-DATA 형식이어야 한다. 반대로 사용자 애플리케이션 프로그램에서 SCS-DATA를 보내는 해당 터미널이 SCS-DATA 프린터 모드가 아니면 정상적으로 동작하지 않는다.

1. 터미널을 Printer 모드(에뮬레이터마다 설정할 수 있는 옵션이 있음)로 설정하고 OpenFrame GW에 접속한다.
2. 로그온할 시스템 서버(Control Region)에 이 터미널의 LU(Netname)가 프린터로 설정되어 있는지 확인한다.
3. 해당 터미널은 Printer 모드이기 때문에 입력이 불가하므로 IP-LU mapping의 자동 로그온 기능을 사용하거나, OSI 명령어 "/OPNDST"를 사용하여 외부에서 로그온해주어야 한다.

4. 해당 터미널로 SCS-DATA를 보내서 출력이 잘 되는지 확인한다.

# Appendix C: 리소스 테이블

본 부록에서는 OpenFrame OSI 시스템에서 관리하는 RDB 테이블에 대한 설명을 기술한다.

OSI에서 사용하는 리소스들을 RDB 테이블에서 관리한다. 테이블들은 **osiinit** 툴로 생성하고 삭제한다. 자세한 내용은 OpenFrame OSI "툴 참조 안내서"를 참고한다.

## • SD 및 RTSD 정보

리소스별 SD 및 RTSD를 관리하는 테이블들은 아래와 같다.

테이블명	설명
OFM_OSI_SD_APPLCTN	프로그램 SD 리소스를 저장한다.
OFM_OSI_SD_DATABASE	데이터베이스 SD 리소스를 저장한다.
OFM_OSI_SD_LTERM	논리 터미널 SD 리소스를 저장한다.
OFM_OSI_SD_TERMINAL	터미널 SD 리소스를 저장한다.
OFM_OSI_SD_TRANSACT	트랜잭션 SD 리소스를 저장한다.
OFM_OSI_RTSD_APPLCTN	프로그램 RTSD 리소스를 저장한다.
OFM_OSI_RTSD_DATABASE	데이터베이스 RTSD 리소스를 저장한다.
OFM_OSI_RTSD_LTERM	논리 터미널 RTSD 리소스를 저장한다.
OFM_OSI_RTSD_TERMINAL	터미널 RTSD 리소스를 저장한다.
OFM_OSI_RTSD_TRANSACT	트랜잭션 RTSD 리소스를 저장한다.
OFM_OSI_RTSD_MODS	MODIFY 커맨드에 의한 변경 대상 RTSD 리소스를 저장한다.

## • CI 정보

CI 리소스(터미널 세션 정보)를 관리하는 테이블은 아래와 같다.

테이블명	설명
OFM_OSI_CI	터미널 세션 정보를 저장한다.

## • MODSTAT 정보

MODSTAT 정보를 관리하는 테이블은 아래와 같다.

테이블명	설명
OFM_OSI_MODSTAT	MODSTAT 정보를 저장한다.

## • Message Queue 정보

MQ 정보를 관리하는 테이블들은 아래와 같다.

테이블명	설명
OFM_OSI_MQ	모든 메시지를 저장 및 관리하는 테이블이다.

#### • Region 정보

Region 정보를 관리하는 테이블은 아래와 같다.

테이블명	설명
OFM_OSI_REGION	DR REGION의 JOB 정보를 저장한다.
OFM_OSI_REGION_S VR	DR REGION의 수행 중인 트랜잭션, 프로그램 정보들을 저장한다.
OFM_OSI_REGION_S TATUS	CTL REGION의 JOBID, STEPSEQ, STATUS, CHKPT 정보들을 저장한다.

#### • Event 정보

Event 정보를 관리하는 테이블은 아래와 같다.

테이블명	설명
OFM_OSI_EVENT	OSI 시스템에서 발생하는 이벤트 정보를 저장한다.

#### • LOG 정보

LOG 정보를 관리하는 테이블은 아래와 같다.

테이블명	설명
OFM_OSI_LOG	DL/I LOG Call을 통해 기록한 로그를 저장한다.