|  |
| --- |
|  |
| **인젠트 ECM NewFace Toy Project**  **운영자 가이드 및 매뉴얼 이미지시스템 구축**  **RL-01** |
| |  |  | | --- | --- | | **업 무 명** | 이미지 구축 | | **작 성 일** | 20xx.00.00 | | **버 전** | 1.0 | |

C:\Users\TY\AppData\Local\Microsoft\Windows\INetCache\Content.Word\inzent_logo_W.PNG

**Copyright Notice**

**Copyright ⓒ 2023 INZENT Co., Ltd. All Rights Reserved.**

**대한민국 서울 영등포구 국제금융로2길 유화증권빌딩 8층**

**Restricted Rights Legend**

**INZENT software and documents are made available under the terms of the INZENT License Agreement and may only be used or copied in accordance with the terms of this agreement. No part of this document may be transmitted, copied, deployed, or reproduced in any form or by any means, electronic, mechanical, or optical, without the prior written consent of INZENT Co., Ltd.**

**사용설명서의 내용과 여기에 설명된 프로그램은 INZENT Co., Ltd.와 사용권 계약 하에서만 사용이 가능하며, 사용권 계약을 준수하는 경우에만 사용 또는 복제할 수 있습니다. 이 사용설명서의 전부 또는 일부분을 INZENT의 사전 서면 동의 없이 전자, 기계, 녹음 등의 수단을 사용하여 전송, 복제, 배포, 2차적 저작물 작성 등의 행위를 하여서는 안 됩니다.**

**안내서 정보**

**발행일: 2023-01-28**

**소프트웨어 버전: XTORM6.0**

안내서 버전: XTORM 운영자 가이드 및 매뉴얼 1.0.0

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **문서 개정 이력표(Project Based)** | | | | |
| **Date** | **Version** | **Description** | **Writer** | **Etc.** |
| 2022. 01. 21 | 0.1 | XTORM 운영자 매뉴얼 초안 작성 | ㈜인젠트 OOO | 최초 작성 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

목 차

[1. XTORM (ECM 엔진) 소개 6](#_Toc120805711)

[1.1 XTORM 개요 6](#_Toc120805712)

[1.2 주요 서비스 기능 7](#_Toc120805713)

[1.3 다양한 컴포넌트 제공 8](#_Toc120805714)

[2. XTORM (ECM 엔진) 주요 구성 9](#_Toc120805715)

[2.1 XTORM 서비스 구성 9](#_Toc120805716)

[2.2 XTORM 엔진 설치 디렉토리 구성 10](#_Toc120805717)

[2.3 XTORM 엔진 서비스 제어 10](#_Toc120805718)

[2.4 XTORM 엔진 실행 컨트롤러 12](#_Toc120805719)

[2.5 XTORM 엔진 서비스 구성 파일 15](#_Toc120805720)

[2.6 XTORM 엔진 파일 관리(내부) 17](#_Toc120805721)

[2.7 XTORM 엔진 인덱스 테이블 18](#_Toc120805722)

[2.8 XTORM 엔진 내부 테이블 19](#_Toc120805723)

[2.9 XTORM 엔진 저장소 구성 개념도 20](#_Toc120805724)

[2.10 XTORM 엔진 메인 테이블 21](#_Toc120805725)

[2.11 XTORM Migration(이관) 관리 정보 23](#_Toc120805726)

[2.12 XTORM Scheduler 관리 정보 25](#_Toc120805727)

[3. XTORM (ECM 엔진) 모니터링 26](#_Toc120805728)

[3.1 XTORM 엔진 서비스 로그 모니터링 26](#_Toc120805729)

[3.2 XTORM 엔진 API 문제 해결 27](#_Toc120805730)

[4. XTORM (ECM 엔진) 관리 정보 29](#_Toc120805731)

[4.1 프로젝트 사이트 정보 29](#_Toc120805732)

[4.2 서버 정보 29](#_Toc120805733)

[4.3 아카이브 · 볼륨 구성 정보 32](#_Toc120805734)

[4.4 엔진 정보 33](#_Toc120805735)

[4.5 WEB / WAS 정보 36](#_Toc120805736)

[4.6 클라이언트 정보 37](#_Toc120805737)

[4.7 암호화 정보 37](#_Toc120805738)

[4.8 배치 정보 38](#_Toc120805739)

[5. 추가 정보 39](#_Toc120805740)

[5.1 기술 지원 이력표 39](#_Toc120805741)

[5.2 참조 자료 경로 39](#_Toc120805742)

**매뉴얼의 대상**

사용자가 XTORM을 사용하기 위한 기본 개념 및 엔진 기동 절차에 대한 설명

**매뉴얼의 전제 조건**

본 안내서는 XTORM 사용을 위한 기본 매뉴얼이다. 따라서 본 매뉴얼을 원활히 이해하기

위해서는 다음과 같은 사항을 미리 알고 있어야 한다.

**● 운영체제 및 시스템 환경의 이해**

**● XTORM 환경의 이해 및 특징**

**● UNIX 계열(Linux 포함)의 기본 지식**

**매뉴얼 작성 규약 (※매뉴얼 작성 후 본 내용 삭제)**

해당 매뉴얼은 사이트 내 구성에 맞게 수정이 필요하다.

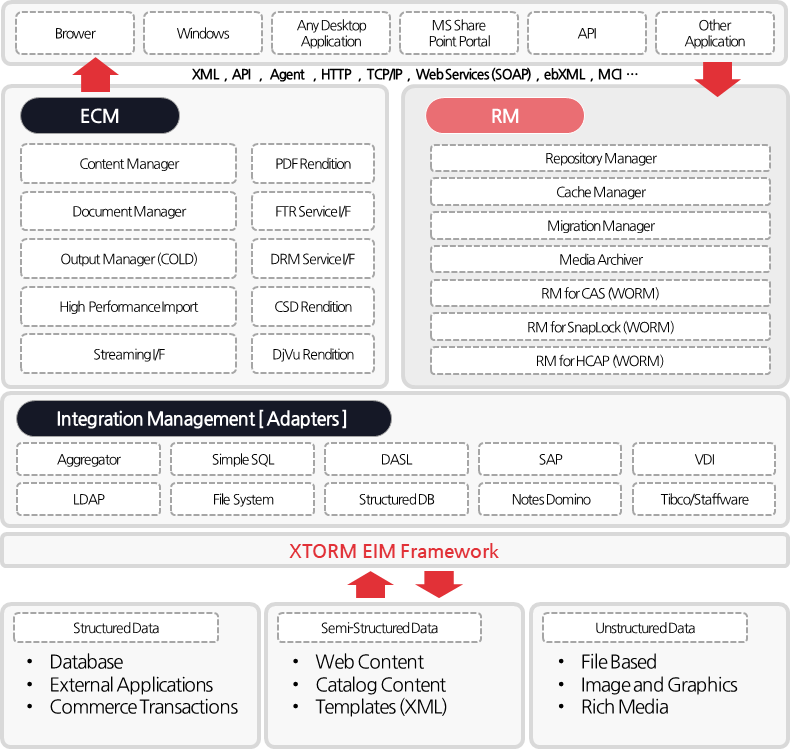
# XTORM (ECM 엔진) 소개

본 장에서는 XTORM (ECM 엔진)에 대한 개요 및 서비스 기능에 대해 설명한다.

## XTORM 개요

XTORM ECM (Enterprise Content Management)은 XTORM EIM 사상을 지향하는 제품 중 하나로, 다양한 기업 환경에서 다양한 콘텐츠 및 제반 서비스를 관리하는 강력한 파일 관리 솔루션이다. 문서 이미지화, 문서 관리, 계층 저장 관리, 전사적 리포트 관리, WebDAV등을 포함한 풍부하고 강력한 저장소(Repository) 서비스를 제공한다.

[그림1-1] XTORM 아키텍처



## 주요 서비스 기능

XTORM ECM은 제품에 있는 저장소(Repository) 서비스로 실행된다. 대용량 볼륨 및 대용량 트랜잭션의 컨텐츠 서비스를 쉽게 구축할 수 있으며 웹 기반의 관리와 운영지원으로 유지관리를 효율적으로 할 수 있다.

[표1-1] 주요 서비스

|  |  |
| --- | --- |
| **구 분** | **세부사항** |
| 데이터베이스 접속 서비스 | * 데이터베이스 Load-Balancing 지원 * 대용량 데이터베이스 분산서비스 * 다양한 DBMS 완벽 지원 * 연결 풀(Pool)을 통한 DB 트랜잭션 향상 |
| 컨텐츠 색인/관리 서비스 | * 고객 업무에 맞는 컨텐츠 관리체계 디자인 * 생성, 조회, 보존, 이관, 폐기 등의 관리 * 컨텐츠 별 유형, 속성 관리 및 Audit Trail 기능 * 계층적 데이터 구조 지원 |
| 보안관리 및 모니터링 서비스 | * 사용자, 그룹 권한 및 글로벌 보안관리 * 단위 컨텐츠까지의 보안체계 지원 * 100% 웹 기반의 원격관리 * 다수의 분산 서버에 대한 모니터링 지원 |
| 스토리지 관리 서비스 | * 계층적 저장장치 관리서비스 * 최신의 SAN, NAS, CAS 환경 지원 * 자료 보존 주기 관리와 자동 Migration * 대용량의 분산스토리지 지원 |
| 트랜잭션 관리 서비스 | * 전용 통신 프로토콜 채용으로 속도 향상 * 자체 네트워크 리소스 제어 기능 지원 * 서버to서버, 서버to클라이언트간 소켓 통신 * 모든 트랜잭션은 XML 및 Thread 방식 처리 |
| 고속 입출력 서비스 | * 대용량 데이터의 고속 배치 입출력시스템 * 스케줄러에 의한 자동 입출력 관리 * 배치 크기 및 Thread 지정 * 입출력에 대한 통계정보 지원 |

## 다양한 컴포넌트 제공

XTORM 엔진에서는 다양한 형태의 컴포넌트를 제공한다. 각 컴포넌트는 ECM에 필요한 기능 등을 포함하고 있으며, 각 필요한 업무에 사용 될 수 있도록 모듈화 되어있다.

[표1-2] 컴포넌트

|  |  |
| --- | --- |
| **컴포넌트** | **설명** |
| Object Management  (Document Imaging) | * 크기, 포맷, 원본 등과 같은 비정형 데이터를 관리하는 컴포넌트 * 강력한 HSM으로 저장된 정보의 Information Lifecycle Management을 지원 * 유연한 색인 기능과 강력한 폴더 링을 제공 |
| Document Management | * 문서 관리를 담당하는 서버 사이드 라이브러리(Library) * 서비스 컴포넌트, Check-in, check-out, version control, revision history, 무제한의 version, revision |
| Content Storage Management | * 가상의 글로벌 컨텐츠(파일) 레파지토리를 담당하는 컴포넌트 * 장치 독립적인 룰 기반의 Migration * 무제한의 분산 스토리지 지원 및 오브젝트의 Intelligent, Proximity, Throttling 기능 |
| Enterprise  Report Management | * 자동적인 저장, 색인 지정, 구조, 저장, 검색을 위한 정형화된 컴퓨터  출력물 관리를 위한 컴포넌트 * 대규모의 분산된 보고서를 구체적인 정보로 변형 가능 * ASCII, EBCDIC, PLC, Meta Code/DJDE, AFP, 그리고 PDF 형식의 프린트 포맷을 위한 향상된 지원 |
| High Performance Import | * 고속 성능을 최적화한 컨텐츠의 import, 저장, 인덱싱을 담당하는 컴포넌트 |
| WebDAV | * 웹 기반의 분산된 Authoring 및 Versioning 표준을 제공하는 컴포넌트 * XVARM의 가상 계층을 제공하며, 문서의 읽기, 쓰기, 업데이트 지원 * Windows Explorer, MS Office, SharePoint 등과 같은 Compliance -  Application 지원 |

# XTORM (ECM 엔진) 주요 구성

본 장에서는 XTORM (ECM 엔진)에 대한 설치 고려사항 및 엔진 주요 구성에 대해 설명한다.

## XTORM 서비스 구성

- 이미지 엔진 구축 시 고려사항

XTORM 서비스를 구성하기 위해서는 다음과 같은 사항을 고려하여 구성해야 한다.

[표2-1] 엔진 설치 고려사항

|  |  |
| --- | --- |
| **항목** | **설명** |
| 서버 | * 동시 사용량 및 저장 용량 등을 고려하여 서버의 최소 사양 등을 고려해야 하며, XTORM 서비스를 구동하기 위한 JVM(Java Virtual Machine)이 기본적으로 탑재 되어야 한다. |
| Database | * XTORM은 서비스가 구동되기 위한 내부정보, 파일을 저장하기 위한 스토리지의 논리 및 물리 볼륨 정보 및 파일의 복제/이관 등을 관리하기 위한 Lifecycle 정보, 그리고 파일의 인덱스 및 위치정보 등을 데이터베이스에 저장한다. * XTORM Data Service를 다수로 운영하고자 한다면 데이터베이스 스키마도 그 개수에 맞추어 생성되어야 한다. |
| 엔지니어 관리 페이지 | * XTORM를 관리 및 모니터링 하기 위해서는 엔지니어 관리 페이지가 필요하다. 관리자만 사용하기 때문에 고성능의 엔진은 필요치 않으며 대부분 XTORM이 설치된 서버(혹은 다른 서버)에 tomcat을 탑재하여 사용한다. |
| 스토리지 | * 파일을 저장하기 위하여 필요한 물리적인 볼륨이며, 추후 확장성을 고려하여 마운트 해야 한다. 개별적인 물리적인 볼륨 크기는 스토리지 업체에서 제안하는 크기 혹은 백업 정책에 따라 정한다. * XTORM은 물리적인 볼륨이 큰 경우에 내부적으로 논리적으로 분할하여 사용하여 볼륨을 유연성 및 연속성을 보장한다. |
| Lifecycle  (보존 주기) | * 파일이 최초로 저장된 이후에 특정한 조건에 의하여 다른 스토리지로 복제/이관 작업 등이 필요한 경우에 이를 미리 정의하여 파일 입력 시 파일의 속성으로 지정해야 한다. |
| ElementId | * XTORM은 내부적으로 입력되는 파일에 ElementId라는 16자리의 키를 부여하여 관리한다. 이는 단일 XTORM Data Service 내에서 파일 단위로 Unique 한 값이며 파일을 다운로드, 수정, 삭제 등의 작업을 수행할 때 사용되는 키이다. |

## XTORM 엔진 설치 디렉토리 구성

[표2-2] 엔진 설치 디렉토리 구성

|  |  |  |
| --- | --- | --- |
| **서버** | **설치위치** | **설명** |
| **이미지 AP #1**  **이미지 AP #2** | **/sw/xtorm/** | XTORM홈디렉토리 |
| **/sw/xtorm/bin/** | XTORM 서비스 control shell |
| **/sw/xtorm/lib/** | XTORM library |
| **/sw/xtorm/msg/** | 메시지 파일 |
| **/sw/xtorm/servers/** | XTORM 서비스 루트 디렉토리 |
| **/log/xtorm/** | XTORM 로그 |

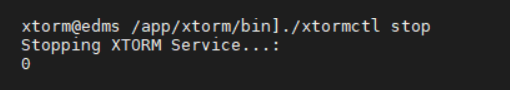
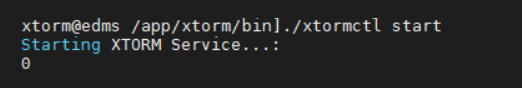
## XTORM 엔진 서비스 제어

- XTORM 서비스 시작 및 종료

모든 XTORM 서비스들은 아래의 명령에 의해 시작 및 종료된다.

[표2-3] 엔진 서비스 제어

|  |  |  |  |
| --- | --- | --- | --- |
| **서버** | **구분** | **명령** | **비고** |
| **이미지 AP #1**  **이미지 AP #2** | **시작** | **/sw/xtorm/bin/xtormctl\_2102 start** | 사이트 특성에 따라 실행 쉘 경로 상이 할 수 있음 |
| **상태 체크** | **/sw/xtorm/bin/xtormctl\_2102 status** |
| **종료** | **/sw/xtorm/bin/xtormctl\_2102 stop** |
| **프로세스 확인** | **ps –ef | grep xtorm | grep –v grep** |

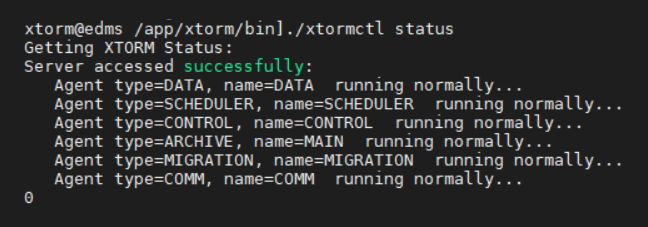
****[그림2-1] 엔진 기동 및 중지 화면 – **예시**

* **XTORM 엔진 서비스 기동 및 중지 화면**
* **고객사 특성에 따라 상이 할 수 있음.**

**- XTORM엔진 서비스 기동 및 중지**

start 혹은 stop 명령어로 기동한다. 실행 및 중지 후 숫자 “0”이 Return 될 경우 정상작동 한 것이다.

명령어는 “[표2-3] 엔진 서비스 제어” 참고

****[그림2-2] 엔진 Status 화면 – **예시**

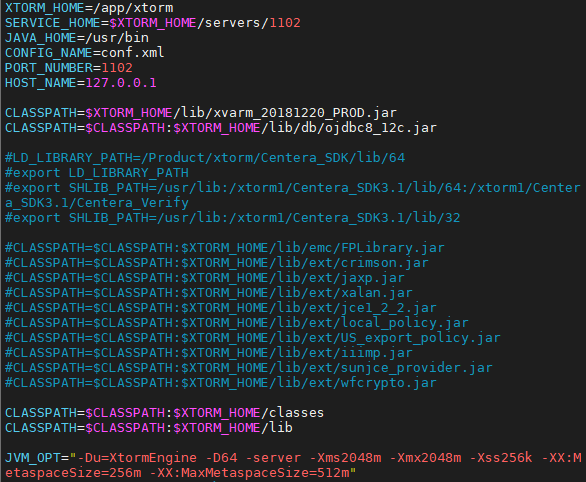
* **XTORM 엔진 서비스 상태 체크 화면**
* **고객사 특성에 따라 상이 할 수 있음.**

**- XTORM엔진 서비스 상태 체크**

status 명령어로 Data 서비스에 상태 체크한다. 위의 캡쳐 화면과 같이 모든 Agent 항목이 running normally……와 같이 화면에 출력이 되어야 정상적으로 XTORM 엔진 에이전트가 가동된 것이다.

명령어는 “[표2-3] 엔진 서비스 제어” 참고

## XTORM 엔진 실행 컨트롤러

****[그림2-3] 엔진 실행 컨트롤러 화면 – **예시**

* **엔진 서비스 xtormctl\_2102 파일 설정(JAVA HOME, 엔진 경로, CLASSPATH, JVM OPTION…)**
* **고객사 특성에 따라 상이 할 수 있음.**

**[설명]**

JAVA\_HOME과 XTORM 서비스 홈을 지정.

XTORM에 필요한 클래스 파일과 라이브러리 파일의 경로를 지정.

Du -2102: 사용하는 포트를 지정

d64: XTORM 비트 수를 지정 (※해당 옵션은 JDK 11 이상에서 제거)

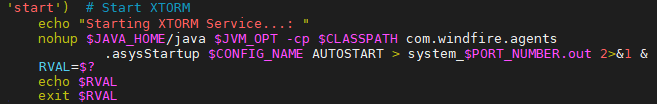
Xms1024m: 최소 Heap 사이즈를 지정, Xmx2048m: 최대 Heap 사이즈를 지정

Xss300k: 최대 Stack 사이즈를 지정

XX:MetaspaceSize: Metaspace Default Size 설정 (※해당 옵션은 JDK 1.8 이상에서 필수)

XX:MaxMetaspaceSize: Metaspace Max Size 설정 (※해당 옵션은 JDK 1.8 이상에서 필수)

XX: + HeapDumpOnOutOfMemoryError: OutOfMemory 에러 발생 시 덤프를 뜸.

****[그림2-4] 엔진 시작 스크립트 화면 – **예시**

* **고객사 특성에 따라 상이 할 수 있음.**

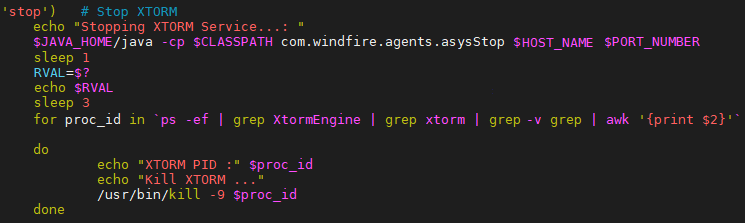
**[설명]**

Xtorm 엔진 수행하기 위한 스크립트이다. 명령어 sh xtormctl\_2102 start를 입력하면 구동된다.

엔진 구동 시 자동적으로 systemout 로그가 생성된다.

자세한 명령어는 “[표2-3] 엔진 서비스 제어” 참고

[그림2-5] 엔진 중지 스크립트 화면 – 예시

* **고객사 특성에 따라 상이 할 수 있음.**

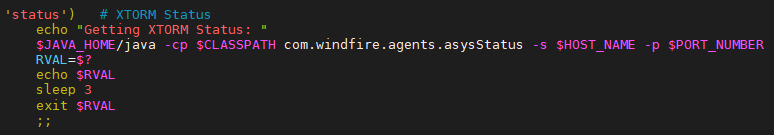
**[설명]**

Xtorm 엔진 정지를 위한 스크립트이다. 명령어sh xtormctl\_2102 stop를 입력하면 구동된다.

엔진이 정상적으로 중지 되지 않을 시, 프로세스 ID를 검색하여 Kill 명령어를 활용하여 중지시킨다.

자세한 명령어는 “[표2-3] 엔진 서비스 제어” 참고

[그림2-6] 엔진 상태 스크립트 화면

****

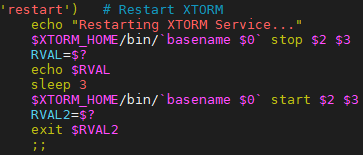
* **고객사 특성에 따라 상이 할 수 있음.**

**[설명]**

Xtorm 엔진 상태 확인을 위한 스크립트이다. 명령어sh xtormctl\_2102 status를 입력하면 구동된다.

명령어는 “[표2-3] 엔진 서비스 제어” 참고

[그림2-7] 엔진 재 시작 스크립트 화면

****

* **고객사 특성에 따라 상이 할 수 있음.**

**[설명]**

Xtorm 엔진 재 시작을 위한 스크립트이다. 명령어sh xtormctl\_2102 restart를 입력하면 구동된다.

## XTORM 엔진 서비스 구성 파일

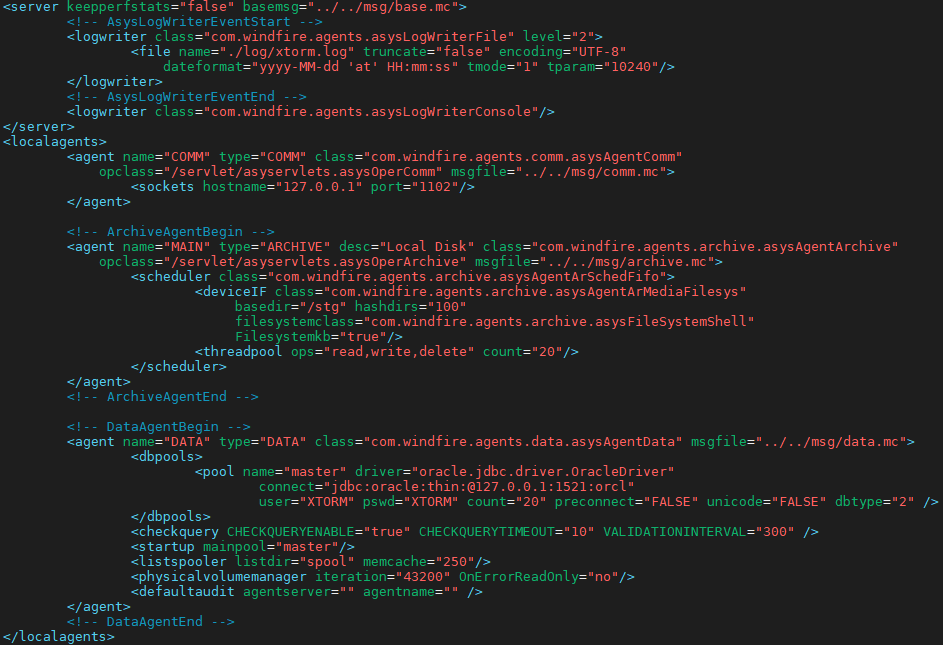
- XTORM Data 엔진 서비스 구성 파일

XTORM Data 엔진 환경 설정과 변경 및 추가 시 사용하는 엔진 환경 구성에 사용되는xml파일입니다.

로그 설정, 포트 설정, DB 설정, 아카이브 설정을 하는 XTORM환경 파일입니다.

명령어는 “[표2-2] 엔진 설치 디렉토리 구성” 참고

[그림2-8] 엔진 서비스 파일 구성 화면

****

**로그 설정 부분**

XTORM로그는 엔진 자체 로그 방식을 사용하며. 로그 레벨을 구분할 수 있도록 지원하며 전체 레벨 수준은 다음과 같이 6 가지로 구분한다.

운영중인 XTORM의 로그 레벨을 높게 잡을 경우 과도한 로그가 쌓일 수 있으므로 “2” 이하를 권장한다.

**[설명<logWriter>]**

“0” : ERROR - 에러내용만 출력

“1” : Normal Warning - 일반적인 경고 내용까지 포함하여 출력

“2“ : Detail Warning - 자세한 경고 내용까지 포함하여 출력

“3” : Normal Information - 일반적인 정보 내용까지 포함하여 출력

“4” : Detail Information - 자세한 정보 내용까지 포함하여 출력

“5” : DEBUG - 디버깅 용도로 모든 내용을 포함하여 출력

**[설명<LocalAgents>]**

hostname: 호스트 명이나 IP로 소켓을 Open 할 수 있도록 지원한다.

port: Transmit 에이전트가 에이전트 상호간 또는 어플리케이션 간에 통신하는 통신 포트 번호

clienttimeout: 일정 시간이 경과하도록 클라이언트 연결이 아무런 작업을 하지 않는 경우, 자동으로 이 클라이언트의 연결을 닫도록 하는 Parameter.

**[추가 가능한 옵션] – 필요한 경우 만 사용한다.**

virtualhost: 대표 IP 접속을 허용하기 위해 대표 호스트 명을 지정한다.

virtualip: 대표 IP 접속을 허용하기 위해 대표 IP 주소를 지정한다.

virtualport: 대표 IP 접속을 허용하기 위해 대표 서비스 포트 번호를 지정한다.

**[설명<Archiveagent>]**

scheduler.class: 장치를 읽고 쓰는 중에 사용될 알고리즘을 포함하고 있는 자바 클래스 이름.

deviceIF.class: 파일 시스템에 파일을 쓰거나 읽을 경우에 사용할 자바 클래스 이름.

filesystemclass: 물리적 볼륨 관리를 위해 볼륨 정보를 취득하기 위한 클래스의 종류를 지정합니다.

basedir: 파일이 저장될 물리적 위치를 지정합니다.

dateformat: basedir로 지정한 디렉토리 아래 날짜형식디렉토리가 생성됩니다.

hashdirs:  웹 서버로 연결된 저장소 상의 사용할 Hash Directory 수를 나타낸다.

**[설명<DataAgent>]**

dbpool: 데이터베이스 연결 풀과 관련된 정보 정의.

name: pool의 논리적 이름, driver: JDBC 드라이버 클래스 이름, connect: 데이터베이스 연결

preconnect: XTORM이 시작될 때 최대 수 만큼 미리 확보할 것인지를 결정한다.

기본값: false, true – 최대 수만큼 확보, false – 필요할 경우 최대 연결

unicode: 데이터베이스 테이블들이 유니코드 컬럼을 포함하고 있는 경우에 사용될 수 있다.

dbtype: 데이터베이스 종류를 정의

mainpool: XTORM 서버 시작 시 마스터 데이터베이스에 접속할 풀 이름 정의

physicalvolumemanager.iteration: 볼륨 정보 관리 Thread의 구동 주기를 지정.

onErrorReadOnly: 물리 볼륨 정보 취득 실패 시 해당 볼륨을 '읽기전용' 변경

Yes: 정보 취득 실패 시 '읽기전용' 변경

No: 정보 취득 실패 후 에러 로그 반환 뒤, 해당 볼륨 재사용 시도

timeout: 검색한 결과 리스트를 얼마 동안 XTORM 서버에 남겨둘지를 결정하는 값

farm.enabled: High-Availability 기능의 활성화 여부를 지정 한다.

ucode: 서버 파밍(farming)을 구성하는 해당 노드의 유일키(Unique Key) 값을 지정 한다.

dserver ucode: 서버 파밍(farming)을 구성하는 다른 노드의 구성 정보를 정의 한다.

address: 서버 파밍(farming)을 구성하는 다른 노드의 IP 정보를 지정 한다.

port: 서버 파밍(farming)을 구성하는 다른 노드의 서비스 포트 정보를 지정 한다.

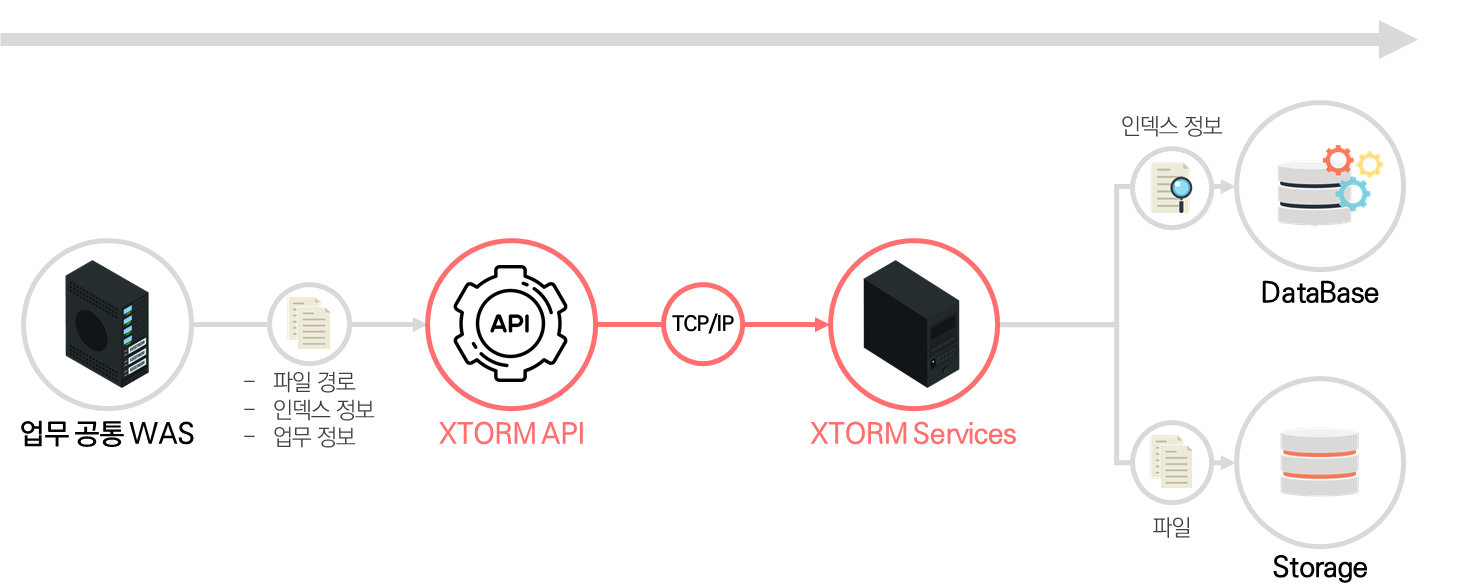
**{주의사항}**

**1) 엔진 환경구성파일 변경 시 에는 XTORM 전문 엔지니어 혹은 유지보수 인원과 협의 후 수행 필요.**

**2) 해당 파일 수정 시 엔진 재기동 이후 결과를 반영 할 수 있다.**

## XTORM 엔진 파일 관리(내부)

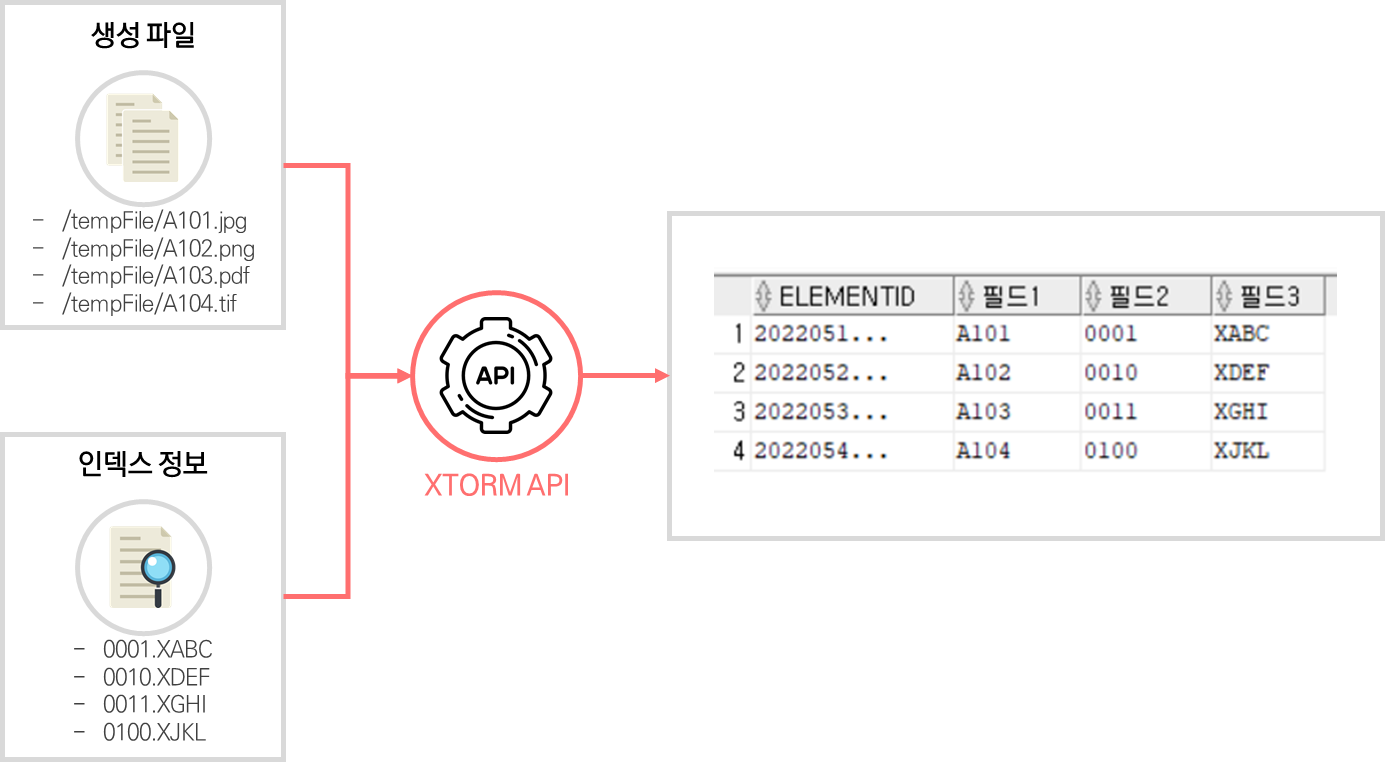
[그림2-9] 인덱스 등록 과정



## XTORM 엔진 인덱스 테이블

XTORM API로 파일을 입력할 때 XTORM은 개별 파일 단위로 “ElementId”라는 키를 유일하게 생성하고 추가적인 필드 정보를 인덱스 테이블에 입력하여 관리한다.

[그림2-10] 인덱스 테이블



**[설명]**

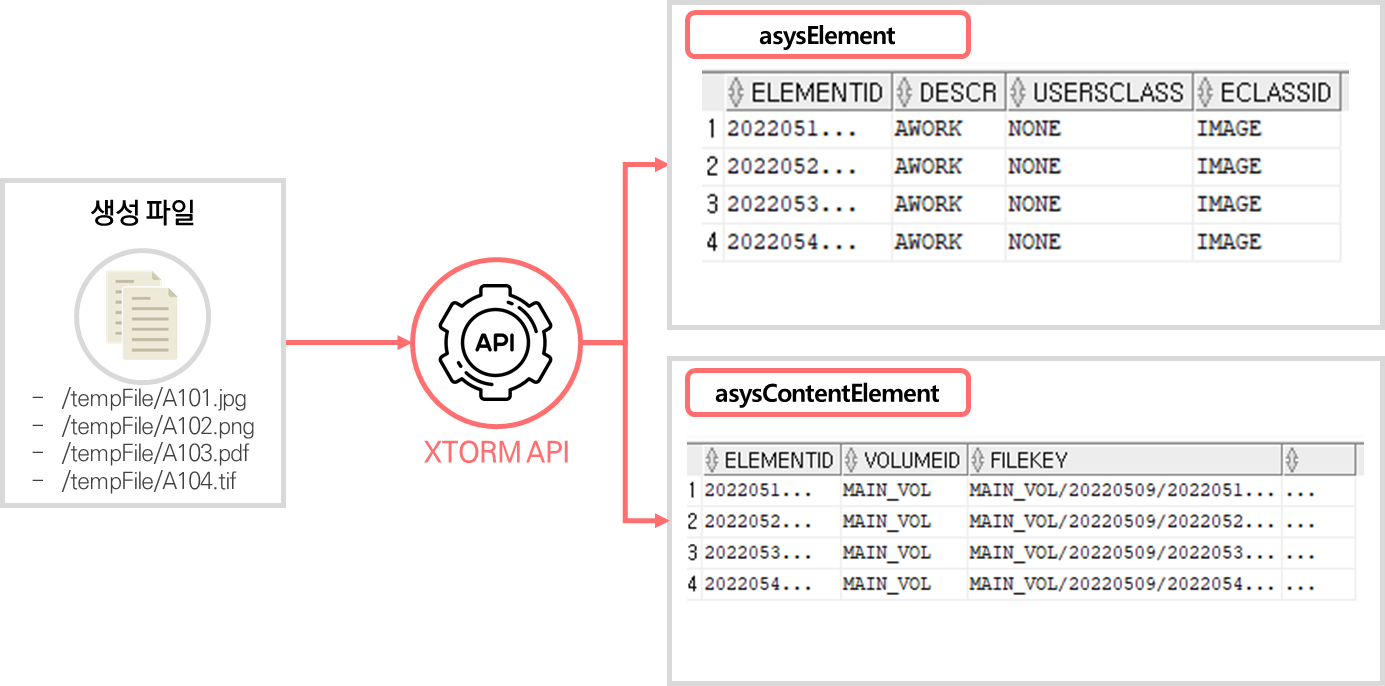
인덱스 테이블의 필드 값들은 1개 이상의 묶음으로 unique 조건이 되어야 한다. 이는 해당 조건으로 인덱스 검색을 수행하였을 때, 조회하려는 파일이 한 개 이상으로 리턴 될 수 있기 때문(중복 조회 가능)이다.

* ElementId는 년/월/일/시/분/초/밀리 초로 구성되며 밀리 초에서는 유일성을 보장하기 위해서 A~Z까지 영문자로 구성된다.

## XTORM 엔진 내부 테이블

XTORM으로 파일을 입력한 경우 파일의 인덱스 정보 및 위치정보, Migration 정보 등을 저장하기 위해서 XTORM은 2개의 내부 엔진 테이블을 사용한다.

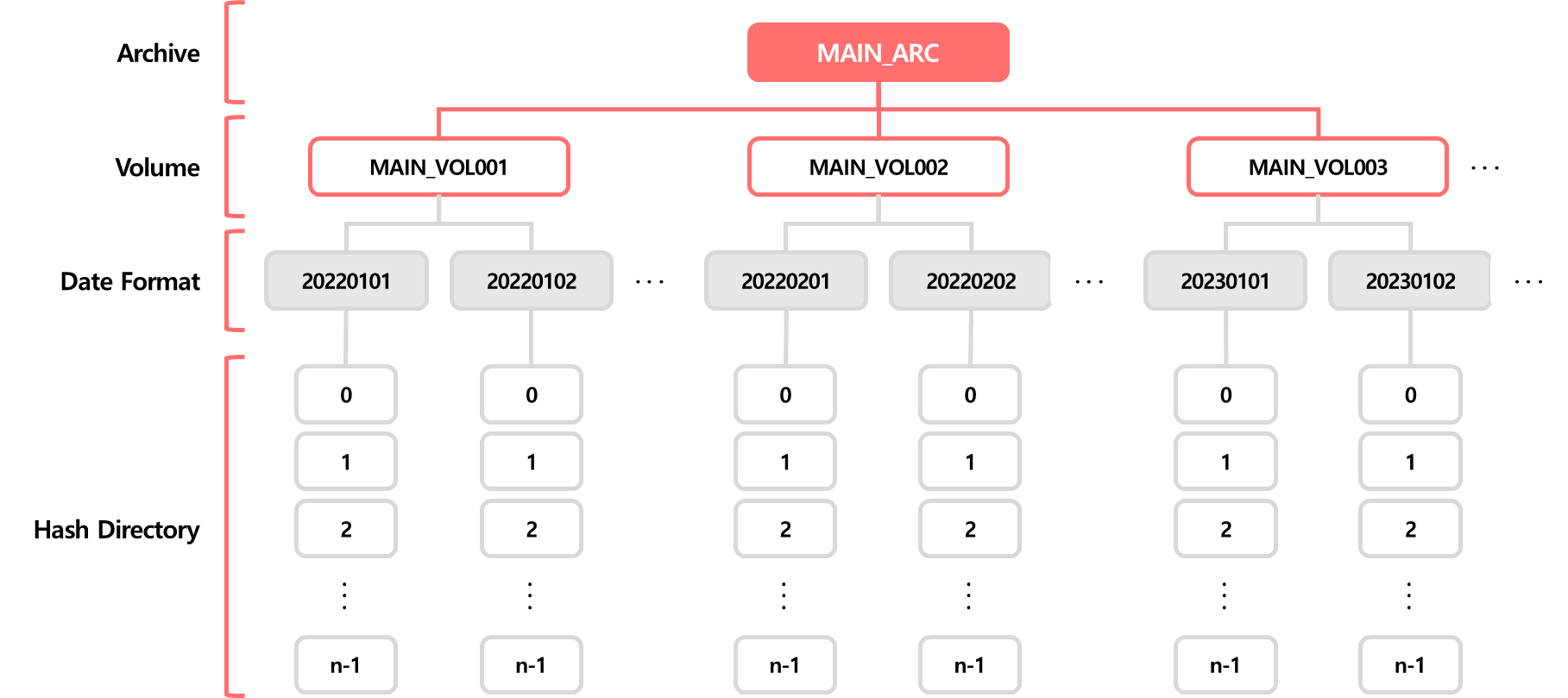
[그림2-11] 엔진 주요 테이블

****

## XTORM 엔진 저장소 구성 개념도

일반적으로 스토리지는 이미지 스토리지 서버에 물리적인 볼륨으로 할당된다. 물리적인 볼륨은 RAID의 구성방법, 서버 볼륨 그룹의 구성방법 등의 의해 일정한 크기로 나누어 할당되며, XTORM은 이 물리적인 볼륨에 파일을 체계적으로 저장하여 관리한다.

[그림2-12] 저장소 개념도



XTORM은 서버에 할당된 물리 볼륨을 논리적으로 분할하여 “VolumeId” 라는 Id를 부여하여 관리한다. VolumeId는 스토리지 파일 시스템 명과 동일하며, 이는 XTORM DB에 insert 하게 된다. 이를 “Archive” 라는 저장소에 할당하여 사용 하게 된다.

각 저장소는 “Basedir” 이라는 물리적인 디렉토리 루트 값을 가지고 있으며, XTORM볼륨의 절대경로는 “저장소의 Basedir + ‘/’ + VolumeId” 라고 간주하면 된다.

XTORM 볼륨은 Date Format이라는 “년월일”로 명명된 디렉토리 내에 해시 디렉토리라는 최하위 디렉토리 밑에 실제 파일을 저장한다. 따라서 볼륨 크기 및 해시 디렉토리 수를 조절하여 최종 해시 디렉토리에 저장되는 파일 개수를 조절할 수 있다. 일반적으로 한 개의 해시 디렉토리에는 500~1,000개의 파일을 저장하여야 조회 시 성능저하를 막을 수 있다.

## XTORM 엔진 메인 테이블

**-** ASYSCONTENTELEMENT Table (Main Table)

[표2-4] AsysContentElement 테이블 정보

|  |  |
| --- | --- |
| **XTORM** | **ASYSCONTENTELEMENT** |
| 테이블 영문 명 | ASYSCONTENTELEMENT |
| 테이블 설명 | KEY 관리 테이블 |

**-** ASYSCONTENTELEMENT Table Fields

[표2-5] AsysContentElement 테이블 컬럼 정보

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **순번** | **필드 명** | **필드TYPE** | **길이** | **NULL** | **필드설명** |
| 1 | ELEMENTID | CHAR | 16 | NN | 엔진에서 관리하는 ID |
| 2 | VOLUMEID | VARCHAR2 | 64 | NN | 볼륨 ID |
| 3 | FILEKEY | VARCHAR2 | 64 | NN | 저장된 오브젝트를 검색. 파일경로 |
| 4 | FILESIZE | NUMBER |  | NN | 파일 크기(bytes) |
| 5 | CREATEDATE | DATE |  | NN | 생성일자 |
| 6 | LASTACCESS | DATE |  | NN | 마지막 접근 일자 |
| 7 | CONTENTTYPE | NUMBER |  | NN | 컨텐츠 유형 |
| 8 | CONTENTCLASSID | CHAR | 16 |  | 콘텐츠 클래스 ID |
| 9 | GENERALDATA | VARCHAR2 | 64 |  | 여러 상태를 함께 지정하기 위한 필드 |
| 10 | CHECKHINT | NUMBER |  |  | Migration 상태 값 |
| 11 | WRITEDATE | DATE |  |  | Storage에 저장된 시간 |
| 12 | RETENTION | NUMBER |  |  | 보존주기(초 단위) |
| 13 | CHECKSUM | VARCHAR2 | 64 |  |  |

- ASYSELEMENT Table (The definition of elements)

[표2-6] AsysElement 테이블 정보

|  |  |
| --- | --- |
| **XTORM** | **ASYSELEMENT** |
| 테이블 영문 명 | ASYSELEMENT |
| 테이블 설명 | XTORM 데이터베이스 안에 저장되는 모든 앨리먼트 정의 |

**-** ASYSELEMENT Table Fields

[표2-7] AsysElement 테이블 컬럼 정보

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **순번** | **필드 명** | **필드TYPE** | **길이** | **NULL** | **PK** | **필드설명** |
| 1 | ELEMENTID | CHAR | 16 | NN | PK | 엔진에서 관리하는 ID |
| 2 | DESCR | VARCHAR2 | 64 | NN |  | 앨리먼트 설명 |
| 3 | USERSCLASS | CHAR | 16 | NN |  | 사용자 보안 클래스 |
| 4 | ECLASSID | CHAR | 16 | NN |  | 앨리먼트 유형 ID |
| 5 | EXPIRATION | DATE |  |  |  | 오브젝트 만료 일자 |

## XTORM Migration(이관) 관리 정보

XTORM에 입력된 파일은 XTORM Migration Component에 의하여 저장소 간, 서버간 파일을 자동으로 복제/이동 관리를 한다. 이는 입력 시에 부여하는 Lifecycle Id(혹은 Content Class Id)에 의해 저장 룰이 지정되며 특정시간 혹은 특정 조건에 따라 XTORM Scheduler Component통해 작업이 수행된다.

[그림2-1] Lifecycle 개념도 예시



**[Migration설명]**

[그림2-1]의 예시와 같이 “이미지 저장 룰” 이라는 Lifecycle이 특정 파일에 지정되었을 때, 최초 입력 시 이미지 저장소에 저장된다. 특정 기간 후에 Archive 저장소에 복제되고, 30일 경과 시 이미지 저장소에서 지워지고 Archive 저장소에 복제된다.

Archive저장소는 같은 이미지 서버만 아니라 원격지 이미지 서버로 구성이 가능하다.

**[복제(백업)/이동 설명]**

XTORM 에 저장된 파일은 XTORM Data Service가 통합관리하고 있는 저장소(archive)간에 파일을 복제/이동할 수 있다. 특정한 상태를 갖는 XTORM Content State를 만들어 원하는 저장소를 지정하고 이를 Lifecycle에 설정하는 것이다.

**[Lifecycle 기준]**

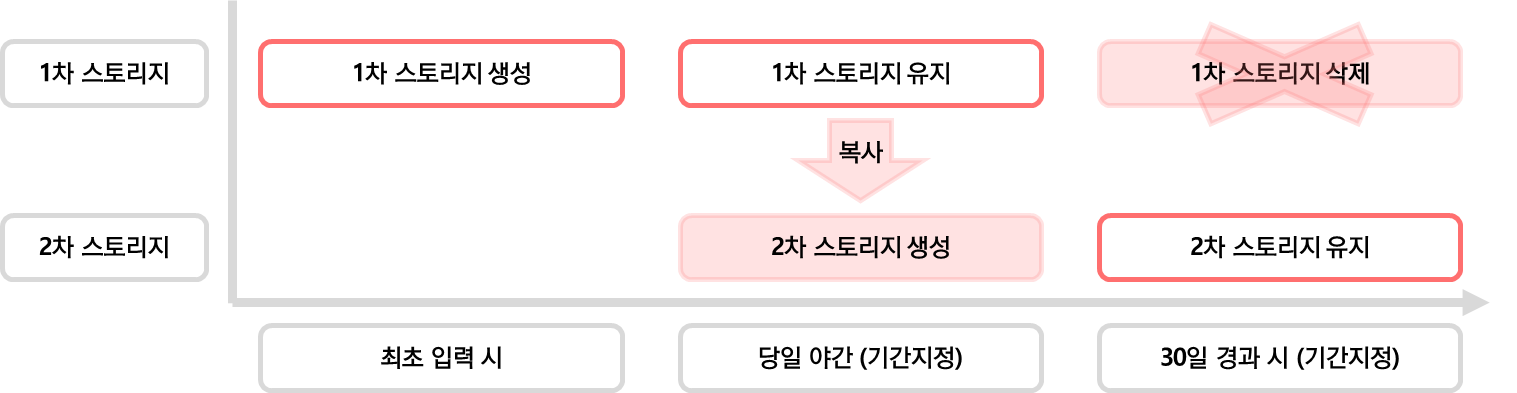
기간 베이스 도는 컨텐츠의 중요도에 따라 기준 설정

[그림2-2] Lifecycle 관리 정책 예시

텍스트이(가) 표시된 사진

자동 생성된 설명

[그림2-3] Lifecycle 관리 흐름 예시



* 업무별 Content 분류 정보

[표2-8] Content Class 정보

|  |  |  |
| --- | --- | --- |
| **Content Class** | **업무명** | **설명** |
| BASIC | 통합 | 통합 시스템 내 저장되는 첨부 파일 및 양식 파일 |
| IFTI\_CC | 연계 | 채널 연계 시스템에서 관리하는 파일 |
| MIG\_CC | 이관 | 시스템 신규 구축 이전 이관 대상 파일 |

- 업무별 Content 라이프사이클 정보

[표2-9] Content State 정보

|  |  |  |
| --- | --- | --- |
| **Content Class** | **목적** | **설명** |
| BASIC | 초기 생성 | 초기 생성시 아카이브에 파일 생성(1차 스토리지 보관) |
| 복제 | 당일 ~ 30일보관 (1차, 2차 스토리지 보관) |
| 이관 | 30일 경과 시 (2차 스토리지 보관) |
| IFTI\_CC | 초기 생성 | 초기 생성시 아카이브에 파일 생성 |
| 이관 | 30일 경과 시 (2차 스토리지 보관) |
| MIG\_CC | 보관 | HCP(WORM) 스토리지 보관 |

## XTORM Scheduler 관리 정보

**-** Scheduler 설정 정보

[표2-10] Scheduler 정보

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Scheduler ID** | **대상 CC** | **유형** | **DATA 서버** | **스케줄러 설정 설명** |
| BASIC\_START | BASIC | Migration 시작 | XTORM #1 | 매주 평일 PM 7시 시작 |
| BASIC\_STOP | BASIC | Migration 종료 | XTORM #1 | 매주 화 ~ 토 AM 7시 중지 |
| IFTI\_START | IFTI\_CC | Migration 시작 | XTORM #2 | 매주 주말 AM 7시 시작 |
| IFTI\_STOP | IFTI\_CC | Migration 종료 | XTORM #2 | 매주 주말 PM 7시 중지 |

# XTORM (ECM 엔진) 모니터링

본 장에서는 XTORM(ECM 엔진)에 대한 로그 모니터링 및 API 문제 해결에 대해 설명한다.

## XTORM 엔진 서비스 로그 모니터링

**-** XTORM 서비스 로그 파일

에러 및 시스템 오류 등이 발생한 경우, XTORM 엔진의 로그를 통해 원인을 파악한다.

log파일은 10MB단위로 새로운 파일을 생성하게 된다.

10MB가 넘으면 기존에 있던 파일은 xtorm\_2102.log\_<년월일시분초> 로 명칭을 바꾸고 새로운 xtorm\_2102.log 파일이 생기게 된다.

Log파일이 커서 vi 편집기로 Open을 할 수 없을 경우에는 more xtorm\_2102.log 명령어를 이용해 Log 내용을 확인할 수 있다.

- XTORM system out 파일

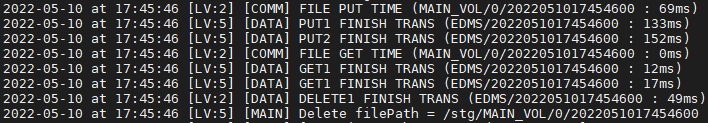
XTORM 기동 시 발생하는 오류에 대해서는 system\_<port>.out에서 확인 할 수 있다.

[표3-1] 로그 파일 위치

|  |  |  |
| --- | --- | --- |
| **서버** | **구분** | **로그파일 위치** |
| **XTORM 서버** | **Data/Archive Agent** | **/log/xtorm/xtorm\_2102.log** |
| **System log** | **/log/xtorm/systemlog/system\_2102.out** |

- 로그 레벨 5의 경우

[그림3-1] 로그 레벨 5 화면



**[설명]**

FILE PUT TIME:

파일 스트림 받은 후 서버에서 파일로 저장 할 경우 해당 로그가 Write 됩니다.

FILE PUT1 TIME:

최초 요청부터 DB에서 볼륨 정보 확인 및 아카이브 확인 할 경우 해당 로그가 Write 됩니다.

FILE PUT2 TIME:

ASYSCONTENTELEMENT INSERT 할 경우 해당 로그가 Write 됩니다.

FILE GET TIME:

읽은 파일 데이터 클라이언트로 내릴 때 해당 로그가 Write 됩니다.

FILE GET1 TIME:

최초 요청부터DB에서 파일 찾은 후 해당 로그가 Write 됩니다.

FILE GET1 TIME:

ASYSCONTENTELEMENT DB업데이트 완료 후 해당 로그가 Write 됩니다.

DELETE1 FINISH TRANS

클라이언트가 XTORM 엔진으로 등록된 파일을 삭제 요청 할 경우 해당 로그가 Write 됩니다.

- 로그 레벨 2의 경우

[그림3-2] 로그 레벨 2 화면

****

**[설명]**

FILE GET TIME:

클라이언트가 XTORM 엔진으로 조회가 일어날 경우 해당 로그가 Write 됩니다.

FILE PUT TIME:

클라이언트가 XTORM 엔진으로 파일을 등록할 경우 해당 로그가 Write 됩니다.

DELETE1 FINISH TRANS

클라이언트가 XTORM 엔진으로 등록된 파일을 삭제 요청 할 경우 해당 로그가 Write 됩니다.

## XTORM 엔진 API 문제 해결

- XTORM API 에러 코드

XTORM 엔진에서 저장하는 에러 로그는 대부분 API를 통한 거래에 의해 발생된 것이기 때문에 해당 API의 상세 오류 메시지로 에러 내용이 반환됩니다.

[표3-2] XTORM API 에러 코드

|  |  |  |
| --- | --- | --- |
| **코드** | **내용** | **비고** |
| 0 | 정상 | 정상 |
| 3 | 엔진 발생 오류 | API 상세 로그 메시지 확인 |
| 5 | 파라미터 오류 |
| 10 | 커넥션 연결 오류 |
| 11 | 커넥션 타임아웃 오류 |

**API 리턴 코드는 위와 같다. API 호출 후 0 이 아닌 경우 전부 오류 이며 API 사용시, getLastError()를 통해 상세 메시지 받아서 오류 확인 해야 한다.**

- XTORM API 에러 메시지

XTORM 엔진에서 발생하는 에러 로그는 엔진 자체 로그부터 DB에서 발생한 오류 등 매우 다양하므로 정확한 진단을 위해서는 XTORM 엔지니어 혹은 유지보수 담당자에게 문의해야 한다.   
XTORM 엔진에서 발생하는 오류 중 가장 빈번히 발생하는 유형은 다음과 같다.

[표3-3] XTORM API 에러 유형

|  |  |  |
| --- | --- | --- |
| **에러 메시지** | **내용** | **조치 방안** |
| [308] [COMM] Communications agent no longer listening on an incoming socket (Socket closed). | COMM 에이전트 기동 종료로 소켓 닫힘 | XTORM 에이전트 재 기동 필요 |
| [6717] [DATA] Error (Error creating local file in file transfer: "FILE\_PATH"  (No such file or directory (errno:2))) retrieving content to a client file from archive "ARCHIVE". Other archives will be tried. | 파일 등록 경로 비 정상 (경로 부적절 또는 해당 경로의 쓰기 권한 문제) | 등록 파일 경로 및 권한 여부 확인 필요 |
| [6714][DATA]Element "elementid" not found in the database | 데이터베이스에 해당 “Elementid” 존재하지 않음 | ElementId 확인 및 Xtorm DB 확인 필요 |
| [6716] [DATA] There are no available archives that can access content for element "elementid", last local path "FILE\_PATH". | 파일 다운로드 실패 | 다운로드 경로 및 쓰기 권한, Archive 목록 확인, 로그 확인 후 조치 필요 |
| [6428] [DATA] A connection to the database has been lost. Closing connection and retrying transaction... | 트랜잭션 도중 DB 연결 끊김 또는 DeadLock 발생 | DB 상태 확인 필요 |

# XTORM (ECM 엔진) 관리 정보

본 장에서는 XTORM(ECM 엔진)을 활용한 프로젝트 내에서 XTORM을 관리하기 위한 정보를 설명한다..

## 프로젝트 사이트 정보

[표4-1] 프로젝트 사이트 정보

|  |  |
| --- | --- |
| **항목** | **내용** |
| 프로젝트 명 | OOOO 차세대 시스템 구축 |
| 프로젝트 수행 기간 | 2022.01.11 ~ 2022.05.18 (4MM) |
| 유지보수 담당자 | OOO 매니저 / 010-0000-0000 / slave@inzent.com |
| 사이트 담당자 | OOO 수석 / 010-0000-0000 / waterstone@banksystem.com |
| 엔지니어 | OOO 책임 / 010-0000-0000 / imbook@inzent.com |
| 엔진 버전 및 JAR | XTORM 5R1 / xvarm\_20181220\_PROD.jar |
| 설치 대상 서버 | 개발(hostname\_dev1), 검증(hostname\_stg1), 운영1(hostname1), 운영2(hostname2) |

- R&R

[표4-2] 사이트 R&R 정보

|  |  |  |
| --- | --- | --- |
| **회사** | **담당자 / 연락처** | **R&R** |
| XTORM 엔지니어 | OOO 책임 / 010-0000-0000 / imbook@inzent.com | 파일 이관 및 ECM 엔진 설치 및 가이드 |
| 마스터 뷰어 | OOO 수석 / 010-0000-0000 / waterstone@vierwer.com | 뷰어 |
| XTORM 유지보수 | OOO 매니저 / 010-0000-0000 / slave@inzent.com | 유지보수 |

## 서버 정보

**-** 운영서버

[표4-3] 운영 서버 정보

|  |  |  |
| --- | --- | --- |
| **서버** | **항목** | **내용** |
| 운영1 | OS | AIX 7.2 |
| IP | 192.168.10.101 |
| User (sudo 권한 여부) | ecmadm (N) |
| Password | Ecmadmin1024! |
| JDK Version | 1.8 |
| 접속 방안 | 현업 팀장 승인 하에 서버실 출입 |
| 운영2 | OS | AIX 7.2 |
| IP | 192.168.10.102 |
| User (sudo 권한 여부) | ecmadm (N) |
| Password | Ecmadmin1024! |
| JDK Version | 1.8 |
| 접속 방안 | 현업 팀장 승인 하에 서버실 출입 |

**-** 개발 및 검증 서버

[표4-4] 개발 및 검증 서버 정보

|  |  |  |
| --- | --- | --- |
| **서버** | **항목** | **내용** |
| 검증 | OS | AIX 7.2 |
| IP | 192.168.20.100 |
| User (sudo 권한 여부) | ecmadm (N) |
| Password | Ecmadmin! |
| JDK Version | 1.8 |
| 접속 방안 | 담당자 PC 접속 |
| 개발 | OS | AIX 7.2 |
| IP | 192.168.20.110 |
| User (sudo 권한 여부) | ecmadm (N) |
| Password | Ecmadmin! |
| JDK Version | 1.8 |
| 접속 방안 | 담당자 PC 접속 |

## 아카이브 · 볼륨 구성 정보

- 운영서버 아카이브 · 볼륨 구성 정보

[표4-5] 운영서버 아카이브 · 볼륨 구성 정보

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **업무** | **스토리지** | **아카이브** | **볼륨** | **용량** | **비고** |
| 파일 첨부 | SAN(/stg1) | MAIN\_ARC | MAIN\_VOL001~003 | 300GB |  |
| 연계 관리 | NAS(/stg2) | IFTI\_ARC | IFTI\_VOL001~002 | 150GB |  |
| IFTI\_VOL003~050 | 1TB |  |
| 증적 파일 관리 | ECS(/stg3) | PROV\_ARC | PROV\_VOL001 | 2TB | 138.10.1.2 |

- 검증서버 아카이브 · 볼륨 구성 정보

[표4-6] 검증서버 아카이브 · 볼륨 구성 정보

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **업무** | **스토리지** | **아카이브** | **볼륨** | **용량** | **비고** |
| 공통 | SAN(/stg1) | MAIN\_ARC  IFTI\_ARC | MAIN\_VOL001  IFTI\_VOL001 | 300GB |  |
|  |  |  |  |  |  |

- 개발서버 아카이브 · 볼륨 구성 정보

[표4-7] 개발서버 아카이브 · 볼륨 구성 정보

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **업무** | **스토리지** | **아카이브** | **볼륨** | **용량** | **비고** |
| 공통 | SAN(/stg1) | MAIN\_ARC  PROV\_ARC | MAIN\_VOL001~003 IFTI\_VOL001 | 30GB |  |
|  |  |  |  |  |  |

## 엔진 정보

- 운영서버 이중화

[표4-8] 운영서버 이중화 정보

|  |  |  |
| --- | --- | --- |
| **이중화 요소** | **항목** | **내용** |
| 로드밸런서 (LB) | IP ( Domain ) | 192.168.10.100 (ecm.pro.com) |
| PORT | 22102 |
| 관리자 페이지 URL | https://ecm.pro.com/speno/xvarm/index.jsp |
| 로드 밸런스 방식 | Hash 방식 |
| VIP | VIP (Domain) | 172.0.3.8 (ecm.pro.com) |
| 관리자 페이지 URL | https://ecm.pro.com/speno/xvarm/index.jsp |
| 전환 방식 | 솔루션 영역 |

**-** 운영서버

[표4-9] 엔진 운영 서버 정보

|  |  |  |
| --- | --- | --- |
| **서버** | **항목** | **내용** |
| 운영1 | IP ( Domain ) | 192.168.10.101 (ecm.pro1.com) |
| PORT | 2102 |
| 엔진 경로 | /sw/xtorm |
| 관리자 페이지 URL | https://ecm.pro1.com/speno/xvarm/index.jsp |
| 운영2 | IP ( Domain ) | 192.168.10.102 (ecm.pro2.com) |
| PORT | 2102 |
| 엔진 경로 | /sw/xtorm |
| 관리자 페이지 URL | https://ecm.pro2.com/speno/xvarm/index.jsp |
| 공통 | 복호화 OR WORM 임시 경로 | /stg/temp |
| 로그 경로 | /log/xtorm/xtorm\_2102.log |

**-** 개발 및 검증 서버

[표4-10] 엔진 개발 및 검증 서버 정보

|  |  |  |
| --- | --- | --- |
| **서버** | **항목** | **내용** |
| 검증 | IP ( Domain ) | 192.168.20.100 (ecm.stg.com) |
| PORT | 2102 |
| 엔진 경로 | /sw/xtorm |
| 관리자 페이지 URL | https://ecm.stg.com/speno/xvarm/index.jsp |
| 복호화 OR WORM 임시 경로 | /stg/temp |
| 로그 경로 | /log/xtorm/xtorm\_2102.log |
| 개발 | IP ( Domain ) | 192.168.20.110 (ecm.dev.com) |
| PORT | 2102 |
| 엔진 경로 | /sw/xtorm |
| 관리자 페이지 URL | https://ecm.dev.com/speno/xvarm/index.jsp |
| 복호화 OR WORM 임시 경로 | /stg/temp |
| 로그 경로 | /log/xtorm/xtorm\_2102.log |

- 엔진 운영 서버 구조도

[그림4-1] 엔진 운영 서버 구조도

|  |
| --- |
| * **작성해서 기입 바람** |

- 엔진 검증 서버 구조도

[그림4-2] 엔진 검증 서버 구조도

|  |
| --- |
| * **작성해서 기입 바람** |

- 엔진 개발 서버 구조도

[그림4-3] 엔진 개발 서버 구조도

|  |
| --- |
| * **작성해서 기입 바람** |

## WEB / WAS 정보

-WEB / WAS 구성 정보

[표4-11] WEB / WAS 구성 정보

|  |  |  |
| --- | --- | --- |
| **서버** | **항목** | **내용** |
| 운영1 | 종류 | WebToB5 / JEUS8 |
| URL | https://edms.bank.com |
| 관리자 페이지 포함 여부 | X |
| 공통 로직 포함 여부 | O (Create, Download API 사용) |
| 운영2 | 종류 | WebToB5 / JEUS8 |
| URL | https://edms.bank.com |
| 관리자 페이지 포함 여부 | X |
| 공통 로직 포함 여부 | O (Create, Download, Delete API 사용) |
| 검증 | 종류 | Tomcat9 |
| URL | http://192.168.20.100:8080 |
| 관리자 페이지 포함 여부 | X |
| 공통 로직 포함 여부 | O (Create, Download API 사용) |
| 개발 | 종류 | Tomcat9 |
| URL | http://192.168.20.110:8080 |
| 관리자 페이지 포함 여부 | X |
| 공통 로직 포함 여부 | O (Create, Download API 사용) |

- API 배포 정보

XTORM API를 통해 엔진에 접근하는 모든 WAS 및 Client 프로그램에 대한 정보를 기술한다.

[표4-12] API 배포 정보

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **서버(hostname)** | **역할** | **API버전** | **사용메소드** | **비고** |
| 127.0.0.1 | WAS1 | xtormapi\_2020.jar | create(),download() |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

## 클라이언트 정보

|  |  |
| --- | --- |
| **항목** | **내용** |
| 회사명 | 미네르바 |
| 담당자 | OOO 대리 / 010-0000-0000 / calldriver@minerba.com |
| 솔루션 | Magic Scanner, Magic Viewer |
| 역할 | 스캔, 뷰어 적용 |
| API | Create, Download, Delete 사용 |

## 암호화 정보

|  |  |
| --- | --- |
| **항목** | **내용** |
| 회사명 | 드림시큐리티 |
| 담당자 | OOO 과장 / 010-0000-0000 / exaggeration@dream.com |
| 솔루션 | SCP-DP |
| 암호화 위치 | Client 암호화 / 엔진 암호화 |
| 암호화 모듈 위치 | /sw/cryptor/ |
| Xtorm Filter 소스 경로 | /sw/xtorm/lib/filter/decrypt.java  /sw/xtorm/lib/filter/encrypt.java |

## 배치 정보

|  |  |
| --- | --- |
| **배치 / 설명** | **기능 / 위치** |
| del\_batch  복호화 파일 삭제 배치 | /sw/xtorm/bin/del\_batch  엔진에서 조회한 파일을 복호화 하는 배치  /stg/temp에 파일을 떨군다. |

# 추가 정보

본 장에서는 프로젝트 내에서 참고할 수 있는 정보에 대해 서술한다.

## 기술 지원 이력표

[표5-1] 기술 지원 이력표

|  |  |  |
| --- | --- | --- |
| **일시** | **지원담당자** | **기술지원내용** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

## 참조 자료 경로

[표5-2] 참조 자료 경로

|  |  |  |
| --- | --- | --- |
| **위치** | **항목** | **내용** |
| 시스템 관리 파일서버 | XTORM 운영자 매뉴얼 | XTORM담당 운영자에게 필요한 내용이 기술되어있음. |
| XTORM 인덱스 테이블 등록  ElementId : 2022051104032300 | 시스템 구성도 | 전체적인 소프트웨어 구성 및 AS-IS, TO-BE 서버의 구성도가 기록되어 있음 |
| /sw/xtorm/lib/filter/decrypt.java  /sw/xtorm/lib/filter/encrypt.java | 필터 암복호화 소스 | XTORM 통해 암복호화 하기 위한 소스 |
|  |  |  |
|  |  |  |
|  |  |  |