목 차

Ⅰ. 시작하기 전에 3
본 매뉴얼 목적 4
본 매뉴얼 구성 4

Ⅱ. 오즈 엔터프라이즈 서버 소개 6
오즈 엔터프라이즈 서버 개요 7
오즈 엔터프라이즈 서버 특징 7
오즈 엔터프라이즈 서버 아키텍처 10

Ⅲ. 오즈 엔터프라이즈 서버 실행 11
시스템 요구 사항 12
오즈 엔터프라이즈 서버 파일 구성 13
오즈 엔터프라이즈 자바 서버 실행 19
오즈 엔터프라이즈 닷넷 서버 실행 23

Ⅳ. 오즈 엔터프라이즈 서버 옵션 설정 26
assembly.properties 27
cachemngr.properties 30
clientlist.xml 32
databind.properties 34
datasource.properties 35
db.properties 37
dbconfig.xml, dbconfig.xsd 40
log.properties 42

mail.properties 52
monitor.properties 54
ozserver.properties 59
ozservice.properties 61
ozudl.properties 62
ozudr.properties 63
repository.properties 64
sap.properties 67
spmgr.properties 69
srmngr.properties 72
uslmngr.properties 73
xml.properties 74

Ⅴ. 오즈 엔터프라이즈 자바 서버 부가 기능 및 기타 설정 75
user, password 암호화 76
빌트인 타입 리파지토리를 RDB 타입으로 변환 78
오즈 엔터프라이즈 자바 서버 Shell 명령어 80
오즈 엔터프라이즈 자바 서버를 윈도우 서비스로 등록 88
오즈 엔터프라이즈 자바 서버와 웹 서비스 연동 91

Ⅵ. 오즈 엔터프라이즈 닷넷 서버 부가 기능 및 기타 설정 97
user, password 암호화 98
빌트인 타입 리파지토리를 RDB 타입으로 변환 100
오즈 엔터프라이즈 닷넷 서버 실행 환경 설정 103
오즈 엔터프라이즈 닷넷 서버를 윈도우 서비스로 등록 105
오즈 엔터프라이즈 닷넷 서버와 웹 서비스 연동 108

Ⅰ. 시작하기 전에

- 본 매뉴얼의 목적
- 본 매뉴얼 구성

본 매뉴얼의 목적

본 매뉴얼은 오즈 엔터프라이즈 자바/닷넷 서버의 모든 구성 요소에 대한 정보 제공과 오즈 엔터프라이즈 자바/닷넷 서버를 설정하고 이용하는 방법에 대한 기술적인 정보를 상세히 설명합니다.

본 매뉴얼 구성

본 매뉴얼 구성과 각 장에서 설명하는 내용은 다음과 같습니다.

오즈 엔터프라이즈 서버 소개

오즈 엔터프라이즈 서버에 대한 개요, 특성, 기능 및 아키텍처를 설명합니다.

오즈 엔터프라이즈 서버 실행

오즈 엔터프라이즈 서버를 실행하기 위한 시스템 요구 사항과 오즈 엔터프라이즈 서버의 실행 방법을 설명합니다.

오즈 엔터프라이즈 서버 옵션 설정

오즈 엔터프라이즈 서버의 동작 옵션을 설정하는 파일 및 동작 옵션에 대해 설명합니다.

오즈 엔터프라이즈 자바 서버 부가 기능 및 기타 설정

오즈 엔터프라이즈 자바 서버에서 제공하는 user, password 암호화 방법, 빌트인 타입 리파지토리 RDB 타입으로 변환하는 방법, Shell 명령어 등의 부가 기능과 오즈 엔터프라이즈 서버와 웹 서비스 연동 방법 및 윈도우 서비스로 등록하는 방법에 대해 설명합니다.

오즈 엔터프라이즈 닷넷 서버 부가 기능 및 기타 설정

오즈 엔터프라이즈 닷넷 서버에서 제공하는 user, password 암호화 방법, 빌트인 타입 리파지토리 RDB 타입으로 변환하는 방법 등의 부가 기능과 오즈 닷넷 서버 동작 환경 설정 방법, 오즈 엔터프라이즈 서버와 웹 서비스 연동 방법 및 윈도우 서비스로 등록하는 방법에 대해 설명합니다.

Ⅱ. 오즈 엔터프라이즈 서버 소개

- 오즈 엔터프라이즈 서버 개요
- 오즈 엔터프라이즈 서버 특징
- 오즈 엔터프라이즈 서버 아키텍처

오즈 엔터프라이즈 서버 개요

오즈 엔터프라이즈 서버는 보고서 폼 관리와 클라이언트의 요청에 따른 보고서 데이터 셋 생성 및 폼과 데이터 셋의 전송을 수행하는 고성능 다중 스레드 서버 엔진입니다.

오즈 엔터프라이즈 서버 특징

우수한 이식성과 확장성

[자바 서버]

- 순수 JAVA/XML 기술로 개발
- JVM (Java Virtual Machine)이 탑재된 모든 운영체제 환경 지원
- 다양한 WAS와 연동하는 통합 환경 제공

[닷넷 서버]

- .NET Framework 기술로 개발
- ADO.NET을 통한 다양한 데이터베이스와 연동 가능
- ADO.NET Framework가 탑재된 모든 운영체제 환경 지원

고성능 서버 아키텍처

서버와 클라이언트 간 분산 처리를 통해 다중 사용자 환경에서 최적의 성능 보장

효과적인 서버 운영관리

- 엔터프라이즈 매니저를 통한 one-point 통합 관리
- 실시간 모니터링 및 서버 설정
- 분산 원격 테스트 및 디버깅

리파지토리를 통한 버전 및 권한 관리

보고서 폼 파일, 데이터 정보 파일 등 다양한 아이템의 저장 및 수정에 대한 히스토리와 버전 관리, 계층적인 카테고리 관리 및 권한 설정 기능을 제공합니다.

고성능 리포팅 엔진

DB Pooling, 다중 스레드, 스레드 재사용, 보고서별 캐시 설정, 디스크/메모리 혼용 캐시 기능 등 다양한 성능 향상 기법을 도입하여 다수 사용자의 동시 접속 환경에서도 최적의 성능을 보장합니다.

다양한 데이터 소스 지원, 유연한 데이터 처리

- JDBC, ODBC, OLEDB, Clear Quest를 지원하는 모든 데이터베이스 지원
- 일반 텍스트 파일, XML(DTD/XSD), XPath 지원
- DB Procedure, SAP R3, Web Service(SOAP, WSDL, UDDI) 지원
- UDS(User Data Store) 연동을 통한 다양한 사용자 데이터 처리

  ※ 참고사항 : UDS란 EJB, Servlet, COM, Legacy Data 등 사용자 애플리케이션을 통해 받은 다양한 형태의 데이터를 연동할 수 있도록 오즈에서 제공하는 인터페이스를 말합니다.

대용량 데이터 실시간 처리

데이터 소스로부터 가져온 데이터의 처리 속도 향상을 위해 실시간으로 뷰어로 전송할 수 있는 동시 전송 모드(Concurrent Mode)를 지원하며, 대용량 데이터 처리 시에 메모리 관리의 효율을

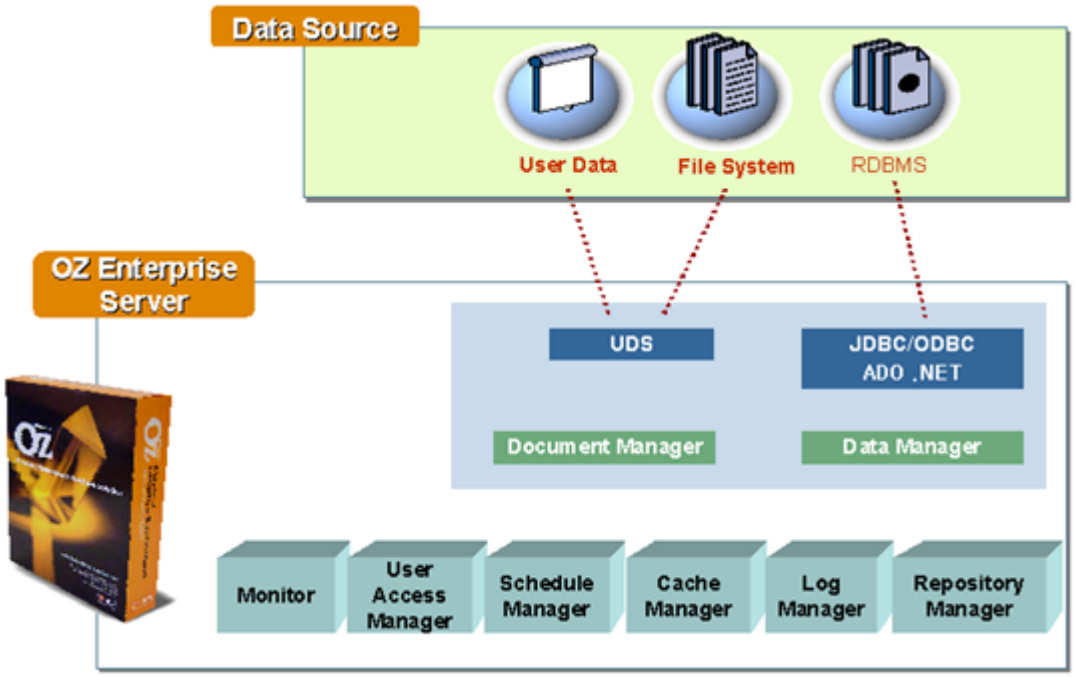
위해 파일 모드를 지원합니다.

데이터 보안 및 인증

보고서의 데이터를 담고 있는 데이터 모듈 파일의 위/변조 방지 기능이 있어, 공문서나 인증서 등의 서비스에 유용하게 적용될 수 있습니다. 또한 사용자 및 그룹별로 계층적 보안을 적용하고 인가된 사용자만이 접근 가능하도록 통제할 수 있어 보고서 보안 유지에 편리합니다.

오즈 엔터프라이즈 서버 아키텍처

오즈 엔터프라이즈 서버 아키텍처는 다음과 같습니다.



UDS	UDS를 이용한 Non-database Source의 이용
JDBC/ODBC ADO. NET	JDBC/ODBC(자바 서버) 또는 ADO.NET(닷넷 서버)을 이용하여 데이터베이스에 연결
Document Manager	보고서 관리 수행
Data Manager	보고서의 데이터 모듈 생성 및 관리 수행
Monitor	서버 프로세스의 운영 상황을 모니터링하여 파일로 로깅
User Access Manager	사용자별/그룹별 보고서에 대한 접근 관리 수행
Schedule Manager	오즈 스케줄러가 생성한 결과 데이터 셋에 대한 관리 수행
Cache Manager	자주 요청하는 결과 데이터 셋이나 보고서 폼 등의 캐싱 처리
Log Manager	오즈 서버의 실행 결과를 로깅
Repository Manager	리파지토리 관련 서비스 수행

Ⅲ. 오즈 엔터프라이즈 서버 실행

- 시스템 요구 사항
- 오즈 엔터프라이즈 서버 파일 구성
- 오즈 엔터프라이즈 자바 서버 실행
- 오즈 엔터프라이즈 닷넷 서버 실행

시스템 요구 사항

플랫폼

[자바 서버]

JRE(Java Runtime Environment) 1.3.X 이상을 지원하는 모든 운영 체제

[닷넷 서버]

.NET Framework 1.1 이상, Microsoft Internet Information Server(IIS) 5.1 이상, ASP.NET 을 지원하는 모든 운영 체제

DBMS

[자바 서버]

Oracle, Informix, Sybase, MS-SQL Server, DB2 등 ODBC/JDBC를 지원하는 모든 상용 DBMS

[닷넷 서버]

Oracle, Informix, Sybase, MS-SQL Server, DB2 등 ADO.NET를 지원하는 모든 상용 DBMS

※ 주의사항 : 닷넷 서버에서 DBMS 사용 시 주의사항

- DB2 : DB2 Runtime Client 프로그램이 설치되어야 합니다.
- Oracle, Sybase, Informix : 해당 Client SDK가 설치되어야 합니다.
- ODBC : IIS를 구동하는 계정이나 시스템 DSN에 등록되어야 합니다.
- User defined : 사용하고자 하는 드라이버 DLL 파일이 UniversalDriver에 위치해야 하고, 참조 가능한 상태이어야 합니다.

오즈 엔터프라이즈 서버 파일 구성

오즈 자바 서버

오즈 자바 서버의 파일 구성은 다음과 같습니다.

※ 참고사항 : %OZ_HOME%은 ozserver.properties 파일에서 path.oz.server.home 옵션에 설정된 값으로, 데몬 타입 서버의 경우 "제품 설치 루트 경로\OZ Server 6.0", 서블릿 타입 서버의 경우 "제품 설치 루트 경로\OZ Servlet 6.0\WEB-INF"가 기본 경로입니다.

%OZ_HOME%\bin	
BuiltinToRDBConvert.*	빌트인 타입 리파지토리를 RDB 타입으로 변환하는 명령어 파일(데몬 타입 서버만 제공)
chk_version.*	오즈 서버의 버전 체크 명령어 파일
ozcntrl.*	오즈 서버 컨트롤 명령어 파일
ozencrypted.bat	ID, 패스워드 암호화 명령어 파일
OZServerInstallService.bat	오즈 서버를 윈도우 서비스로 등록하는 명령어 파일(데몬 타입 서버만 제공)
OZServerUninstallService.bat	오즈 서버를 윈도우 서비스에서 삭제하는 명령어 파일(데몬 타입 서버만 제공)
OZServiceSV.exe	오즈 서버를 윈도우 서비스로 등록하거나 서비스에서 삭제하는 실행 파일(데몬 타입 서버만 제공)
ozstart.*	오즈 서버 시작 명령어 파일(데몬 타입 서버만 제공)
ozstop.*	오즈 서버 정지 명령어 파일(데몬 타입 서버만 제공)
server.ico	오즈 서버 아이콘 파일(데몬 타입 서버만 제공)
%OZ_HOME%\classes(서블릿 타입 서버만 제공)	
HelloWorld.class	샘플 클래스 파일
ozserver.properties	오즈 서버 설정 파일
%OZ_HOME%\conf	
cachemngr.properties	캐시 설정 파일
databind.properties	데이터 바인딩 설정 파일
datasource.properties	데이터 소스 설정 파일

db.properties	DB 연결 설정 파일
dbconfig.xml	DBMS별 기본 JDBC 드라이버 설정 파일
dbconfig.xsd	
log.properties	로그 설정 파일
mail.properties	메일 설정 파일
monitor.properties	모니터 로그 설정 파일
ozserver.properties	오즈 서버 설정 파일(데몬 타입 서버만 제공. 서블릿 타입 서버는 %OZ_HOME%\classes 경로에 위치)
ozservice.properties	서비스 등록 설정 파일
ozudl.properties	사용자 정의 로그 설정 파일
ozudr.properties	사용자 정의 리소스 설정 파일
repository.properties	리파지토리 설정 파일
sap.properties	SAP 설정 파일
server_error_msg_en_us.xml	오즈 서버 에러 메시지 파일
spmgr.properties	보안 설정 파일
srmngr.properties	스케줄 데이터 처리 설정 파일
uslmngr.properties	사용자 보안 설정 파일
xml.properties	XML 파서 설정 파일
%OZ_HOME%\conf\sample	
clientlist.xml	clientlist.xml 샘플 파일
db.properties.*	데이터베이스별 db.properties 샘플 파일
repository.properties.*	데이터베이스별 repository.properties 샘플 파일
%OZ_HOME%\Expand(데몬 타입 서버만 제공)	
daemon_redirect.asp	ASP용 브릿지 서버 모듈 샘플 웹페이지 파일
OZMBAServer.dll	ASP용 브릿지 서버 모듈 파일
sample.html	ASP용 브릿지 서버 모듈 샘플 웹페이지 파일
%OZ_HOME%\lib	
activation.jar	오즈 메일 라이브러리 파일
crimson.jar	XML 크림손 파서 라이브러리 파일
dom4j-1.6.1.jar	XML 파서 라이브러리 파일
dtdparser120.jar	XML 스토어를 사용하기 위한 클래스 라이브러리 파일
hsqldb.jar	빌트인 타입의 리파지토리를 사용하기 위한 라이브러리 파일
jaxen-1.1.6.jar	XML 파서 라이브러리 파일

jazzlib.jar	파일 압축 라이브러리 파일
jcert.jar	HTTP/SSL 지원용 J2EE 라이브러리 파일(JRE 1.3.X의 경우에만 필요)
jnet.jar	
js.jar	Java Script 구동 라이브러리 파일
jsr173_api.jar	File/HTTP 스토어의 XML 지원 라이브러리 파일
jsse.jar	HTTP/SSL 지원 J2EE 라이브러리 파일(JRE 1.3.X의 경우에만 필요)
log4j.jar	로그 처리 라이브러리 파일
mail.jar	오즈 메일 라이브러리 파일
NTEventLogAppender.dll	윈도우 NT에 이벤트 로그를 남기기 위한 DLL 파일
ozrresource.jar	서버 바인딩을 실행하는 리소스 파일
ozsfw60.jar	오즈 서버 라이브러리 파일
OZUSLServerDES128.jar	USL 라이브러리 파일
OZUSLServerSession.jar	
ozviewer_lib.jar	서버 바인딩을 실행하는 라이브러리 파일
parser.jar	XML 파서 라이브러리 파일
poi-scratchpad-3.0.1.jar	서버 바인딩 시 보고서를 PPT/XLS 파일로 저장하는 라이브러리 파일
servlet.jar_	서블릿 라이브러리 파일
servlet2_3.jar	서블릿 2.3용 라이브러리 파일
soap2_3_1.jar	SOAP 라이브러리 파일
stax-1.2.0.jar	File/HTTP 스토어의 XML 지원 라이브러리 파일
UDSTestPackage.jar	UDS 테스트 패키지 라이브러리 파일
uk.co.mmscomputing.imageio.tiff.jar	서버 바인딩 시 보고서를 TIFF 파일로 저장하는 라이브러리 파일
xalan.jar	XML XPath 라이브러리 파일
xerces.jar	XML 파서 라이브러리 파일
xercesImpl.jar	XML XPath 라이브러리 파일
xml-apis.jar	
%OZ_HOME%\license	
오즈 서버 라이선스 폴더	
%OZ_HOME%\logs	
서버 로그, 모니터 로그 폴더	
%OZ_HOME%\repository_files	

오즈 서버 리파지토리 기본 폴더	
%OZ_HOME%\scheduled_data	
스케줄링된 데이터 저장 폴더	
%OZ_HOME%\SilverStream(서블릿 타입 서버만 제공)	
Servlet.*	서블릿 구현 샘플 파일
%OZ_HOME%\UniversalDriver	
classes12.jar	Oracle 8i, 9i JDBC 드라이버 파일
classes12.zip	
db2java.jar	IBM DB2 JDBC 드라이버 파일
db2java.zip	
db2jcc.jar	IBM DB2 버전 8 이상 JDBC 드라이버 파일
db2jcc_license_cisuz.jar	
db2jcc_license_cu.jar	
ifxjdbc.jar	Infomix JDBC 드라이버 파일
ijava.jar	Oracle 7i JDBC 드라이버 파일
jconn2.jar	Sybase ASE JDBC 드라이버 파일
jkjdbc.jar	MSSQL 97 JDBC 드라이버 파일
jt400.jar	IBM JT400 JDBC 드라이버 파일
msbase.jar	MSSQL 2000 JDBC 드라이버 파일
mssqlserver.jar	
msutil.jar	
mysql5.1.5.jar	Mysql JDBC 드라이버 파일
ozsapjco2.jar	SAPJCO2 드라이버 파일
ozsapjco3.jar	SAPJCO3 드라이버 파일(JRE 1.5 이상에서 동작)
sqljdbc.jar	MSSQL 2008 JDBC 드라이버 파일
tibero4-jdbc.jar	Tibero4 JDBC 드라이버 파일
tibero5-jdbc.jar	Tibero5 JDBC 드라이버 파일
unisqljdbc10.jar	UniSQL JDBC 드라이버 파일
unisqljdbc205.zip	

※ 주의사항 : UniversalDriver 폴더의 파일은 오즈 서버 설치 시 기본적으로 제공되는 JDBC 드라이버 파일로, 기존에 사용 중인 RDB의 JDBC 드라이버와 호환이 되지 않거나 충돌이 발생할 수 있습니다. 그러므로 기존에 사용 중인 RDB의 JDBC 드라

이버를 사용할 것을 권장합니다.

오즈 닷넷 서버

오즈 닷넷 서버의 파일 구성은 다음과 같습니다.

※ 참고사항 : %OZ_HOME%은 ozserver.properties 파일에서 path.oz.server.home 옵션에 설정된 값으로, "제품 설치 루트 경로\OZ Server.NET 6.0"이 기본 경로입니다.

%OZ_HOME%	
Global.asax	웹 애플리케이션의 시작과 종료 시점에서의 오즈 서버 동작(리소스 생성 및 해제)을 위한 클래스 맵핑 파일
ozserver.properties	오즈 서버 설정 파일
Server.aspx	오즈 서버에 연결되는 포인트를 설정하는 파일
Web.config	오즈 서버 추가 옵션 설정 파일
%OZ_HOME%\assembly	
Tibero.DbAccess.dll	Tibero4 연동 라이브러리 파일
OZTibero5.DbAccess.dll	Tibero5 연동 라이브러리 파일
%OZ_HOME%\bin	
BuiltinToRDBConvert.exe	빌트인 타입 리파지토리를 RDB 타입으로 변환하는 실행 파일
OZDatabaseHandler.dll	데이터베이스 연동 라이브러리 파일
OZEncryptor.exe	ID, 패스워드 암호화 실행 파일
OZSchedulerAPI.dll	오즈 스케줄러 라이브러리 파일
OZServer.NET.dll	오즈 서버 라이브러리 파일
OZServiceLauncher.exe	오즈 닷넷 서버 윈도우 서비스 실행 파일
OZServiceLauncher.exe.config	OZServiceLauncher.exe 설정 파일
OZServiceManager.exe	오즈 닷넷 서버 윈도우 서비스를 관리하는 실행 파일
OZServiceManager.exe.config	OZServiceManager.exe 설정 파일
OZUSLServer.dll	USL 라이브러리 파일
OZWPFViewer_Lib.dll	서버 바인딩을 실행하는 라이브러리 파일
%OZ_HOME%\conf	
assembly.properties	어셈블러 정보 설정 파일

cachemngr.properties	캐시 설정 파일
databind.properties	데이터 바인딩 설정 파일
datasource.properties	데이터 소스 설정 파일
db.properties	DB 연결 설정 파일
dbconfig.xml	DBMS별 기본 ADO.NET 드라이버 설정 파일
dbconfig.xsd	
log.properties	로그 설정 파일
mail.properties	메일 설정 파일
monitor.properties	모니터 로그 설정 파일
ozservice.properties	서비스 등록 설정 파일
ozudl.properties	사용자 정의 로그 설정 파일
ozudr.properties	사용자 정의 리소스 설정 파일
repository.properties	리파지토리 설정 파일
sap.properties	SAP 설정 파일
server_error_msg_en_us.xml	오즈 서버 에러 메시지 파일
spmgr.properties	보안 설정 파일
srmngr.properties	스케줄 데이터 처리 설정 파일
uslmngr.properties	사용자 보안 설정 파일
%OZ_HOME%\conf\sample	
clientlist.xml	clientlist.xml 샘플 파일
db.properties.*	데이터베이스별 db.properties 샘플 파일
repository.properties.*	데이터베이스별 repository.properties 샘플 파일
%OZ_HOME%\license	
오즈 서버 라이선스 폴더	
%OZ_HOME%\logs	
서버 로그, 모니터 로그 폴더	
%OZ_HOME%\scheduled_data	
스케줄링된 데이터 저장 폴더	

오즈 엔터프라이즈 자바 서버 실행

데몬 타입 오즈 자바 서버 실행

■ **오즈 서버 경로, 포트 설정**

%OZ_HOME%\config 폴더에 있는 ozserver.properties 파일을 편집 프로그램으로 열어 오즈 서버 홈 경로와 포트 번호를 설정합니다.

```
server.portnumber=8003
path.oz.server.home=../
...
```

※ 참고사항 : 오즈 서버 홈 경로는 절대 경로 뿐만 아니라 상대 경로로도 설정할 수 있습니다.

■ **자바 옵션 설정**

%OZ_HOME%\bin 폴더에 있는 ozcntrl.bat(유닉스 계열인 경우 ozcntrl.sh) 파일을 편집 프로그램으로 열어 JAVA_HOME 옵션에 오즈 서버를 구동할 때 연동할 JDK 또는 JRE 루트 경로를 설정합니다. MAX_MEMORY, MIN_MEMORY, JAVA_OPTION을 이용하여 최대, 최소 메모리, 그 외 실행 옵션을 설정합니다.

```
...
set JAVA_HOME=..\..\JRE
set MAX_MEMORY=256
set MIN_MEMORY=128
set JAVA_OPTION=
...
```

■ **라이선스 파일 설치**

발급받은 라이선스 파일을 %OZ_HOME%\license 폴더에 ozlicense.xml 파일로 복사합니다.

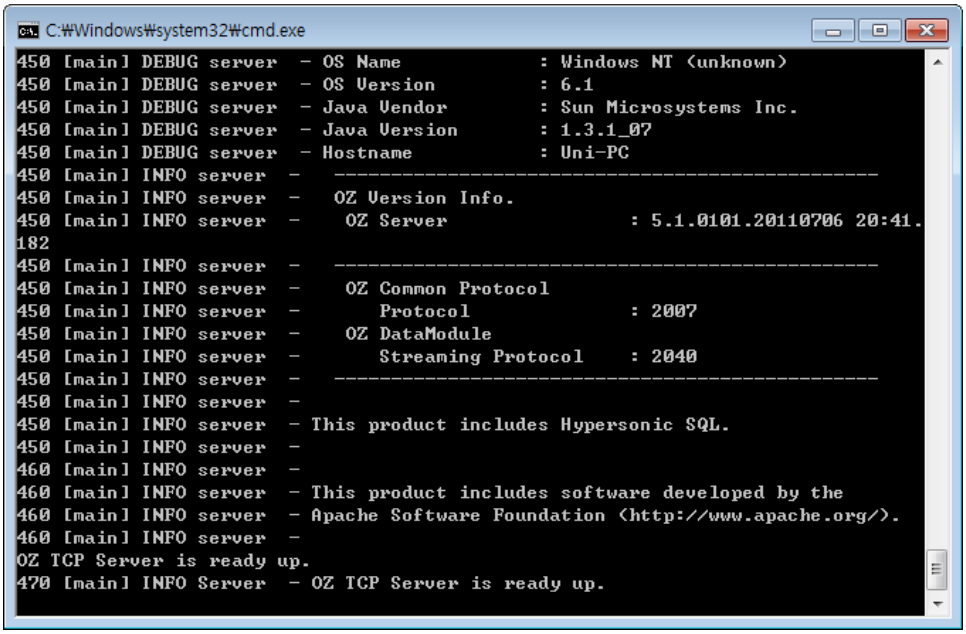
라이선스 파일의 설치 경로와 파일 이름을 변경하려면 %OZ_HOME%\config 폴더에 있는 ozserver.properties 파일의 path.license.file 옵션을 이용하시기 바랍니다.

```
server.portnumber=8003
path.oz.server.home=../

path.license.file=%OZ_HOME%/license/ozlicense.xml
...
```

■ **오즈 서버 실행**

%OZ_HOME%\bin 폴더에 있는 ozstart.bat(유닉스 계열인 경우 ozstart.sh) 파일을 실행합니다.



서블릿 타입 오즈 자바 서버 실행

■ **오즈 서버 경로 설정**

%OZ_HOME%\classes 폴더에 있는 ozserver.properties 파일을 편집 프로그램으로 열어 오즈 서버 홈 경로를 설정합니다.

```
path.oz.server.home=%OZ_RUNNING_PATH%/WEB-INF
...
```

※ 참고사항 : 오즈 서버 홈 경로는 절대 경로 뿐만 아니라 상대 경로로도 설정할 수 있습니다. 경로 설정 시 사용하는 %OZ_RUNNING_PATH%는 오즈 서버가 설치된 루트 경로를 나타내는 예약어입니다.

■ **라이선스 파일 설치**

발급받은 라이선스 파일을 %OZ_HOME%\license 폴더에 ozlicense.xml 파일로 복사합니다.

라이선스 파일의 설치 경로와 파일 이름을 변경하려면 %OZ_HOME%\classes 폴더에 있는

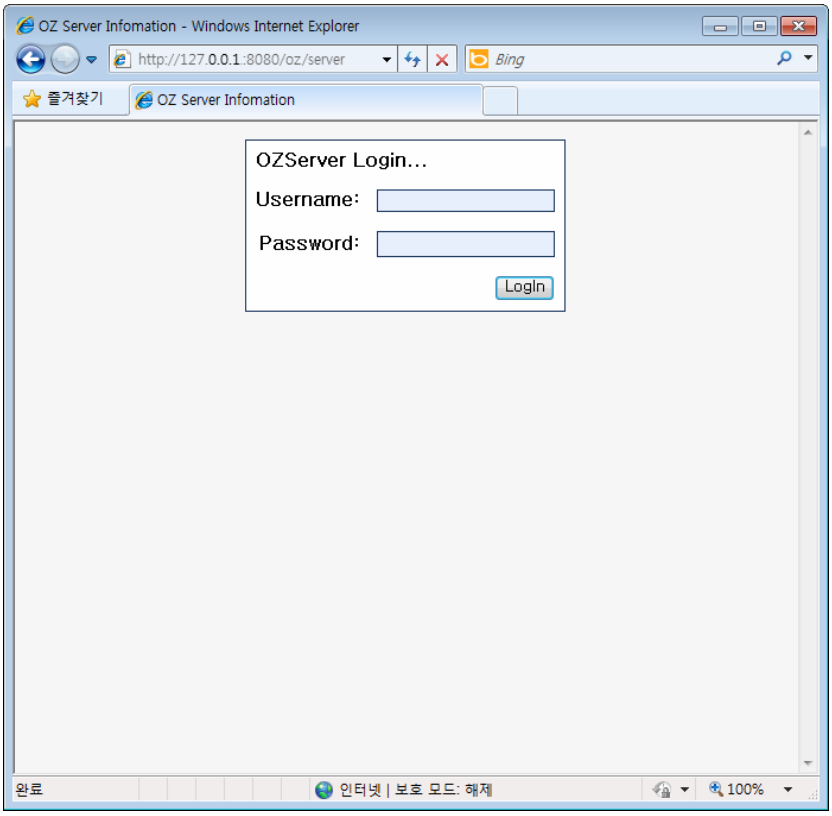
ozserver.properties 파일의 path.license.file 옵션을 이용하시기 바랍니다.

```
path.oz.server.home=%OZ_RUNNING_PATH%/WEB-INF

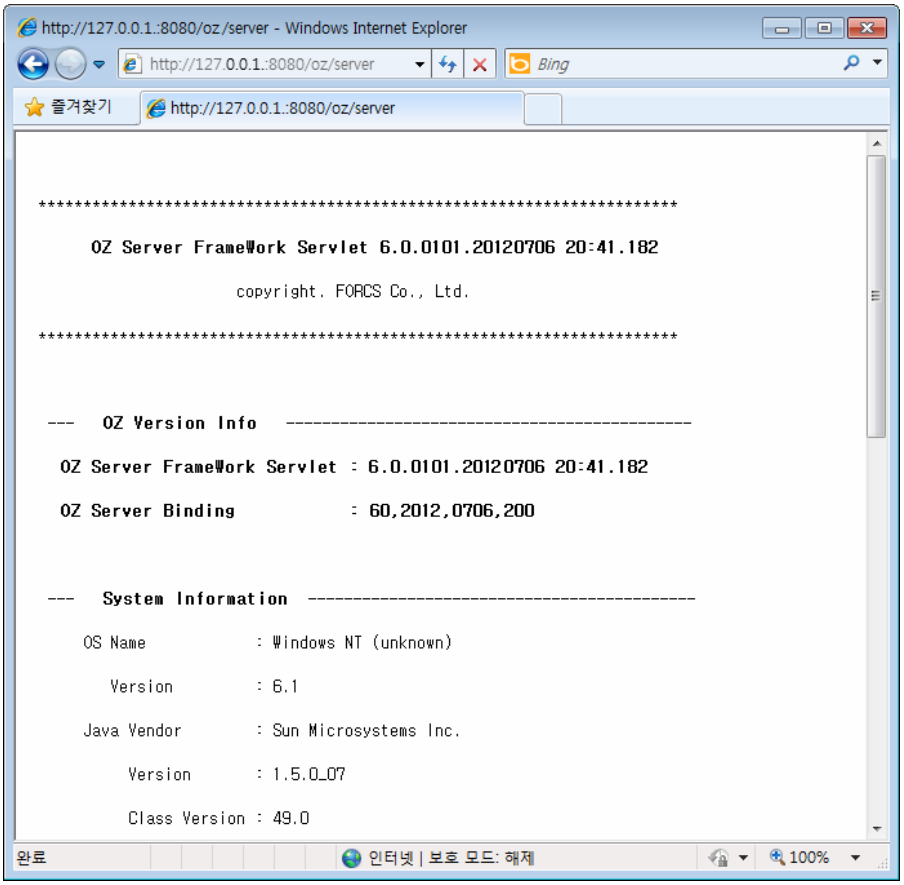
path.license.file=%OZ_HOME%/license/ozlicense.xml
...
```

■ **오즈 서버 실행**

오즈 서버를 WAS에 등록하고 WAS를 구동한 후 오즈 서버 URL을 실행합니다.



Username에는 admin 계정 ID를, Password에는 admin 계정의 패스워드를 입력한 후 [Login] 버튼을 클릭하면 오즈 서버 정보를 나타내는 화면이 표시됩니다.



오즈 서버를 WAS에 등록하는 방법은 WAS 종류마다 내용이 상이하여 본 매뉴얼에서는 별도로 기술하지 않습니다. 자세한 내용은 오즈 기술 지원 센터에 문의하시기 바랍니다.

※ 주의사항 : 서버 바인딩 라이브러리와 JVM 버전

서블릿 타입 오즈 서버가 구동될 때 서버 바인딩 라이브러리가 로딩되는 경우 WAS에 설정된 JVM 버전이 1.5보다 낮으면 에러가 발생합니다.

이럴 경우 서버 바인딩 라이브러리 파일을 삭제하거나 JVM 버전을 1.5 이상으로 설정한 후 WAS를 재시작하시기 바랍니다. 서버 바인딩 라이브러리 파일은 %OZ_HOME%\lib 폴더의 ozrresource.jar, ozviewer_lib.jar, poi-scratchpad-3.0.1.jar, uk.co.mmscomputing.imageio.tiff.jar 파일입니다.

오즈 엔터프라이즈 닷넷 서버 실행

데몬 타입 오즈 닷넷 서버 실행

오즈 닷넷 서버를 데몬 타입으로 실행하려면 먼저 오즈 닷넷 서버를 윈도우 서비스로 등록한 후 윈도우 서비스를 시작합니다.

오즈 닷넷 서버를 윈도우 서비스로 등록하는 방법은 본 매뉴얼의 "오즈 엔터프라이즈 닷넷 서버를 윈도우 서비스로 등록" 부분을 참고하시기 바랍니다.

서블릿 타입 오즈 닷넷 서버 실행

- **오즈 서버 경로 설정**

  %OZ_HOME% 폴더에 있는 ozserver.properties 파일을 편집 프로그램으로 열어 오즈 서버 홈 경로를 절대 경로로 설정합니다.

```
path.oz.server.home=c:/Forcs/OZ Server.NET 6.0
...
```

- **라이선스 파일 설치**

  발급받은 라이선스 파일을 %OZ_HOME%\license 폴더에 ozlicense.xml 파일로 복사합니다.

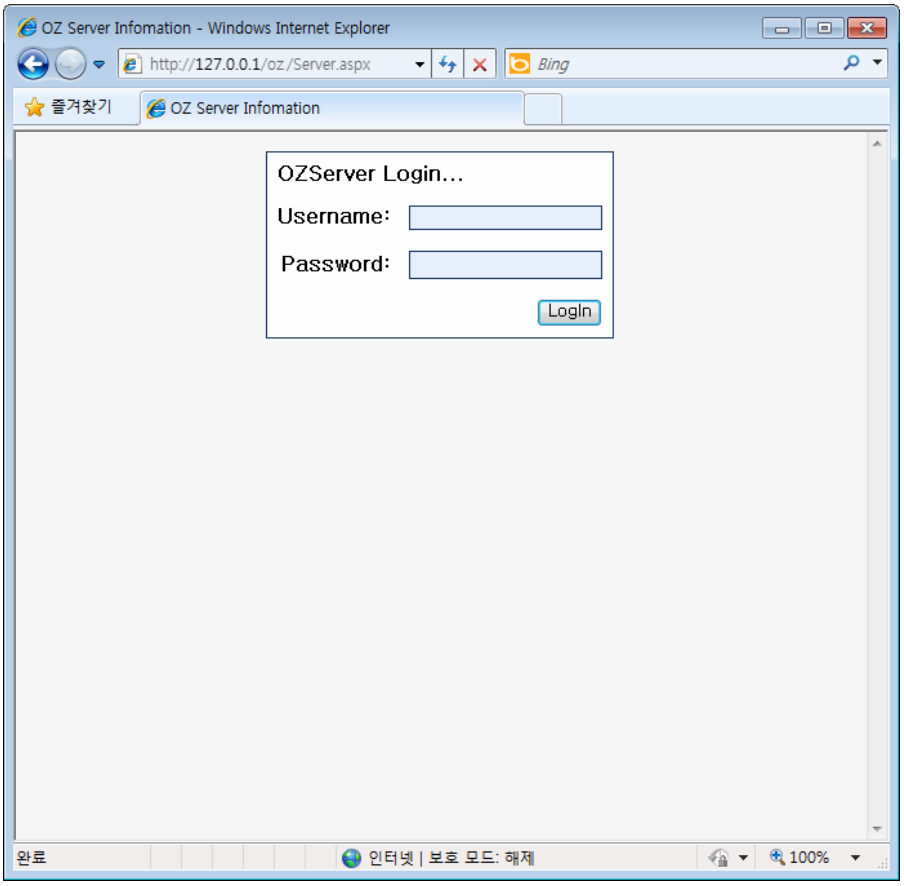
  라이선스 파일의 설치 경로와 파일 이름을 변경하려면 %OZ_HOME% 폴더에 있는 ozserver.properties 파일의 path.license.file 옵션을 이용하시기 바랍니다.

```
path.oz.server.home=c:/Forcs/OZ Server.NET 6.0

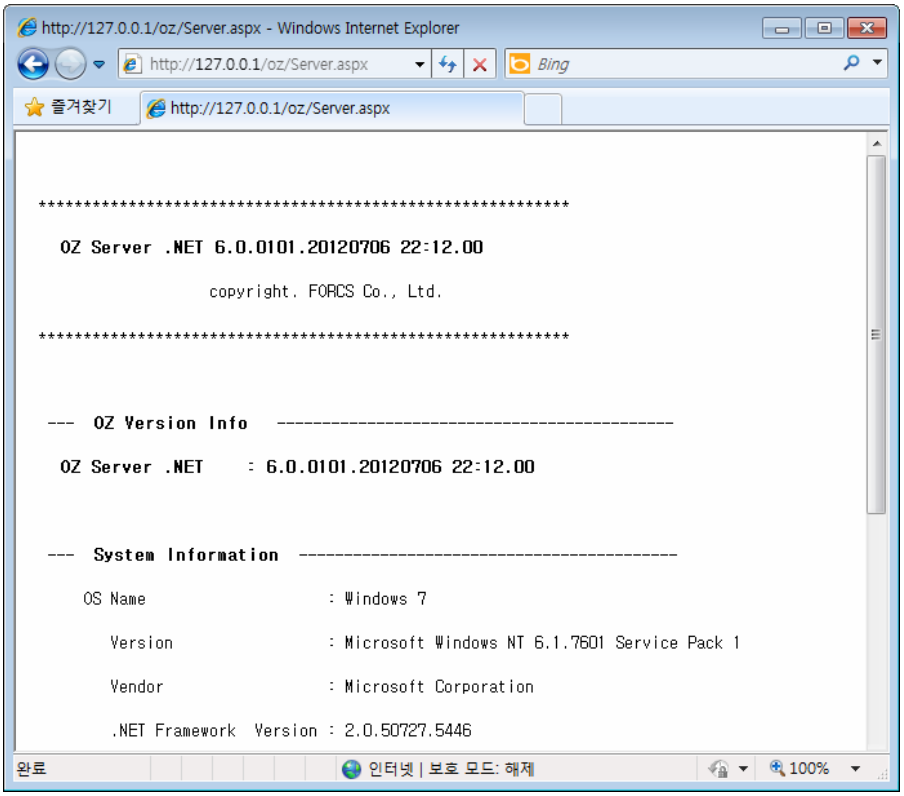
path.license.file=%OZ_HOME%/license/ozlicense.xml
...
```

- **오즈 서버 실행**

  오즈 서버가 설치된 폴더를 IIS의 기본 웹 사이트에 등록하고, ASP.NET 사용자 권한을 설정한 후 IIS를 구동한 후 오즈 서버 URL을 실행합니다.



Username에는 admin 계정의 ID를, Password에는 admin 계정의 패스워드를 입력한 후 [Login] 버튼을 클릭하면 오즈 서버 정보를 나타내는 화면이 표시됩니다.



오즈 서버가 설치된 폴더를 IIS에 등록하고, ASP.NET 사용자 권한을 설정하는 방법은 본 매뉴얼의 오즈 엔터프라이즈 닷넷 서버 실행 환경 설정 부분을 참고하시기 바랍니다.

※ 주의사항 : 서버 바인딩 라이브러리와 .NET Framework 버전

오즈 서버가 구동될 때 서버 바인딩 라이브러리가 로딩되는 경우 .NET Framework 버전이 3.5보다 낮으면 에러가 발생합니다.

이럴 경우 서버 바인딩 라이브러리 파일을 삭제하거나 .NET Framework 버전을 3.5 이상으로 설치한 후 IIS를 재시작하시기 바랍니다. 서버 바인딩 라이브러리 파일은 %OZ_HOME%\bin 폴더의 OZWPFViewer_Lib.dll 파일입니다.

Ⅳ. 오즈 엔터프라이즈 서버 옵션 설정

- assembly.properties
- cachemngr.properties
- clientlist.xml
- databind.properties
- datasource.properties
- db.properties
- dbconfig.xml, dbconfig.xsd
- log.properties
- mail.properties
- monitor.properties
- ozserver.properties
- ozservice.properties
- ozudl.properties
- ozudr.properties
- repository.properties
- sap.properties
- spmgr.properties
- srmngr.properties
- uslmngr.properties
- xml.properties

assembly.properties

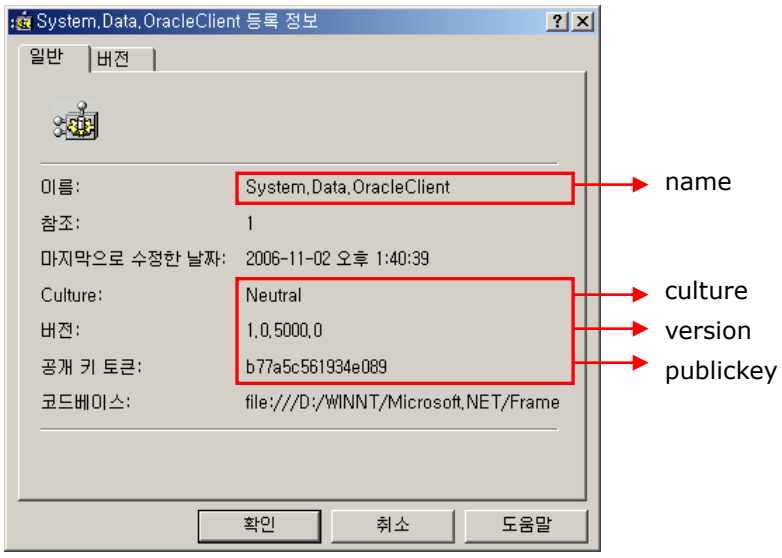
오즈 닷넷 서버에서 사용할 어셈블러 정보를 설정합니다.
어셈블리 파일을 사용할 경우 서버 PC에 어셈블리를 포함하는 클라이언트 툴을 설치하고, 클라이언트 리스너를 구동시켜야 합니다.

assembly.properties 옵션

옵션	설명
home	어셈블리 파일의 홈 경로를 설정합니다. ※ 주의사항 : 오즈 서버가 구동될 때 홈으로 설정한 디렉토리를 생성하지 못할 경우 오즈 서버가 중지됩니다.
alias.filepath	어셈블리 파일 경로를 절대 경로 또는 상대 경로로 설정합니다. ※ 주의사항 : filepath를 설정할 경우 name, version, culture, publickey에 설정된 값은 무시됩니다.
alias.name	어셈블리 이름을 설정합니다.
alias.version	어셈블리 버전을 설정합니다.
alias.culture	어셈블리 문화권 또는 언어 정보를 설정합니다.
alias.publickey	어셈블리 공개 키 토큰을 설정합니다.
e.g.sessionusl.filepath	USL 어셈블리 파일 즉, OZUSLServer.dll 파일의 경로를 절대 경로로 설정합니다. ※ 참고사항 : OZUSLServer.dll 파일이 닷넷 서버가 설치된 경로의 /bin 폴더가 아닌 다른 경로에 있을 경우에만 해당 옵션을 설정하시기 바랍니다.
<ProtocolVersion>.isexternal module	어셈블리가 외부 모듈인지 여부를 설정합니다.
<ProtocolVersion>.active	외부 모듈의 사용 여부를 설정합니다.

<ProtocolVersion>.description	외부 모듈의 프로토콜 설명을 설정합니다. ※ 참고사항 : <ProtocolVersion>에서 설정할 수 있는 프로토콜 대역폭은 30001 ~ 39999 사이의 값입니다.
<ProtocolVersion>.filepath	외부 모듈의 파일 경로를 절대 경로 또는 상대 경로로 설정합니다. ※ 주의사항 : filepath를 설정할 경우 name, version, culture, publickey에 설정된 값은 무시됩니다.
<ProtocolVersion>.name	외부 모듈 이름을 설정합니다.
<ProtocolVersion>.version	외부 모듈 버전을 설정합니다.
<ProtocolVersion>.culture	외부 모듈 문화권 또는 언어 정보를 설정합니다.
<ProtocolVersion>.publickey	외부 모듈 공개 키 토큰을 설정합니다.
<ProtocolVersion>.classname	외부 모듈을 구현한 클래스 이름을 설정합니다.

※ 참고사항 : assembly.properties 파일에 설정할 어셈블리 옵션은 등록 정보 창에서 확인할 수 있습니다.



DBMS별 assembly.properties 설정 예

■ MS-SQL

```
#
```

```
#MSSQL ADO.NET Provider
#
mssql_DB.name=System.Data
mssql_DB.version=1.0.50000.0
mssql_DB.culture=Neutral
mssql_DB.publickey=b77a5c561934e089
```

※ 주의사항 : db.properties에 사용할 어셈블리를 설정해야 합니다.

ex) mssql_example.assembly= mssql_DB

- **Oracle**

```
#
#Microsoft Oracle ADO.NET Provider
#
oracle_DB.name= System.Data.OracleClient
oracle_DB.version=1.0.5000.0
oracle_DB.culture=Neutral
oracle_DB.publickey=b77a5c561934e089
```

※ 주의사항 : db.properties에 사용할 어셈블리를 설정해야 합니다.

ex) oracle_example.assembly=oracle_DB

- **UDS**

```
uds1.filepath=%OZ_HOME%/assembly/UDSSample.dll
```

cachemngr.properties

캐시 매니저와 관련된 정보를 설정합니다.
여러 명의 사용자가 동시에 연속적으로 동일한 내용의 보고서, 애플리케이션 폼, 데이터 모듈을 요청하는 경우 캐시 기능을 사용하여 속도 개선 효과를 기할 수 있습니다.

cachemngr.properties 옵션

옵션	설명
CACHE_FILE_PATH	캐시 저장 경로를 설정합니다.
DM_CACHE_FILE_PATH	데이터 모듈의 캐시 대상 저장 경로를 설정합니다.
대상.active	캐시 여부를 설정합니다.
대상.memoryCacheValidTime	캐시를 메모리에 저장할 때 보관하는 시간을 설정합니다.(단위:초)
대상.freeMemoryPercentage	캐시를 메모리에 저장할 때 메모리 여유 용량을 설정합니다.(단위:퍼센트) 시스템의 메모리 여유 용량이 설정한 값보다 작으면 디스크로 캐시합니다.
대상.diskCacheValidTime	캐시를 디스크에 저장할 때 보관하는 시간를 설정합니다.(단위:초)
serverbind.active	서버 바인딩 기능이 실행될 때 캐시 여부를 설정합니다.
serverbind.validtime	서버 바인딩 기능으로 생성된 캐시를 보관하는 시간을 설정합니다.(단위:초)

대상 설정 방법

■ 리파지토리

- 전체 리파지토리 캐시

```
repository.active = true
```

- 특정 카테고리 캐시(카테고리가 "/sales/car"인 경우)

```
repository.[/sales/car].active = true
```

- 특정 파일 캐시(카테고리가 "/sales/car", 파일 이름이 "dept1.ozr"인 경우)

```
repository.[/sales/car].[dept1.ozr].active = true
```

■ **데이터 모듈**

데이터 모듈을 캐시할 때에는 같은 데이터 모듈이어도 오즈 리포트와 오즈 애플리케이션에서 각각 설정하여야 합니다. 오즈 애플리케이션은 데이터 셋 단위까지 설정할 수 있습니다.

- 전체 데이터 모듈 캐시
  - 오즈 리포트

```
datamodule.report.active = true
```

  - 오즈 애플리케이션

```
datamodule.application.active = true
```

- 특정 카테고리 캐시(카테고리가 "/sales/car"인 경우)
  - 오즈 리포트

```
datamodule.report.[/sales/car].active = true
```

  - 오즈 애플리케이션

```
datamodule.application.[/sales/car].active = true
```

- 특정 파일 캐시(카테고리가 "/sales/car", 파일 이름이 "dept1.odi"인 경우)
  - 오즈 리포트

```
datamodule.report.[/sales/car].[dept1.odi].active = true
```

  - 오즈 애플리케이션

```
datamodule.application.[/sales/car].[dept1.odi].[DataSet_1].[0]
.active = true
```

- Fetch 타입, 패러미터 값 별로 특정 파일 캐시(카테고리가 "/sales/car", 파일 이름이 "dept1.odi", Fetch 타입이 "DM_BATCH_FETCH", 패러미터 이름이 "Code", 패러미터 값이 "A001"인 경우)
  - 오즈 리포트

```
datamodule.report.[/sales/car].[dept1.odi].[DM_BATCH_FETCH].[Co
de=A001].active = true
```

  - 오즈 애플리케이션

```
datamodule.application.[/sales/car].[dept1.odi].[DataSet_1].[0]
.[DM_BATCH_FETCH].[Code=A001].active = true
```

clientlist.xml

오즈 뷰어에서 OZA, OZR 파일을 호출하거나 오즈 스케줄러에서 뷰어를 이용하여 파일로 익스포트할 때 사용자의 Mac Address, IP Address, HDD Serial Number의 정보를 체크하여 사용자가 요청한 기능을 실행시킬지 여부를 설정합니다.

해당 기능은 자주 사용하지 않는 기능으로, ozserver.properties 파일에 clientlist.xml 파일이 등록되어있지 않습니다. 해당 기능을 사용하려면 clientlist.xml 파일의 옵션 설정과 별개로 ozserver.properties 파일에 아래와 같은 형태로 clientlist.xml 파일의 경로를 추가하시기 바랍니다.

```
path.clientinfo.properties.file=%OZ_HOME%/conf/clientlist.xml
```

clientlist.xml 옵션

옵션	설명
ClientList use	사용자 정보(MAC Address, IP Address, HDD Serial Number)를 체크하여 사용자가 요청한 기능을 실행시킬지 여부를 설정합니다.
Client checkAll	사용자의 MAC Address, IP Address, HDD Serial Number정보를 모두 체크할지 여부를 설정합니다. true인 경우 세 가지 정보가 모두 일치할 경우에만 기능을 실행하고, false인 경우 하나라도 일치하면 기능을 실행합니다.
MACAddress checkAll	<Item>에 입력된 모든 MAC Address와 체크할지 여부를 설정합니다. true인 경우 <Item>에 입력된 모든 MAC Address와 사용자의 MAC Address가 일치할 경우에만 기능을 실행하고, false인 경우 하나라도 일치하면 기능을 실행합니다.
IPAddress checkAll	<Item>에 입력된 모든 IP Address와 체크할지 여부를 설정합니다. true인 경우 <Item>에 입력된 모든 IP Address와 사용자의 IP Address가 일치할 경우에만 기능을 실행하고, false인 경우 하나라도 일치하면 기능을 실행합니다.

HDDSerial checkAll	<Item>에 입력된 모든 HDD Serial Number와 체크할지 여부를 설정합니다. true인 경우 <Item>에 입력된 모든 HDD Serial Number와 사용자의 HDD Serial Number가 일치할 경우에만 기능을 실행하고, false인 경우 하나라도 일치하면 기능을 실행합니다.
Item	기능 실행을 허용하는 Mac Address, IP Address, HDD Serial Number를 설정합니다.

주의사항

- 오즈 자바 서버의 경우 JDK 1.4.1 버전을 사용할 경우 해당 기능이 지원되지 않습니다.
- 옵션 이름은 대소문자를 구분합니다. 예를 들어, "HDDSerial"을 "HddSerial"로 입력하면 에러가 발생합니다.
- clientlist.xml 파일을 로딩할 때 에러가 난 경우 서버가 중지되오니 사용 시 주의바랍니다.

databind.properties

데이터 바인딩 옵션을 설정합니다.

databind.properties 옵션

옵션	설명
ConcurrentFetchSize	데이터를 Concurrent 타입으로 뷰어로 전송할 때 Stream 버퍼의 크기를 설정합니다.(단위:byte) 음수 또는 잘못된 값으로 설정하면 기본 값(4096)으로 적용되고, 최소 값보다 작은 양수 값으로 설정하면 최소 값(256)으로 적용됩니다.
ConcurrentFirstRow	데이터를 Concurrent 타입으로 뷰어로 전송할 때 맨 처음 전송되는 데이터의 행 수를 설정합니다. 0보다 작은 값으로 설정하면 기본 값(0)으로 적용됩니다.
NumericDecimalDisplayZero	데이터 값이 1보다 작을 경우 소수점 앞에 0을 추가할지 여부를 설정합니다. DB 서버 또는 ODI의 데이터 트리에 설정된 속성 값에 따라 적용하려면 "default"로, 무조건 0을 추가하려면 "true"로, 0을 추가하지 않으려면 "false"로 설정하시기 바랍니다. ※ 주의사항 • 오즈 자바 서버에서만 설정 가능합니다. • 필드 타입이 "Numeric", "Decimal"인 경우에만 적용됩니다.

datasource.properties

XML, SOAP, 파일, HTTP 스토어의 데이터 소스 정보를 설정합니다.

datasource.properties 옵션

옵션	설명
alias.CHARACTERSET	CSV 데이터 셋의 문자 셋을 설정합니다. 설정하지 않을 경우 시스템에 설정된 값으로 적용됩니다.
alias.CHECKERROR	CSV 데이터 셋의 에러 표시 코드를 설정합니다.
alias.DATASETNODENAME	XML 데이터 셋의 데이터 셋 노드 이름을 설정합니다.
alias.DESIGNTIMEFILENAME	CSV/XML 데이터 셋의 디자인 시 사용할 임시 CSV/XML 파일 경로를 설정합니다.
alias.DOPOST	HTTP 스토어의 포스트 타입 전송 여부를 설정합니다.
alias.ENDPOINT	SOAP 스토어의 웹 서비스 호출 지점을 설정합니다.
alias.ERRORRETRYCOUNT	HTTP 스토어의 재시도 횟수를 설정합니다.
alias.FIELDDELIMETER	CSV 데이터 셋의 필드 구분자를 설정합니다.
alias.FILENAME	파일 스토어의 파일 경로를 설정합니다.
alias.FILENAMETOLOG	XML 스토어와 HTTP 스토어에 추가된 CSV/XML 데이터 셋의 서버 로그 파일 이름을 설정합니다.
alias.HASFIELDTYPE	CSV 데이터 셋의 필드 타입 정보 포함 여부를 설정합니다.
alias.ID	SOAP 스토어의 웹 서비스가 HTTP 인증을 사용하는 경우 사용자 ID를 설정합니다.
alias.LINEDELIMITER	CSV 데이터 셋의 라인 구분자를 설정합니다.
alias.NULLMARK	CSV 데이터 셋의 널 마크를 설정합니다.
alias.PASSWORD	SOAP 스토어의 웹 서비스가 HTTP 인증을 사용하는 경우 사용자 패스워드를 설정합니다.
alias.RECORDNODENAME	XML 데이터 셋의 레코드 셋 노드 이름을 설정합니다.
alias.SAMPLEXMLURL	XML 스토어의 디자인 시 사용할 임시 XML 파일 경로를 설정합니다.
alias.SCHEMAPATH	XML 스토어의 디자인 시 사용할 정의 파일 경로(DTD 또는 XSD)를 설정합니다.

alias.STARTINDEX	다이나믹 데이터 셋의 시작 행 번호를 설정합니다.
alias.URL	HTTP 스토어의 사용자 애플리케이션 주소(URL)을 설정합니다.
alias.URLENCODE	HTTP 스토어의 URL 인코드 설정 여부를 설정합니다.
alias.VENDOR	벤더를 설정합니다.
alias.WSDLURL	SOAP 스토어의 WSDL 위치를 설정합니다.
alias.XMLURL	XML 스토어의 XML 주소를 설정합니다.
alias.XMLPARSER	XML 스토어의 XML 파서를 설정합니다.

주의사항

- 옵션 값을 사용자 지정 패러미터로 설정할 경우 DB 접속 정보를 앨리어스로 설정한 ODI에 같은 이름의 사용자 지정 패러미터가 추가되어있어야 합니다.
- 파일 경로는 상대 경로와 OZ URL(ozp://Category/FileName)로 설정할 수 있습니다. 상대 경로로 설정할 경우 디자인 시에는 디자이너 설치 폴더를 기준으로한 상대 경로에 파일을 위치시키고, 운영 시에는 오즈 서버 HOME 경로를 기준으로한 상대 경로에 파일을 위치시켜야 합니다. OZ URL 형태로 설정할 경우에는 리파지토리 루트를 기준으로 한 카테고리 경로에 맞게 파일을 위치시켜야 합니다.
- datasource.properties 파일 저장 시 시스템에서 사용하는 인코딩으로 설정한 후 저장하여야 하며, 한글을 지원하지 않는 OS일 경우 파일 안에 있는 한글을 삭제한 후 저장하시기 바랍니다.

db.properties

오즈 서버가 사용할 데이터베이스의 연결 정보를 설정합니다.

오즈 서버는 서버 성능 향상을 위해 서버 시작 시에 DB Connection Pool을 생성하고, 이를 공유하여 사용할 수 있도록 하는 DB Connection Pool을 지원합니다.

db.properties 옵션

옵션	설명
alias.dbName	데이터베이스 이름을 설정합니다.
alias.decodecharset	디코딩 문자 셋을 설정합니다.
alias.doConnectionCheck	Connection Pool에서 Connection을 가져올 때마다 테스트 쿼리문를 실행하여 에러가 발생하면 Connection을 새로 생성할지 여부를 설정합니다.
alias.doConnectionCheckMaxTime	Connection 상태를 체크하는 최대 시간을 설정합니다.(단위:초) Connection을 사용한 최근 시점이 doConnectionCheckMaxTime 옵션 값보다 큰 경우 doConnectionCheck 옵션 값과 상관 없이 Connection을 새로 생성합니다.
alias.doConnectionCheckMinTime	Connection 상태를 체크하는 최소 시간을 설정합니다.(단위:초) doConnectionCheck 옵션 값이 true인 경우 Connection을 사용한 최근 시점이 doConnectionCheckMinTime보다 작은 경우 기존 Connection을 사용하고, doConnectionCheckMinTime보다 크고 doConnectionCheckMaxTime보다 작은 경우 testQueryString을 사용하여 Connection 여부를 체크합니다.
alias.driver	JDBC/ADO.NET 드라이버의 클래스 이름을 설정합니다.
alias.dsn	ODBC 데이터베이스의 소스 이름을 설정합니다.
alias.encodecharset	인코딩 문자 셋을 설정합니다.

alias.initconns	오즈 서버의 초기 Connection 개수를 설정합니다.
alias.maxconns	오즈 서버가 가질 수 있는 최대 Connnection 개수를 설정합니다.
alias.password	데이터베이스에 접속할 사용자의 패스워드를 설정합니다. db.properties 파일이 외부에 유출될 가능성이 있을 경우 ozencrypted.bat 파일을 이용하여 패스워드를 암호화한 후 설정하시기 바랍니다.
alias.portNo	데이터베이스 포트 번호를 설정합니다.
alias.props	JDBC/ADO.NET 드라이버의 Connection Property String을 설정합니다.(User/Password 설정)
alias.serverAddress	데이터베이스 서버 주소를 설정합니다.
alias.serverName	데이터베이스 서버 이름을 설정합니다.
alias.sessionQuery	모니터 로그에서 사용하는 Session ID를 가져오기 위한 쿼리문을 설정합니다.
alias.sid	Oracle 데이터베이스의 SID를 설정합니다.
alias.testQueryString	Connection Pool에서 Connection을 가져올 때마다 수행할 테스트 쿼리문을 설정합니다.
alias.timeout	최대 Connnection 개수를 초과했을 때 사용 중인 Connnection이 반환되기를 기다리는 최대 시간을 설정합니다.(단위:초)
alias.querytimeout	쿼리 실행 제한 시간을 설정합니다.(단위:초) 특정 쿼리의 실행 속도가 늦어져 세션이 홀딩되고 시스템의 과부하로 인해 다른 쿼리의 응답 속도도 느려지는 문제를 개선하려면 querytimeout 옵션을 0보다 큰 값으로 적절히 설정하시기 바랍니다.
alias.url	JDBC/ADO.NET 드라이버의 연결 URL을 설정합니다.
alias.user	데이터베이스에 접속할 사용자 이름을 설정합니다. db.properties 파일이 외부에 유출될 가능성이 있을 경우 ozencrypted.bat 파일을 이용하여 사용자 이름을 암호화한 후 설정하시기 바랍니다.
alias.vendor	데이터베이스 벤더를 설정합니다.
alias.fetchrow	JDBC 전송 ROW 즉, 결과 셋 반환 시 한꺼번에 가져올 수 있는 행 수를 설정합니다.
alias.ignoreQueryError	쿼리 실행 에러 발생 시 필드 정보만 있고 결과 셋은 없는 빈 데이터 셋을 전송할지 여부를 설정합니다.

주의사항

- 인코딩/디코딩 문자 셋을 설정하지 않은 경우 오즈 서버의 환경에 따라 적용됩니다.
- 인코딩/디코딩 문자 셋을 같은 값으로 설정한 경우 문자 셋이 적용되지 않습니다.
- 인코딩/디코딩 문자 셋을 쿼리 디자이너와 db.properties에서 모두 설정한 경우 쿼리 디자이너에서 설정한 값으로 적용됩니다. 단, 쿼리 디자이너에서 설정한 인코딩/디코딩 문자 셋이 잘못되었거나 같은 값인 경우 db.properties에 설정된 값으로 적용됩니다.
- NT 서비스로 등록하여 ODBC를 사용할 경우 시스템 데이터 원본에 등록하여야 사용할 수 있습니다.

sessionQuery 작성 예

session ID를 얻는 쿼리가 있는 모든 데이터베이스는 db.properties의 alias.sessionQuery에 세션 ID를 달리 등록해야 합니다. 쿼리 설정이 없을 경우 session ID가 공백 문자로 로깅됩니다.

```
alias.sessionQuery = "[DB별 Query]"

<infomix>
SELECT DBINFO(\"sessionid\") FROM SysTables WHERE TabID = 1

<mssql>
select @@spid

<oracle>
SELECT sid FROM V$SESSION WHERE audsid = userenv('sessionid')

<sybase>
select @@spid
```

dbconfig.xml, dbconfig.xsd

각 DBMS에서 사용할 JDBC/ADO.NET 드라이버를 설정합니다.

주의사항

- 오즈 서버와 연동하는 모든 오즈 제품에서 동일한 dbconfig.xml을 사용해야 합니다.
- VENDOR 이름은 중복될 수 없으며 모두 대문자로 인식합니다. 중복하여 정의할 경우 나중에 정의된 VENDOR를 사용합니다.
- 데이터베이스의 User ID와 Password 값은 ITEM 뿐만 아니라 뷰어 패러미터를 이용하여 설정할 수 있습니다. ITEM과 패러미터 모두 정의할 경우 패러미터로 적용됩니다.
- ITEM의 KEY 값은 대/소문자를 구분합니다.
- ITEM의 KEY 값은 영문 알파벳, 정수형 숫자, 그 밖의 기호(~, !, @, #, $, %, ^, *, (), _, +, |, \, =, -, {}, [], :, ;, ., ,, /, ?)로 설정하여야 합니다.
- ITEM의 KEY 값으로 사용할 수 없는 단어는 아래와 같습니다.

  alias, vendor, key, user, password, initconns, maxconns, timeout, initSqls, closeSqls, doAutoCommit, doConnectionCheck, testQueryString, sessionQuery, _encrypted로 끝나는 문자열, NAME, INCLUDE, VENDOR, USERNAME, ENCYPW, PASSWORD, USEALIAS, POOLALIAS, ALIASFILENAME, AUTOCOMMIT, ENCODECHARSET, DECODECHARSET, TIMEFORMAT, DATEFORMAT, TIMESTAMPFORMAT
- 서버 리파지토리 타입이 RDB일 경우 Vendor별로 ADDITIONAL을 추가하여 아래와 같이 DATE_EXPRESSION을 설정해야합니다. 값을 지정하지 않거나 Vendor가 설정한 포맷을 지원하지 않을 경우 에러가 발생합니다.

Vendor	DATE Type	DATE_EXPRESSION
ORACLE	DATE	TO_DATE('@YYYY/@MM/@DD @hh:@mm:@ss', 'YYYY/MM/DD hh24:mi:ss')
MSSQL	DATETIME	CAST('@YYYY.@MM.@DD @hh:@mm:@ss' AS DATETIME)
DB2	TIMESTAMP	TO_DATE('@YYYY-@MM-@DD @hh:@mm:@ss', 'YYYY-MM-DD HH24:MI:SS')
SYBASE	DATETIME	CONVERT(DATETIME, '@YYYY.@MM.@DD @hh:@mm:@ss')

기본 값	DATE	TO_DATE('@YYYY/@MM/@DD @hh:@mm:@ss', 'YYYY/MM/DD hh24:mi:ss')

- 오즈 닷넷 서버에서 마스터-디테일 형태의 RDB 스토어 바인딩 시 하나의 커넥션을 사용할 커넥션 사용량을 최소화하기 위해서는 MARS(Multiple Active Result Sets) 옵션을 설정하여야 합니다.

  이 기능은 ORACLE인 경우는 .NET Framework 1.1 이상, MSSQL 2005인 경우는 .NET Framework 2.0 이상에서 지원됩니다. 또한 MSSQL 2005에서 해당 기능을 사용하기 위해서는 mssql2005_mars 연결 정보를 사용하여야 합니다.

  - ORACLE

    MARS ACTIVE를 true로 설정하고 MAJOR_VESRION과 MINOR_VERSION을 .NET Framework 1.1로 설정합니다.

    ```
    <MARS ACTIVE="true" MAJOR_VERSION="1" MINOR_VERSION="1"/>
    ```

  - MSSQL 2005

    MARS ACTIVE를 true로 설정하고 MAJOR_VESRION과 MINOR_VERSION을 .NET Framework 2.0으로 설정합니다.

    ```
    <MARS ACTIVE="true" MAJOR_VERSION="2" MINOR_VERSION="0"/>
    ```

log.properties

로그의 기록 형태와 방법을 설정합니다.
서버 에러, 서버 동작 상황, 트랜잭션 정보 뿐만 아니라 HTTP 스토어 데이터 셋, XML 스토어를 바인딩하는 중 발생한 에러에 대해서도 로그를 기록할 수 있습니다.
로그는 콘솔과 파일 형태로 기록할 수 있으며, 윈도우 NT 이벤트와 유닉스 시스템 로그 및 메일로 전송할 수 있습니다.

로그 레벨 옵션

옵션	설명	
Priority	로그 레벨을 설정합니다. 로그 레벨을 낮은 단계에서 높은 단계로 정렬하면 DEBUG, INFO, WARN, ERROR 순서입니다.	
	DEBUG	실제 에러, 서버의 트랜잭션 정보 뿐만 아니라 전반적인 서버 동작 상황을 로그에 기록
	ERROR	실제 에러 상황에서만 로그에 기록
	INFO	실제 에러와 서버의 트랜잭션 정보를 로그에 기록
	WARN	실제 에러와 경고 수준의 에러를 로그에 기록

로그 실행 옵션

■ CONSOLE(자바 서버만 지원)

옵션	설명	
CONSOLE.Type	실행 타입을 설정합니다.	
	Stdout	로그를 표준 입출력 장치에 출력
	RollingFile	로그를 파일에 출력
CONSOLE.Layout	로그의 표시 형태를 설정합니다. 표시 형태에 설정할 수 있는 문자는 아래 "로그 Layout 문자"를 참고하시기 바랍니다.	
CONSOLE.Async	버퍼링 여부를 설정합니다.	

CONSOLE.File	로그를 파일에 출력할 때 파일의 경로와 이름을 설정합니다.

■ ROLLING_FILE

옵션	설명	
ROLLING_FILE.Type	로그 파일의 생성 타입을 설정합니다.	
	DailyRollingFile	ROLLING_FILE.DatePattern 옵션에 설정한 대로 로그 파일 생성
	RollingFile	ROLLING_FILE.File 옵션에서 설정한 파일 이름으로 로그 파일 생성
ROLLING_FILE.DatePattern	로그 파일 이름의 날짜 패턴을 설정합니다. 날짜 패턴에 설정할 수 있는 문자는 아래 "날짜 패턴 문자"를 참고하시기 바랍니다.	
ROLLING_FILE.Layout	로그의 표시 형태를 설정합니다. 표시 형태에 설정할 수 있는 문자는 아래 "로그 Layout 문자"를 참고하시기 바랍니다.	
ROLLING_FILE.Append	로그 이어쓰기 여부를 설정합니다. 서버가 재시작될 때 ROLLING_FILE.File 옵션에 설정한 파일에 새로운 로그를 기록할 때 ROLLING_FILE.File의 기존 내용에 이어서 기록할 때에는 true로, 기존 내용을 지우고 기록할 때에는 false로 설정하시기 바랍니다.	
ROLLING_FILE.MaxBackupIndex	백업되는 로그 파일의 일련 번호 최대 값을 설정합니다. 백업 파일의 일련 번호가 최대 값을 넘길 경우 가장 오래된 파일이 삭제됩니다.	
ROLLING_FILE.MaxFilesize	로그 파일의 최대 크기를 설정합니다. 로그 파일의 크기가 최대 크기를 넘길 경우 현재 파일은 일련 번호를 추가하여 다른 이름으로 변경하고, 다시 로그를 기록합니다. 예를 들어, 로그 파일 이름이 server.log인 경우 server.log.1로 파일 이름을 변경하고 server.log 파일을 새로 만들어 로그를 기록합니다.	
ROLLING_FILE.ImmediateFlush	버퍼에 있는 로그를 로그 파일에 바로 추가할지 여부를 설정합니다.	
ROLLING_FILE.File	로그 파일의 경로와 이름을 설정합니다.	

■ NT_EVENT(자바 서버만 지원)

옵션	설명

NT_EVENT.Type	실행 타입을 설정합니다. 해당 기능을 사용하려면 오즈 서버의 lib 폴더에 있는 NTEventLogAppender.dll 파일을 윈도우 시스템 폴더로 이동시키기 바랍니다.	
	NTEvent	로그를 윈도우 NT 이벤트에 등록
NT_EVENT.Layout	로그의 표시 형태를 설정합니다. 표시 형태에 설정할 수 있는 문자는 아래 "로그 Layout 문자"를 참고하시기 바랍니다.	
NT_EVENT.Threshold	윈도우 NT 이벤트로 보낼 로그 레벨을 설정합니다. Priority 옵션에서 설정한 값보다 높은 단계의 라벨로 설정하시기 바랍니다.	

■ SYSLOG(자바 서버만 지원)

옵션	설명	
SYSLOG.Type	실행 타입을 설정합니다.	
	Syslog	로그를 유닉스 시스템 로그로 전송
SYSLOG.Layout	로그의 표시 형태를 설정합니다. 표시 형태에 설정할 수 있는 문자는 아래 "로그 Layout 문자"를 참고하시기 바랍니다.	
SYSLOG.Threshold	유닉스 시스템 로그로 전송할 로그 레벨을 설정합니다. Priority 옵션에서 설정한 값보다 높은 단계의 라벨로 설정하시기 바랍니다.	
SYSLOG.SyslogHost	유닉스 호스트 IP를 설정합니다.	
SYSLOG.Facility	해당 메시지가 발생하는 Facility를 설정합니다.	
SYSLOG.FacilityPrinting	로그에 Facility 추가 여부를 설정합니다.	

■ HTTPSTORE_LOG_FILE

옵션	설명	
HTTPSTORE_LOG_FILE	ODI에 추가된 HTTP 스토어의 데이터 셋을 바인딩하는 중 발생한 에러에 대한 로그 기록 옵션을 설정합니다.	
	AUTO	모든 HTTP 스토어 데이터 셋에 대해 로그를 기록 데이터 셋 디자인 시 서버 로그 파일 이름을 설정하지 않은 데이터 셋은 "OZ_HTTPSTATMENT.LOG.#" 형태로 로그 파일을 생성(#은 1부터 시작되는 일련 번호)

	MANUAL	HTTP 스토어 데이터 셋 중 서버 로그 파일 이름이 설정된 데이터 셋만 로그를 기록(쿼리 디자이너의 HTTP 스토어 데이터 셋 추가/수정 다이얼로그 참고)
	NONE	HTTP 스토어 데이터 셋에 대한 로그를 기록 안 함
HTTPSTORE_LOG_FILE.LOCATION	로그 파일의 저장 경로를 설정합니다.	

■ **XMLSTORE_LOG_FILE**

옵션	설명	
XMLSTORE_LOG_FILE	ODI에 추가된 XML 스토어에서 XML 주소가 http:// 또는 https://일 때 파싱 에러에 대한 로그 기록 옵션을 설정합니다.	
	AUTO	XML 주소가 http:// 또는 https://인 모든 XML 스토어에 대해 로그를 기록 스토어 디자인 시 서버 로그 파일 이름을 설정하지 않은 스토어는 "OZHTTP_XPATH.LOG.#" 형태로 로그 파일을 생성(#은 1부터 시작되는 일련 번호)
	MANUAL	XML 주소가 http:// 또는 https://인 XML 스토어 중 서버 로그 파일 이름이 설정된 스토어만 로그를 기록(쿼리 디자이너의 XML 스토어 추가/수정 다이얼로그 참고)
	NONE	XML 스토어에 대한 로그를 기록 안 함
XMLSTORE_LOG_FILE.MAXCOUNT	백업되는 로그 파일의 최대 개수를 설정합니다.	
HTTPSTORE_LOG_FILE.LOCATION	로그 파일의 저장 경로를 설정합니다.	

■ **BUILTIN_LOGGING(자바 서버만 지원)**

BUILTIN_LOGGING은 오즈 서버 로그 처리를 위해 자체 개발한 로깅 엔진입니다. 고객 사이트의 특정 상황에 의해 오즈 서버에서 Log4j 방식으로 로그를 남길 수 없는 경우 BUILTIN_LOGGING 방식으로 사용하시기 바랍니다.

옵션	설명
USE_BUILTIN_LOGGING_ENGINE	BUILTIN_LOGGING 방식의 사용 여부를 설정합니다.

<table>
<tr><td>BUILTIN_LOGGING_CONSOLE</td><td colspan="2">로그를 콘솔에 출력할지 여부를 설정합니다.</td></tr>
<tr><td>BUILTIN_LOGGING_CONSOLE.CharSet</td><td colspan="2">콘솔에 출력될 로그 내용의 문자 셋을 설정합니다.
값을 설정하지 않을 경우 시스템에 설정된 문자 셋으로 적용됩니다.</td></tr>
<tr><td>BUILTIN_LOGGING_FILE</td><td colspan="2">로그를 파일로 출력할지 여부를 설정합니다.</td></tr>
<tr><td rowspan="3">BUILTIN_LOGGING_FILE.TYPE</td><td colspan="2">로그 파일의 생성 타입을 설정합니다.</td></tr>
<tr><td>DailyRollingFile</td><td>BUILTIN_LOGGING_FILE.DatePattern 옵션에 설정한 대로 로그 파일 생성</td></tr>
<tr><td>RollingFile</td><td>BUILTIN_LOGGING_FILE.LOCATION 옵션에서 설정한 파일 이름으로 로그 파일 생성</td></tr>
<tr><td>BUILTIN_LOGGING_FILE.DatePattern</td><td colspan="2">로그 파일 이름의 날짜 패턴을 설정합니다.
날짜 패턴에 설정할 수 있는 문자는 아래 "날짜 패턴 문자"를 참고하시기 바랍니다.</td></tr>
<tr><td>BUILTIN_LOGGING_FILE.CharSet</td><td colspan="2">콘솔에 출력될 로그 내용의 문자 셋을 설정합니다.
값을 설정하지 않을 경우 시스템에 설정된 문자 셋으로 적용됩니다.</td></tr>
<tr><td>BUILTIN_LOGGING_FILE.APPEND</td><td colspan="2">로그가 생길 때마다 로그 파일에 추가할지 여부를 설정합니다.</td></tr>
<tr><td>BUILTIN_LOGGING_FILE.MaxBackupIndex</td><td colspan="2">백업되는 로그 파일의 일련 번호 최대 값을 설정합니다.
백업 파일의 일련 번호가 최대 값을 넘길 경우 가장 오래된 파일이 삭제됩니다.</td></tr>
<tr><td>BUILTIN_LOGGING_FILE.MaxFileSize</td><td colspan="2">로그 파일의 최대 크기를 설정합니다.
로그 파일의 크기가 최대 크기를 넘길 경우 현재 파일은 일련 번호를 추가하여 다른 이름으로 변경하고, 다시 로그를 기록합니다.
예를 들어, 로그 파일 이름이 server_oz.log인 경우 server_oz.log.1로 파일 이름을 변경하고 server_oz.log 파일을 새로 만들어 로그를 기록합니다.</td></tr>
</table>

BUILTIN_LOGGING_FILE.ImmediateFlush	버퍼에 있는 로그를 로그 파일에 바로 추가할지 여부를 설정합니다.
BUILTIN_LOGGING_FILE.LOCATION	로그 파일의 경로와 이름을 설정합니다.

■ Mail

옵션	설명
Mail.alias	mail.properties에서 설정한 앨리어스 이름을 설정합니다. 메일 정보를 log.properties에서 설정하지 않고, mail.properties에서 설정한 값으로 적용할 때 사용합니다.
Mail.active	서버 시작, 서버 종료, 에러 발생 시 해당 로그를 메일로 전송할지 여부를 설정합니다.
Mail.fromSend	보낸 사람 메일 주소를 설정합니다.
Mail.toSend	받는 사람 메일 주소를 설정합니다.
Mail.SMTPServer	보내는 메일 서버(SMTP) 주소를 설정합니다.
Mail.SMTPServerPort	보내는 메일 서버(SMTP) 포트를 설정합니다.
Mail.SMTPUserID	보내는 메일 서버(SMTP)의 인증이 필요할 때 사용자 이름을 설정합니다.
Mail.SMTPUserPassword	보내는 메일 서버(SMTP)의 인증이 필요할 때 사용자 암호를 설정합니다.
Mail.SMTPUserID_encrypted	보내는 메일 서버(SMTP)의 인증이 필요할 때 암호화된 사용자 이름을 설정합니다. 암호화된 사용자 이름을 설정할 경우 Mail.SMTPUserID에서 설정한 값은 무시됩니다.
Mail.SMTPUserPassword_encrypted	보내는 메일 서버(SMTP)의 인증이 필요할 때 암호화된 사용자 암호를 설정합니다. 암호화된 사용자 암호를 설정할 경우 Mail.SMTPUserPassword에서 설정한 값은 무시됩니다.
Mail.SendRetryCount	메일 전송을 실패했을 때 재전송할 횟수를 설정합니다.
Mail.SendRetryPeriodTime	메일 전송을 실패했을 때 재전송할 주기를 설정합니다.(단위 : 초)
Mail.PrefixSubjectMessage	메일 제목 접두어를 설정합니다. 접두어를 설정할 경우 메일 제목은 "[접두어] 에러 메시지" 형태로 설정됩니다.
Mail.FilterType	Mail.FilterServiceCode에서 설정한 서비스 코드의 필터링 규칙을 설정합니다.

	negative	Mail.FilterServiceCode에서 설정한 서비스 코드에서 에러 발생 시 메일 전송 안 함. 그 외 경우에는 메일 전송함
	positive	Mail.FilterServiceCode에서 설정한 서비스 코드에서 에러 발생 시에만 메일 전송함. 그 외 경우에는 메일 전송 안 함
Mail.FilterServiceCode	서비스 코드를 설정합니다. 설정할 수 있는 서비스 코드는 monitor.properties 부분을 참고하시기 바랍니다.	

※ 참고사항 : 오즈 닷넷 서버와 연결된 .NET Framework 버전이 1.1인 경우 로그 파일 이름이 log.properties에서 설정한 이름이 아니라 임의의 이름으로 변경되어 첨부됩니다.

log.properties 옵션 설정 시 사용 가능한 패턴, 문자

■ **날짜 패턴 문자**

패턴	스케줄	적용 예
yyyy-MM	매달 초	6월의 마지막 밤 12시에 server.log 파일이 server.log.2010-06 파일로 바뀌고, 다음 로그는 server.log 파일을 새로 만들어 기록합니다.
yyyy-ww	매주 초	매주 초가 일요일인 경우 1월 첫째 주 토요일 밤 12시에 server.log 파일이 server.log.2010-01 파일로 바뀌고, 다음 로그는 server.log 파일을 새로 만들어 기록합니다.
yyyy-MM-dd	매일 자정	7월 1일 자정에 server.log 파일이 server.log.2010-07-01 파일로 바뀌고, 다음 로그는 server.log 파일을 새로 만들어 기록합니다.
yyyy-MM-dd-HH	매 시 정각	7월 1일 10시 정각에 server.log 파일이 server.log.2010-07-01-09 파일로 바뀌고, 다음 로그는 server.log 파일을 새로 만들어 기록합니다.

yyyy-MM-ddt	12시간	7월 1일 밤 12시에 server.log 파일이 server.log.2010-07-01-09P 파일로 바뀌고, 다음 로그는 server.log 파일을 새로 만들어 기록합니다.
yyyy-MM-dd-HH-mm	매분	7월 1일 10시 정각에 server.log 파일이 server.log.2010-07-01-09-59 파일로 바뀌고, 다음 로그는 server.log 파일을 새로 만들어 기록합니다.

※ 주의사항 : BUILTIN_LOGGING_FILE.DatePattern에서 패턴 설정 시 주의사항

- 오즈 서버 종료 시점과 재시작 시점이 다른 경우 서버 종료 시 남아 있는 기존 로그 파일은 서버가 시작될 때 롤링이 되지 않고 다음 롤링 시점에 롤링됩니다. 예를 들어, DatePattern을 yyyy-MM-dd로 설정한 경우 서버 종료가 2010년 1월 17일에 되고, 재구동이 2010년 1월 18일에 된 경우 server.log.2010-01-17 파일은 만들어지지 않고, 2010년 1월 19일이 되는 동시에 17일과 18일의 로그 파일 내용이 server.log.2010-01-18 파일로 롤링됩니다.
- DatePattern이 yyyy-MM인 경우 매달 초에 롤링되지만 시간적 오차가 1분 ~ 59분까지 발생할 수 있습니다.
- DatePattern이 yyyy-ww인 경우 매주 초에 롤링되지만 시간적 오차가 1분 ~ 59분까지 발생할 수 있습니다. 또한 올해 12월 마지막 주와 내년 1월 첫째 주가 겹치는 경우 시간 계산 오류로 인한 잘못된 파일 이름으로 롤링됩니다. 예를 들어, 2010년 12월 31일은 2010-53 또는 2011-01로 롤링되어야 하지만 2010-01로 롤링됩니다.

■ 로그 Layout 문자

문자	설명
c	오즈 서버 카테고리 이름을 출력합니다. 카테고리 이름은 계층 구조로, %c{2}처럼 출력 단계를 지정할 수 있습니다. 예) %c %c{1} %c{2}

d	날짜를 출력합니다. 패턴 문자를 조합하여 직접 설정하거나 Log4j의 패턴 문자(ABSOLUTE, DATE, ISO8601)를 이용할 수 있습니다. 예) [%d{dd MM yyyy HH:mm:ss,SSS}] [%d{ISO8601}]
m	로그 메시지를 발생시키는 오즈 서버 내부적으로 정해진 이벤트 이름을 출력합니다. 예) %d{ISO8601} [%t] %-5p %c{1} %x-%m
n	개행 문자를 출력합니다.
p	로그 레벨을 출력합니다. %p 앞에 숫자 값을 입력하여 칸 수를 설정할 수 있습니다. 예) %d{ISO8601} [%t] %-5p %c{1} %x-%m (-5p는 5칸의 공간을 차지하고 왼쪽으로 정렬해 출력하라는 의미임)
r	오즈 서버가 구동 중인 시간을 출력합니다.(단위 : millisecond) 예) [%r]
t	쓰레드 이름을 출력합니다. 예) [%t]
%	% 기호를 출력합니다.

■ **Log 파일 날짜 문자**

문자	설명	형태	예
G	Era designator	Text	AD
y	Year	Number	2001
M	Month	Text & Number	May & 05
d	Day	Number	05
h	Hour in am/pm (1~12)	Number	12
H	Hour in day (0~23)	Number	0
m	Minute	Number	30
s	Second	Number	55
S	Millisecond	Number	978
E	Day in week	Text	Tuesday
D	Day in year	Number	189
F	Day of week in month	Number	2 (2nd)

w	Week in year	Number	27
W	Week in month	Number	2
a	Am/pm marker	Text	PM
k	Hour in day (1~24)	Number	24
K	Hour in am/pm (0~11)	Number	0
z	Time zone	Text	Pacific Standard Time
'	Escape for text	Delimiter	
"	Single quote sign	Literal	'

mail.properties

메일 전송 관련 정보 설정 파일입니다.

mail.properties 옵션

옵션	설명
alias.active	메일 전송 기능 사용 여부를 설정합니다.
alias.fromSend	발신자 메일 주소를 설정합니다.
alias.toSend	수신자 메일 주소를 설정합니다.
alias.SMTPServer	SMTP 서버 URL을 설정합니다.
alias.SMTPServerPort	SMTP 서버 포트 번호를 설정합니다.
alias.SMTPUserID	SMTP 인증이 필요한 메일 서버를 이용할 때 인증할 사용자 ID를 설정합니다.
alias.SMTPUserPassword	SMTP 인증이 필요한 메일 서버를 이용할 때 인증할 사용자 ID의 패스워드를 설정합니다.
alias.SMTPUserID_encrypted	SMTP 인증이 필요한 메일 서버를 이용할 때 인증할 암호화된 사용자 ID를 설정합니다.
alias.SMTPUserPassword_encrypted	SMTP 인증이 필요한 메일 서버를 이용할 때 인증할 암호화된 사용자 패스워드를 설정합니다.
alias.EnableSSL	SSL 사용 여부를 설정합니다. ※ 주의사항 : 자바 서버일 경우 jre1.4.2_12 버전 이상에서만 동작합니다.
alias.SendRetryCount	메일 전송 실패 시 재전송 횟수를 설정합니다.
alias.SendRetryPeriodTime	메일 전송 실패 시 재전송 주기를 설정합니다. (단위 : 초)
alias.PrefixSubjectMessage	메일 제목 앞에 사용할 접두어를 설정합니다.

■ **mail.properties 설정 예**

- 일반 메일 서버 설정 예

```
ozmail.active=true
ozmail.fromSend=oz@forcs.com
ozmail.toSend=
```

```
ozmail.SMTPServer=mail.forcs.com
ozmail.SMTPServerPort=25
ozmail.SMTPUserID=UserID
ozmail.SMTPUserPassword=UserPassword
#ozmail.SMTPUserID_encrypted=
#ozmail.SMTPUserPassword_encrypted=
#ozmail.EnableSSL=false
#ozmail.SendRetryCount=
#ozmail.SendRetryPeriodTime=
#ozmail.PrefixSubjectMessage=
```

- SSL 보안이 설정되어 있는 메일 서버 설정 예

```
ozmail.active=true
ozmail.fromSend=oz@forcs.com
ozmail.toSend=
ozmail.SMTPServer=smtp.gmail.com
ozmail.SMTPServerPort=465
ozmail.SMTPUserID=oz@forcs.com
ozmail.SMTPUserPassword=UserPassword
#ozmail.SMTPUserID_encrypted=
#ozmail.SMTPUserPassword_encrypted=
ozmail.EnableSSL=true
#ozmail.SendRetryCount=1
#ozmail.SendRetryPeriodTime=30
#ozmail.PrefixSubjectMessage=[OZ]
```

monitor.properties

오즈 서버 트랜잭션 관련 로그 정보 설정 파일입니다.

monitor.properties 옵션

옵션	설명
MONITOR.Active	모니터 로그를 남길지 여부를 설정합니다.
MONITOR.File	로그 파일의 경로와 이름을 설정합니다.
MONITOR.Append	로그가 생길 때마다 로그 파일에 추가할지 여부를 설정합니다.
MONITOR.MaxBackupIndex	백업되는 로그 파일의 일련 번호 최대 값을 설정합니다. 백업 파일의 일련 번호가 최대 값을 넘길 경우 가장 오래된 파일이 삭제됩니다.
MONITOR.MaxFileSize	로그 파일의 최대 크기를 설정합니다. 로그 파일의 크기가 최대 크기를 넘길 경우 현재 파일은 일련 번호를 추가하여 다른 이름으로 변경하고, 다시 로그를 기록합니다. 예를 들어, 로그 파일 이름이 server.log인 경우 server.log.1로 파일 이름을 변경하고 server.log 파일을 새로 만들어 로그를 기록합니다.
MONITOR.ImmediateFlush	버퍼에 있는 로그를 로그 파일에 바로 추가할지 여부를 설정합니다.
MONITOR.TimePattern	로그에 기록할 시간 패턴을 설정합니다.

■ **파일 포맷 구성 내역**

오즈 서버에서 제공하는 서비스의 트랜잭션 정보를 CSV 포맷으로 따로 저장할 수 있으며 탭 형식으로 구분됩니다.

파일 포맷은 다음과 같습니다.

```
MARK/THR_NAME/TIME/F_MEM/T_MEM/S_CODE/S_STATUS/S_PARAMS/C_IP/DB_SS/E_TIME/DBCONNS
```

항목	설명
MARK	서비스의 시작 및 종료 상태를 표시합니다. 서비스 시작 시 "start", 종료 시 "end"로 표시됩니다.
THR_NAME	서비스 스레드 이름을 표시합니다.
TIME	서비스 시작 및 종료 시간을 표시합니다. 시간은 monitor.properties의 "MONITOR.TimePattern" 옵션에서 설정한 패턴으로 표시됩니다.
F_MEM	사용 가능한 메모리를 표시합니다. (단위 : MB)
T_MEM	전체 JVM 메모리를 표시합니다. (단위 : MB)
S_CODE	서비스 코드를 표시합니다. 아래 "서비스 코드" 부분을 참고하시기 바랍니다.
S_STATUS	상태 코드를 표시합니다. 성공 시 "9001", 실패 시 "9002"로 표시됩니다.
S_PARAMS	클라이언트에서 서비스를 요청하면서 넘긴 패러미터를 표시합니다.
C_IP	서비스를 요청한 클라이언트의 IP를 표시합니다.
DB_SS	DBMS 세션 ID를 표시합니다.
E_TIME	처리 시간을 표시합니다. 시간은 monitor.properties의 "MONITOR.TimePattern" 옵션에서 설정한 패턴으로 표시됩니다.
DBCONNS	DB 커넥션을 사용하여 데이터 바인딩이나 데이터 액션을 실행할 때 DB 커넥션 사용 개수를 표시합니다. DB 커넥션 사용 개수는 "앨리어스1[개수],앨리어스2[개수],...,앨리어스n[개수]" 형태로 표시됩니다.

■ 서비스 코드

서비스 코드	서비스 내용
380	데이터 모듈 형태의 데이터 바인딩 서비스
381	ODI에 추가된 패러미터 정보(name, value)를 가져오는 서비스
382	SDM 요청 서비스
383	데이터 셋 형태의 데이터 바인딩 서비스
384	SDM 요청 서비스(클라이언트 캐시되어 부분적 데이터 호출)
385	databind.properties 정보를 가져오는 서비스
386	databind.properties 정보를 설정하는 서비스
387	ODI에 추가된 패러미터 정보(name, value, type, desc)를 가져오는 서비스
400	DB Pool 정보를 가져오는 서비스
401	DB Pool 상태 리스트를 가져오는 서비스

402	DB Pool 정보 리스트를 가져오는 서비스
403	DB Pool을 추가하는 서비스
404	DB Pool을 삭제하는 서비스
405	db.properties 파일을 저장하는 서비스
406	로그 설정 정보를 가져오는 서비스
407	로그 속성 값을 가져오는 서비스
408	로그 속성 값을 String 형태로 가져오는 서비스
409	로그 파일을 가져오는 서비스
410	로그 레벨을 설정하는 서비스
411	오즈 서버 정보를 가져오는 서비스
412	오즈 서버 Garbage Collection을 수행하는 서비스
413	오즈 서버의 메모리 정보를 가져오는 서비스
414	오즈 서버에 접속된 Concurrent 사용자 개수를 가져오는 서비스
415	오즈 서버의 메모리 상태를 가져오는 서비스
419	모니터 로그 내용을 가져오는 서비스
420	캐시 정보를 가져오는 서비스
412	캐시 정보를 설정하는 서비스
422 / 423	실시간 오즈 서버 로그를 가져오는 서비스
424	실시간 오즈 서버 모니터 로그를 가져오는 서비스
431	라이선스 정보를 가져오는 서비스
432	오즈 서버의 시스템 속성을 가져오는 서비스
433	오즈 서버의 Thread Dump 정보를 가져오는 서비스
500	오즈 서버를 중지하는 서비스
501	오즈 서버를 재시작하는 서비스
600	스케줄러 결과 리스트를 가져오는 서비스
601	단일 보고서 스케줄러 결과를 삭제하는 서비스
602	멀티 보고서 스케줄러 결과를 삭제하는 서비스
603	스케줄러에서 SDM을 생성하고 호출하는 서비스
700	리파지토리 설정 정보를 가져오는 서비스
701	리파지토리 설정 정보를 설정하는 서비스
702	리파지토리를 다시 로딩하는 서비스
703	리파지토리 타입을 가져오는 서비스
704	라파지토리 검색 옵션 가능 여부를 가져오는 서비스

705	리파지토리에 있는 보고서의 리소스를 동기화하는 서비스
800	데이터 모듈의 버전 정보를 가져오는 서비스
900	USL 정보를 가져오는 서비스
901	USL 사용 여부를 가져오는 서비스
1000	데이터 액션 중 Insert 서비스
1001	데이터 액션 중 Delete 서비스
1002	데이터 액션 중 Update 서비스
1100	캐시된 데이터를 삭제하는 서비스
1201	스크립트를 체크하는 서비스
1202	수식 필드 스크립트를 체크하는 서비스
1300	최근 로그 내용을 가져오는 서비스
1301	로그 파일을 가져오는 서비스
1302	로그 파일 리스트를 가져오는 서비스
1303	최근 모니터 로그 내용을 가져오는 서비스
1304	모니터 로그 파일을 가져오는 서비스
1305	모니터 로그 파일 리스트를 가져오는 서비스
1306	최근 로그 내용을 가져오는 서비스
1400	데이터 바인딩 웹 서비스
1402	데이터 액션 웹 서비스
1501	UDL 속성 값을 가져오는 서비스
1502	UDL 속성 값을 설정하는 서비스
1511	데이터 소스 앨리어스 리스트를 가져오는 서비스
1512	데이터 소스 속성 값을 가져오는 서비스
1513	데이터 소스에 앨리어스를 추가하는 서비스
1514	데이터 소스에 앨리어스에 해당되는 속성 값을 수정하는 서비스
1515	데이터 소스에 앨리어스를 삭제하는 서비스
1517	spmgr.properties 정보를 가져오는 서비스
1518	spmgr.properties 정보를 저장하는 서비스
1521	USL 속성 값을 가져오는 서비스
1522	USL 속성 값을 설정하는 서비스
1531	외부 모듈 리스트를 가져오는 서비스
1532	외부 모듈을 추가하는 서비스
1533	외부 모듈을 수정하는 서비스

1534	외부 모듈을 삭제하는 서비스
1535	외부 모듈 속성 리스트를 가져오는 서비스
1536	외부 모듈 숨김 여부를 알려주는 서비스
1541	SAP 앨리어스 리스트를 가져오는 서비스
1542	SAP 속성 리스트를 가져오는 서비스
1543	SAP 앨리어스를 추가하는 서비스
1544	SAP 앨리어스를 수정하는 서비스
1545	SAP 앨리어스를 삭제하는 서비스
1601	메일 앨리어스 리스트를 가져오는 서비스
1602	메일 앨리어스를 추가하는 서비스
1603	메일 앨리어스를 수정하는 서비스
1604	메일 앨리어스를 삭제하는 서비스
1701	UDR 속성 값을 가져오는 서비스

ozserver.properties

오즈 서버와 관련된 통신 포트 설정 및 서버가 사용하는 각종 환경 파일에 대한 경로를 지정합니다. 이 파일의 내용이 변경되면 반드시 서버를 재실행해야 합니다.

ozserver.properties 옵션

옵션	설명
server.portnumber	오즈 서버가 사용할 포트 번호를 설정합니다. 서버 타입이 데몬인 경우만 값을 설정하시기 바랍니다.
path.oz.server.home	오즈 서버 홈 디렉토리를 설정합니다. ※ 주의사항 : 오즈 닷넷 서버의 경우 절대 경로로 설정하시기 바랍니다.
path.assembly.properties.file	assembly.properties 파일 경로를 설정합니다. ※ 주의사항 : 오즈 닷넷 서버에서만 설정 가능합니다.
path.cache.properties.file	cachemngr.properties 파일 경로를 설정합니다.
path.clientinfo.properties.file	clientlist.xml 파일 경로를 설정합니다.
path.connpool.properties.file	db.properties 파일 경로를 설정합니다.
path.databind.properties.file	databind.properties 파일 경로를 설정합니다.
path.datasource.properties.file	datasource.properties 파일 경로를 설정합니다.
path.license.file	ozlicense.xml 파일 경로를 설정합니다.
path.log.properties.file	log.properties 파일 경로를 설정합니다.
path.mail.properties.file	mail.properties 파일 경로를 설정합니다.
path.monitor.properties.file	monitor.properties 파일 경로를 설정합니다.
path.ozdbconfig.properties.file	dbconfig.xml 파일 경로를 설정합니다.
path.ozservice.properties.file	ozservice.properties 파일 경로를 설정합니다.
path.ozudl.properties.file	ozudl.properties 파일 경로를 설정합니다.
path.ozudr.properties.file	ozudr.properties 파일 경로를 설정합니다.
path.repository.properties.file	repository.properties 파일 경로를 설정합니다.

path.sapclient.properties.file	sap.properties 파일 경로를 설정합니다.
path.schedule.properties.file	srmngr.properties 파일 경로를 설정합니다.
path.securitypolicy.properties.file	spmgr.properties 파일 경로를 설정합니다.
path.usl.properties.file	uslmngr.properties 파일 경로를 설정합니다.
path.xml.properties.file	xml.properties 파일 경로를 설정합니다.

참고사항

- %OZ_HOME%

  %OZ_HOME%은 오즈 서버 홈 디렉토리를 가르키는 키워드로 ozserver.properties 파일 내의 path.oz.server.home 옵션에서 설정한 경로를 나타냅니다.
- %OZ_RUNNING_PATH%

  %OZ_RUNNING_PATH%는 오즈 서버의 실행 경로를 나타내며, 오즈 서버의 홈 경로를 상대 경로로 설정할 때 사용합니다.

ozservice.properties

오즈 서버 서비스와 웹 서비스의 사용 여부를 설정합니다.

웹 서비스 기능은 서블릿 타입 서버에서만 지원되며, 웹 서비스 기능이 지원되는 라이선스 파일을 발급받아야 합니다.

오즈 서버는 다음과 같은 웹 서비스 기능을 제공합니다.

- Post – SOAP 요청에 대해서 DataModule에 대한 웹 서비스 제공
- Post – SOAP 요청에 대해서 DataAction에 대한 웹 서비스 제공
- Get 요청에 대해서 해당 서비스의 wsdl 제공

ozservice.properties 옵션

옵션	설명
enable_ozserver	오즈 서버 서비스 기능 사용 여부를 설정합니다.
enable_webservice	웹 서비스 사용 여부를 설정합니다.

ozudl.properties

사용자 정의 로그 사용 여부와 관련 클래스 파일의 경로를 설정합니다.
사용자 정의 로그 기능은 RDB 스토어를 사용한 경우에만 지원됩니다.

ozudl.properties 옵션

옵션	설명
OZ_USER_DEFINED_LOG.Active	사용자 정의 로그 사용 여부를 설정합니다.
OZ_USER_DEFINED_LOG.Class	사용자 정의 로그 인터페이스를 구현한 클래스를 설정합니다.
OZ_UDL_MONITOR.Active	사용자 정의 모니터 로그 사용 여부를 설정합니다.
OZ_UDL_MONITOR.Class	사용자 정의 모니터 로그 인터페이스를 구현한 클래스를 설정합니다.

ozudr.properties

보고서의 리소스 필드와 동기화할 데이터베이스 정보를 설정합니다.

ozudr.properties 옵션

옵션	설명
alias.active	리소스 동기화 기능 사용 여부를 설정합니다.
alias.class	동기화할 정보를 가진 클래스 이름을 설정합니다.
alias.dbaliasname	리소스 정보를 가진 데이터베이스의 앨리어스 이름을 설정합니다. db.properties에서 설정한 앨리어스 이름으로 설정하시기 바랍니다.
alias.props	리소스를 동기화할 때 전달할 값을 설정합니다.

repository.properties

서버 리파지토리 동작 환경을 설정합니다.

repository.properties 옵션

옵션	설명	
REPOSITORY_TYPE	리파지토리 타입을 설정합니다.	
	RDB	상용 데이터베이스와 연동
	BUILTIN	오즈 서버에서 자체적으로 제공하는 데이터베이스 시스템 사용
	NONE	일반 파일 시스템으로 사용
REPOSITORY_DB_ALIAS	리파지토리 타입이 RDB일 때 리파지토리로 사용할 데이터베이스의 앨리어스 이름을 설정합니다. db.properties에서 설정한 앨리어스 이름으로 설정하시기 바랍니다.	
REPOSITORY_FILE_PATH	BUILTIN 타입 리파지토리에서 생성된 파일 또는 NONE 타입에서 업로드한 아이템의 저장 경로를 설정합니다.	
REPOSITORY_NONE_BACKUP	NONE 타입 리파지토리에서 아이템의 백업 기능을 사용할지 여부를 설정합니다.	
REPOSITORY_NONE_BACKUP_PATH	NONE 타입 리파지토리에서 아이템을 백업할 때 백업 경로를 설정합니다.	
REPOSITORY_NONE_BACKUP_ITEM_COUNT	NONE 타입 리파지토리에서 아이템을 백업할 때 백업할 아이템의 개수를 설정합니다.	
REPOSITORY_NONE_BACKUP_ITEM_VALIDDAYS	NONE 타입 리파지토리에서 아이템을 백업할 때 백업 파일을 보관할 기간를 설정합니다.	
REPOSITORY_ITEM_NUMBER_PER_DIRECTORY	디렉토리당 최대 아이템 개수를 설정합니다.	
REPOSITORY_HISTORY_ITEM_VALID_DAYS	아이템의 히스토리를 남길 기간을 설정합니다.	
REPOSITORY_TEMP_DIRECTORY	임시 파일(*.czo, *.obj_stream.zip 등)의 저장 경로를 설정합니다.	
REPOSITORY_ACCESS_BY_HTTP_GET	서블릿 타입의 오즈 서버에서 아이템을 Get 방식으로 가져올지 여부를 설정합니다.	

REPOSITORY_HTTP_GET_CHECK_SESSION	서블릿 타입의 오즈 서버에서 아이템을 Get 방식으로 가져올 때 세션 체크 여부를 설정합니다.
REPOSITORY_RDB_BUILTIN_GUEST_ACCESS	RDB 또는 BUILTIN 타입 리파지토리 서버를 서버 guest 계정으로 접속 가능하도록할지 여부를 설정합니다.

참고사항

- REPOSITORY_TYPE이 "NONE"일 경우
  - 그룹과 사용자 추가 및 권한 설정할 수 없으며, 기본 사용자인 admin 또는 guest 계정만 사용할 수 있습니다.
  - 서버 리파지토리의 아이템은 항상 체크아웃 상태로 표시되며, 기본적으로 아이템 히스토리 및 롤백 기능을 제공하지 않습니다.
  - 아이템 히스토리 및 롤백 기능을 사용하려면 REPOSITORY_NONE_BACKUP 옵션을 true로 설정하시기 바랍니다.

- REPOSITORY_TYPE이 "'RDB'"일 경우
  - db.properties에서 vender 옵션을 User로 설정한 경우는 리파지토리를 RDB 타입으로 사용할 수 없으므로 User 이외의 다른 벤더로 설정하시기 바랍니다.
  - RDB가 MS-SQL일 때 MS-SQL을 대소문자를 체크하지 않는 옵션으로 설치한 경우 카테고리 이름과 아이템 이름만 대소문자를 구별하지 않고, 그 외 경우에는 대소문자를 구별합니다.

- 서버 guest 계정
  - 서버 guest 계정은 오즈 서버 내부적으로 추가된 계정으로, BuiltIn 또는 RDB 타입인 리파지토리 서버만 사용할 수 있습니다.
  - 서버 guest 계정은 오즈 서버 관리자가 별도로 추가할 수 없고, 리파지토리 매니저에도 표시되지 않습니다. 만일, guest 계정이 리파지토리 매니저에서 표시된다면 해당 guest 계정은 서버 guest 계정이 아니라 일반 guest 계정입니다.
  - 서버 guest 계정과 일반 guest 계정은 사용자 ID가 "guest"로 이름은 같지만 웹에서 보고서 또는 폼 파일을 id, password를 지정하지 않고 호출할 때 서버 guest의 경우 각각 아이템에 대해 권한을 설정하지 않아도 표시되고, 일반 guest의 경우 guest 계정에 대한 보기, 읽기 권한을 설정하여야 표시됩니다.
  - guest 계정에 대한 아이템 보기, 읽기 권한을 일일이 설정하는 불편을 없애고자할 경우 REPOSITORY_RDB_BUILTIN_GUEST_ACCESS 옵션 값을 true로 설정하여

서버 guest 계정 기능을 사용하시기 바랍니다. 만일 이미 운영 중인 제품에서 일반 guest 계정이 추가된 경우 리파지토리 매니저를 이용하여 일반 guest 계정을 삭제한 후 REPOSITORY_RDB_BUILTIN_GUEST_ACCESS 옵션 값을 true로 설정하고, 오즈 서버를 재시작하시기 바랍니다.

sap.properties

오즈 서버가 사용할 SAP의 연결 정보를 설정하는 파일로 다수의 연결 정보를 설정할 수 있습니다. 오즈 서버는 서버 성능 향상을 위해 서버 시작 시에 SAP 커넥션 풀을 생성하고, 이를 공유하여 사용할 수 있도록 하는 SAP 커넥션 풀을 지원합니다.

sap.properties 옵션은 대소문자를 구별하오니 입력에 주의하시기 바랍니다.

sap.properties 옵션

옵션	설명
<alias>.proxy	SAP Proxy 클래스 이름을 설정합니다. ※ 주의사항 : 오즈 닷넷 서버에서만 설정 가능합니다.
<alias>.client	SAP 클라이언트를 설정합니다.
<alias>.user	SAP 사용자 이름을 설정합니다.
<alias>.password	SAP 사용자 패스워드를 설정합니다.
<alias>.language	SAP 언어를 설정합니다.
<alias>.host	SAP 호스트 이름을 설정합니다.
<alias>.sysno	SAP 시스템 번호를 설정합니다.
<alias>.r3name	SAP 시스템 이름을 설정합니다. ※ 주의사항 : 오즈 자바 서버에서만 설정 가능합니다.
<alias>.group	애플리케이션 서버의 그룹 이름을 설정합니다.
<alias>.gwhost	SAP 게이트웨이 호스트 이름을 설정합니다. ※ 주의사항 : 오즈 자바 서버에서만 설정 가능합니다.
<alias>.gwserv	SAP 게이트웨이 서비스 번호를 설정합니다. ※ 주의사항 : 오즈 자바 서버에서만 설정 가능합니다.
<alias>.tpname	JCO 서버 프로그램에 등록되어 있는 프로그램 ID를 설정합니다. ※ 주의사항 : 오즈 자바 서버에서만 설정 가능합니다.
<alias>.tphost	외부 서버의 호스트 이름을 설정합니다. ※ 주의사항 : 오즈 자바 서버에서만 설정 가능합니다.

<alias>.maxcnt	오즈 서버가 가질 수 있는 SAP 커넥션 최대 개수를 설정합니다. ※ 주의사항 : 오즈 자바 서버에서만 설정 가능합니다.
<alias>.timeout	SAP 타임아웃을 설정합니다. (단위 : ms) ※ 주의사항 : 오즈 자바 서버에서만 설정 가능합니다.
<alias>.timeoutcheckperiod	SAP 타임아웃 체크 간격을 설정합니다. (단위 : ms) ※ 주의사항 : 오즈 자바 서버에서만 설정 가능합니다.
<alias>.istrace	SAP 시스템의 로그 파일에 기록할지 여부를 설정합니다.
<alias>.doConnectionCheck	SAP이 재구동될 때 커넥션이 유효하지 않을 경우 오즈 서버를 재동하지 않고 자동으로 커넥션을 맺어줄지 여부를 설정합니다. ※ 주의사항 : 오즈 자바 서버에서만 설정 가능합니다.
<alias>.assembly	SAP 연결 시 사용할 어셈블리를 설정합니다. assembly.properties 파일에 설정된 어셈블리 앨리어스 이름으로 설정합니다. ※ 주의사항 : 오즈 닷넷 서버에서만 설정 가능합니다.
<alias>.sid	SAP Netweaver Server에 설정된 Pool의 sid 값 ※ 주의사항 : 오즈 자바 서버에서만 설정 가능하며, SAP Netweaver Server 7.x 이상, sapjco 3.0 이상에서만 적용됩니다.

spmgr.properties

에러 메시지와 서버 정보 표시 레벨 및 오즈 서버 접속 허용 여부 등을 설정합니다.

spmgr.properties 옵션

옵션	설명	
client_message_secure_level	클라이언트에서 오즈 서버로 전송하는 메시지의 전송 레벨을 설정합니다. ※ 주의사항 : 오즈 서버가 서블릿 형태로 실행된 경우에만 설정 가능합니다.	
	high	클라이언트는 에러 코드만 전송. 서버는 아무 정보도 출력 안 함
	middle	클라이언트는 에러 코드와 Exception Name만 전송. 서버는 서버 버전 정보만 출력
	low	클라이언트는 모든 에러 정보를 전송. 서버는 모든 정보를 출력
show_browser_secure_level	웹 브라우저에 표시할 오즈 서버 정보 표시 레벨을 설정합니다. show_browser_secure_level 옵션을 입력하지 않은 경우 client_message_secure_level 옵션에 설정된 값으로 적용됩니다. ※ 주의사항 : 오즈 서버가 서블릿 형태로 실행된 경우에만 설정 가능합니다.	
	high	웹 브라우저에 아무 정보도 출력 안 함
	middle	로그인 화면이 표시되고, 로그인된 경우 아래 정보를 표시 오즈 서버 버전 정보 오즈 서버 Security 정보(Admin 계정 패스워드 변경 여부, Guest 계정 세션 체크 여부, USL 사용 여부) 웹 서비스 리스트

	low	로그인 화면이 표시되고, 로그인된 경우 아래 정보를 표시 오즈 서버 버전 정보 시스템 정보(OS 이름, 호스트 이름, 오즈 서버 경로, IP 등) 오즈 서버 Security 정보(Admin 계정 패스워드 변경 여부, Guest 계정 세션 체크 여부, USL 사용 여부) 웹 서비스 리스트
user.multilogin	같은 사용자에 대한 멀티 로그인 허용 여부를 설정합니다.	
user.sessionTimeout	세션 타임 아웃을 설정합니다. (단위 : 초) ※ 주의사항 : 단일 로그인 즉, user.multilogin 옵션 값이 "false"인 경우에만 설정 가능합니다.	
enable_service_repository	디자이너에서 리파지토리 창을 이용하여 오즈 서버 접속 시 접속을 허용할지 여부를 설정합니다.	
allow_service_qd	오즈 서버에서 ODI 파일을 열었을 때 쿼리문 실행 허용 여부를 설정합니다. ※ 주의사항 : enable_service_repository 옵션 값이 "false"인 경우 무조건 "false"로 적용됩니다.	
allow_service_em	엔터프라이즈 매니저를 이용하여 오즈 서버 접속 시 접속을 허용할지 여부를 설정합니다.	
allow_service_rm	리파지토리 매니저를 이용하여 오즈 서버 접속 시 접속을 허용할지 여부를 설정합니다.	
allow_service_rv	오즈 서버에서 OZR 파일을 열었을 때 미리보기 허용 여부를 설정합니다. ※ 주의사항 : enable_service_repository 옵션 값이 "false"인 경우 무조건 "false"로 적용됩니다.	
allow_service_av	오즈 서버에서 OZA 파일을 열었을 때 미리보기 허용 여부를 설정합니다. ※ 주의사항 : enable_service_repository 옵션 값이 "false"인 경우 무조건 "false"로 적용됩니다.	
allow_service_api	오즈 서버 API를 이용하여 오즈 서버 접속 시 접속을 허용할지 여부를 설정합니다.	
allow_service_sh	오즈 스케줄러를 이용하여 오즈 서버 접속 시 접속을 허용할지 여부를 설정합니다.	
allow_service_ps	푸시 서버를 이용하여 오즈 서버 접속 시 접속을 허용할지 여부를 설정합니다.	

allow_exportbind_service	서버 바인딩 기능(OZR 파일을 오즈 서버에서 바인딩하여 파일로 저장하는 기능)을 허용할지 여부를 설정합니다. ※ 주의사항 : 자바 서버인 경우 자바 버전이 1.5 이상인 서블릿 환경에서만 가능하고, 닷넷 서버인 경우 Microsoft .NET Framework 버전이 3.5 이상인 경우에만 설정 가능합니다.
exportbind_sessionkey	서버 바인딩 기능 사용 시 적용할 세션 키를 설정합니다. 콤마(,)를 구분자로 하여 여러 개 설정할 수 있으며, OR 조건으로 체크됩니다. allow_exportbind_service 옵션 값이 true일 때 exportbind_sessionkey 옵션을 입력하지 않은 경우에는 "user_id"가 세션 키로 동작되고, exportbind_sessionkey 옵션 값을 공백으로 설정한 경우(exportbind_sessionkey=)에는 세션을 체크하지 않습니다.
guest_account_sessionkey	guest 계정으로 오즈 서버에 접속 시 적용할 세션 키를 설정합니다. 콤마(,)를 구분자로 하여 여러 개 설정할 수 있으며, OR 조건으로 체크됩니다. guest_account_sessionkey 옵션을 입력하지 않거나 옵션 값을 공백으로 설정한 경우(guest_account_sessionkey=)에는 세션을 체크하지 않습니다. ※ 주의사항 • 오즈 서버가 서블릿 형태로 실행된 경우에만 설정 가능합니다. • guest 계정에 대한 세션 키가 설정된 오즈 서버를 오즈 스케줄러나 오즈 서버 API를 이용하여 접속할 경우 guest_account_session_allow_ip 옵션에 오즈 스케줄러 또는 오즈 서버 API를 실행하는 시스템의 IP를 추가하시기 바랍니다.
guest_account_session_allow_ip	guest 계정 세션 체크 시 세션 체크 없이 항상 접근을 허용할 IP를 설정합니다. 세미콜론(;)를 구분자로 하여 여러 개 설정할 수 있습니다. 허용 IP는 *를 이용하여 여러 개의 한꺼번에 설정할 수 있으며, *로 설정 시 IPv4 또는 IPv6 형식의 구분자 개수에 맞게 설정하시기 바랍니다. 예를 들어, 192로 시작하는 모든 IP를 허용할 경우 192.*.*.*은 적용되나 192.*, 192.*.100 등의 형태는 적용되지 않습니다. 127.0.0.1 또는 0:0:0:0:0:0:0:1은 허용 IP로 설정하지 않아도 허용됩니다.

srmngr.properties

오즈 스케줄러에 의한 SDM 파일 생성 및 SDM 파일의 사용 여부를 설정합니다.

srmngr.properties 옵션

옵션	설명
use_scheduled_data	오즈 스케줄러에 의한 SDM 파일 생성 및 SDM 파일의 사용 여부를 설정합니다. use_scheduled_data 옵션 값이 "true"인 경우 datamodule_path 옵션에서 설정한 경로에 SDM 파일이 있을 경우 데이터 바인딩을 다시 하지 않고, SDM 파일을 사용합니다. ※ 주의사항 : 데이터 모듈의 Fetch 타입이 "BATCH"인 경우에만 설정 가능합니다.
datamodule_path	SDM 파일이 저장될 서버 경로를 설정합니다.

uslmngr.properties

USL 동작 환경을 설정합니다.

uslmngr.properties 옵션

옵션	설명
UseUserSecurityLogicVer30	USL 사용 여부를 설정합니다.
OZDefault_SERVER	오즈 서버의 기본 USL 이름을 설정합니다.
OZDefault_CLIENT	클라이언트의 기본 USL 이름을 설정합니다.

xml.properties

오즈 서버에서 사용할 xml 파서에 대한 정보를 설정합니다.

xml.properties 파일은 오즈 자바 서버에서만 지원합니다.

xml.properties 옵션

옵션	설명
sax.parser	OZR 파일을 로드하는데 사용할 파서를 SUN, CRIMSON, XERCES, SYSTEM_DEFAULT 중 하나의 값으로 설정합니다. ※ 참고사항 : SYSTEM_DEFAULT로 설정할 경우 자바 서버와 연동된 JDK의 javax.xml.parsers.DocumentBuilderFactory에서 넘겨주는 파서로 적용됩니다.
dom.parser	라이선스 파일과 XML 데이터 셋 파일을 읽을 때 사용할 파서를 SUN, CRIMSON, XERCES, SYSTEM_DEFAULT 중 하나의 값으로 설정합니다. ※ 참고사항 : SYSTEM_DEFAULT로 설정할 경우 자바 서버와 연동된 JDK의 javax.xml.parsers.DocumentBuilderFactory에서 넘겨주는 파서로 적용됩니다.
xpath.parser	XML 데이터 셋의 XPath를 처리할 때 사용할 파서를 XERCES, DOM4J 중 하나의 값으로 설정합니다. ※ 참고사항 : 대용량 데이터를 바인딩할 경우 DOM4J로 설정하면 바인딩 속도를 향상시킬 수 있습니다.

주의사항

- 파서를 설정할 경우 해당 파서의 jar 파일이 오즈 서버가 구동될 때 로딩되도록 설정하여야 합니다.
- 일반적으로 SUN이 가장 속도가 빠르므로 SUN으로 설정할 것을 권장합니다. 단, 기존 시스템에서 CRIMSON을 사용하고 있을 경우에는 CRIMSON으로 설정하시기 바랍니다.
- IBM의 JVM을 사용할 경우는 XERCES의 사용을 권장합니다.
- XML 데이터 셋에서 사용하는 XML 파일에 Entity가 사용된 경우 dom.parser 옵션 값은 CRIMSON 또는 XERCES로 설정하시기 바랍니다.

Ⅴ. 오즈 엔터프라이즈 자바 서버 부가 기능 및 기타 설정

- user, password 암호화
- 빌트인 타입 리파지토리를 RDB 타입으로 변환
- 오즈 엔터프라이즈 자바 서버 Shell 명령어
- 오즈 엔터프라이즈 자바 서버를 윈도우 서비스로 등록
- 오즈 엔터프라이즈 자바 서버와 웹 서비스 연동

user, password 암호화

오즈 엔터프라이즈 자바 서버는 DB 서버, SAP 등에 접속할 때 user와 password를 암호화하여 설정할 수 있도록 ozencrypted.bat 파일을 제공합니다.

DB 서버 접속 정보 암호화

DB 서버에 접속하는 user와 password를 암호화할 때는 아래와 같은 형식으로 ozencrypted.bat 파일을 실행한 후 명령어 창에서 암호화된 결과 값을 복사하여 사용하시기 바랍니다.

- **user 암호화**

```
ozencrypted.bat DB앨리어스이름 user 사용자이름 [db.properties파일절대경로]
```

ex)

```
ozencrypted.bat oracle_example user forcs
```

```
ozencrypted.bat oracle_example user forcs c:\Forcs\OZ Servlet 6.0\conf\db.properties
```

- **password 암호화**

```
ozencrypted.bat DB앨리어스이름 password 암호 [db.properties파일절대경로]
```

ex)

```
ozencrypted.bat oracle_example password forcs
```

```
ozencrypted.bat oracle_example password forcs c:\Forcs\OZ Servlet 6.0\conf\db.properties
```

※ 참고사항 : db.properties 파일 절대 경로

db.properties 파일 절대 경로는 선택 옵션으로, 파일 경로를 설정하여 ozencrypted.bat 파일을 실행할 경우 암호화되지 않은 정보는 삭제하고 암호화된 정보로 db.properties 파일을 수정합니다.

예를 들어, db.properties 파일이 아래와 같이 설정된 경우

```
...
oracle_example.user=forcs
oracle_example.password=forcs
...
```

ozencrypted.bat 파일을 db.properties 파일 경로를 지정하여 user와 password 암호화 명령어를 실행하면 다음과 같이 변경됩니다.

```
...
oracle_example.user_encrypted=GdKcQdRdFbGgHaPaJbJaIb
oracle_example.password_encrypted=GdKcQdRdFbGgHaPaJbJaIb
...
```

SAP 접속 정보 암호화

SAP에 접속하는 user와 password를 암호화할 때는 아래와 같은 형식으로 ozencrypted.bat 파일을 실행한 후 명령어 창에서 암호화된 결과 값을 복사하여 사용하시기 바랍니다.

- **user 암호화**

```
ozencrypted.bat SAP앨리어스이름 user 사용자이름
```

ex)

```
ozencrypted.bat sap_example user forcs
```

- **password 암호화**

```
ozencrypted.bat SAP앨리어스이름 password 암호
```

ex)

```
ozencrypted.bat sap_example password forcs
```

빌트인 타입 리파지토리를 RDB 타입으로 변환

오즈 엔터프라이즈 자바 서버는 빌트인 타입의 리파지토리를 RDB 타입으로 변환할 경우 리파지토리에 저장된 사용자 정보, 카테고리, 아이템, 사용 권한 등을 일일이 새로 만들지 않고, 기존 상태 그대로 변환할 수 있도록 BuiltinToRDBConvert.bat/BuiltinToRDBConvert.sh 파일을 제공합니다.

BuiltinToRDBConvert.bat/BuiltinToRDBConvert.sh 옵션

옵션	설명
JAVA_HOME	자바 플랫폼이 설치된 경로를 설정합니다.
USER_CLASSPATH	사용자 클래스 경로를 설정합니다.
MAX_MEMORY	JVM이 사용할 수 있는 최대 메모리를 설정합니다.
MIN_MEMORY	JVM이 사용할 수 있는 최소 메모리를 설정합니다.
JAVA_OPTION	메모리 외에 별도에 설정할 자바 옵션을 설정합니다.
PATH	빌트인 타입 리파지토리 아이템의 경로를 설정합니다.
DRIVER	변환할 RDB 타입 리파지토리의 JDBC 드라이버 클래스 이름을 설정합니다.
URL	변환할 RDB 타입 리파지토리의 JDBC 드라이버 연결 URL을 설정합니다.
ID	변환할 RDB 타입 리파지토리의 DB 서버 사용자 ID를 설정합니다.
PWD	변환할 RDB 타입 리파지토리의 DB 서버 사용자 패스워드를 설정합니다.
DATEFIELDTYPENAME	변환할 RDB 타입 리파지토리의 DB 종류에 따른 날짜형 데이터 타입 이름을 설정합니다. 예를 들어, MSSQL인 경우 DATETIME, ORACLE인 경우 DATE, DB2인 경우 TIMESTAMP, SYBASE인 경우 DATETIME으로 설정하시기 바랍니다.

빌트인 타입 리파지토리를 RDB 타입으로 변환

본 매뉴얼에서는 빌트인 타입의 리파지토리 경로가 "C:\OZRepository"이고, 변경할 RDB 타입 리파지토리에서 사용할 DB가 MSSQL인 경우를 예로 들어 설명합니다.

먼저 오즈 자바 서버를 중지합니다.

BuiltinToRDBConvert.bat 파일을 텍스트 편집기로 열어 리파지토리 경로와 RDB 드라이버 이름 등을 설정합니다.

```
...
set PATH=C:\OZRepository
set DRIVER=com.microsoft.jdbc.sqlserver.SQLServerDriver
set URL=jdbc:microsoft:sqlserver://127.0.0.1:1433;DatabaseName=Master
set ID=forcs
set PWD=forcs
set DATEFIELDTYPENAME=DATETIME
...
```

명령어 창에서 BuiltinToRDBConvert.bat 파일을 실행합니다. RDB 타입으로 변환이 완료되면 "Builtin repository to RDB repository convert ok." 메시지가 표시됩니다.

주의사항

- BuiltinToRDBConvert.bat/BuiltinToRDBConvert.sh 파일을 실행하기 전에 반드시 오즈 자바 서버를 중지하시기 바랍니다.
- 다음과 같은 경우 에러가 발생하며 RDB 타입으로 변환되지 않습니다.
  - PATH에 설정한 리파지토리 경로에 빌트인 리파지토리 관련 내용이 없을 경우
  - 변환할 DB 서버에 접근이 불가한 경우
  - 변환할 DB에 동일한 테이블 이름이 있을 경우

오즈 엔터프라이즈 자바 서버 Shell 명령어

기본 사용 방법

■ 사용 방법

ozcntrl [-command] [-arg_name:arg_value...] [-option...]

■ 예

- 여러 개의 옵션을 아규먼트로 입력하는 경우

```
ozcntrl -version -ip:127.0.0.1 -port:8001 -id:admin -pwd:admin
```

- 여러 개의 옵션을 대화식으로 입력하는 경우

```
ozcntrl -testsdm
ozcntrl
```

위와 같이 명령어만 입력하면 ozcntrl 명령어는 필요한 옵션을 차례로 입력받습니다.

- 각 옵션을 파일에 저장하는 경우

ozcntrl –regsrv 명령을 이용하여 관리할 오즈 서버 정보를 파일에 기록할 수 있습니다. ozcntrl –regsrv 명령을 통해 생성된 정보는 ozashell.properties 파일에 저장되고, 이후부터는 오즈 서버에 관한 정보(url or ip/port, id, password)를 따로 입력할 필요가 없습니다. 단, 이 경우라도 명령어 창에서 -ip, -port 등의 옵션을 지정하는 경우에는 명령어 창에서 입력한 옵션이 우선적으로 적용됩니다.

Arguments

■ 아규먼트 Source

아규먼트는 세 가지 Source로부터 입력받을 수 있습니다.

- Source A

ozcntrl –regsrv 명령어를 통해 기록된 파일(ozashell.properties : 서버의 주소 및 사용자 ID/Password를 저장하고 있음)

- Source B

  Command argument

- Source C

  Interactive한 User Input

■ **아규먼트 Type**

아규먼트는 위의 Source로부터 읽어 들이는 순서에 따라 세 가지 Type으로 나눌 수 있습니다.

- Type 1

  Source B → Source A → Source C의 순으로 아규먼트를 검색합니다.

  Server Address 아규먼트가 이에 해당합니다.

- Type 2

  Source B → Source C의 순으로 아규먼트를 검색합니다.

  Server Address 외에 작업을 수행하는데 반드시 필요한 아규먼트가 이에 해당합니다.

  예) 보고서 이름, 그룹 이름, User Input Parameter 등

- Type 3

  Source B에서만 아규먼트를 검색합니다. 옵션으로 주어진 아규먼트가 이에 해당합니다.

  예) -xml 등

■ **공통 아규먼트**

아규먼트	설명	타입
url	서블릿 타입 오즈 서버의 URL ex) -url:http://127.0.0.1:8080/oz/server	1
ip	데몬 타입 오즈 서버의 IP ex) -ip:127.0.0.1	1
port	데몬 타입 오즈 서버의 포트 번호 ex) -port:8003	1
id	사용자 ID ex) -id:admin	1
pwd	사용자 패스워드 ex) -pwd:admin	1
fn	파일 이름 ex) -fn:d:/test.xml	2

rn	보고서 이름 ex) -rn:/susb/report1	2
gn	보고서 그룹 이름 ex) -gn:/susb	2
pm	패러미터 ex) -pm:arg1=ttt;arg2=123 ex) -pm:default	2
usl	USL 사용 여부 (repository.dn 명령어 사용 시에만 적용됨) ex) -rpt.dn.-url:http://127.0.0.1:8080/oz/server-usl	없음

Commands

■ **형식 표기법**

Command를 포함한 모든 아규먼트는 생략 가능합니다.

[ ] 는 Type 1 아규먼트를 나타냅니다.

{} 는 Type 2 아규먼트를 나타냅니다.

() 는 Type 3 아규먼트를 나타냅니다.

■ **기본 명령어**

-help

사용법을 표시합니다.

```
ozcntrl -help
```

-regsrv

서버 주소를 등록합니다.

한번 등록한 후에는 아규먼트를 지정하지 않아도 등록한 서버 주소를 사용합니다.

```
ozcntrl -regsrv
```

-start

오즈 데몬 서버를 실행합니다.

```
ozcntrl -start {configuration_file_name}
```

configuration_file_name은 오즈 데몬 서버의 설정 파일입니다. 값을 지정하지 않을 경우 ozserver.properties 파일을 찾습니다.

예)

```
ozcntrl -start
```

```
ozcntrl -start d:/server2.properties
```

-stop

오즈 데몬 서버를 중지합니다.

```
ozcntrl -stop [SERVER-LOCATION]
```

SERVER-LOCATION는 다음과 같은 형식으로 설정합니다.

-ip : ip_address -port:port_number

예)

```
ozcntrl -stop
```

```
ozcntrl -stop -ip:127.0.0.1 -port:8003
```

-version

오즈 서버 및 서버가 설치된 시스템의 버전 정보를 표시합니다. 오즈 서버에 접속하여 정보를 가져오므로 서버 정보는 반드시 명시하여야 합니다.

```
ozcntrl -version[SERVER-INFO]
```

예)

```
ozcntrl -version
```

```
ozcntrl -version -url:http://127.0.0.1:8080/oz/server
```

```
ozcntrl -version -ip:127.0.0.1 -port:8080 -id:admin -pwd:admin
```

-local.version

ozcntrl이 사용하는 로컬 시스템의 오즈 서버 버전을 표시합니다.

```
ozcntrl -local.version
```

예)

```
ozcntrl -local.version
```

-testsdm

SDM 형식의 파일을 읽어 데이터 내용을 확인합니다.

```
ozcntrl -testsdm {-fn:sdm_file_name} (-xml)
```

-fn : 테스트할 sdm 형식의 파일을 지정합니다.

xml : 데이터 모듈을 XML 형태로 파일에 기록합니다. 기록되는 파일의 위치는 ./sdmtest.xml입니다.

예)

```
ozcntrl -testsdm
```

```
ozcntrl -testsdm -fn:/temp/report1.sdm -xml
```

-property

오즈 서버가 설치된 시스템의 자바 정보를 표시합니다.

```
ozcntrl -property:[Java-Prorerty-Name]
```

예)

```
ozcntrl -property : java.runtime.name
```

※ 참고사항 : 해당 속성에 대한 값이 없을 경우 값이 "null"로 표시됩니다.

■ 리파지토리 매니저 관련 명령어

-repository.getcfg

리파지토리 매니저 설정 값을 표시합니다.

```
ozcntl -repository.getcfg [SERVER-LOCATION]
```

예)

```
ozcntl -rpt.getcfg
```

```
ozcntl -rpt.getcfg -url:http://127.0.0.1:8080/oz/server
```

-repository.dn

서버 리파지토리에 있는 아이템을 다운로드합니다.

```
ozcntl -repository.dn [SERVER-INFO]
```

예)

```
ozcntl -rpt.dn
```

```
ozcntl -rpt.dn -url:http://127.0.0.1:8080/oz/server
```

※ 참고사항 : USL을 사용하는 오즈 서버에서 다운로드할 경우에는 오즈 서버에서 사용하는 USL Server.class와 대응하는 USL Client.class가 Classpath에 설정되어 있어야 하며, 아래와 같은 형식으로 명령어를 실행하여야 합니다.

```
ozcntrl -rpt.dn -url:http://127.0.0.1:8080/oz/server-usl
```

■ 캐시 매니저 관련 명령어

-cache.getcfg

캐시 매니저 설정 값을 표시합니다.

```
ozcntl -cache.getcfg [SERVER-INFO]
```

예)

```
ozcntl -cache.getcfg
```

```
ozcntl -cache.getcfg -url:http://127.0.0.1:8080/oz/server
```

■ 커넥션 풀 관련 명령어

-cp.status

데이터베이스 커넥션 풀의 설정 값과 현재 상태를 표시합니다.

```
ozcntrl -cp.status [SERVER-INFO]
```

예)

```
ozcntrl -cp.status
```

```
ozcntrl -cp.status -ip:127.0.0.1 -port:8003
```

```
ozcntrl -cp.status -url:http://127.0.0.1:8080/oz/server
```

■ 로그 매니저 관련 명령어

-log.getcfg

로그 매니저 설정 값을 표시합니다.

```
ozcntrl -log.getcfg [SERVER-INFO]
```

예)

```
ozcntrl -log.getcfg
```

```
ozcntrl -log.getcfg -ip:127.0.0.1 -port:8003
```

```
ozcntrl -log.getcfg -url:http://127.0.0.1:8080/oz/server
```

-log.dn

로그 파일을 다운로드합니다.

```
ozcntrl -log.dn [SERVER-INFO] {-fn:filename}
```

filename : 로그를 저장할 파일 이름

예)

```
ozcntrl -log.dn
```

```
ozcntrl -log.dn -lp:INFO
```

```
ozcntrl -log.dn -url:http://127.0.0.1:8080/oz/server
```

-log.realtime

로그를 실시간으로 표시합니다.

```
ozcntrl –log.realtime
```

예)

```
ozcntrl –log.realtime
```

■ 모니터 로그 관련 명령어

-mntr.dn

트랜잭션 모니터 로그를 다운로드합니다.

```
ozcntrl –mntr.dn [SERVER-INFO] {–fn:filename}
```

filename : 로그를 저장할 파일 이름

예)

```
ozcntrl -mntr.dn
```

```
ozcntrl -mntr.dn -fn:d:/monitor.log
```

```
ozcntrl -mntr.dn -url:http://127.0.0.1:8080/oz/server
```

■ Thread Dump 관련 명령어

–threaddump

Thread Dump를 표시합니다.

오즈 서버와 연동 중인 자바 버전이 1.5 이상인 경우에만 사용 가능합니다.

```
ozcntrl –threaddump
```

예)

```
ozcntrl –threaddump
```

■ Builtin 타입 리파지토리 복구 관련 명령어

-builtin.repair

query.log 파일을 이용하여 db_file.script의 내용을 복구합니다.

```
ozcntrl -builtin.repair [SERVER-INFO]
```

예)

```
ozcntrl -builtin.repair
```

```
ozcntrl -builtin.repair -ip:127.0.0.1 -port:8003
```

```
ozcntrl -builtin.repair -url:http://127.0.0.1:8080/oz/server
```

■ 데이터 바인딩 관련 명령어

-databind.getcfg

오즈 서버에 설정된 ConcurrentFetchSize와 ConcurrentFirstRow 옵션 값을 표시하며, 조회된 내용을 파일로 저장합니다.

```
ozcntrl -databind.getcfg [SERVER-INFO]
```

filename : 로그를 저장할 파일 이름

예)

```
ozcntrl -databind.getcfg
```

```
ozcntrl -databind.getcfg -ip:127.0.0.1 -port:8003
```

```
ozcntrl -databind.getcfg -url:http://127.0.0.1:8080/oz/server
```

오즈 엔터프라이즈 자바 서버를 윈도우 서비스로 등록

오즈 엔터프라이즈 자바 서버는 오즈 자바 서버를 윈도우 서비스로 등록 및 제거할 수 있도록 OZServerInstallService.bat/OZServerUninstallService.bat 파일을 제공합니다.

OZServerInstallService.bat 옵션

옵션	설명	
--Description	윈도우 서비스 설명을 설정합니다. 최대 1024 바이트 문자까지 입력할 수 있습니다.	
--DisplayName	윈도우 서비스에 표시할 이름을 설정합니다. 필수 설정 옵션은 아니나 해당 옵션을 설정하기를 권장합니다.	
--Startup	윈도우 서비스 시작 유형을 설정합니다.	
	Manual	수동 시작
	Auto	자동 시작
--Jvm	윈도우 서비스가 구동될 자바의 jvm.dll 파일의 경로를 설정합니다. 반드시 설정해야할 필수 옵션입니다. 예) --Jvm="%JRE_PATH%"	
--JvmOptions	Jvm 구동 시 설정할 옵션을 설정합니다.	
--Classpath	오즈 서버를 구동하기 위한 클래스 파일 경로를 설정합니다. 반드시 설정해야할 필수 옵션입니다.	
--JvmMs	초기화 메모리를 설정합니다. (단위 : MB)	
--JvmMx	최대 메모리를 설정합니다. (단위 : MB)	
--JvmSs	쓰레드 스택 크기를 설정합니다. (단위 : KB)	
--StartPath	윈도우 서비스를 구동할 오즈 서버 작업 경로를 설정합니다. 반드시 설정해야할 필수 옵션입니다. 예) --StartPath="%OZSERVER_HOME_BIN%"	
--StartClass	윈도우 서비스를 시작하는 클래스 이름을 설정합니다. 반드시 설정해야할 필수 옵션입니다. 예) --StartClass=oz.server.OZServiceBootstrap	
--StartParams	윈도우 서비스를 시작하기 위한 아규먼트를 설정합니다. 반드시 설정해야할 필수 옵션입니다. 예) --StartParams=OZServerStart	

--StartMode	윈도우 서비스 시작 모드를 설정합니다. 반드시 설정해야할 필수 옵션입니다.	
	jvm	Jvm을 사용하여 서비스 시작
--StopPath	윈도우 서비스를 종료할 오즈 서버 작업 경로를 설정합니다. 반드시 설정해야할 필수 옵션입니다. 예) --StopPath="%OZSERVER_HOME_BIN%"	
--StopClass	윈도우 서비스를 종료하는 클래스 이름을 설정합니다. 반드시 설정해야할 필수 옵션입니다. 예) --StopClass=oz.server.OZServiceBootstrap	
--StopParams	윈도우 서비스를 종료하기 위한 아규먼트를 설정합니다. 반드시 설정해야할 필수 옵션입니다. 예) --StopParams=OZServerStop	
--StopMode	윈도우 서비스 종료 모드를 설정합니다. 반드시 설정해야할 필수 옵션입니다.	
	jvm	Jvm을 사용하여 서비스 시작
--LogPath	윈도우 서비스 로그 파일 절대 경로를 설정합니다. 예) --LogPath="%OZSERVER_HOME_BIN%"	
--LogPrefix	윈도우 서비스 로그 파일 접두어를 설정합니다. 예) OZ로 설정할 경우 OZ_yyyyMMdd.log 이름으로 파일이 생성됨	
--LogLevel	로그 레벨을 설정합니다.	
	ERROR	실제 에러 상황에서만 로그에 기록
	WARN	실제 에러와 경고 수준의 에러를 로그에 기록
	INFO	실제 에러와 서버의 트랜잭션 정보를 로그에 기록
	DEBUG	실제 에러, 서버의 트랜잭션 정보 뿐만 아니라 전반적인 서버 동작 상황을 로그에 기록

오즈 엔터프라이즈 자바 서버를 윈도우 서비스로 등록

오즈 자바 서버가 구동될 수 있도록 환경 설정하고, OZServerInstallService.bat 파일을 텍스트 편집기로 열어 필수 옵션을 설정한 후 명령어 창에서 OZServerInstallService.bat 파일을 실행합니다.

등록된 윈도우 서비스 제거

먼저 오즈 자바 서버를 중지시킨 후 명령어 창에서 OZServerUninstallService.bat 파일을 실행합니다.

주의사항

- 윈도우 서비스 등록 기능은 64비트 윈도우 환경에서는 정상적으로 동작하지 않을 수 있습니다.
- 윈도우 서비스로 등록하면 로그를 콘솔에 남길 수 없습니다. 로그를 콘솔로 남기려면 윈도우 서비스로 등록하지 않고, ozstart.bat 파일을 시작 프로그램으로 등록하시기 바랍니다.

오즈 엔터프라이즈 자바 서버와 웹 서비스 연동

본 매뉴얼은 웹 서비스 기능에 대한 기본적인 개념을 숙지하고 있는 개발자를 대상으로 작성되었으며, 웹 서비스 중 Apache Axis와 연동하는 방법으로 설명합니다.

오즈에서 제공하는 웹 서비스

오즈 엔터프라이즈 서버는 데이터 모듈 생성과 DataAction 관련하여 다음과 같은 웹 서비스를 제공합니다.

- Post – SOAP 요청에 대해서 DataModule에 대한 웹 서비스 제공
- Post – SOAP 요청에 대해서 DataAction에 대한 웹 서비스 제공
- Get 요청에 대해서 해당 서비스의 wsdl 제공

오즈 서버와 웹 서비스는 HttpRequest 헤더의 "SOAPAction" 키의 유무로 체크합니다. 만일 HttpRequest 헤더에 "SOAPAction" 키가 없으면 오즈 서버가 요청을 처리합니다.

서비스 URL에 "?wsdl" 패러미터로 "GET" 요청을 보내면 Response로 해당 서비스의 wsdl이 리턴됩니다. 서비스 URL에 "POST"로 SOAP 요청을 보냄으로써 실제 웹 서비스를 사용할 수 있습니다.

- GET 요청을 보내는 예
  http://localhost:8080/oz60/server/ODISERVICE?wsdl
- POST로 SOAP 요청을 보내는 예
  http://localhost:8080/oz60/server/ODISERVICE

웹 서비스 기능은 오즈 서버가 서블릿 형태로 구동될 때만 가능합니다.

관련 클래스 함수

■ **OZServlet 클래스 함수**

- preProcess

Prototype	public void preProcess(javax.servlet.http.HttpServletRequest request, OZAttribute[] attr) throws OZWebServiceException	
Definition	oz.server.OZServlet을 상속받아 preProcess 메서드를 Overwrite해서 사용자 정의 SOAP Header 처리 로직을 추가할 수 있습니다. preProcess에서 Exception이 발생하면 클라이언트에 SOAP Fault 메시지가 결과로 전송됩니다.	
Argument	*request*	요청 객체
	Attr	요청 SOAP 메시지의 Header OZAttribute
Exception	*OZWebServiceException*	사용자 로직 처리 중 예외 상황이 발생했을 때 발생됨

■ **OZAttribute 클래스 함수**

- getName

Prototype	public String getName()
Definition	OZAttribute 이름을 가져옵니다.

- getValue

Prototype	public String getValue()
Definition	OZAttribute 값을 가져옵니다.

■ **ODIService**

- requestDAC2

Prototype	requestDAC2(data)
Definition	DataAction을 실행한 후에 결과로 데이터 모듈을 가져옵니다. ※ 참고사항 : 요청 처리 중 예외 상황이 발생하면 데이터 모듈이 반환되지 않고 "<FAULT>"라는 문자열이 반환됩니다.
Argument	*data* DataAction을 실행할 데이터 모듈

오즈 엔터프라이즈 자바 서버와 웹 서비스 연동 예

오즈 서블릿 서버에서 Apache Axis 웹 서비스를 이용하여 ODI에 대한 DataAction 서비스를 제공한 후 변경된 데이터 모듈을 가져오는 방법에 대해 설명합니다.

➢ 오즈 자바 서버를 구동시키는 WAS가 웹 서비스를 구동할 수 있도록 설정합니다.

- ozservice.properties 파일을 텍스트 편집기로 열어 "enable_webservice" 옵션을 "true"로 설정합니다.

```
...
enable_ozserver=true
enable_webservice=true
...
```

- 웹 서비스 기능을 사용하기 위해서는 웹 서비스 기능을 지원하는 라이브러리 파일(soap2_3_1.jar, mail.jar, activation.jar)을 오즈 서버 라이브러리 파일(ozsfw60.jar)과 같은 경로에 복사합니다.

- 웹 서비스 기능이 지원되는 라이선스 파일을 발급받아 오즈 자바 서버의 license 폴더에 복사합니다.

- WAS를 다시 구동한 후 웹 브라우저에 ODIService의 URL(http://127.0.0.1:8080/oz60/server/ODISERVICE)을 실행하여 ODIService에서 사용할 수 있는 Operation이 웹 브라우저 하단에 표시되는지 확인합니다.

- Apache Axis 1.4 버전을 다운로드합니다.

- 텍스트 편집기를 이용하여 아래와 같이 작성한 후 Axis 1.4 파일의 lib 폴더에 wsdl2.java.bat 파일로 저장합니다.

```
@echo off

set JAVA_HOME=C:\jdk1.4.2_12
set PATH=%JAVA_HOME%\bin;.

set              WEBSERVICE_CLASSPATH=wsdl4j-1.5.1.jar;commons-logging-
1.0.4.jar;commons-discovery-0.2.jar;log4j-
1.2.8.jar;saaj.jar;jaxrpc.jar;axis.jar

echo WSDL2Java is beginning...
%JAVA_HOME%/bin/java         -classpath         "%WEBSERVICE_CLASSPATH%"
org.apache.axis.wsdl.WSDL2Java     -o     .     -d     Application     -s
http://127.0.0.1:8080/oz60/server/ODISERVICE?wsdl
pause
```

- wsdl2.java.bat 파일을 실행하여 클라이언트용 소스를 생성합니다.

- 클라이언트 프로젝트에 클라이언트용 소스를 임포트하고, lib 폴더에 있는 *.lib 파일과 servlet2_3.jar 파일이 클라이언트 프로그램 실행 시 로딩될 수 있도록 classpath에 추가합니다.

- 텍스트 편집기를 이용하여 아래와 같이 작성한 후 OZAxisDataAction2Client.java 파일로 저장합니다.

```
package sample;

import java.io.ByteArrayOutputStream;
import java.io.FileInputStream;
import javax.xml.rpc.ServiceException;
import javax.xml.rpc.Stub;
import sun.misc.BASE64Encoder;
import com.forcs.ws.ODIService;
import com.forcs.ws.ODIServiceServiceLocator;
import com.forcs.ws.meta.Const_DAC;
import com.forcs.ws.meta.Const_FIELD;
import com.forcs.ws.meta.DataActionInfo;
import com.forcs.ws.meta.OZData;
import com.forcs.ws.meta.OZDataSet;
import com.forcs.ws.meta.Param;
import com.forcs.ws.meta.Record;

public class OZAxisDataAction2Client {
  public      static      final      String      _END_POINT_ADDRESS      =
"http://127.0.0.1:8080/oz60/server/ODISERVICE";
  public static void main(String[] args) {
    System.out.println("Endpoint address = " + _END_POINT_ADDRESS);
    try {
      ODIService stub = (ODIService)createProxy();
      OZData data = new OZData();
      data.setODINAME("DACTest.odi");
      data.setCATEGORY("/Category1/");
      DataActionInfo[] info = new DataActionInfo[1];
      info[0] = new DataActionInfo();
      info[0].setSETNAME("DataSet");
      info[0].setTYPE(Const_DAC.Insert);
      Param[] sources = new Param[7];
      sources[0] = new Param();
      sources[0].setNAME("name");
      sources[0].setTYPE(Const_FIELD.VARCHAR);
      sources[0].setVALUE("Sample1");
      sources[1] = new Param();
```

```
sources[1].setNAME("address");
sources[1].setTYPE(Const_FIELD.VARCHAR);
sources[1].setVALUE("Sample1");
sources[2] = new Param();
sources[2].setNAME("birthday");
sources[2].setTYPE(Const_FIELD.DATE);
sources[2].setVALUE("1980-09-02");
sources[3] = new Param();
sources[3].setNAME("job");
sources[3].setTYPE(Const_FIELD.VARCHAR);
sources[3].setVALUE("Sample1");
sources[4] = new Param();
sources[4].setNAME("age");
sources[4].setTYPE(Const_FIELD.INTEGER);
sources[4].setVALUE("1001");
sources[5] = new Param();
sources[5].setNAME("picture");
sources[5].setTYPE(Const_FIELD.BINARY);
FileInputStream fis = new FileInputStream("me.jpg");
ByteArrayOutputStream baos = new ByteArrayOutputStream();
byte[] buffer = new byte[2048];
int len =0;
while((len=fis.read(buffer)) != -1) {
  baos.write(buffer, 0 , len);
}
BASE64Encoder encoder = new BASE64Encoder();
sources[5].setVALUE(encoder.encode(baos.toByteArray()));
fis.close();
baos.close();
sources[6] = new Param();
sources[6].setNAME("picture_path");
sources[6].setTYPE(Const_FIELD.VARCHAR);
sources[6].setVALUE("Image1.jpg");
info[0].setSOURCE(sources);
data.setOZDATAACTIONS(info);

OZData result = (OZData)stub.requestDAC2(data);
OZDataSet[] datas = result.getOZDATARESULT();
for(int i=0 ; i < datas.length ; i++) {
  Record[] record = datas[i].getRECORDS();
  for(int j=0 ; j < record.length ; j++) {
    String[] cols = record[j].getCOLS();
    for(int k=0 ; k < cols.length ; k++) {
      if(cols[k].length() >100) {
```

```
            continue;
          }
          System.out.println(cols[k]);
        }
        System.out.println();
      }
    }
   } catch (Exception ex) {
    ex.printStackTrace();
   }
  }

  private static Stub createProxy() throws ServiceException{
   ODIServiceServiceLocator impl = new ODIServiceServiceLocator();
   impl.setODIServiceEndpointAddress(_END_POINT_ADDRESS);
   return (Stub)impl.getODIService();
  }
}
```

- 작성한 OZAxisDataAction2Client.java 파일을 Eclipse에서 열어 실행하고, 실행 후 변경된 데이터 모듈이 콘솔 창에 표시되는지 확인합니다.

Ⅵ. 오즈 엔터프라이즈 닷넷 서버 부가 기능 및 기타 설정

- user, password 암호화
- 빌트인 타입 리파지토리를 RDB 타입으로 변환
- 오즈 엔터프라이즈 닷넷 서버 실행 환경 설정
- 오즈 엔터프라이즈 닷넷 서버를 윈도우 서비스로 등록
- 오즈 엔터프라이즈 닷넷 서버와 웹 서비스 연동

user, password 암호화

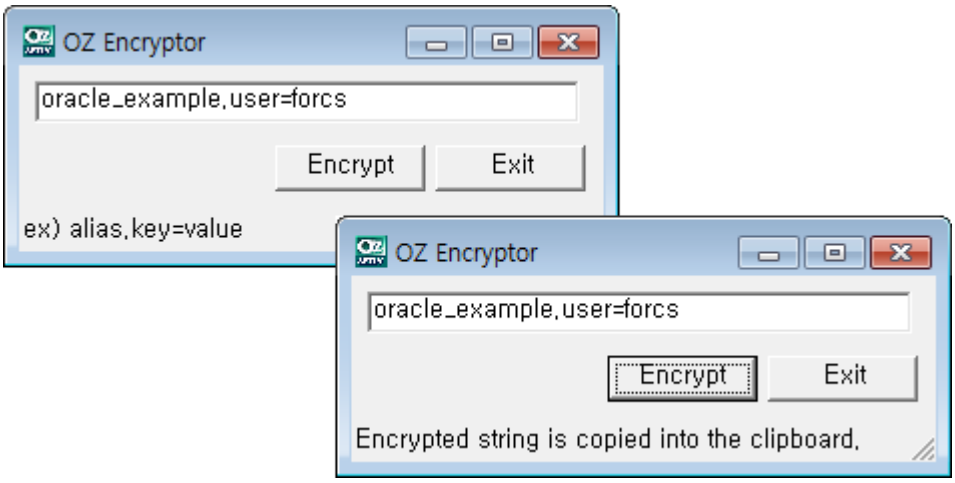
오즈 엔터프라이즈 닷넷 서버는 DB 서버, SAP 등에 접속할 때 user와 password를 암호화하여 설정할 수 있도록 OZEncryptor.exe 파일을 제공합니다.

암호화 방법

OZEncryptor.exe 파일을 실행한 후 텍스트박스에 아래와 같은 형식으로 명령어를 입력한 후 키보드의 <Enter> 키를 입력하거나 [Encrypt] 버튼을 클릭하면 암호화된 값이 클립보드에 복사됩니다.

- **user 암호화**

```
앨리어스이름.user=사용자이름
```



- **password 암호화**

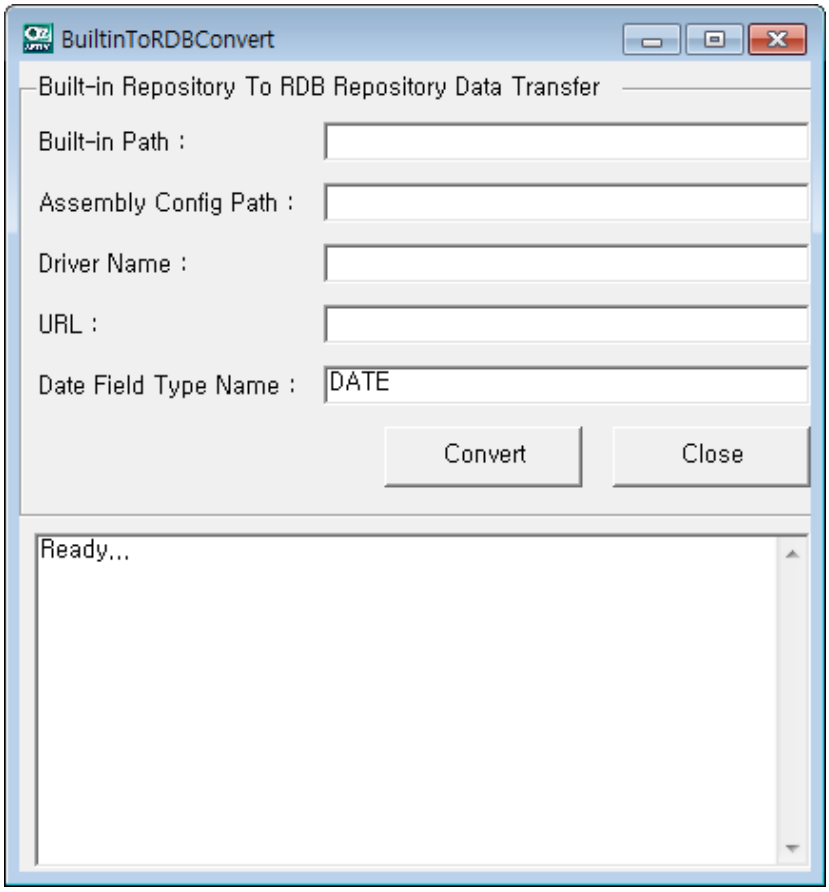
```
앨리어스이름.password=암호
```


OZ Encryptor
oracle_example,password=forcs
Encrypt
Exit
ex) alias,key=value
OZ Encryptor
oracle_example,password=forcs
Encrypt
Exit
Encrypted string is copied into the clipboard.

빌트인 타입 리파지토리를 RDB 타입으로 변환

오즈 엔터프라이즈 닷넷 서버는 빌트인 타입의 리파지토리를 RDB 타입으로 변환할 경우 리파지토리에 저장된 사용자 정보, 카테고리, 아이템, 사용 권한 등을 일일이 새로 만들지 않고, 기존 상태 그대로 변환할 수 있도록 BuiltinToRDBConvert.exe 파일을 제공합니다.

BuiltinToRDBConvert.exe 옵션



옵션	설명
Builti-in Path	빌트인 타입 리파지토리 아이템의 경로를 설정합니다.

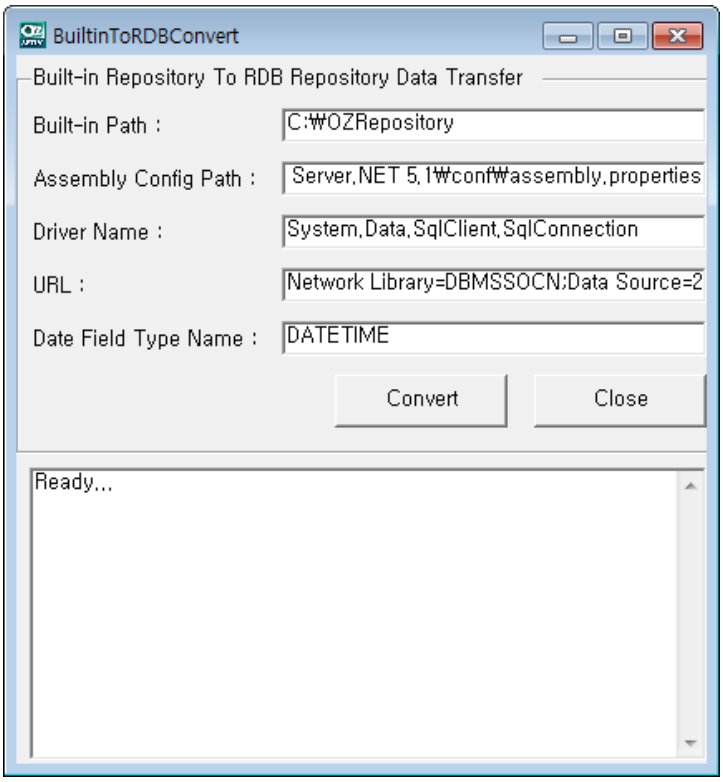
Assembly Config Path	assembly.properties 파일의 경로를 설정합니다.
Driver Name	변환할 RDB 타입 리파지토리의 ADO.NET 드라이버 클래스 이름을 설정합니다.
URL	변환할 RDB 타입 리파지토리의 ADO.NET 드라이버 연결 URL을 설정합니다.
DATE Field Type Name	변환할 RDB 타입 리파지토리의 DB 종류에 따른 날짜형 데이터 타입 이름을 설정합니다. 예를 들어, MSSQL인 경우 DATETIME, ORACLE인 경우 DATE, DB2인 경우 TIMESTAMP, SYBASE인 경우 DATETIME으로 설정하시기 바랍니다.

빌트인 타입 리파지토리를 RDB 타입으로 변환

본 매뉴얼에서는 빌트인 타입의 리파지토리 경로가 "C:\OZRepository"이고, 변경할 RDB 타입 리파지토리에서 사용할 DB가 MSSQL인 경우를 예로 들어 설명합니다.

먼저 오즈 닷넷 서버를 중지합니다.

BuiltinToRDBConvert.exe 파일을 실행하여 나타나는 다이얼로그에서 리파지토리 경로, 클래스 이름 등의 옵션을 설정합니다.



[Convert] 버튼을 클릭합니다. RDB 타입으로 변환이 완료되면 "Builtin repository to RDB repository convert ok." 메시지가 표시됩니다.

주의사항

- BuiltinToRDBConvert.exe 파일을 실행하기 전에 반드시 오즈 닷넷 서버를 중지하시기 바랍니다.
- 다음과 같은 경우 에러가 발생하며 RDB 타입으로 변환되지 않습니다.
  - Builti-in Path에 설정한 리파지토리 경로에 빌트인 리파지토리 관련 내용이 없을 경우
  - 변환할 DB 서버에 접근이 불가한 경우
  - 변환할 DB에 동일한 테이블 이름이 있을 경우

오즈 엔터프라이즈 닷넷 서버 실행 환경 설정

오즈 엔터프라이즈 닷넷 서버를 실행하기 위해서는 .NET Framework 1.1 이상, Microsoft Internet Information Server (IIS) 5.1 이상, ASP.NET이 설치된 환경에서 아래 세 가지 환경을 설정하여야 실행됩니다.

- IIS와 ASP.NET 연동
- IIS의 기본 웹 사이트에 오즈 닷넷 서버 폴더 추가
- 오즈 닷넷 서버 폴더의 접근 권한에 ASP.NET 계정 추가

IIS와 ASP.NET 연동

■ **역할 추가 기능을 제공하지 않는 플랫폼인 경우**

➢ 명령어 창을 실행하여 .NET Framework이 설치된 폴더 중 aspnet_regiis.exe 파일이 있는 폴더로 이동한 후 아래와 같은 명령어를 실행합니다.

```
aspnet_regiis.exe -i
```

■ **역할 추가 기능을 제공하는 플랫폼인 경우**

➢ 서버에 역할을 추가하는 마법사를 실행한 후 서버 역할에 "IIS"를 추가합니다.

※ 참고사항 : 네트워크 정책 및 액세스 서비스

역할 추가 화면에서 "네트워크 정책 및 액세스 서비스"가 표시되는 플랫폼인 경우 "네트워크 정책 및 액세스 서비스"도 추가하고, 다음 단계로 표시되는 역할 서비스 선택에서 "네트워크 정책 서버"를 선택하시기 바랍니다.

➢ IIS 역할 서비스 추가 마법사를 제공하는 플랫폼(ex. Windows 2008)인 경우 "응용 프로그램 개발" 및 그 하위 항목을 모두 체크하고, 응용 프로그램 서버 옵션 다이얼로그가 표시되는 플랫폼(ex. Windows 2003)인 경우 "ASP.NET 사용"을 체크하고 [다음] 버튼을 클릭합니다.

IIS의 기본 웹 사이트에 오즈 닷넷 서버 폴더 추가

■ **IIS 버전이 6.0 이하인 경우**

오즈 닷넷 서버 폴더를 IIS의 기본 웹 사이트에 가상 디렉터리로 추가합니다.

- 인터넷 정보 서비스 다이얼로그에서 "기본 웹 사이트"의 팝업 메뉴 중 [새로 만들기] > [가상 디렉터리] 메뉴를 클릭하여 가상 디렉터리 만들기 마법사를 실행합니다.
- 가상 디렉터리에 사용할 별칭을 입력합니다. 별칭은 오즈 닷넷 서버를 인식하는 이름으로, 서버 URL은 "http://IP/별칭/Server.aspx" 형태로 설정됩니다.
- 웹 사이트에서 게시할 컨텐트 위치는 오즈 닷넷 서버의 %OZ_HOME% 경로로 설정합니다.
- 가상 디렉터리에 설정할 액세스 권한을 읽기, 스크립트 실행, 실행, 쓰기로 선택합니다.
- 설정 완료 후 [마침] 버튼을 클릭하여 가상 디렉터리 만들기 마법사를 종료합니다.

■ **IIS 버전이 7.0 이상인 경우**

오즈 닷넷 서버 폴더를 IIS의 기본 웹 사이트에 응용 프로그램으로 추가합니다.

- 인터넷 정보 서비스 관리자에서 "Default Web Site"의 팝업 메뉴 중 [응용 프로그램 추가] 메뉴를 클릭하여 응용 프로그램 추가 다이얼로그를 실행합니다.
- 응용 프로그램 추가 다이얼로그에서 별칭을 설정합니다. 별칭은 오즈 닷넷 서버를 인식하는 이름으로, 서버 URL은 "http://IP/별칭/Server.aspx" 형태로 설정됩니다.
- 응용 프로그램 추가 다이얼로그에서 실제 경로를 설정합니다. 실제 경로는 오즈 닷넷 서버의 %OZ_HOME% 경로로 설정합니다.
- 설정 완료 후 [확인] 버튼을 클릭하여 응용 프로그램 추가 다이얼로그를 종료합니다.

오즈 닷넷 서버 폴더에 사용 권한 추가

■ **IIS 버전이 6.0 이하인 경우**

- 오즈 닷넷 서버가 설치된 폴더의 속성 다이얼로그를 실행한 후 [보안] 탭으로 이동하여 사용자에 ASPNET, NETWORK SERVICE을 추가하고 ASPNET, NETWORK SERVICE의 사용 권한에 모든 권한을 추가합니다.

■ **IIS 버전이 7.0 이상인 경우**

- 오즈 닷넷 서버가 설치된 폴더의 속성 다이얼로그를 실행한 후 [보안] 탭으로 이동하여 사용자에 IIS_IUSRS를 추가하고 IIS_IUSRS의 사용 권한에 모든 권한을 추가합니다.

오즈 엔터프라이즈 닷넷 서버를 윈도우 서비스로 등록

오즈 엔터프라이즈 닷넷 서버는 오즈 닷넷 서버를 윈도우 서비스로 등록 및 제거할 수 있도록 OZServiceManager.exe, OZServiceLauncher.exe 파일을 제공합니다.
OZServiceManager.exe 파일을 실행하기 전에 OZServiceManager.exe.config, OZServiceLauncher.exe.config 파일을 이용하여 서비스 이름, 설명, 실행 파일 경로 등을 설정하여야 합니다.

OZServiceManager.exe 실행 화면



버튼	설명
Install Service	오즈 닷넷 서버를 윈도우 서비스로 등록합니다.
Uninstall Service	오즈 닷넷 서버를 윈도우 서비스에서 제거합니다.
Start Service	오즈 닷넷 서버 윈도우 서비스를 시작합니다.
Stop Service	오즈 닷넷 서버 윈도우 서비스를 중지합니다.

OZServiceManager.exe.config 옵션

옵션	설명
name	윈도우 서비스 이름을 설정합니다.
executable_binary	윈도우 서비스 실행 파일 즉, OZServiceLauncher.exe 파일의 경로를 설정합니다. 절대 경로 뿐만 아니라 OZServiceManager.exe 파일의 경로를 기준으로 상대 경로로 설정할 수 있습니다. OZServiceManager.exe 파일의 경로를 기준으로 상대 경로로 설정할 때에는 %MNGR_HOME%을 사용하여 설정하시기 바랍니다.
description	윈도우 서비스 설명을 설정합니다.

OZServiceLauncher.exe.config 옵션

옵션	설명
name	윈도우 서비스 이름을 설정합니다.
ozserver.properties	오즈 닷넷 서버 설정 정보 파일인 ozserver.properties 파일의 절대 경로를 설정합니다.

오즈 엔터프라이즈 닷넷 서버를 윈도우 서비스로 등록

■ **환경 설정**

- ozserver.properties 파일을 텍스트 편집기로 열어 오즈 닷넷 서버 실행 시 사용할 포트 번호를 server.portnumber 옵션에 설정합니다.

```
server.portnumber=8003
path.oz.server.home=c:/Forcs/OZ Server.NET 6.0
...
```

- OZServiceManager.exe.config 파일을 텍스트 편집기로 열어 서비스 이름, 설명, 실행 파일 경로를 설정합니다.

```
...
<add key="name" value="OZServer.NET"/>
<add                                          key="executable_binary"
```

```
value="%MNGR_HOME%\OZServiceLauncher.exe"/>
<add key="description" value="OZ Server .NET Windows Service"/>
...
```

- OZServiceLauncher.exe.config 파일을 텍스트 편집기로 열어 서비스 이름과 ozserver.properties 파일의 절대 경로를 설정합니다.

```
...
<add key="name" value="OZServer.NET"/>
<add key="ozserver.properties" value="c:\Forcs\OZ Server.NET
6.0\ozserver.properties"/>
...
```

- OZServiceManager.exe 파일을 실행한 후 [Install Service] 버튼을 클릭합니다.

등록된 윈도우 서비스 제거

OZServiceManager.exe 파일을 실행한 후 [Uninstall Service] 버튼을 클릭합니다.

주의사항

- 윈도우 서비스 등록 기능은 64비트 윈도우 환경에서는 정상적으로 동작하지 않을 수 있습니다.

오즈 엔터프라이즈 닷넷 서버와 웹 서비스 연동

본 매뉴얼은 웹 서비스 기능에 대한 기본적인 개념을 숙지하고 있는 개발자를 대상으로 작성되었습니다.

오즈에서 제공하는 웹 서비스

오즈 엔터프라이즈 닷넷 서버는 데이터 모듈 생성과 DataAction 관련하여 다음과 같은 웹 서비스를 제공합니다.

- Post – SOAP 요청에 대해서 DataModule에 대한 웹 서비스 제공
- Post – SOAP 요청에 대해서 DataAction에 대한 웹 서비스 제공
- Get 요청에 대해서 해당 서비스의 wsdl 제공

오즈 서버와 웹 서비스는 HttpRequest 헤더의 "SOAPAction" 키의 유무로 체크합니다. 만일 HttpRequest 헤더에 "SOAPAction" 키가 없으면 오즈 서버가 요청을 처리합니다.

서비스 URL에 "?wsdl" 패러미터로 "GET" 요청을 보내면 Response로 해당 서비스의 wsdl이 리턴됩니다. 서비스 URL에 "POST"로 SOAP 요청을 보냄으로써 실제 웹 서비스를 사용할 수 있습니다.

웹 서비스 기능은 오즈 서버가 서블릿 형태로 구동될 때만 가능합니다.

관련 클래스 함수

■ **OZIISPage 클래스 함수**

- Preprocess

Prototype	`public void Preprocess(System.Web.HttpRequest request, OZAttribute[] attr) throws OZWebServiceException`
Definition	oz.server.OZIISPage를 상속받아 Preprocess 메서드를 Overwrite해서 사용자 정의 SOAP Header 처리 로직을 추가할 수 있습니다. Preprocess 에서 Exception이 발생하면, 클라이언트에 SOAP Fault 메시지가 결과로 전송된다.

Argument	*request*	요청 객체
	attr	요청 SOAP 메시지의 Header OZAttribute
Exception	*OZWebServiceException*	사용자 로직 처리 중 예외 상황이 발생했을 때 발생됨

■ **OZATTRIBUTES 클래스 함수**

- Name

Prototype	`public string Name{get;}`
Definition	OZAttribute 이름을 가져옵니다.

- Value

Prototype	`public string Value{get;}`
Definition	OZAttribute 값을 가져옵니다.

■ **ODIService**

- requestDAC2

Prototype	`requestDAC2(data)`	
Definition	DataAction을 실행한 후에 결과로 데이터 모듈을 가져옵니다. ※ 참고사항 : 요청 처리 중 예외 상황이 발생하면 데이터 모듈이 반환되지 않고 "<FAULT>"라는 문자열이 반환됩니다.	
Argument	*data*	DataAction을 실행할 오즈 데이터 모듈

오즈 엔터프라이즈 닷넷 서버와 웹 서비스 연동 예

➢ IIS가 웹 서비스를 구동할 수 있도록 설정합니다.

➢ ozservice.properties 파일을 텍스트 편집기로 열어 "enable_webservice" 옵션을 "true"로 설정합니다.

```
..
enable_ozserver=true
enable_webservice=true
...
```

➢ 웹 서비스 기능이 지원되는 라이선스 파일을 발급받아 오즈 닷넷 서버의 license 폴더에 복

사합니다.

- IIS를 다시 구동합니다.

- .NET Framework SDK를 설치하여 ASP.NET용 웹 서비스 유틸리티인 wsdl.exe 파일을 아래와 같이 실행하면

```
wsdl http://hostServer/WebserviceRoot/WebServiceName.asmx?WSDL
```

ODIServiceService.cs 파일이 생성됩니다.(wsdl.exe 버전이 1.1.4322.2032인 경우)

```
//----------(----------------------------------------------------------------
-----
// <autogenerated>
//    This code was generated by a tool.
//    Runtime Version: 1.1.4322.2032
//
//    Changes to this file may cause incorrect behavior and will be lost if
//    the code is regenerated.
// </autogenerated>
//------------------------------------------------------------------------------
----Ü
using System.Diagnostics;
using System.Xml.Serialization;
using System;
using System.Web.Services.Protocols;
using System.ComponentModel;
using System.Web.Services;

/// <remarks/>
[System.Diagnostics.DebuggerStepThroughAttribute()]
[System.ComponentModel.DesignerCategoryAttribute("code")]
[System.Web.Services.WebServiceBindingAttribute(Name="ODIServiceSoapBinding
", Namespace="http://ws.forcs.com")]
public               class               ODIServiceService               :
System.Web.Services.Protocols.SoapHttpClientProtocol {

   public             static             string             URI             =
"http://127.0.0.1/oz60/server.aspx/ODIService";

   public OZATTRIBUTES OZATTRIBUTESValue;

   /// <remarks/>
   public ODIServiceService() {
```

```
        this.Url = URI;
    }

    /// <remarks/>
    [System.Web.Services.Protocols.SoapHeaderAttribute("OZATTRIBUTESValue")]
    [System.Web.Services.Protocols.SoapDocumentMethodAttribute("",
RequestNamespace="http://ws.forcs.com",
ResponseElementName="responseDataModule",
ResponseNamespace="http://ws.forcs.com",
Use=System.Web.Services.Description.SoapBindingUse.Literal,
ParameterStyle=System.Web.Services.Protocols.SoapParameterStyle.Wrapped)]
    public                                                          void
requestDataModule([System.Xml.Serialization.XmlElementAttribute(Form=System
.Xml.Schema.XmlSchemaForm.Qualified, IsNullable=true)] ref OZData OZDATA) {
        object[] results = this.Invoke("requestDataModule", new object[] {
                    OZDATA});
        OZDATA = ((OZData)(results[0]));
    }

    /// <remarks/>
    public    System.IAsyncResult    BeginrequestDataModule(OZData    OZDATA,
System.AsyncCallback callback, object asyncState) {
        return this.BeginInvoke("requestDataModule", new object[] {
                    OZDATA}, callback, asyncState);
    }

    /// <remarks/>
    public  void  EndrequestDataModule(System.IAsyncResult  asyncResult,  out
OZData OZDATA) {
        object[] results = this.EndInvoke(asyncResult);
        OZDATA = ((OZData)(results[0]));
    }

    /// <remarks/>
    [System.Web.Services.Protocols.SoapHeaderAttribute("OZATTRIBUTESValue")]
    [System.Web.Services.Protocols.SoapDocumentMethodAttribute("",
RequestNamespace="http://ws.forcs.com",   ResponseElementName="responseDAC",
ResponseNamespace="http://ws.forcs.com",
Use=System.Web.Services.Descriptios.SoapBindingUse.Literal,
ParameterStyle=System.Web.Services.Protocols.SoapParameterStyle.Wrapped)]
    public                                                          void
requestDAC([System.Xml.Serialization.XmlElementAttribute(Form=System.Xml.Sc
hema.XmlSchemaForm.Qualified, IsNullable=true)] ref OZData OZDATA) {
        object[] results = this.Invoke("requestDAC", new object[] {
```

```
                OZDATA});
        OZDATA = ((OZData)(results[0]));
    }

    /// <remarks/>
    public      System.IAsyncResult      BeginrequestDAC(OZData      OZDATA,
System.AsyncCallback callback, object asyncState) {
        return this.BeginInvoke("requestDAC",Mnew object[] {
                OZDATA}, callback, asyncState);
    }

    /// <remarks/>
    public  void  EndrequestDAC(System.IAsyncResult  asyncResult,  out  OZData
OZDATA) {
        object[] results = this.EndInvoke(asyncResult);
        OZDATA = ((OZData)(results[0]));
    }

    /// <remarks/>
    [System.Web.Sxrvices.Protocols.SoapHeaderAttribute("OZATTRIBUTESValue")]
    [System.Web.Services.Protocols.SoapDocumentMethodAttribute("",
RequestNamespace="http://ws.forcs.com",   ResponseElementName="rsponseDAC2",
ResponseNamespace="http://ws.forcs.com",
Use=System.Web.Services.Description.SoapBindingUse.Literal,
ParameterStyle=System.Web.Services.Protocols.SoapParameterStyle.Wrapped)]
    public                                                              void
requestDAC2([System.Xml.Serialization.XmlElementAttribute(Form=System.Xml.S
chema.XmlSchemaForm.Qualified, IsNullable=true)] ref OZData OZDATA) {
        object[] results = this.Invoke("requestDAC2", new object[] {
                OZDATA});
        OZDATA = ((OZData)(results[0]));
    }

    /// <remarks/>
    public      System.IAsyncResult      BeginrequestDAC2(OZData      OZDATA,
System.AsyncCallback callback, object asyncState) {
        return this.BeginInvoke("requestDAC2", new object[] {
                OZDATA}, callback, asyncState);
    }

    /// <remarks/>
    public  void  EndrequestDAC2(System.IAsyncResult  asyncResult,  out  OZData
OZDATA) {
        object[] results = this.EndInvoke(asyncResult);
```

```
        OZDATA = ((OZData)(results[0]));
    }
}

/// <remarks/>
[System.Xml.Serialization.XmlTypeAttribute(Namespace="http://ws.forcs.com")
]
[System.Xml.Serialization.XmlRootAttribute(Namespace="http://ws.forcs.com",
IsNullable=false)]
public class OZATTRIBUTES : System.Web.Services.Protocols.SoapHeader {

    /// <remarks/>

[System.Xml.SPrialization.XmlArrayAttribute(Form=System.Xml.Schema.XmlSchem
aForm.Qualified, IsNullable=true)]
    [System.Xml.Serialization.XmlArrayItemAttribute("OZATTRIBUTE",
Form=System.Xml.Schema.XmlSchemaForm.Qualified)]
    public OZAttribute[] OZATTRIBUTE;
}

/// <remarks/>
[System.Xml.Serialization.XmlTypeAttribute(Namespace="http://meta.ws.forcs.
com")]
public class OZAttribute {

    /// <remarks/>

[System.Xml.Serialization.XmlElementAttribute(Form=System.Xml.Schema.XmlSch
emaForm.Qualified, IsNullable=true)]
    public string NAME;

    /// <remarks/>

[System.Xml.Serialization.XmlElementAttribute(Form=System.Xml.Schema.XmlSch
emaForm.Qualified, IsNullable=true)]
    public string VALUE;
}

/// <remarks/>
[System.Xml.Serialization.XmlTypeAttribute(Namespace="http://meta.ws.forcs.
com")]
public class DataActionInfo {

    /// <remarks/>
```

```
[System.Xml.Serialization.XmlElementAttribute(Form=System.Xml.Schema.XmlSch
emaForm.Qualified, IsNullable=true)]
   public string SETNAME;

   /// <remarks/>

[System.Xml.Serialization.XmlElementAttribute(Form=System.Xml.Schema.XmlSch
emaForm.Qualified)]
   public Const_DAC TYPE;

   /// <remarks/>
   [System.Xml.Serialization.XmlIgnoreAttribute()]
   public bool TYPESpecified;

   /// <remarks/>

[System.Xml.Serialization.XmlElementAttribute(Form=System.Xml.Schema.XmlSch
emaForm.Qualified, IsNullable=true)]
   public string EXTARG;

   /// <remarks/>

[System.Xml.Serialization.XmlArrayAttribute(Form=System.Xml.Schema.XmlSchem
aForm.Qualified, IsNullable=true)]
   [System.Xml.Serialization.XmlArrayItemAttribute("S",
Form=System.Xml.Schema.XmlSchemaForm.Qualified,
Namespace="http://ws.forcs.com")]
   public Param[] SOURCE;

   /// <remarks/>

[System.Xml.Serialization.XmlArrayAttribute(Form=System.Xml.Schema.XmlSchem
aForm.Qualified, IsNullable=true)]
   [System.Xml.Serialization.XmlArrayItemAttribute("D",
Form=System.Xml.Schema.XmlSchemaForm.Qualified,
Namespace="http://ws.forcs.com")]
   public Param[] DESTINATION;

   /// <remarks/>

[System.Xml.Serialization.XmlArrayAttribute(Form=System.Xml.Schema.XmlSchem
aForm.Qualified, IsNullable=true)]
   [System.Xml.Serialization.XmlArrayItemAttribute("MASTERPARAMETER",
```

```
Form=System.Xml.Schema.XmlSchemaForm.Qualified,
Namespace="http://ws.forcs.com")]
    [System.Xml.Serialization.XmlArrayItemAttribute("PARAM",
Form=System.Xml.Schema.XmlSchemaForm.Qualified,
Namespace="http://ws.forcs.com", NestingLevel=1)]
    public Param[][] MASTERPARAMETERLIST;

    /// <remarks/>

[System.Xml.Serialization.XmlElementAttribute(Form=System.Xml.Schema.XmlSch
emaForm.Qualified, IsNullable=true)]
    public string MESSAGE;
}

/// <remarks/>
[System.Xml.Serialization.XmlTypeAttribute(Namespace="http://meta.ws.forcs.
com")]
public enum Const_DAC {

    /// <remarks/>
    Insert,

    /// <remarks/>
    RowUpdate,

    /// <remarks/>
    Delete,
}

/// <remarks/>
[System.Xml.Serialization.XmlTypeAttribute(Namespace="http://meta.ws.forcs.
com")]
public class Param {

    /// <remarks/>

[System.Xml.Serialization.XmlElementAttribute(Form=System.Xml.Schema.XmlSch
emaForm.Qualified, IsNullable=true)]
    public string NAME;

    /// <remarks/>

[System.Xml.Serialization.XmlElementAttribute(Form=System.Xml.Schema.XmlSch
emaForm.Qualified)]
```

```
    public Const_FIELD TYPE;

    /// <remarks/>
    [System.Xml.Serialization.XmlIgnoreAttribute()]
    public bool TYPESpecified;

    /// <remarks/>

[System.Xml.Serialization.XmlElementAttribute(Form=System.Xml.Schema.XmlSch
emaForm.Qualified, IsNullable=true)]
    public string VALUE;
}

/// <remarks/>
[System.Xml.Serialization.XmlTypeAttribute(Namespace="http://meta.ws.forcs.
com")]
public enum Const_FIELD {

    /// <remarks/>
    BIT,

    /// <remarks/>
    TINYINT,

    /// <remarks/>
    SMALLINT,

    /// <remarks/>
    INTEGER,

    /// <remarks/>
    INT,

    /// <remarks/>
    BIGINT,

    /// <remarks/>
    FLOAT,

    /// <remarks/>
    REAL,

    /// <remarks/>
    DOUBLE,
```

```
    /// <remarks/>
    NUMERIC,

    /// <remarks/>
    DECIMAL,

    /// <remarks/>
    CHAR,

    /// <remarks/>
    VARCHAR,

    /// <remarks/>
    LONGVARCHAR,

    /// <remarks/>
    DATE,

    /// <remarks/>
    TIME,

    /// <remarks/>
    TIMESTAMP,

    /// <remarks/>
    BINARY,

    /// <remarks/>
    VARBINARY,

    /// <remarks/>
    LONGVARBINARY,

    /// <remarks/>
    BLOB,

    /// <remarks/>
    CLOB,

    /// <remarks/>
    OTHER,
}
```

```
/// <remarks/>
[System.Xml.Serialization.XmlTypeAttribute(Namespace="http://meta.ws.forcs.
com")]
public class Record {

   /// <remarks/>

[System.Xml.Serialization.XmlArrayAttribute(Form=System.Xml.Schema.XmlSchem
aForm.Qualified, IsNullable=true)]
   [System.Xml.Serialization.XmlArrayItemAttribute("COL",
Form=System.Xml.Schema.XmlSchemaForm.Qualified,
Namespace="http://ws.forcs.com")]
   public string[] COLS;

   /// <remarks/>

[System.Xml.Serialization.XmlArrayAttribute(Form=System.Xml.Schema.XmlSchem
aForm.Qualified, IsNullable=true)]
   [System.Xml.Serialization.XmlArrayItemAttribute("OZDATASET",
Form=System.Xml.Schema.XmlSchemaForm.Qualified,
Namespace="http://ws.forcs.com")]
   public OZDataSet[] DETAILS;
}

/// <remarks/>
[System.Xml.Serialization.XmlTypeAttribute(Namespace="http://meta.ws.forcs.
com")]
public class OZDataSet {

   /// <remarks/>

[System.Xml.Serialization.XmlElementAttribute(Form=System.Xml.Schema.XmlSch
emaForm.Qualified, IsNullable=true)]
   public string NAME;

   /// <remarks/>

[System.Xml.Serialization.XmlArrayAttribute(Form=System.Xml.Schema.XmlSchem
aForm.Qualified, IsNullable=true)]
   [System.Xml.Serialization.XmlArrayItemAttribute("RECORD",
Form=System.Xml.Schema.XmlSchemaForm.Qualified,
Namespace="http://ws.forcs.com")]
   public Record[] RECORDS;
}
```

```
/// <remarks/>
[System.Xml.Serialization.XmlTypeAttribute(Namespace="http://meta.ws.forcs.
com")]
public class Field {

   /// <remarks/>

[System.Xml.Serialization.XmlElementAttribute(Form=System.Xml.Schema.XmlSch
emaForm.Qualified, IsNullable=true)]
   public string NAME;

   /// <remarks/>

[System.Xml.Serialization.XmlElementAttribute(Form=System.Xml.Schema.XmlSch
emaForm.Qualified)]
   public Const_FIELD TYPE;
}

/// <remarks/>
[System.Xml.Serialization.XmlTypeAttribute(Namespace="http://meta.ws.forcs.
com")]
public class DataSetInfo {

   /// <remarks/>

[System.Xml.Serialization.XmlElementAttribute(Form=System.Xml.Schema.XmlSch
emaForm.Qualified, IsNullable=true)]
   public string NAME;

   /// <remarks/>

[System.Xml.Serialization.XmlElementAttribute(Form=System.Xml.Schema.XmlSch
emaForm.Qualified, IsNullable=true)]
   public string MASTERSET;

   /// <remarks/>

[System.Xml.Serialization.XmlArrayAttribute(Form=System.Xml.Schema.XmlSchem
aForm.Qualified, IsNullable=true)]
   [System.Xml.Serialization.XmlArrayItemAttribute("FIELD",
Form=System.Xml.Schema.XmlSchemaForm.Qualified,
Namespace="http://ws.forcs.com")]
   public Field[] FIELDINFO;
```

```
}

/// <remarks/>
[System.Xml.Serialization.XmlTypeAttribute(Namespace="http://meta.ws.forcs.
com")]
public class DataSet {

    /// <remarks/>

[System.Xml.Serialization.XmlElementAttribute(Form=System.Xml.Schema.XmlSch
emaForm.Qualified, IsNullable=true)]
    public string NAME;

    /// <remarks/>

[System.Xml.Serialization.XmlElementAttribute(Form=System.Xml.Schema.XmlSch
emaForm.Qualified, DataType="integer", IsNullable=true)]
    public string MAXROW;

    /// <remarks/>

[System.Xml.Serialization.XmlElementAttribute(Form=System.Xml.Schema.XmlSch
emaForm.Qualified, DataType="integer", IsNullable=true)]
    public string NO;
}

/// <remarks/>
[System.Xml.Serialization.XmlTypeAttribute(Namespace="http://meta.ws.forcs.
com")]
public class OZData {

    /// <remarks/>

[System.Xml.Serialization.XmlElementAttribute(Form=System.Xml.Schema.XmlSch
emaForm.Qualified)]
    public string VER;

    /// <remarks/>

[System.Xml.Serialization.XmlElementAttribute(Form=System.Xml.Schema.XmlSch
emaForm.Qualified, IsNullable=true)]
    public string ODINAME;

    /// <remarks/>
```

```
[System.Xml.Serialization.XmlElementAttribute(Form=System.Xml.Schema.XmlSch
emaForm.Qualified, IsNullable=true)]
   public string CATEGORY;

   /// <remarks/>

[System.Xml.Serialization.XmlElementAttribute(Form=System.Xml.Schema.XmlSch
emaForm.Qualified)]
   public bool REFRESH;

   /// <remarks/>
   [System.Xml.Serialization.XmlIgnoreAttribute()]
   public bool REFRESHSpecified;

   /// <remarks/>

[System.Xml.Serialization.XmlElementAttribute(Form=System.Xml.Schema.XmlSch
emaForm.Qualified)]
   public Const_DM CACHEDMTYPE;

   /// <remarks/>
   [System.Xml.Serialization.XmlIgnoreAttribute()]
   public bool CACHEDMTYPESpecified;

   /// <remarks/>

[System.Xml.Serialization.XmlElementAttribute(Form=System.Xml.Schema.XmlSch
emaForm.Qualified)]
   public Const_FETCH FETCHUNIT;

   /// <remarks/>
   [System.Xml.Serialization.XmlIgnoreAttribute()]
   public bool FETCHUNITSpecified;

   /// <remarks/>

[System.Xml.Serialization.XmlArrayAttribute(Form=System.Xml.Schema.XmlSchem
aForm.Qualified, IsNullable=true)]
   [System.Xml.Serialization.XmlArrayItemAttribute("PARAM",
Form=System.Xml.Schema.XmlSchemaForm.Qualified,
Namespace="http://ws.forcs.com")]
   public Param[] OZPARAMS;
```

```
    /// <remarks/>

[System.Xml.Serialization.XmlElementAttribute(Form=System.Xml.Schema.XmlSch
emaForm.Qualified, IsNullable=true)]
    public DataSet DATASET;

    /// <remarks/>

[System.Xml.Serialization.XmlArrayAttribute(Form=System.Xml.Schema.XmlSchem
aForm.Qualified, IsNullable=true)]
    [System.Xml.Serialization.XmlArrayItemAttribute("DATASET",
Form=System.Xml.Schema.XmlSchemaForm.Qualified,
Namespace="http://ws.forcs.com")]
    public DataSet[] INVALIDSET;

    /// <remarks/>

[System.Xml.Serialization.XmlArrayAttribute(Form=System.Xml.Schema.XmlSchem
aForm.Qualified, IsNullable=true)]
    [System.Xml.Serialization.XmlArrayItemAttribute("MASTERPARAMETER",
Form=System.Xml.Schema.XmlSchemaForm.Qualified,
Namespace="http://ws.forcs.com")]
    [System.Xml.Serialization.XmlArrayItemAttribute("PARAM",
Form=System.Xml.Schema.XmlSchemaForm.Qualified,
Namespace="http://ws.forcs.com", NestingLevel=1)]
    public Param[][] MASTERPARAMETERLIST;

    /// <remarks/>

[System.Xml.Serialization.XmlArrayAttribute(Form=System.Xml.Schema.XmlSchem
aForm.Qualified, IsNullable=true)]
    [System.Xml.Serialization.XmlArrayItemAttribute("DATASET",
Form=System.Xml.Schema.XmlSchemaForm.Qualified,
Namespace="http://ws.forcs.com")]
    public DataSet[] MAXROWSOFSET;

    /// <remarks/>

[System.Xml.Serialization.XmlArrayAttribute(Form=System.Xml.Schema.XmlSchem
aForm.Qualified, IsNullable=true)]
    [System.Xml.Serialization.XmlArrayItemAttribute("DATASETINFO",
Form=System.Xml.Schema.XmlSchemaForm.Qualified,
Namespace="http://ws.forcs.com")]
    public DataSetInfo[] OZDATASETINFOS;
```

```
    /// <remarks/>

[System.Xml.Serialization.XmlArrayAttribute(Form=System.Xml.Schema.XmlSchem
aForm.Qualified, IsNullable=true)]
    [System.Xml.Serialization.XmlArrayItemAttribute("OZDATASET",
Form=System.Xml.Schema.XmlSchemaForm.Qualified,
Namespace="http://ws.forcs.com")]
    public OZDataSet[] OZDATARESULT;

    /// <remarks/>

[System.Xml.Serialization.XmlArrayAttribute(Form=System.Xml.Schema.XmlSchem
aForm.Qualified, IsNullable=true)]
    [System.Xml.Serialization.XmlArrayItemAttribute("DATAACTIONINFO",
Form=System.Xml.Schema.XmlSchemaForm.Qualified,
Namespace="http://ws.forcs.com")]
    public DataActionInfo[] OZDATAACTIONS;
}

/// <remarks/>
[System.Xml.Serialization.XmlTypeAttribute(Namespace="http://meta.ws.forcs.
com")]
public enum Const_DM {

    /// <remarks/>
    FILE,

    /// <remarks/>
    MEMORY,
}

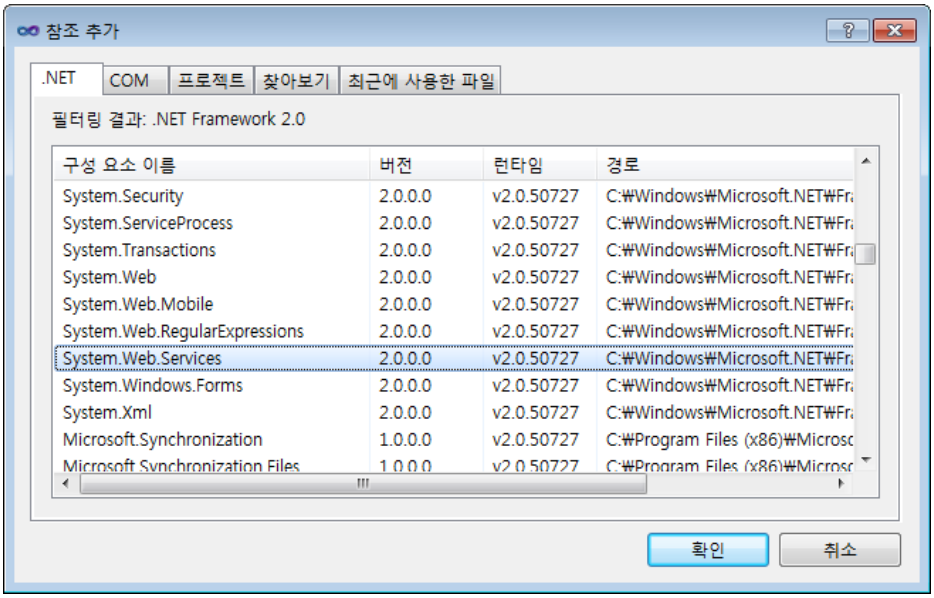
/// <remarks/>
[System.Xml.Serialization.XmlTypeAttribute(Namespace="http://meta.ws.forcs.
com")]
public enum Const_FETCH {

    /// <remarks/>
    PER_DATAMODULE,

    /// <remarks/>
    PER_DATASET,
}
```

- Visual studio 2010을 실행하여 .NET Framework 2.0을 설정하고, 콘솔 응용 프로그램을 선택하여 새 프로젝트를 생성합니다.

- wsdl.exe로 생성한 ODIServiceService.cs 파일을 프로젝트에 추가하고, System.Web.Services 참조를 추가합니다.



- 아래와 같이 스크립트 소스를 작성하고, 프로젝트를 실행합니다.

```
using System;
using System.Collections.Generic;
using System.Text;
using System.IO;

namespace WebServiceTest
{
    class Program
    {
        static void Main(string[] args)
        {
            try{
          OZData data = new OZData();
          data.ODINAME = "DACTest.odi";
          data.CATEGORY = "/REQ/258/";
          DataActionInfo[] info = new DataActionInfo[1];
          info[0] = new DataActionInfo();
          info[0].SETNAME = "DataSet";
          info[0].TYPE = Const_DAC.Insert;
```

```
info[0].TYPESpecified = true;
Param[] sources = new Param[7];
sources[0] = new Param();
sources[0].NAME = "name";
sources[0].TYPE = Const_FIELD.VARCHAR;
sources[0].VALUE = "Sample2";
sources[0].TYPESpecified = true;
sources[1] = new Param();
sources[1].NAME = "address";
sources[1].TYPE = Const_FIELD.VARCHAR;
sources[1].VALUE = "Sample2";
sources[1].TYPESpecified = true;
sources[2] = new Param();
sources[2].NAME = "birthday";
sources[2].TYPE = Const_FIELD.DATE;
sources[2].VALUE = "1980-09-02";
sources[2].TYPESpecified = true;
sources[3] = new Param();
sources[3].NAME = "job";
sources[3].TYPE = Const_FIELD.VARCHAR;
sources[3].VALUE = "Sample1";
sources[3].TYPESpecified = true;
sources[4] = new Param();
sources[4].NAME = "age";
sources[4].TYPE = Const_FIELD.INTEGER;
sources[4].VALUE = "1002";
sources[4].TYPESpecified = true;
sources[5] = new Param();
sources[5].NAME = "picture";
sources[5].TYPE = Const_FIELD.BINARY;
sources[5].TYPESpecified = true;
FileStream fis = new FileStream("c:/me.png", FileMode.Open);
MemoryStream baos = new MemoryStream();
byte[] buffer = new byte[2048];
int read;
while((read = fis.Read(buffer, 0, buffer.Length)) > 0)
   baos.Write(buffer, 0, read);
sources[5].VALUE = System.Convert.ToBase64String(baos.ToArray());
fis.Close();
baos.Close();
sources[6] = new Param();
sources[6].NAME = "picture_path";
sources[6].TYPE = Const_FIELD.VARCHAR;
sources[6].VALUE = "me.png";
```

```
        sources[6].TYPESpecified = true;
        info[0].SOURCE = sources;
        data.OZDATAACTIONS = info;
        ODIServiceService svc = new ODIServiceService();
        svc.requestDAC2(ref data);
        //DataAction을 실행한 후 변경된 데이터 모듈의 내용을 콘솔에 출력합니다.
        OZDataSet[] datas = data.OZDATARESULT;
        for(int i=0 ; i < datas.Length ; i++) {
          Record[] record = datas[i].RECORDS;
          for (int j = 0; j < record.Length; j++)
          {
            String[] cols = record[j].COLS;
            for (int k = 0; k < cols.Length; k++)
            {
              if (cols[k].Length > 100)
              {
                continue;
              }
              Console.WriteLine(cols[k]);
            }
            Console.WriteLine("");
          }
        }
      } catch (Exception ex) {
        Console.WriteLine(ex.Message);
      }
      Console.WriteLine("END");
    }
  }
}
```