

OZ Data Push Server Manual

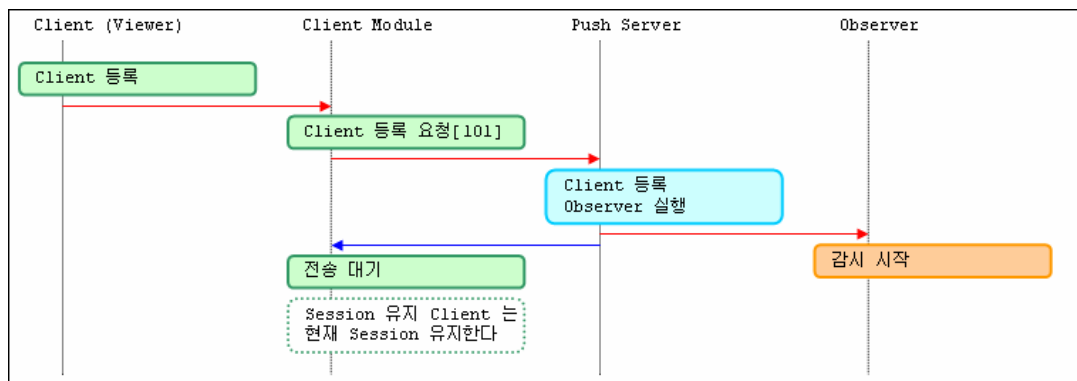
오즈 Push 서버 개요	2
오즈 Push 서버 파일 구성	4
오즈 Push 서버 설정	6
오즈 Push 서버 콘솔 명령어	14

오즈 Push 서버 개요

연결 흐름도

오즈 Push 서버(Daemon 서버)는 서버에서 클라이언트로 특정 데이터의 변경 상태를 실시간으로 전달하는 서버로, 오즈 Push 서버는 데이터 Push 서버에 등록한 Observer가 데이터의 변환을 감지하여 클라이언트 모듈에 데이터를 전송하는 서버입니다.

클라이언트와 오즈 Push 서버의 연결 흐름도는 다음과 같습니다.



- 최초 클라이언트 모듈의 요청을 접수하고, Observer를 실행 및 관리합니다.
- 뷰어와의 Session은 최초 클라이언트의 요청 소켓을 유지하는 방법과 연결 정보를 관리하는 방법을 제공합니다.
- Thread 개수를 최소화하기 위하여 클라이언트로 데이터 전송이 필요한 시점에 전송 Thread를 생성하고 전송 완료 후에 자동 소멸합니다.
- Observer ID를 생성하는 로직과 데이터 변경 체크를 위한 로직은 사용자가 직접 개발합니다.
- Observer 목록 및 해당 클래스/로그 정보/Session interval 등을 pushserver.xml 파일로 관리합니다.
- Log4j를 이용한 로깅 기능을 제공합니다.

※ 참고사항 : 오즈 Push 서버에 많은 수의 클라이언트가 등록될 경우 성능 저하가 발생할 수 있습니다.

Observer

- 대상 데이터의 변화 여부를 감지하고, 변화가 감지될 경우 결과를 전송할 클라이언트 목록을 생성하여 Queue에 넣습니다.
- 전송할 클라이언트를 결정한 후 sdm 또는 사용자가 정의한 메시지를 클라이언트로 전송합니다.
- ObserverID와 데이터 변화 감지 로직은 Interface로 제공하고, 사용자가 개발합니다.

※ 참고사항 : Observer 인터페이스 구현시 "_ObserverImpl"을 클래스 이름으로 사용할 수 없습니다.

Client Module

- Push 서버와의 통신을 담당합니다.

오즈 Push 서버 파일 구성

디렉토리	파일명	설명
bin	InstallService.bat	오즈 Push 서버를 NT 서버에 등록시키는 파일
	ozpushcntrl.bat	윈도우(.bat) 또는 유닉스(.sh)용 오즈 Push 서버 컨트롤 파일
	ozpushcntrl.sh	
	OZPushServer.ini	오즈 Push 서버 실행을 위한 환경 파일
	ozpushstart.bat	윈도우(.bat) 또는 유닉스(.sh)용 오즈 Push 서버를 실행시키는 파일
	ozpushstart.sh	
	ozpushstop.bat	윈도우(.bat) 또는 유닉스(.sh)용 오즈 Push 서버를 종료시키는 파일
	ozpushstop.sh	
	ozService.exe	오즈 서비스 실행 파일
	ozservice_create.bat	서비스 환경 파일을 생성해주는 파일
	Profile.exe	환경 생성 실행 파일
	Service.bat	서비스 환경 설정 파일
	ServiceManager.exe	NT 서비스를 등록, 해제, 시작, 정지시키는 실행 파일
	StartService.bat	서비스 시작 파일
	StopService.bat	서비스 종료 파일
	UninstallService.bat	등록된 오즈 Push 서버를 NT 서비스에서 제거하는 파일
conf	pushserver.properties	오즈 Push 서버 기본 설정 파일
	pushserver.xml	오즈 Push 서버의 로그, session, transmission, Observer를 설정
lib	crimson.jar	XML 크림슨 파서 라이브러리 파일
	log4j.jar	로그 사용에 대한 라이브러리 파일
	ozpush10.jar	오즈 Push 서버 라이브러리 파일
	ozsfw35.jar	오즈 서버 라이브러리 파일
	parser.jar	XML 파서 클래스 라이브러리 파일
	xalan.jar	XML Xerces 파서 클래스 라이브러리 파일
	xerces.jar	XML Xerces 파서 클래스 라이브러리 파일

	xercesImpl.jar	XML Xerces 파서 클래스 라이브러리 파일
	xml-apis.jar	XML Xerces 파서 클래스 라이브러리 파일
license	ozlicense.xml	오즈 Push 서버 라이선스 파일
logs	pushserver.log	오즈 Push 서버 로그 파일

오즈 Push 서버 설정

License 설정

오즈 Push 서버 라이선스는 ozlicense.xml 파일 형태로 발급되며 이 파일을 오즈 Push 서버가 설치된 디렉토리의 License 폴더에 복사하여 설정합니다. 각 License 파일에는 License 유효 기간, 서버 IP 주소, 운영 체제 종류 등이 설정되어 있습니다. 오즈 Push 서버를 운영하던 중 IP 주소 및 운영 체제가 변경될 경우에는 License 파일을 재발급 받아야 합니다.

오즈 Push 서버 환경 설정

■ ozpushcntrl.bat

JRE 또는 오즈 Push 서버를 운영하기 위한 각종 클래스 라이브러리를 설정합니다. 기본적으로 Windows 환경의 경우 자체 JVM을 내장하여 이를 이용하고 있으며, UNIX 환경의 경우 ozpushcntrl.sh 파일에 JRE Path를 직접 설정하여야 합니다.

```
@echo off
rem -----
rem
rem          OZ Push Server Contorl(OZA SHELL)
rem
rem
rem This script file written and tested under Windows 2000
rem
rem User should set following variables:
rem
rem JAVA_HOME      - Determines the version of Java used to start
rem                  OZ Server. This variable must point to the
rem                  root directory of a JDK or JRE installation.
rem                  for an up-to-date list of supported JVMs your platform.
rem MAX_MEMORY      - Maximum memory used by OZ Push Server. (The unit is mega-byte.)
rem MIN_MEMORY      - Minimum memory used by OZ Push Server. (The unit is mega-byte.)
rem -----
```

```
rem -----
rem      Set user-defined variables
rem -----

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

set OZPUSHSERVER_HOME=..
set OZLIBALL=%OZPUSHSERVER_HOME% \lib

rem -----
rem      OZ Main Class
rem -----
set OZLIB=%OZLIBALL% \ozpush10.jar;%OZPUSHSERVER_HOME% \conf \;

rem -----
rem      OZ Server Class
rem -----
set OZLIB=%OZLIB%;%OZLIBALL% \ozsfw35.jar

rem -----
rem      Library for Logging
rem -----
set OZLIB=%OZLIB%;%OZLIBALL% \log4j.jar

rem -----
rem      Library for XML
rem -----
set OZLIB=%OZLIB%;%OZLIBALL% \xalan.jar
set OZLIB=%OZLIB%;%OZLIBALL% \xercesImpl.jar
set OZLIB=%OZLIB%;%OZLIBALL% \xml-apis.jar
set OZLIB=%OZLIB%;%OZLIBALL% \parser.jar
set OZLIB=%OZLIB%;%OZLIBALL% \xerces.jar
set OZLIB=%OZLIB%;%OZLIBALL% \crimson.jar

rem -----
rem      User Library
rem -----
rem set OZLIB=%OZLIB%;%OZLIBALL% \sample.jar

echo %OZLIB%
```



```
MAX_MEMORY=256
MIN_MEMORY=128
JAVA_OPTION=-hotspot

OZPUSHSERVER_HOME=..
OZLIB=.:$OZPUSHSERVER_HOME/conf
OZLIB=$OZLIB:$OZPUSHSERVER_HOME/lib/ozpush10.jar

# -----
#           OZ Server Library
# -----
OZLIB=$OZLIB:$OZPUSHSERVER_HOME/lib/ozsfw35.jar

# -----
#           Library for Logging
# -----
OZLIB=$OZLIB:$OZPUSHSERVER_HOME/lib/log4j.jar

# -----
#           Library for XML
# -----
OZLIB=$OZLIB:$OZPUSHSERVER_HOME/lib/xalan.jar
OZLIB=$OZLIB:$OZPUSHSERVER_HOME/lib/xercesImpl.jar
OZLIB=$OZLIB:$OZPUSHSERVER_HOME/lib/xml-apis.jar
OZLIB=$OZLIB:$OZPUSHSERVER_HOME/lib/parser.jar
OZLIB=$OZLIB:$OZPUSHSERVER_HOME/lib/xerces.jar
OZLIB=$OZLIB:$OZPUSHSERVER_HOME/lib/crimson.jar

rem -----
rem           User Library
rem -----
rem set OZLIB=%OZLIB%;%OZLIBALL% W sample.jar

if [ "$1" != "-start" ]; then
    JAVA_OPTION=
else
    JAVA_OPTION="$JAVA_OPTION -ms$MIN_MEMORY"m
    JAVA_OPTION="$JAVA_OPTION -mx$MAX_MEMORY"m
fi

echo ""
echo $OZLIB
echo ""
```

```
$JAVA_HOME/bin/java $JAVA_OPTION -classpath $OZLIB oz.push.server.Console $1 $2 $3 $4
```

※ 참고사항 : MAX_MEMORY, MIN_MEMORY는 자바 옵션으로 JVM이 사용할 수 있는 최대/최소 메모리를 설정하며, 그 외 기타 자바 옵션은 JAVA_OPTION에서 설정할 수 있습니다. 해당 운영체제의 JAVA_OPTION에서 자세한 옵션 사항을 참조하여 서버 성능에 도움이 되는 옵션 사용을 권장합니다.

■ pushserver.properties

오즈 Push 서버의 홈 디렉토리를 설정합니다.

```
#-----
# Main configuration of 오즈push서버
#-----

path.oz.push.home=..₩
```

옵션	설명
path.oz.push.home	오즈 Push 서버 홈 디렉토리 경로를 설정합니다. (상대/절대 경로 설정 가능)

■ pushserver.xml

오즈 Push 서버의 로그, session, transmission, Observer를 설정 홈 디렉토리를 설정합니다.

```
<?xml version="1.0" encoding="EUC-KR"?>
<ozpushserver port="8088">
  <log priority="DEBUG" path="../logs/">
    <logger type="Console" aync="false" />
    <!--
      DailyRollingFile 로거를 설정할때 사용
    -->
    <logger type="DailyRollingFile" filename="pushserver.log" append="true" immediateFlush="true"
datepattern="yyyy-MM-dd" />
    <!--
    <logger type="RollingFile" filename="pushserver.log" append="true" immediateFlush="true"
maxbackupindex="5" maxFileSize="10KB" />
  </log>
  <session interval='5000'/>
  <transmission multi='false'/>
  <observers>
```

```

<!--
  사용자가 개발한 observer 등록
  <observer name="OB1" classname="sample.observer.OB1" checkInterval="5000" />
  <observer name="OB2" classname="sample.observer.OB2" checkInterval="5000"/>
-->
</observers>
</ozpushserver>#-----
# Main configuration of ozpushserver
#-----

```

Element	옵션	설명	
ozpushserver	port	오즈 Push 서버의 포트를 설정합니다. (기본값:8088)	
log	priority	로그 레벨을 설정합니다. (기본값:DEBUG)	
		ERROR	log level 중 낮은 단계이며, 실제로 에러 상황에서만 에러 메시지를 log에 기록합니다.
		WARN	"ERROR" level에서의 log 출력과 함께 에러는 발생하나 프로그램 동작에 크게 영향을 미치지 않는 에러에 대해서 log 기록합니다.
		INFO	"ERROR" level에서의 log 출력과 함께 서버의 트랜잭션 정보를 log에 기록합니다.
		DEBUG	"ERROR", "INFO" level에서의 log 출력과 함께 기타 전반적인 서버 동작 정보를 log에 기록합니다.
	path	로그 파일이 생성될 디렉토리를 설정합니다. (기본값 : %OZPushHome%\log)	
logger	type	logger 타입 즉, 로그의 출력 형태를 설정합니다. (기본값:Console) ※ 참고사항 : logger 타입은 복수로 설정할 수 있으며, 이때 Console과 RollingFile 형태 또는 Console과 DaillyRollingFile 형태로만 설정할 수 없습니다.	
		Console	로그를 콘솔로 출력합니다.
		RollingFile	로그를 기본 형태의 로그 파일로 출력합니다. ex) pushserver.log, pushserver.log.1, pushserver.log.2, ...

		DaillyRollingFile	로그 파일로 출력하되 파일을 날짜 패턴 (DatePattern)에 설정한대로 생성합니다. ex) pushserver.log, pushserver.log.날짜패턴, pushserver.log.날짜패턴, ...
	async		logger 타입이 Console일 경우 버퍼링 여부를 설정합니다. (기본값:false)
	filename		logger 타입이 RollingFile 또는 DaillyRollingFile일 경우 로그 파일의 기본값을 설정합니다. (기본값:pushserver.log)
	append		logger 타입이 RollingFile 또는 DaillyRollingFile일 경우 서버가 재구동될 때 기존의 로그 파일에 로그를 추가할지 여부를 설정합니다. (기본값:true)
	immediateFlush		logger 타입이 RollingFile 또는 DaillyRollingFile일 경우 버퍼에 있는 로그를 바로 기록할 것인지를 설정합니다. (기본값:false)
	datepattern		logger 타입이 DaillyRollingFile일 경우 날짜별로 로그 파일을 생성할 때 생성할 날짜 패턴을 설정합니다. 날짜 패턴은 아래 "DatePattern" 부분을 참조하시기 바랍니다. (기본값:YYYY-MM-DD)
	maxBackupIndex		logger 타입이 RollingFile일 경우 Backup될 로그 파일의 최대 개수를 정의합니다. 생성된 로그 파일이 최대 파일 개수를 초과할 경우에는 가장 오래된 파일이 삭제되고 새로운 파일이 생깁니다. (기본값:5)
	maxFileSize		logger 타입이 RollingFile일 경우 로그 파일의 최대 사이즈를 설정합니다. (기본값:1MB)
session	interval		세션 체크 간격을 밀리세컨드 단위로 설정합니다. (기본값:5000)
transmission	multi		데이터 전송 시 새로운 스레드를 생성하여 전송할지 여부를 설정합니다. (기본값:false)
Observer	name		Observer의 이름을 설정합니다.
	classname		Observer의 클래스 이름을 설정합니다.
	checkInterval		데이터 변화 감지 체크 간격을 밀리세컨드 단위로 설정합니다. (기본값:5000)

- DatePattern

날짜 패턴	로그 파일 생성 시간	예
-------	-------------	---

yyyy-MM	매달 초	6월의 마지막 밤 12시에 pushserver.log는 pushserver.log.2005-06으로 바뀌며 이후의 로그는 다시 pushserver.log에 기록됩니다.
yyyy-ww	매주 초 (매주 초는 지역에 따라 다를 수 있음)	주초가 일요일이며, 1월 첫째 주 토요일 밤 12시에 pushserver.log는 pushserver.log.2005-01로 바뀌며 이후의 로그는 다시 pushserver.log에 기록됩니다.
yyyy-MM-dd	매일 자정 (기본값)	7월 1일 자정에 pushserver.log는 pushserver.log.2005-07-01로 바뀌며 이후의 로그는 pushserver.log에 기록됩니다.
yyyy-MM-dd-HH	매 시 정각	7월 1일 10시 정각에 pushserver.log는 pushserver.log.2005-07-01-09로 바뀌며 이후의 로그는 pushserver.log에 기록됩니다.

오즈 Push 서버 콘솔 명령어

■ -help

콘솔 명령어의 사용법을 출력합니다.

예) `ozpushcntrl -help`

■ -obinfo

해당 ObserverID의 정보를 출력합니다.

출력 형식은 다음과 같습니다.

ObserverID | ObserverName | ObserverClassName | 등록된 client 의 개수

예) `ozpushcntrl -obinfo ip:127.0.0.1 port:8088 id:ObserverID`

■ -oblist

요청한 ip 와 포트에 있는 모든 Observer 정보를 출력합니다.

출력 형식은 다음과 같습니다.

ObserverID | ObserverName | ObserverClassName | 등록된 client 의 개수

예) `ozpushcntrl -oblist ip:127.0.0.1 port:8088`

■ -obruninfo

현재 동작중인 모든 Observer 의 정보를 출력합니다.

출력 형식은 다음과 같습니다.

ObserverID | ObserverName | ObserverClassName | 등록된 client 의 개수

예) `ozpushcntrl -obruninfo ip:127.0.0.1 port:8088`

■ -obstop

ObserverID를 가지는 Observer의 구동을 중지시킵니다.

예) `ozpushcntrl -obstop ip:127.0.0.1 port:8088 id:ObserverID`

■ -stop

OZ Push 서버를 중지합니다

예) `ozpushcntrl -stop ip:127.0.0.1 port:8088`

■ **-usage**

start 명령의 사용법을 보여줍니다.

예) `ozpushcntrl -usage`

■ **-version**

OZ Push 서버의 버전 정보를 확인합니다.

예) `ozpushcntrl -version -ip:127.0.0.1 -port:8080`