



XDB

XDB / JAVA

※ XDB 설치 가이드는 **RDBMS**에 관한 기본적 지식을 가진 사람은 누구나 설치 가능하도록 작성되었습니다. 문서내용을 순서대로 확인하면서 XDB 를 설치하시기를 당부 드립니다.



RDBMS란?

행과 열로 된 2차원의 표로 데이터를 표현하는 데이터베이스 관리 시스템. 대규모 컴퓨터 시스템을 대상으로 많은 이용자가 대량의 데이터를 다룰 때 데이터베이스를 관리하는 시스템으로서, PC용 데이터베이스에 비해 대량의 데이터를 취급할 수 있고 신뢰성 측면에서도 뛰어난 업무 시스템을 구축할 수 있다. 구조화 조회 언어(SQL)로 작성된 데이터 조작용 명령에 따라 서버상의 데이터를 조작하며, 수정 등의 명령을 서버로 보낼 수 있는 것이면 어떤 클라이언트에서도 서버의 데이터를 조작할 수 있다.



XDB for JAVA 란?

XDB for JAVA는 ㈜쏜다넷의 SMS/URL/MMS/BARCODE G/W 서비스를 사용하기 위한 고객이 고객의 시스템과 쉽게 연동할 수 있도록 도와 주는 SMS/URL/MMS/BARCODE G/W 도우미로서, 고객사의 DB 서버 및 JDBC연결이 가능한 시스템에 설치된다.



설치 가능 OS는?

XDB for JAVA는 **Java 5 이상**이 운영 가능한 Windows 또는 Linux 시스템에서 설치하실 수 있습니다. (**Java 10 이상은 지원하지 않습니다.**)



설치 가능 RDBMS는?

지원RDBMS	지원버전	비고
MYSQL	3.23.xx 이상	MYSQL 4.xx.xx이상 최신버전 권장
MSSQL	6.5, 7.0, 2000, 2005, 2008	SP3 설치권장
ORACLE	7i 이상 (8i , 9i, 10g)	반드시 서버버전과 클라이언트 버전이 동일해야 함
INFORMIX	9.x 이상	최신버전 권장
DB2	9.xx 이상	최신버전 권장
SYBASE	15 이상	최신버전 권장
CUBRID	7.3, 2008 R1.0, 2008 R2.0	최신버전 권장

【참고】

1. RDBMS 서버의 위치는 리모트인 경우는 RDBMS가 설치된 운영체제(OS)와는 상관이 없습니다.
2. XDB 설치서버에 JDK 및 RDBMS 클라이언트 설치 및 설정이 필요합니다.
3. RDBMS가 XDB와 같은 시스템에 설치, 운영이 된다면 별도의 클라이언트 프로그램이 필요하지 않을 수도 있습니다.
4. XDB는 고객사의 DB에 접속하여 전송, 결과 및 MMS 테이블을 생성합니다.

XDB를 설치하기 전 Database, DB User를 우선 생성하셔야 합니다. (각 DBMS 매뉴얼 및 책자 참조)

RDBMS XDB for JAVA – Installation (UNIX, LINUX)

1. 필요 S/W: XDB for JAVA용 Package, DATABASES, JDK

2. 다운로드 받은 XDB 파일 확인 후, 적당한 위치에 압축을 해제합니다.

XDB 파일형식 예: **xdb2_for_java_yyyymmdd.tgz** 또는 **xdb2_for_java_yyyymmdd.zip**

로그 폴더와 MMS콘텐츠 폴더는 자동으로 생성이 됩니다.

위치를 변경하고자 하는 경우에는 xdb.conf를 수정하여 주십시오.

설치된 파일들의 사이즈는 XDB파일의 버전에 따라 다를 수 있습니다.

3. JDK 설치

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

상기 경로에서 서버 환경에 맞는 JDK를 다운로드 받으신 후 XDB를 설치할 서버에 먼저 설치하셔야 합니다. (Java 5~8 권장 / **Java 10** 이상은 지원하지 않습니다. / 설치 후 자동업데이트 해제 필수!!)

4. 2번과 같이 다운받은 XDB파일의 압축을 풀었다면, xdb.conf.sample 파일을 xdb.conf 로 변경 혹은 copy하여, 아래내용을 수정합니다.

```
#####  
#  
# XDB Config File  
#  
#####  
#  
# Host Info(접속정보)  
#  
sms_ip=SMRS IP Address(쏜다넷 G/W 서버 IP)  
sms_port=4000  
sms_id=고객사 고유ID  
sms_pwd=고객사 비밀번호  
  
#####  
# 로그 패스  
# log file이 기록될 디렉토리 설정 생략시 현재 실행 디렉토리  
# 예) 윈도우 => C:\\XDB\\Log 또는 C:/XDB/Log 로 해야 디렉토리가 생성됨.  
log_path=
```

```
#####
# MMS 패스
# 셋팅시 MMS메시지로 전송 할 SKN파일이 만들어 진다.
# 값이 없을시 skn 파일을 생성하지 않고 전송한다.
# skn 팩킹 파일이 만들어질 디렉토리
# 예) 윈도우 => C:\XDB\SKNFile\ 또는 C:/XDB/SKNFile/ 로 해야 디렉토리가 생성됨.
# SKNFile 하위에 날짜별 디렉토리가 생성되어 skn 파일이 저장됨
# 경로뒤에 / 또는 \ 가 없을 경우 SKNFile20150101 과 같이 생성되므로 / 또는 \ 를 포함해야 함
sknfile_path=

#####
# Que 패스
# XDB 내부에서 사용할 Que 저장 디렉토리
# 예) 윈도우 => C:\\XDB\\QUE 또는 C:/XDB/QUE 로 해야 디렉토리가 생성됨.
que_path=

#####
# 바코드 이미지 패스
# 예) 윈도우 => C:\\XDB\\BARCODE_FILES 또는 C:/XDB/BARCODE_FILES 로 해야 디렉토리가 생성됨.
barcode_files_path=

#####
# 임시저장 패스
# 예) 윈도우 => C:\\XDB\\TEMP 또는 C:/XDB/TEMP 로 해야 디렉토리가 생성됨.
temp_path=

#####
# 로그 파일 삭제 옵션 (단위:일 | 기본 : 60)
# log_path 에 설정한 폴더에 있는 log 파일을 삭제한다.
# 생략시 60일
# 해당 파일을 삭제하지 않을 시 0 로 입력
# 예) log_file_life_time=60 (log_path 폴더에 있는 log 파일 중 60일 이후 파일은 삭제함)
# log_file_life_time=0 (log_path 폴더에 있는 log 파일을 삭제하지 않음)
# log_file_life_time=60

#####
```

```
# MMS 파일 삭제 옵션 (단위:일 | 기본 : 60)
# sknfile_path 에 설정한 폴더에 있는 SKN 파일을 삭제한다.
# 생략시 60일
# 해당 파일을 삭제하지 않을 시 0 로 입력
# 예) skn_file_life_time=60 (sknfile_path 폴더에 있는 SKN 파일 중 60일 이후 파일은 삭제함)
#     skn_file_life_time=0 (sknfile_path 폴더에 있는 SKN 파일을 삭제하지 않음)
# skn_file_life_time=60

#####
# 바코드 이미지 파일 삭제 옵션 (단위:일 | 기본 : 60)
# barcode_files_path 에 설정한 폴더에 있는 이미지 파일을 삭제한다.
# 생략시 60일
# 해당 파일을 삭제하지 않을 시 0 로 입력
# 예) barcode_file_life_time=60 (barcode_files_path 폴더에 있는 이미지 파일 중 60일 이후 파일은 삭제함)
#     barcode_file_life_time=0 (barcode_files_path 폴더에 있는 이미지 파일을 삭제하지 않음)
# barcode_file_life_time=60

#####
# 접속 암호 및 전송시 수신번호 암호와 옵션
# npro_sign On , Off
# 예) npro_sign=Off
#
npro_sign=On

#####
# log 테이블 이동 모드(기본 : BATCH)
# BATCH : 전송 완료된 메시지를 일정시간마다 log 테이블로 이동
# REAL  : 전송 완료된 메시지를 실시간으로 log 테이블로 이동
# 예) npro_logmode=BATCH
#
npro_logmode=BATCH

#####
# 로그테이블을 MSG_LOG_YYYYMM 형태로 여러 개를 할 것인지
# MSG_LOG 하나로 할 것인지 선택 모드(기본 : DEFAULT)
# DEFAULT : 로그테이블을 기존 방식과 같이 매년매월로 나누어 생성함.
```

```
# ONE : 로그테이블을 MSG_LOG 테이블로 하나만 생성함.
# 예, npro_logmakemode=ONE
#
npro_logmakemode=DEFAULT

#####
# XDB의 장애시 동작 지속 시간 추가(단위:시간)
# Socket 및 DB 연결이 안되고 장애가 발생하고 있을 때 동작 지속 시간이
# 초과하면 XDB의 기동을 모두 중지함
# 생략시 36시간
# 해당 장애시간에 상관없이 계속 연결을 원할시 0시로 입력
# 예) npro_life_time=1 (1시간 동안만 동작을 지속하고도 장애시 XDB로 종료함)
#     npro_life_time=0 (장애가 있어 연결이 안되고 있어서 XDB의 동작을 계속 유지함)
npro_life_time=36

#####
#레포트를 기다리는 시간 , 이 시간이 지날경우 더 이상 레포트를 기다리지 않고 ,
#해당레코드를 로그 테이블로 이동시킨다.
record_life_time=24

#####
# 데이터베이스 유형(db_name)
# ORACLE, MSSQL, MYSQL, SYBASE, DB2, INFOMIX, CUBRID
#
# Database Driver
#
# ORACLE
# db_name=ORACLE
# db_driver=oracle.jdbc.driver.OracleDriver
# db_url=jdbc:oracle:thin:@DB서버주소:1521:ORCL
#
# MSSQL (Microsoft SQL Server 6.5, 7, 2000, 2005 and 2008)까지 지원
# db_name=MSSQL
# db_driver=net.sourceforge.jtds.jdbc.Driver
# db_url=jdbc:jtds:sqlserver://DB서버주소:1433/데이터베이스명
#
# MYSQL (MYSQL 3.x.xx , MYSQL 5.x.xx 이상)
```

```
# db_name=MYSQL
# db_driver=com.mysql.jdbc.Driver
#db_url=jdbc:mysql://DB서버주소:3306/데이터베이스
명?autoReconnect=true&useUnicode=true&characterEncoding=euc-kr
#
# MYSQL (MYSQL 4.x.xx)
# db_name=MYSQL
# db_driver=com.mysql.jdbc.Driver
#db_url=jdbc:mysql://DB서버주소:3306/데이터베이스
명?autoReconnect=true&useUnicode=true&characterEncoding=euckr
#
# SYBASE
# db_name=SYBASE
# db_driver=com.sybase.jdbc3.jdbc.SybDriver
# db_url=jdbc:sybase:Tds:DB서버주소:5000/데이터베이스명?CHARSET=eucksc
#
# DB2
# db_name=DB2
# db_driver=com.ibm.db2.jcc.DB2Driver
# db_url=jdbc:db2://DB서버주소:50000/데이터베이스명
#
# INFORMIX
# db_name=INFORMIX
# db_driver=com.informix.jdbc.IfxDriver
# db_url=jdbc:informix-sqli://DB서버주소:555/데이터베이스:informixserver=INFORMIXSERVER_NAME
#
# CUBRID
# db_name=CUBRID
# db_driver=cubrid.jdbc.driver.CUBRIDDriver
# db_url=jdbc:cubrid:DB서버주소:33000:데이터베이스명:::
#

db_name=ORACLE
db_driver=oracle.jdbc.driver.OracleDriver
db_url=jdbc:oracle:thin:@DB서버주소:1521:ORCL

#####
```



```
# 데이터베이스 CharSet(db_charset)
# MSSQL, ORACLE, MYSQL, SYBASE, DB2, INFOMIX, CUBRID 공통사용
# DBMS에 설정된 CharSet으로 설정한다.
# DBMS마다 euckr, euc_kr, eucksc 으로 명칭이 틀릴경우 EUC-KR로 사용.
db_charset=EUC-KR

#####
# 데이터베이스의 계정설정(테이블생성권한필요함)
# MSSQL, ORACLE, MYSQL, SYBASE, DB2, INFOMIX 공통사용
#
db_userid=데이터베이스 접속아이디
db_password=데이터베이스 접속암호

#####
# 사용하는 테이블 명을 변경하는 옵션
# 기존에 테이블 명을 사용자가 임의의 테이블 명으로 변경할 수 있다.
# 아래 예와 같이 설정을 하면
# MSG_DATA, MSG_LOG, MMS_CONTENTS_INFO 와 같이 기본적으로 생성되는 테이블 명이
# MSG_DATA_2, MSG_LOG_2, MMS_CONTENTS_INFO_2 와 같이 생성되어 XDB가 기동된다.
# 예) db_table_name.MSG_DATA=MSG_DATA_2
#db_table_name.MSG_LOG=MSG_LOG_2
#db_table_name.MMS_CONTENTS_INFO=MMS_CONTENTS_INFO_2
#db_table_name.MSG_SPAM=MSG_SPAM
#
#db_table_name.MSG_DATA=MSG_DATA
#db_table_name.MSG_LOG=MSG_LOG
#db_table_name.MMS_CONTENTS_INFO=MMS_CONTENTS_INFO
#db_table_name.MSG_SPAM=MSG_SPAM

#####
# 사용하는 시퀀스 명을 변경하는 옵션
# 기존에 시퀀스 명을 사용자가 임의의 시퀀스 명으로 변경할 수 있다.
# 아래 예와 같이 설정을 하면
# MSG_DATA_SEQ, MSG_LOG_SEQ, MMS_CONTENTS_INFO_SEQ 와 같이 기본적으로 생성되는 시퀀스
명이
# MSG_DATA_SEQ2, MSG_LOG_SEQ2, MMS_CONTENTS_INFO_SEQ2 와 같이 생성되어 XDB가 기동된
```

다.

```
# 예) db_table_name.MSG_DATA=MSG_DATA_SEQ2
```

```
#db_table_name.MMS_CONTENTS_INFO=MMS_CONTENTS_INFO_SEQ2
```

```
#db_table_name.MSG_SPAM=MSG_SPAM_SEQ2
```

```
#
```

```
#db_sequence_name.MSG_DATA=MSG_DATA_SEQ
```

```
#db_sequence_name.MMS_CONTENTS_INFO=MMS_CONTENTS_INFO_SEQ
```

```
#db_sequence_name.MSG_SPAM=MSG_SPAM_SEQ
```

```
#####
```

```
# 데이터베이스의 사용자 필드, 사용자 인덱스 추가 가능한 총량
```

```
# 필드 및 인덱스는 최대 10개만 가능하다.
```

```
# 예) db_max_count=5 <-- db_field,db_index 를 5개 이상 설정해 놓아도
```

```
# 5개까지만 추가가 된다.(최대 10개)
```

```
#
```

```
#db_max_count=10
```

```
#####
```

```
# 데이터베이스의 사용자 필드추가 옵션
```

```
# 생성할 필드명과 데이터 타입 및 옵션 사이엔 스페이스로 구분해 줘야 한다.
```

```
# 아래 예와 같이 설정을 하면 2개의 필드가가 생성되며
```

```
# 테이블 맨 마지막에 MSG_ETC1, MSG_ETC2 필드가 추가된다.
```

```
# 예) db_field=MSG_ETC1 varchar(10)
```

```
# db_field=MSG_ETC2 varchar(10)
```

```
#
```

```
#db_field=
```

```
#####
```

```
# 데이터베이스의 사용자 인덱스추가 옵션
```

```
# 생성할 인덱스의 필드와 필드 사이는 콤마로 구분해 줘야 하며
```

```
# 정렬(오름차순:ASC, 내림차순:DESC) 도 설정할 수 있다.
```

```
# 아래 예와 같이 설정을 하면 3개의 인덱스가 생성되며
```

```
# 인덱스명은 "IDX_로그테이블명_순서"로 생성이된다.
```

```
# 예) db_index=MSG_SEQ
```

```
# db_index=CUR_STATE DESC, REQ_DATE ASC
```

```
# db_index=CALL_TO
```

```
#
#db_index=

#####
# The time when sender query to DBMS (sec)
# 발송 데이터 조회 time(초)
#
tran_term=1

#####
# The time when connection query to DBMS (sec)
# DBMS의 연결유지를 위한 조회 time(초)
# 생략시 60초
#
#db_ping_term=60

#####
# The time when sender and receiver send ping to Gateway (sec)
# 연결 상태 확인 term(초)
#
ping_term=60

#####
# The time when sender query to DBMS
# "ORDER BY MSG_DATA.MSG_SEQ" 사용여부(On, Off | 기본 : On)
# SELECT 시 ORDER BY MSG_DATA.MSG_SEQ 를 사용하면
# DB 서버에 부하가 약간 생기는 문제로 업체가 요청한 옵션.
# Off 하면 SELECT 시 CUR_STATE = 1의 값을 무작위로 가져오므로
# 메시지 사이사이 늦게 전송되는 것이 있음.
#
# db_orderby : "ORDER BY MSG_DATA.MSG_SEQ" 사용여부(On, Off)
#
#db_orderby = On

#####
# Sender, Receiver, Spam On , Off
# Off일 경우에만 해당 쓰레드가 작동하지 않는다...
```

```
#
sender_switch=ON
receiver_switch=ON
spam_switch=ON

#####
# SPAM 리로드 설정 (단위 분)
# 생략시 480분
#spam_reload_term=480

#####
# SMS 전송문자에 특정키워드가 없으면 SPAM 처리하여
# 메시지가 전송되지 않는다.
# 예) include_word=TEST
#      include_word=테스트
#
#include_word=

#####
# 하루에 한 수신번호에 보낼수 있는 메시지 수량(기본 : 10개)
# 생략시 10개
# 0개로 셋팅시 사용안함
#duplicate_phone_count=10

#####
# Limit Time Zone (발송제한 시간대 설정)
# 발송제한 시작 시간 (0100 ~ 2400) 분 단위까지 적용가능, 생략시 24시간 전송 가능.
# 예) limit_begin=2100
# 예) limit_end=0600
#
limit_begin=
limit_end=

#####
# Before Time (단위 시간)
# 현재 시간으로부터 설정 시간 이전 시간 까지만 발송.
# 생략시 3시간
```

before_time=

#####

Table Move (em_tran의 값을 em_log_YYYYMM 테이블로 이동 유무 설정)

db_nomove : 결과 수신이후 LOG테이블로 이동하지 않음(1)

db_npbatch : 정기 BATCH작업을 하지 않는다.(1)

db_nomove, db_npbatch : 동일한 값을 지정을 설정

db_nomove=

db_nobatch=

#####

log 기록 범위 설정

FATAL : XDB의 치명적에러

ERROR : XDB가 수행가능정도의에러

WARN : XDB가 시스템에 문제를 일으킬가능성이 있을때 사용

INFO : XDB의 정보용으로 사용

ex)loglevel="FATAL","ERROR","WARN","INFO"

전송클라이언트로그, 송신클라이언트로그, 정보로그

consol.log_level=INFO

system.log_level=INFO

send.log_level=INFO

recv.log_level=INFO

info.log_level=INFO

5. Xdb.conf 정상 수정 여부를 터미널에서 확인합니다.

- 실행 방법

```
$nohup java -jar ./xdb2.jar ./xdb.conf > /dev/null 2>&1 &
```

- XDB가 설치되어 있는 디렉토리에서 터미널을 연 후, 위와 같이 실행하여 Exception이 나오지 않으면 정상적으로 xdb.conf 정보를 수정한 것이고 Exception이 나오면 해당 에러의 로그 부분은 수정하시면 됩니다.
- 서비스 실행시 자동 생성되는 디렉토리가 root 또는 다른 계정으로 이미 생성되어 있는 상태에서 일반 계정으로 실행할 때 권한 문제로 인한 Exception이 발생할 수 있습니다.

6. XDB Log 파일 확인

- XDB Log 파일은 동작 내용 및 에러 로그 수집을 위해 일자 별로 지정된 디렉터리에 생성됩니다.
생성되는 위치는 xdb.conf 파일의 log_path 에서 설정하실 수 있습니다.
- SMS Gateway 접속 오류 및 DB오류가 연속적으로 발생할 경우 XDB 정상 설치 여부를 확인하셔야 합니다.

7. DB Table 생성 확인

- XDB 프로그램이 4개의 테이블을 자동 생성합니다.
MSG_DATA, MMS_CONTENTS_INFO, MSG_LOG_YYYYMM, MSG_SPAM
실제 메시지 전송을 하면 이동 통신사로부터 전송 결과를 받아 MSG_DATA와
같은 형식으로 Log Table을 월별 생성합니다.

8. DB Table이 정상적으로 생성 되었다면 사용자설명서를 참고하여 전송 테스트를 하시면 됩니다.

- 장문/멀티 전송시 한글이 깨져서 수신되거나, RSLT_CODE(400), RSLT_CODE2(C) 와 같은 결과로 전송실패 할 경우 제목에 포함된 특수문자(<,>,&)를 제거하거나, 현재 세션의 로케일 확인 후 OS에서 지원되는 한글 캐릭터셋으로 변경해야 합니다. (요약설명서 참고)
- 단문 전송시 한글이 깨져서 수신된다면 xdb.conf 에서 DB 캐릭터셋을 확인해주세요.

9. 추가적인 예외 상황 또는 자동 실행 설정 등의 작업이 필요할 경우 요약설명서를 참고하세요.

RDBMS XDB for JAVA – Installation (Windows / NT)

1. 필요 S/W: XDB for JAVA용 Package, DATABASES

2. 다운로드 받은 XDB 파일 확인 후, 적당한 위치에 압축을 해제합니다.

XDB 파일형식 예: **xdb2_for_java_yyyymmdd.tgz** 또는 **xdb2_for_java_yyyymmdd.zip**

로그 폴더와 MMS콘텐츠 폴더는 자동으로 생성이 됩니다.

위치를 변경하고자 하는 경우에는 xdb.conf를 수정하여 주십시오.

설치된 파일들의 사이즈는 XDB파일의 버전에 따라 다를 수 있습니다.

(설치가이드에서는 D:\XDB\ 를 예로 설명하겠습니다.)

이름	크기	유형
JavaService32.exe	96KB	응용 프로그램
JavaService64.exe	136KB	응용 프로그램
serviceinstall32.bat	1KB	Windows 배치 파일
serviceinstall64.bat	1KB	Windows 배치 파일
serviceuninstall32.bat	1KB	Windows 배치 파일
serviceuninstall64.bat	1KB	Windows 배치 파일
xdb.conf.sample	12KB	SAMPLE 파일
xdb_process.sh.sample	3KB	SAMPLE 파일
xdb_run.sh.sample	1KB	SAMPLE 파일
xdb2.jar	6,509KB	JAR 파일
XDB2 for JAVA 사용자설명서v2.0.pdf	769KB	Adobe Acrobat Document
XDB2 for JAVA 설치가이드v2.0.pdf	594KB	Adobe Acrobat Document
XDB2 for JAVA 추가설명서.pdf	368KB	Adobe Acrobat Document

3. JDK 설치

RDBMS XDB for JAVA - Installation (UNIX, LINUX) 3번 설명 동일함.

4. 2번과 같이 다운받은 XDB파일의 압축을 풀었다면, xdb.conf.sample 파일을 xdb.conf 로 변경 혹은 copy하여, 아래내용을 수정합니다.

RDBMS XDB for JAVA - Installation (UNIX, LINUX) 4번 설명 동일함.

5. xdb.conf 정상 수정여부를 CMD(Command Mode)에서 확인합니다.

D:\XDB>java -jar xdb2.jar xdb.conf 를 실행하여 Exception이 나오지 않으면 정상적으로 **xdb.conf** 정보를 수정한 것이고 Exception이 나오면 해당 에러의 로그 부분은 수정하시면 됩니다.

6. XDB 실행 파일(jar)을 서비스에 등록 및 시작

- 명령 프롬프트(CMD)에서 XDB 파일이 위치한 디렉터리로 이동한 후, 다음을 실행합니다.
- 아래 XDB 파일이 D:\XDB\ 에 있는 경우에 대한 예시입니다.

서비스 등록 방법 (서비스 이름을 xonda XDB2 라고 한 경우)

윈도우 32bit 시스템의 경우 **serviceinstall32.bat**를

윈도우 64bit 시스템의 경우 **serviceinstall64.bat**를 사용합니다.

(64bit 시스템에 JDK 32bit 설치한 경우 serviceinstall32.bat 사용)

아래 설명은 **32bit**의 시스템으로 설명합니다.

serviceinstall32.bat 을 **notepad**로 열고 아래처럼 수정합니다.

```

1 set JAVA_HOME=C:\java\jdk1.5.0_22
2
3 set XDB2_HOME=D:\XDB
4 set XDB2_SERVICE_NAME=xonda XDB2
5
6 set XDB2_NAME=xdb2.jar
7 set XDB2_CONFIGING=xdb.conf
8
9 "%XDB2_HOME%\JavaService32.exe" -install "%XDB2_SERVICE_NAME%" "%JAVA_HOME%\jre\bin\server\
10 jvm.dll" -Djava.class.path="%XDB2_HOME%\%XDB2_NAME%" -start kr.co.iheart.NPROMain -params
11 %XDB2_CONFIGING% -current "%XDB2_HOME%" -auto -description "%XDB2_SERVICE_NAME% Java Service"

```

1번째 줄 : set JAVA_HOME=자바 JDK를 설치한 폴더의 절대 패스를 기입합니다.

3번째 줄 : set XDB2_HOME=XDB를 압축 해제한 폴더의 절대 패스를 기입합니다.

4번째 줄 : set XDB2_SERVICE_NAME=서비스에서 사용할 XDB 명을 기입합니다.

6번째 줄 : set XDB2_NAME=XDB 실행 jar 모듈을 기입합니다.

7번째 줄 : set XDB2_CONFIGING=XDB의 xdb.conf 명을 기입합니다.

위 사항 중에 기본 적으로 1번째 줄과 3번째 줄을 수정하여 서비스에 등록하면 됩니다.

D:\XDB>serviceinstall32.bat

```
D:\XDB>serviceinstall32.bat

D:\XDB>set JAVA_HOME=C:\java\jdk1.5.0_22

D:\XDB>set XDB2_HOME=D:\XDB

D:\XDB>set XDB2_SERVICE_NAME=xonda XDB2

D:\XDB>set XDB2_NAME=xdb2.jar

D:\XDB>set XDB2_CONFIG=xdb.conf

D:\XDB>"D:\XDB\JavaService32.exe" -install "xonda XDB2" "C:\java\jdk1.5.0_22\jre\bin\server\jvm.dll" -Djava.class.path="D:\XDB\xdb2.jar" -start kr.co.iheart.NPROMain -params xdb.conf -current "D:\XDB" -auto -description "xonda XDB2 Java Service"
The xonda XDB2 automatic service was successfully installed

D:\XDB>
```

서비스 등록 확인 방법

D:\XDB>sc query "xonda XDB2"

```
D:\XDB>sc query "xonda XDB2"

SERVICE_NAME: xonda XDB2
        종류               : 10  WIN32_OWN_PROCESS
        상태                : 1  STOPPED
        WIN32_EXIT_CODE      : 1077  (0x435)
        SERVICE_EXIT_CODE  : 0  (0x0)
        검사점              : 0x0
        WAIT_HINT           : 0x0

D:\XDB>
```

서비스 시작 방법 (서비스 시작은 임의 경로에서 실행하셔도 됩니다)

D:\XDB>net start "xonda XDB2"

```
D:\XDB>net start "xonda XDB2"
xonda XDB2 서비스를 시작합니다..
xonda XDB2 서비스가 잘 시작되었습니다.

D:\XDB>
```

서비스 실행 실패시 (지정되지 않은 오류가 발생하여 시작되지 않는 경우)

- **방법 1. 환경변수 Path 설정**
C:\java\jdk1.5.0_22\bin

[컴퓨터] -> [속성] -> [고급] -> [환경 변수] -> [시스템 변수] -> [Path] 에 위 경로를 추가한 후 재부팅 하면 자동시작

- **방법 2.** msvcr100.dll 파일 복사

C:\java\jdk1.5.0_22\bin\msvcr100.dll 파일을 XDB 설치경로에 복사 후 서비스 시작

- **방법 3.** msvcr100.dll 파일 복사

C:\java\jdk1.5.0_22\bin\msvcr100.dll 파일을 system32 폴더에 복사 후 서비스 시작

- **기타.** serviceinstall32(또는 64).bat 파일내에 **jvm.dll** 경로 확인

"%JAVA_HOME%\jre\bin\server\jvm.dll"

경로가 다를 경우 올바르게 수정 하신 후 서비스를 제거하고 다시 등록하세요.

서비스 중지 방법

D:\XDB>net stop "xonda XDB2"

```
D:\XDB>net stop "xonda XDB2"
xonda XDB2 서비스를 멈춥니다.....
xonda XDB2 서비스를 잘 멈추었습니다.

D:\XDB>
```

서비스 삭제 방법 (서비스 등록 방법에서 서비스 이름을 XDB2라고 한 경우)

serviceuninstall32.bat 을 notepad로 열고 아래처럼 수정합니다.

```
1 set XDB2_HOME=D:\XDB
2 set XDB2_SERVICE_NAME=xonda XDB2
3
4 "%XDB2_HOME%\JavaService32.exe" -uninstall "%XDB2_SERVICE_NAME%"
```

1번째 줄 : set XDB2_HOME=XDB를 압축 해제한 폴더의 절대 패스를 기입합니다.

2번째 줄 : set XDB2_SERVICE_NAME=서비스에 등록한 XDB명을 기입합니다.

D:\XDB>serviceuninstall32.bat

```
D:\XDB>serviceuninstall32.bat

D:\XDB>set XDB2_HOME=D:\XDB

D:\XDB>set XDB2_SERVICE_NAME=xonda XDB2

D:\XDB>"D:\XDB\JavaService32.exe" -uninstall "xonda XDB2"
Successfully uninstalled service xonda XDB2

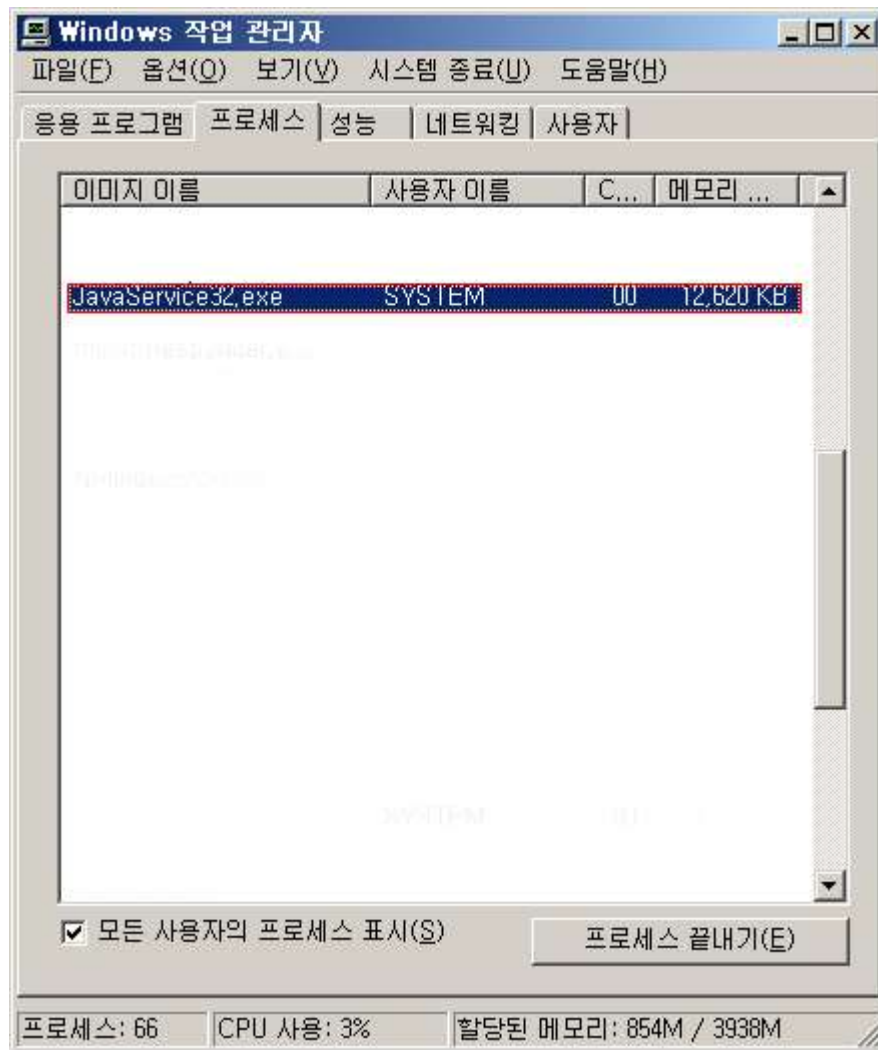
D:\XDB>
```

서비스 등록 현황은 [제어판] -> [관리도구] -> [서비스] 에서 확인할 수 있습니다.

7. XDB 서비스를 시작하였다면 작업관리자에서 프로세스를 확인합니다.

아래와 같이 JavaService32.exe가 실행되어야 합니다.

(윈도우 64Bit의 경우는 JavaService64.exe가 실행)



10.XDB Log 파일 확인

RDBMS XDB for JAVA - Installation (UNIX, LINUX) 6번 설명과 동일함.

11. DB Table 생성 확인

RDBMS XDB for JAVA - Installation (UNIX, LINUX) 7번 설명과 동일함.

12. DB Table이 정상적으로 생성 되었다면 사용자설명을 참고하여 전송 테스트를 하시면 됩니다.