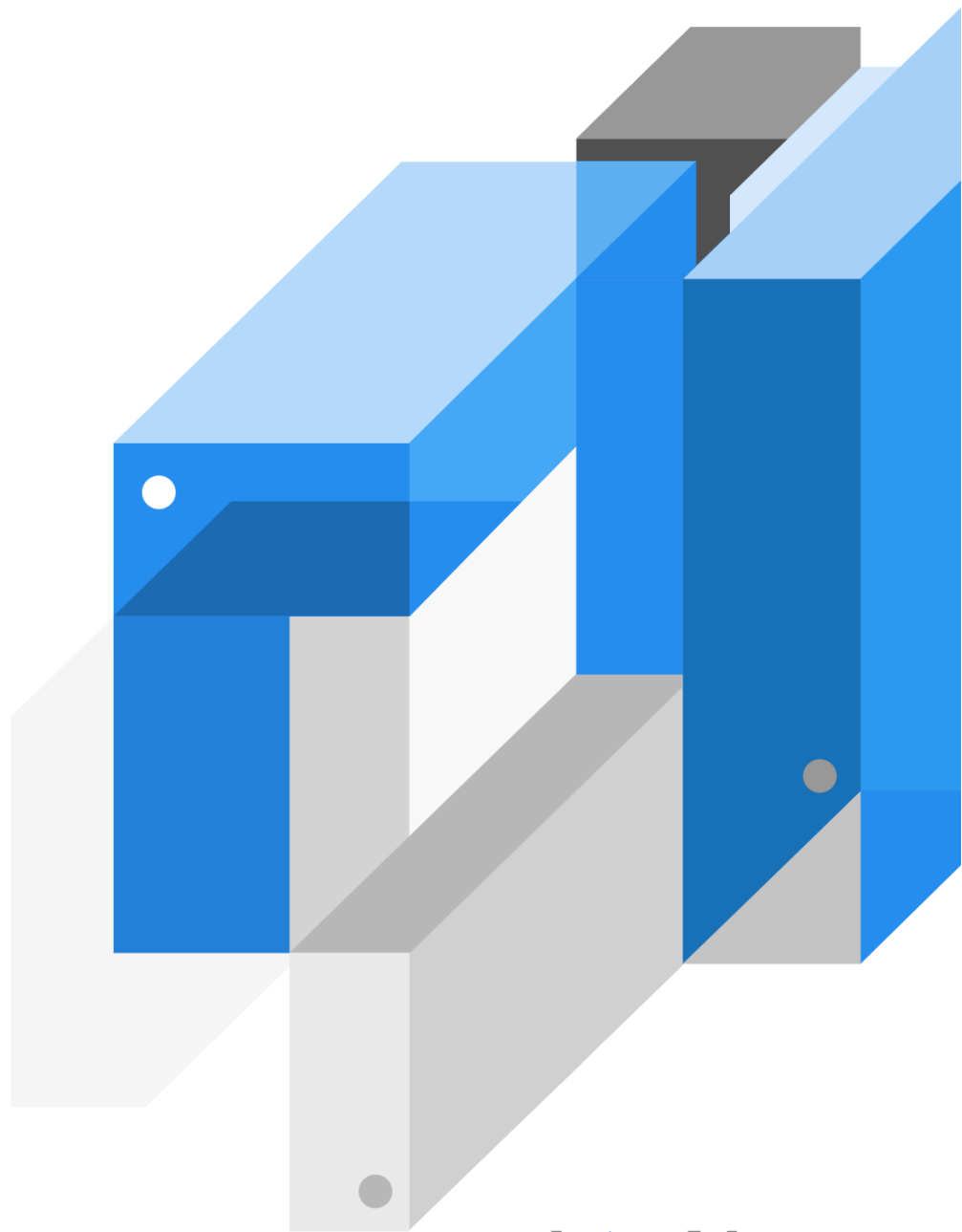


# InterMax

## Installation & Architecture Guide



InterMax

# Table of Contents

1. InterMax Architecture .....	6
1.1. InterMax Network Connection .....	8
Data Collection Layer .....	8
InterMax AP Server Layer .....	9
서비스 포트.....	9
1.2 Compatibility .....	10
JAVA 환경 .....	10
.Net 환경 .....	10
TP 환경.....	10
InterMax 제품 지원 현황 .....	11
1.3. InterMax License.....	12
1.3.1. Trial License Key .....	12
1.3.2. Formal License Key .....	12
2. Data Collection Layer 설치 및 구성 .....	13
2.1 InterMax WAS Agent Set (JSPD) .....	13
2.1.1 사전 준비사항.....	13
Network Port (Windows).....	13
Network Port (Unix / Linux) .....	13
WAS Agent Set 구성.....	14
2.1.2. 설치 절차 .....	14
Windows 환경.....	14
Unix / Linux 환경 .....	15

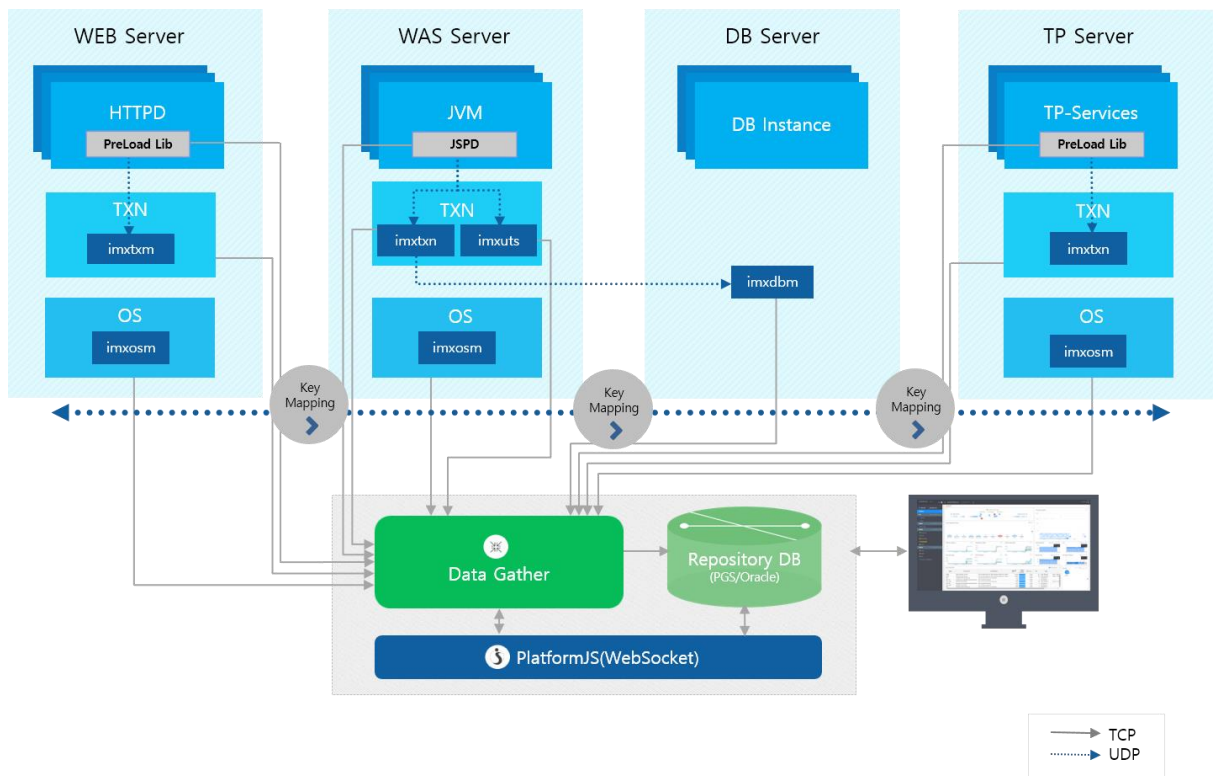
DB 모니터링 연동 시 추가 설정 .....	16
2.1.3. OS 별 추가 설정 .....	16
AIX .....	16
Sun Solaris .....	17
2.1.4 기동 방법 .....	18
2.1.5. InterMax WAS Agent 기동 .....	18
2.2 InterMax DB Agent Set (IMXDBM) .....	18
2.2.1 사전 준비사항 .....	18
OS 사용자 권한 .....	19
InterMax Profile Setting (Windows 환경에서는 생략) .....	19
Oracle Version .....	19
Oracle Instance .....	19
Oracle Numa Segment .....	20
Shared Memory IPC key .....	20
Oracle PMON .....	21
Network Port .....	21
2.2.2. 설치 절차 .....	21
Windows 환경 .....	21
Unix / Linux 환경 .....	24
2.2.3. 기동 방법 .....	28
Windows 환경 .....	28
Unix / Linux 환경 .....	28
2.2.4. 예외 처리 .....	29
MakeConf Script Error (Windows 환경 해당 없음) .....	29

Run by sys.sql Error.....	29
Env & List.conf Error .....	30
3. AP Server 및 Data Storage Layer 설치 및 구성 .....	31
3.1. 사전 준비 사항.....	31
3.1.1. AP 서버 사양.....	31
3.2. Windows 환경 .....	32
3.2.1. 사전 준비사항 .....	32
3.2.2. 설치 절차 (자동-Installer).....	32
3.2.3. 설치 절차 (수동).....	34
PostgreSQL 수동 설치 .....	34
Oracle 수동 설치 .....	38
Data Gatherer 수동 설치 .....	39
Platform.JS 수동 설치 .....	44
3.2.4. 기동 및 접속 확인 .....	45
InterMax Local Services.....	45
InterMax 접속 확인 .....	46
3.2.5. 사용자 정의 옵션 .....	47
Slave Gatherer Process 추가 .....	47
PostgreSQL Tablespace 설정.....	48
3.3. Unix/Linux 환경.....	51
3.3.1. 사전 준비사항.....	51
3.3.2. 설치 절차 (수동).....	51

PostgreSQL 수동 설치 .....	51
Oracle 수동 설치 .....	52
Data Gatherer 수동 설치 .....	53
Platform.JS 수동 설치 .....	57
3.3.3. 기동 방법 .....	58
InterMax PlatformJS 기동.....	58
InterMax 설정 및 확인 방법 .....	58
4. Appendix.....	60
4.1 WAS 벤더 별 InterMax Option 설정 .....	60
4.1.1. JEUS InterMax Option 설정.....	60
4.1.2. WebLogic InterMax Option 설정 .....	61
4.1.3. WebSphere InterMax Option 설정.....	62
4.1.4. Tomcat InterMax Option 설정 .....	63
4.1.5. JBoss InterMax Option 설정.....	63
4.1.6. Resin InterMax Option 설정 .....	64
4.1.7. OC4J(Oracle Containers for J2EE) InterMax Option 설정.....	64
4.1.8. GlassFish InterMax Option 설정 .....	65

# 1. InterMax Architecture

기본적인 WEB~WAS~DB 환경의 모니터링 구성뿐 만 아니라 사용자 단말단부터 기업의 핵심 백엔드 시스템까지 전체 구간을 End-To-End 관점에서 동시에 모니터링 할 수 있게 설계되어 있다.

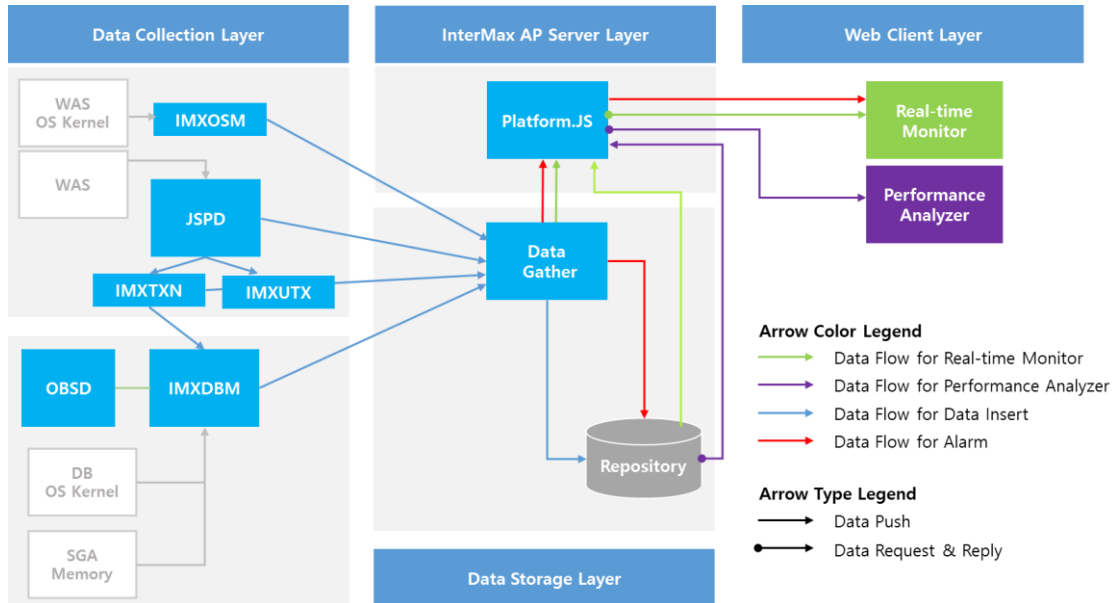


기업의 어플리케이션 모니터링 대상 시스템에 해당 Agent 가 설치되어 성능 데이터를 전달하는 역할을 수행하며, 데이터 수집용 서버와 저장용 데이터베이스가 추가적으로 설치 구성되어야 한다.

WAS(JVM) 모니터링 기준으로 InterMax 내부 엔진 모듈은 다음과 같은 4 개의 기본 Layer 로 구성된다.

- 1 **Data Collection Layer:** 기업의 어플리케이션을 직접 모니터링 하기 위해 설치되는 Agent 영역으로 다양한 성능 데이터를 수집하고 이를 서버에 전달하는 역할을 한다.
- 2 **InterMax Application Server Layer:** InterMax 서버 영역으로 Agent로부터 전송받은 성능 데이터를 수집/분석/가공하여 데이터베이스에 저장하고 구성정보를 관리하는 전용 Web Daemon Server 이다
- 3 **Data Storage Layer:** InterMax 서버에서 가공된 성능 데이터를 저장하는 저장소로 각종 성능 정보 및 분석을 위한 데이터들이 저장되는 영역이다.

- 4 **Web Client Layer:** 수집된 성능 데이터를 실시간 모니터링 및 성능 분석을 위한 웹기반의 사용자 인터페이스를 제공하는 영역이다.



InterMax 에이전트(JSPD)와 독립 프로세스(IMXTXN, IMXUTX)는 다양한 성능 데이터를 수집하고 이를 수집서버(Data Gather)에 전송한다. 수집서버는 전송받은 성능 데이터를 분석하고 가공하여 데이터 저장소(Repository DB)에 저장하며, 각종 구성 정보 등을 관리하게 된다. 저장된 데이터는 웹 클라이언트단의 사용자 인터페이스(HTML5기반)를 통하여 각종 실시간 모니터링과 성능 지표 및 통계 분석 등의 인터페이스를 제공한다.

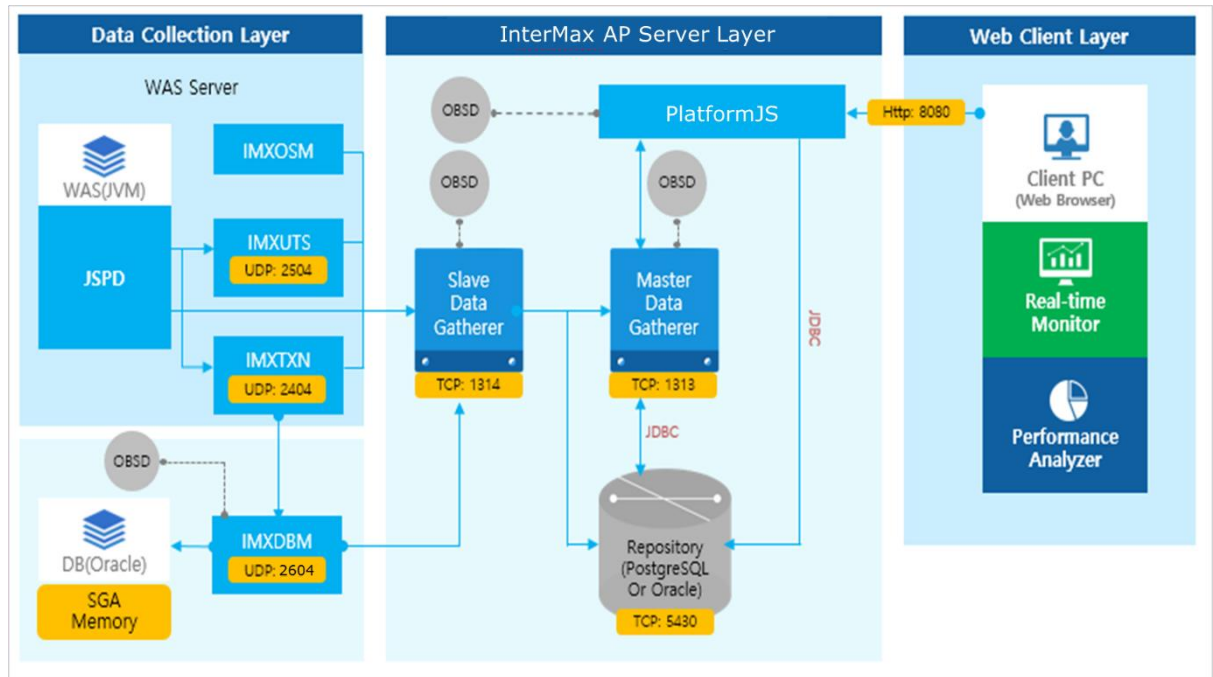
---

**Note.** InterMax AP Server Layer 및 Data Storage Layer 는 Logical 한 구분이다. 두 개의 Layer 는 하나의 서버 내에서 구성할 수 있다. InterMax 아키텍처에 대한 상세 설명은 "[InterMax Administration Guide](#)"를 참고한다.

---

# 1.1. InterMax Network Connection

InterMax 각 Layer 간 다양한 네트워크 통신을 처리하기 위한 필요한 네트워크 포트 등의 구성에 대해 설명한다. InterMax 설치 시, 필요 네트워크 포트는 다음과 같다. 주로 포트 번호 설정과 관련된 것으로 기본 포트 번호를 다른 애플리케이션에서 이미 사용중 일 경우에는 다른 포트로 변경하여 사용해야 한다. 또한 동일한 하드웨어에 복수의 InterMax 환경을 설치한 경우에도 해당 포트 번호가 중복되지 않도록 설정에 주의해야 한다.



## Data Collection Layer

범 주	설 명
JSPD	JVM 내부 thread 로 구동되며 WAS 관련 대부분의 주요 성능 데이터 수집
IMXOSM	OS 의 시스템 리소스(Memory, CPU etc) 및 통계 정보 수집, WAS container 생사 Check.
IMXTXN	SQL 쿼리 정보 수집
IMXDBM	SQL OWI 기준 통계&이벤트 데이터 수집
IMXUTS	Remote 정보 수집 (EtoE)
OBSD	내부 process 감시 역할(30 초 주기로 감시하여 down 되면 재기동 역할)



## InterMax AP Server Layer

범 주	설 명
Slave DataGatherer	Data Collection Layer 에서 보내준 데이터를 수집 처리하는 모듈로 수집 대상수와 부하량에 따라 복수개로 확장하여 구성 가능 일반적으로 50 개 instances 모니터링 당 1 개의 Slave 구성을 권장한다. (Slave 수=instance 수/50)
Master DataGatherer	Slave DataGatherer 관리 역할 및 PlatformJS 로 부터 요청에 대한 정보 제공
PlatformJS	사용자의 웹브라우저를 통한 실시간 모니터링 정보 및 분석 정보 제공 역할
OBSD	내부 process 감시 역할(30 초 주기로 감시하여 down 되면 구동)

## 서비스 포트

Source	Target	Port	Protocol	설 명
JSPD	IMXTXN	2404	UDP	SQL 관련 정보 전송
JSPD	IMXUTS	2504	UDP	Remote 관련 정보 전송 (EtoE)
IMXTXN	IMXDBM	2604	UDP	DB 연계 관련 정보 전송
JSPD	Slave Data Gather	1314	TCP	JVM 주요 성능 정보 전송
IMXOSM				OS 리소스 정보 전송
IMXTXN				SQL 관련 정보 전송
IMXUTS				Remote 관련 정보 전송 (EtoE)
IMXDBM				DB 연계 관련 정보 전송
Slave Data Gather	Master Data Gather	1313	UDP	Server 및 DB Agent 의 정보 전송
Slave Data Gather	Repository	5430	TCP	Server 및 DB Agent 의 정보 저장
Master Data Gather				통계성 정보 저장
PlatformJS	Web Client	8080	TCP	브라우저에 표시되는 정보 전송

## 1.2 Compatibility

InterMax 제품의 지원 범위 및 호환되는 버전은 다음과 같다.

### JAVA 환경

운영체제 (OS)	어플리케이션 서버 (WAS)	지원 DB
AIX 5.x 이상 (32/64bit)	WebLogic 10.x 이상	Oracle
HP-UX IA64	WebSphere 6.1 이상	DB2
Linux (32/64bit)	JEUS 5.x 이상	MS SQL Server
Solaris SPARC (32/64bit)	Tomcat 5.x 이상	Mysql
Solaris (x86/x64)	Oracle Application Server(OC4J)	Postgres
Windows Server 2003 이상 (x86/x64)	Resin 3.x 이상	Sybase
	Jboss 5.x 이상	Tibero
	GlassFish 2.x 이상(JDK 1.5 이상)	

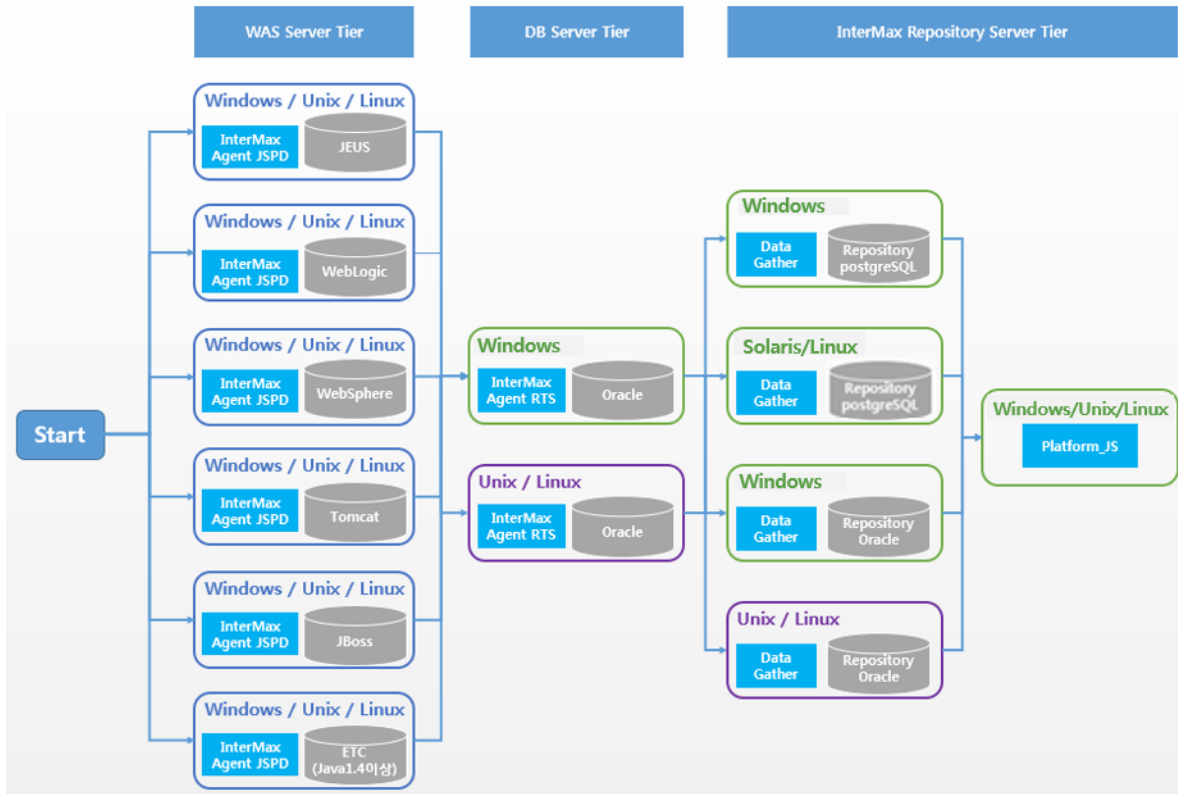
### .Net 환경

운영체제 (OS)	웹서버	어플리케이션 서버	지원 DB
Windows Server 2003 이상 (x86/x64 )	IIS 6.0 이상	.NET Framework 2.0 이상	MS SQL Server 2008 이상

### TP 환경

운영체제 (OS)	어플리케이션 서버 (WAS)	지원 DB
AIX 5.x 이상 (32/64bit)	TMAX 5.x 이상	Oracle
HP-UX IA64	TUXEDO 10.x 이상	DB2
Linux (32/64bit)	TIBCO 5.x 이상	MS SQL Server
Solaris SPARC (32/64bit)		Mysql
Solaris (x86/x64)		Postgres
		Sybase
		Tibero

## InterMax 제품 지원 현황



## 1.3. InterMax License

InterMax License Key 는 InterMax Agent Set 을 구동하기 위해 필요하다.

### 1.3.1. Trial License Key

Trial License Key 는 테스트 목적으로 한정된 기간 동안만 사용할 수 있다.

### 1.3.2. Formal License Key

Formal License Key 는 제품 계약 이후에 발급되며, License Key 요청 시 다음과 같은 정보를 제공해야 한다.

범 주	항 목
업무명	업무 명
OS 정보	Unix Type Unix Version Unix Bit Level
Database 정보 (Oracle 일 경우)	Oracle Version Oracle Bit Level Oracle SID
Host Server 정보	IP Address Host ID Real CPU Dual Core Count

**Note.** InterMax License 정책은 CPU Core 단위이며, Server 의 Host ID 와 CPU Core 수를 이용하여 Formal License Key 의 유효성을 체크한다. 따라서, 발급 받은 Formal License Key 는 해당 서버에서만 사용 가능하며, 해당 서버의 CPU Core 수가 증가한 경우에는 Formal License Key Validation 체크 오류가 발생하므로, InterMax Agent Set 이 정상 동작하지 않는다. 따라서 CPU Core 수가 증가하는 경우에는, 사전에 Formal License Key 를 재 신청해야만 한다. (경우에 따라, 재 계약이 필요할 수도 있다)

## 2. Data Collection Layer 설치 및 구성

### 2.1 InterMax WAS Agent Set (JSPD)

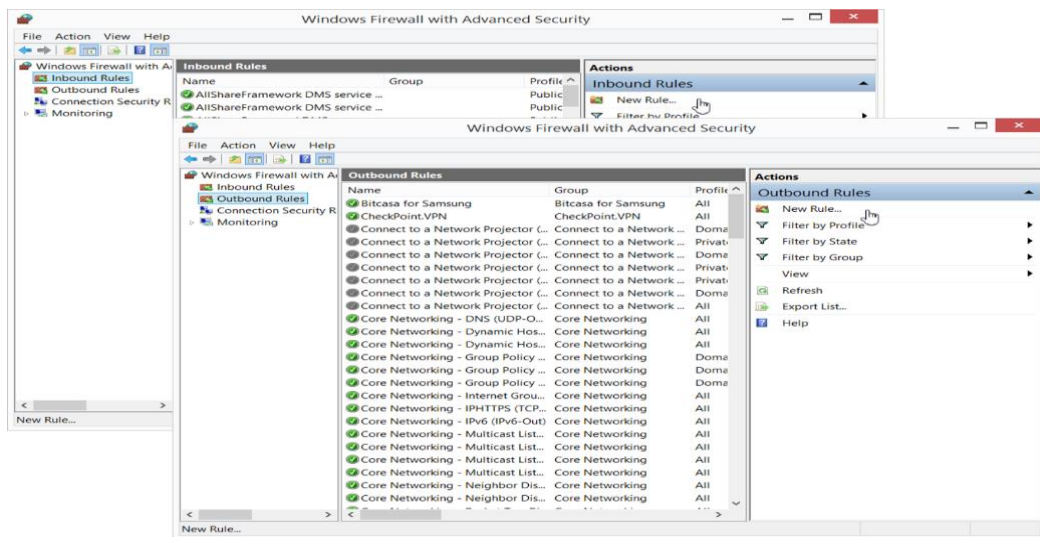
#### 2.1.1 사전 준비사항

항목	표준 권장 사양
WAS Type	J2EE 기반
Java Version	Java 1.4 이상 ~ 1.8 지원
OS Disk Size	Agent Set Size : 100MB

**Note.** JAVA 기반 데몬 또한 모니터링이 가능하다.

#### Network Port (Windows)

WAS Agent 는 1314 TCP 포트를 사용하여 Slave Data Gatherer 와 통신을 한다. 제어판에서 1314 TCP 포트를 Inbound/Outbound 로 모두 허용해준다.



#### Network Port (Unix / Linux)

JSPD 는 1314 TCP 포트를 사용하여 Slave Data Gatherer 와 통신한다. 포트 사용여부 확인 방법은 다음과 같다.

```
$ netstat -an | grep 1314
```

**Note.** DB Server 에 대한 2404(Default) UDP 포트도 오픈되어 있어야 한다.

## WAS Agent Set 구성

WAS(JVM) 데몬에 추가되는 JSPD 모듈 외에 해당 서버에 설치되는 agent 프로세스로 각 기능별로 다음과 같이 세가지 agent set 으로 구성된다.

- **IMXOSM** : 해당 OS 의 Memory 나 CPU 등 OS 리소스에 대한 정보를 수집한다.
- **IMXTXN** : SQL 관련 정보를 수집한다.
- **IMXUTS** : Remote Data 관련 정보를 수집한다.

### 2.1.2. 설치 절차

#### Windows 환경

- 1 WAS Agent 설치를 위해서는 다음과 같은 설치 파일들이 필요하다. 다음 파일들을 WAS 서버에 업로드 한다.

파일 명	설 명
InterMax_Agent_YYMMDD.tar	WAS Agent 설치 파일
License_.key	라이선스 파일

- 2 업로드한 파일의 압축을 WAS Os User Home Directory 에 해제한다.

{압축 해제 경로}\intermax\jspd 디렉토리를 이하 %JSPD\_HOME%이라고 하겠다.

**Note.** 압축 해제 위치는 변경 될 수 있다.

- 3 내부 프로세스 정보 및 Data Gatherer 정보를 입력하기 위해  
서 %JSPD\_HOME%\cfg\agent\jspd.prop 파일을 수정한다. 기본 설정 파라미터는 다음과 같다.

항 목	설 명
WR_ADDR	<b>Slave Data Gatherer</b> 의 IP:Port 정보를 입력한다.
TXN_ADDR	IMXTXN port 정보를 입력한다.
UTS_ADDR	IMXUTS port 정보를 입력한다.

#### 수행 예제

```
# WR_ADDR
WR_ADDR=192.168.123.52:1314

#{UDP_PORT|UDP_PORT}
TXN_ADDR=2404

#{UDP_PORT|UDP_PORT}
UTS_ADDR=2504
```

- 4 InterMax WAS Agent 기동을 위해서는 InterMax 옵션을 각 WAS 의 Start Batch 파일에 적용해야 한다. InterMax 옵션은 다음과 같다.

**Java Version 1.7 이상**

```
-noverify -Djspd.wasid={WAS_ID} -javaagent:%JSPD_HOME%\lib\jspd.jar
```

**Java Version 1.5 이상**

```
-Djspd.wasid={WAS_ID} -javaagent:%JSPD_HOME%\lib\jspd.jar
```

**Java Version 1.4**

```
-Djspd.wasid={WAS_ID} -Xbootclasspath/p:%JSPD_HOME%\lib\jspd.jar;%JSPD_HOME%\lib\jspd-common.jar;JSPD_HOME%\lib\jspd-pool.jar
```

**Note1.** WAS\_ID 는 각 WAS 와 각 Agent 를 매핑하기 위한 용도로 사용되며 1 부터 65535 까지 부여 할 수 있다. 동일한 번호가 중복 설정 되지 않도록 각별히 주의 해야 한다.

**Note2.** Java Version 1.4 이하인 환경에서는 %JSPD\_HOME%\build-jdk 폴더로 이동하여 build.bat jdk 실행 해야 한다.

- 5 자세한 내용은 각 ["Appendix. WAS 벤더 별 InterMax Option 설정"](#)을 참고한다.
- 6 라이선스 적용을 위해서 %JSPD\_HOME%\cfg\ 디렉터리로 **License file** 을 복사한다.

**Unix / Linux 환경**

- 1 WAS Agent 설치를 위해서는 다음과 같은 설치 파일들이 필요하다. 해당 파일을 Binary 형식으로 업로드 한다.

파일 명	설 명
InterMax_Agent_YYMMDD.tar	WAS Agent 설치 파일
License_.key	라이선스 파일

- 1 업로드 한 파일의 압축을 WAS OS User Home Directory 에 해제한다. 압축 해제 방법은 다음과 같다. {압축 해제 경로}/Jspd 디렉토리를 이하 \$JSPD\_HOME 이라고 하겠다.

```
$ tar -xvf InterMax_Agent_YYMMDD.tar
```

- 2 내부 프로세스 정보 및 Data Gatherer 정보를 입력하기 위해서 \$JSPD\_HOME/cfg/agent/jspd.prop 파일을 수정한다. 기본 설정 파라미터는 다음과 같다.

항 목	설 명
WR_ADDR	<b>Slave Data Gatherer</b> 의 IP:Port 정보를 입력한다.
TXN_ADDR	IMXTXN port 정보를 입력한다.
UTS_ADDR	IMXUTS port 정보를 입력한다.

**3 수행 예제**

```
# WR_ADDR
WR_ADDR=192.168.123.52:1314

#{UDP_PORT|UDP_PORT}
TXN_ADDR=2404

#{UDP_PORT|UDP_PORT}
```

```
UTS_ADDR=2504
```

- 4 InterMax WAS Agent 기동을 위해서는 InterMax 옵션을 각 WAS의 start script 파일에 적용해야 한다. InterMax 옵션은 다음과 같다.

#### Java Version 1.7 이상

```
-noverify -Djspd.wasid={WAS_ID} -javaagent:$JSPD_HOME/lib/jspd.jar
```

#### Java Version 1.5 이상

```
-Djspd.wasid={WAS_ID} -javaagent:$JSPD_HOME/lib/jspd.jar
```

#### Java Version 1.4

```
-Djspd.wasid={WAS_ID} -Xbootclasspath/p:$JSPD_HOME/lib/jspd.jar;%JSPD_HOME%\lib\jspd-common.jar;%JSPD_HOME%\lib\jspd-pool.jar
```

**Note1.** WAS\_ID는 각 WAS와 각 Agent를 매핑하기 위한 용도로 사용되며 1부터 65536까지 부여할 수 있다. 동일한 번호가 중복 설정되지 않도록 각별히 주의해야 한다.

**Note2.** Java Version 1.4 이하인 환경에서는 \$JSPD\_HOME/build-jdk 폴더로 이동하여 build.sh jdk 실행해야 한다.

- 5 라이선스 적용을 위해서 \$JSPD\_HOME/cfg/ 디렉터리로 **License file**을 복사한다.

## DB 모니터링 연동 시 추가 설정

DB 모니터링을 위해서는 \$JSPD\_HOME/cfg/agent/imx.prop 파일을 수정한다.

항목	설명
IMX_ADDR	IMXDBM 주소와 UDP PORT
DB_ADDR	모니터링 대상 DB의 주소(VIP).포트. SID 정보

#### 수행 예제

```
# IMX_ADDR = DB_ADDR
10.10.202.183:2604=10.10.202.183.1521.ora112
```

**Note1.** IMX\_ADDR은 Virtual IP가 아닌 실제 IP를 입력해야 한다.

**Note2.** SID는 소문자로 입력한다.

**Note3.** IMX\_ADDR 포트는 IMXDBM의 jspd.prop에 설정한 UDP\_PORT이다

## 2.1.3. OS 별 추가 설정

OS 별로 다음과 같이 추가적인 설정이 필요하다.

### AIX



GC 관련 데이터(수행시간, 수행횟수)를 수집해야 하는 경우 다음과 같은 설정이 필요하다.

항 목	설 명
JAVA 1.4 이하	<p>\$JAVA_HOME 에서 find . -name *.so command 를 수행한다.</p> <p>하위 디렉토리 중 *.so 파일이 모여 있는 곳을 확인한다.</p> <p>JAVA 32 bit 인 경우 libXmJvmpiSvc_32.so 파일을 ppc 하위 디렉토리 중 *.so 파일이 모여 있는 곳에 복사한다.</p> <p>JAVA 64 bit 인 경우 libXmJvmpiSvc_64.so 파일을 ppc64 하위 디렉토리 중 *.so 파일이 모여 있는 곳에 복사한다.</p>
JAVA 1.5 이상	<p>\$JAVA_HOME 에서 find . -name *.so command 를 수행한다.</p> <p>하위 디렉토리 중 *.so 파일이 모여 있는 곳을 확인한다.</p> <p>JAVA 32 bit 인 경우 libXmJvmtiSvc_32.so 파일을 ppc 하위 디렉토리 중 *.so 파일이 모여 있는 곳에 복사한다.</p> <p>JAVA 64 bit 인 경우 libXmJvmtiSvc_64.so 파일을 ppc64 하위 디렉토리 중 *.so 파일이 모여 있는 곳에 복사한다.</p>

### 수행 예제

#### Java Version 1.4

```

$JSPD_HOME/lib/jni/libXmJvmpiSvc.so
32 bit 인 경우
$ cp $JSPD_HOME/lib/jni/libXmJvmpiSvc.so ${JAVA_HOME}/../ppc/
64bit 인 경우
$ cp $JSPD_HOME/lib/jni/libXmJvmpiSvc.so ${JAVA_HOME}/../ppc64/
파일 복사 후 사용자 권한을 해당 디렉토리 권한과 동일하게 변경한다.
$ cd ${JAVA_HOME}/../ppc[_64]
$ chown root:root libXmJvmpiSvc.so
    
```

#### Java Version 1.5 이상

```

$JSPD_HOME/lib/jni/libXmJvmtiSvc.so
32 bit 인 경우
$ cp $JSPD_HOME/lib/jni/libXmJvmtiSvc.so ${JAVA_HOME}/../ppc/
64bit 인 경우
$ cp $JSPD_HOME/lib/jni/libXmJvmtiSvc.so ${JAVA_HOME}/../ppc64/
파일 복사 후 사용자 권한을 해당 디렉토리 권한과 동일하게 변경한다.
$ cd ${JAVA_HOME}/../ppc[_64]
$ chown root:root libXmJvmtiSvc.so
    
```

**Note1.** Java 를 설치한 소유자가 root 인 경우 root 권한이 필요하다.

**Note2.** 위에서 언급한 \$JAVA\_HOME 은 실제 WAS 가 사용하는 JAVA 를 말한다.

### Sun Solaris

OS 가 Sun 일 경우 다음과 같은 추가 적인 설정이 필요하다.

항 목	설 명
Sun	\$JAVA_HOME 에서 find . -name *.so command 를 수행한다.

	<p>하위 디렉토리 중 *.so 파일이 모여 있는 곳을 확인한다.</p> <p>JAVA 32 bit 인 경우 libgcc_s.so.1_32 파일을 sparc 하위 디렉토리 중 *.so 파일이 모여 있는 곳에 복사한다.</p> <p>JAVA 64 bit 인 경우 libgcc_s.so.1_64 파일을 sparc9 하위 디렉토리 중 *.so 파일이 모여 있는 곳에 복사한다.</p>
--	---

### 수행 예제

```

$JSPD_HOME/lib/jni/libgcc_s.so.1
32 bit 인 경우
$ cp $JSPD_HOME/lib/jni/ libgcc_s.so.1 ${WAS_JAVA}/../sparc/
64bit 인 경우
$ cp $JSPD_HOME/lib/jni/ libgcc_s.so.1 ${WAS_JAVA}/../sparc9v/
파일 복사 후 사용자 권한을 해당 디렉토리 권한과 동일하게 변경한다.
$ cd ${WAS_JAVA}/../sparc{9v}
$ chown root:root libgcc_s.so.1

```

**Note1.** Java 를 설치한 소유자가 root 인 경우 root 권한이 필요하다.

**Note2.** 위에서 언급한 \$JAVA\_HOME 은 실제 WAS 가 사용하는 JAVA 를 말한다.

## 2.1.4 기동 방법

OS 별 추가 설정 작업을 진행한 뒤 WAS 를 재기동 한다. InterMax Agent 는 WAS 와 생사를 같이 하기 때문에 특별한 관리 포인트가 없다. 따라서 기동은 기존 WAS 기동 방법에 따라 기동한다.

## 2.1.5. InterMax WAS Agent 기동

1. JSPD 가 기동되면 IMXOSM 를 기동 시킨다. (JSPD 는 WAS(JVM) 기동시 함께 기동된다)
2. IMXOSM 기동되면 IMXTXN 를 기동 시킨다.
3. IMXOSM 기동되면 IMXUTS 를 기동 시킨다

## 2.2 InterMax DB Agent Set (IMXDBM)

### 2.2.1 사전 준비사항

항목	표준 권장 사양
Oracle Version	Oracle 9i 이상
OS Disk Size	Agent Set Size : 10MB

## OS 사용자 권한

**Oracle** 설치 사용자와 동일한 권한을 가진 사용자 또는 **DBA** 그룹에 속해 있는 사용자를 생성하여 **DB Agent** 를 설치한다. Linux 계열은 **Bash**, Unix 계열은 **Ksh** 을 사용한다. 생성 방법은 다음과 같다.

```
# useradd -d {home-dir} -s {shell Path} -g {oracle gid} -G {oracle groups} intermax
# passwd intermax
```

---

**Note1.** Maxgauge 가 설치 되어 있는 경우 OS User 를 생성할 필요 없이 MaxGauge 의 OS User 로 설치를 진행하면 된다.

**Note2.** Windows 환경에서의 사용자 생성은 제어판 > 사용자 계정에서 할 수 있다.

---

## InterMax Profile Setting (Windows 환경에서는 생략)

DBMS 접속을 위해 Oracle 사용자의 .profile 중 ORACLE\_HOME, ORACLE\_BASE, ORACLE\_SID, PATH 를 intermax 사용자의 .profile 에 추가 한다.

```
PATH=$PATH:$HOME/bin
export PATH
#Oracle config
export ORACLE_BASE=/app/oracle
export ORACLE_HOME=$ORACLE_BASE/product/11.2/db_01
export ORACLE_SID=orcl
#export EDITOR=vi
#Linux config
export CLASSPATH=$ORACLE_HOME/JRE/lib:$ORACLE_HOME/jlib
export PATH=$PATH:$ORACLE_HOME/bin
export $ORACLE_HOME/lib/libclntsh.*
export LANG=en-US.UTF-8
```

## Oracle Version

해당 Instance 의 Oracle Version 에 대한 정보를 확인한다. 방법은 다음과 같다

```
SQL> select * from v$version;
```

### 수행 예제

```
BANNER
-----
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - Production
PL/SQL Release 11.2.0.1.0 - Production
CORE      11.2.0.1.0      Production
TNS for Linux: Version 11.2.0.1.0 - Production
NLSRTL Version 11.2.0.1.0 - Production
```

## Oracle Instance

해당 Instance 이름을 확인한다. 확인 방법은 다음과 같다.

```
SQL> select instance_name from v$instance;
```

**수행 예제**

```
INSTANCE_NAME
-----
intermax
```

**Oracle Numa Segment**

인터맥스는 Uniform Memory Access 와 Non-uniform Memory Access (NUMA) 두 가지 방식 모두 지원하기 때문에 서버의 NUMA 여부 확인이 필요하다. SID 배열을 통한 NUMA 확인 방법은 다음과 같다.

```
SQL> select sid from v$session;
```

**수행 예제**

```
SID
-----
 21
 22
126      <- SID 배열이 증가하는 부분이 있는 경우 Numa Segment 사용
127
128
```

- 
- Note1.** NUMA 구조의 서버는 분산된 segment 를 사용하기 때문에 SID 의 배열이 10~100 단위로 증가한다. 일반적으로 Oracle 11g 이후 버전에서는 대부분 NUMA segment 를 사용한다.
  - Note2.** 여기서 말하는 NUMA 는 NUMA 아키텍처를 의미하는 것은 아니다. Oracle Session Structure Array 가 연속적인 메모리 공간에 위치하는지, 아니면 2 개 이상의 메모리 공간에 분산되어 위치하는지의 여부에 따라, 편의상 UMA, NUMA 라고 지칭한다는 점에 유의하도록 한다.
- 

**Shared Memory IPC key**

인터맥스의 DB Agent 는 Shared Memory 의 IPC key Address 를 통해 SGA Direct Access 한다. 해당 Instance 의 IPC Key 확인 방법은 다음과 같다. (Windows 환경에서는 SID 로 대체)

```
Unix OS (Linux)
$ ipcs -mb (ipcs -m)
```

**수행 예제**

```
----- Shared Memory Segments -----
Key shmid   owner      perms      bytes      nattch     status
0x00000000 3702785    root       644        80         2
...
0x00000000 4751378    oracle     640        4096       0
0x992513cc 4784147    oracle     640        4096       0
```

- 
- Note.** 한 대의 Instance 에 2 개 이상의 IPC Key 값이 존재하는 경우 **Oradebug** 를 이용하여 올바른 IPC Key 값을 확인한다.
-

Oradebug 를 사용하여 IPC Key 확인 방법은 다음과 같다.

```
SYS> oradebug setmypid
Statement processed.
SYS> oradebug ipc
Information written to trace file.
SYS> oradebug tracefile name
/u01/app/oracle/admin/orcl/udump/orcl_ora_00000.trc
SYS> ! cat /u01/app/oracle/admin/orcl/udump/orcl_ora_00000.trc
```

**수행 예제**

```
...
Area #5 `skgm overhead' containing Subareas 5-5
Total size 00000000000003000 Minimum Subarea size 00000000
Area Subarea Shmid Stable Addr Actual Addr
5 5 4784147 0x00000092000000 0x00000092000000
...
```

---

**Note.** 'skgm overhead' 구간의 shmid 값을 확인하여, 해당 shmid 의 IPC key 값을 ipcs command 를 이용하여 확인한다.

---

**Oracle PMON**

해당 Instance 의 Oracle PMON 의 이름과 소유자를 확인한다.  
 확인 방법은 다음과 같다. (Windows 환경에서는 Oracle Process 명으로 대체)

```
$ ps -ef | grep pmon
```

**수행 예제**

```
$ ps -ef | grep pmon
oracle 45410 1 0 10:12 ? 00:00:01 ora_pmon_orcl
```

**Network Port**

WAS Agent 로 부터 트랜잭션 정보를 받기 위해 2404 UDP 포트를 사용한다.  
 다른 프로세스에 의해 해당 포트가 사용 중인지 여부를 사전에 다음과 같은 명령어로 확인한다.

```
$ netstat -an | grep 2404
```

**2.2.2. 설치 절차**

**Windows 환경**

1. 설치 파일 업로드

인터맥스는 다음과 같은 설치 파일들이 필요하며, 해당 파일을 서버에 복사한다.

파일 명	설 명
------	-----

InterMax_DBM_[OS Ver]_[Oracle Ver].tar	InterMax DB Agent Set 설치 파일
License_.key	라이선스 파일

## 2. 설치 파일 압축해제.

업로드한 파일의 압축을 해제한다. {압축 해제 경로}\intermax\를 이하 %IMX\_HOME% 이라고 하겠다.

---

**Note.** 압축 해제 위치는 변경 될 수 있다.

---

현재 Window 버전에서는 자동 설치는 지원하지 않는다. 수동 설치를 수행한다.

### 수행 예제

```
\> md %IMX_HOME%\cfg\{SID}RTS
\> copy %IMX_HOME%\cfg\sample\* %IMX_HOME%\cfg\{SID}RTS
```

## 3. 환경 파일 설정

DB Agent 설정을 하기 위해 %IMX\_HOME%\cfg\{SID}RTS\ 위치의 환경설정파일을수정한다

### Jspd.prop

기본 설정 파라미터는 다음과 같다.

항 목	설 명
WR_ADDR	Data Gatherer 의 IP 정보와 포트를 입력한다.
TXN_ADDR	IMXDBM 연결 Port 정보를 입력한다.

### 수행 예제

```
# ${IP};${TCP_PORT}
WR_ADDR=10.10.202.182:1314

# ${UDP_PORT|UDP_PORT}
TXN_ADDR=2604
```

### Imx.prop

기본 설정 파라미터는 다음과 같다.

항 목	설 명
DB_ADDR	DB IP.LISTENER PORT.sid

### 수행 예제

```
# DB Address, copy address from imx.dbm
# DB_ADDR=127.0.0.1.1521.orcl(IP.PORT.SID)
DB_ADDR=10.10.202.183.1521.ora112
```

---

**Note1.** DB IP 는 Virtual IP 가 아닌 실제 IP 를 입력해야 한다.

---

**Note2.** PORT 는 Oracle 의 LISTENER PORT 이다.

**Note3.** SID 는 소문자로 입력해야 한다.

**common.conf**

기본 설정 파라미터는 다음과 같다.

항 목	설 명
ipc_key	SID 명을 입력한다.
pmon_name	Oracle 프로세스 명을 입력한다.

**수행 예제**

```
# Oracle shared memory key
ipc_key=ora112

# Oracle PMON process name
pmon_name=oracle.exe
```

**Note1.** InterMax common.conf 설정에 대한 상세 설명은 ["InterMax Administration Guide"](#) 를 참고한다.

4. 환경 파일 생성

Windows 환경에서는 IMXDBM 이 사용하는 환경 파일을 수동으로 생성 해야 한다.

%INTERMAX\_HOME%\util\db\_setup 폴더로 이동한다.

**Maxgauge User 생성**

Intermax 에서 사용하는 DB User 를 생성하고 권한을 부여한다.

SQL\*PLUS 에 SYS User 로 접속하여 **run\_by\_sys.sql** 실행한다.

**수행 예제**

```
D:\Intermax\IXMDBM\util\db_setup>sqlplus "/ as sysdba"

Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options
SQL> @run_by_sys.sql
Enter MaxGauge USER :maxgauge
Enter password for maxgauge :maxgauge
Enter Default Tablespace for maxgauge :users
Enter Temporary Tablespace for maxgauge :temp
```

**List.conf 생성**

5. Intermax 에서 사용하는 DB 의 Stat, Event 정보 수집을 위한 환경 파일을 생성한다.

SQL\*PLUS 에 maxgauge User 로 접속하여 **listconf3.sql** 실행한다.

수행 예제

```
D:\Intermax\IXMDBM\util\db_setup>sqlplus maxgauge/maxgauge

Copyright (c) 1982, 2010, Oracle. All rights reserved.
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> @listconf3.sql
```

### env 생성

6. Intermax 에서 사용하는 환경파일을 생성한다. mkenv.exe 을 실행한다.

수행 예제

```
D:\Intermax\IXMDBM\util\db_setup>mkenv.exe
```

---

**Note.** Env 파일과 list.conf 파일이 생성되면 해당 파일을 다음위치로복사한다.  
 %INTERMAX\_HOME%\cfg\{SID}RTS

---

7. 서비스 등록 및 삭제

서비스 등록 명령어는 다음과 같다. 반드시 관리자권한을 가진 입력창에서 실행 해야 한다.

```
%IMX_HOME%\lib\imx\imxdbm -c {SID}RTS -install -H {IMX_HOME}
```

수행 예제

```
c:\intermax> Imxdbm -c IM_RTS -install -H c:\intermax
```

서비스 삭제 명령어는 다음과 같다.

```
%IMX_HOME%\lib\imx\imxdbm -c {SID}RTS -remove -H {IMX_HOME}
```

수행 예제

```
c:\intermax> Imxdbm -c IM_RTS -remove -H c:\intermax
```

8. 라이선스 파일 적용

%IMX\_HOME%\cfg 디렉토리로 License file 을 이동한다.

## Unix / Linux 환경

1. 설치 파일 업로드

인터맥스는 다음과 같은 설치 파일들이 필요하며, 해당 파일을 Binary 형식으로 업로드한다.

파일 명	설 명
InterMax_DBM_[OS Ver]_[Oracle Ver].tar	InterMax DB Agent Set 설치 파일
License_.key	라이선스 파일

수행 예제



Ex) OS : Linux 6.2, Oracle Version : 11.2.0.1, Numa Segment

```
FTP> put InterMax_DBM_linux_64_ora_112_160928.tar
```

```
FTP> put License_.key
```

2. 설치 파일 압축해제

업로드한 파일의 압축을 maxgauge 사용자 Home Directory 에 해제한다. 압축 해제 방법은 다음과 같다.

```
$ tar -xvf InterMax_DBM_[OS Ver]_[Oracle Ver].tar
```

수행 예제

```
$ tar -xvf InterMax_DBM_linux_64_ora_112_160928.tar
```

3. InterMax 환경 파일 실행

인터맥스 홈으로 이동하여 환경변수(.mxgrc) 파일을 실행한다.

```
$ cd /home/maxaguge/intermax
```

```
.$ .mxgrc.
```

4. Install Script 실행

Install 폴더 안에 있는 **install.sh** 를 사용하여 자동설치를 수행한다.

항 목	설 명
DBM setup Type	모니터링 대상 DB Type
Database owner	Oracle Instance 를 구동한 OS 유저
Conf name	[ORACLE_SID]RTS ORACLE_SID 대문자로 입력
IPC Key	설치 필요 체크사항의 Oracle Shared Memory Key
PMON process	설치 필요 체크사항의 Oracle PMON Name
DBM UDP port	WAS Agent 와 통신 포트 (Default 2404)
Data Gather IP address	DG Slave 의 설치 IP 주소
Data Gather Port	DG Slave 와 통신 포트 (Default 1314)
DBM ENV Server port	DBM 내부 통신 포트 (Default 2405)
DB_ADDR IP ADDRESS	Database 설치 주소
DB_ADDR PORT	Database 의 LISTENER PORT
DB_ADDR Database Name	Database 의 SID
Oracle Database user	MaxaGauge DB 유저 생성
Oracle Database Password	MaxGauge DB 유저 암호
Default Tablespace	MaxGauge User 의 Default Tablespace
Temporary Tablespace	MaxGauge User 의 Temporary Tablespace
Conf file	서버 에이전트 Configuration 파일 생성
Run_by_sys	MaxGauge DB User 생성 및 권한 부여
Env	Agent 의 필요 환경 File 생성
List.conf	Agent 의 필요 환경 File 생성

**수행 예제**

```
$ cd $INTERMAX_HOME/install
$. install.sh

Welcome to Intermax DBM setup

Enter DBM setup Type: [1:oracle, 2:db2]
1

Enter Database owner: [oracle]
oracle

Enter Maxgauge conf name: [ora112]
ORA112RTS

1) 0xd3ac6c80
Select ipc key: 1
ipc key : d3ac6c80

ora_pmon_orcl
1) ora_pmon_orcl
Select pmon process name: 1
pmon name : ora_pmon_orcl

DBM UDP Port number : [2604]
2604

DataGather IP Address : []
192.168.0.10

DataGather Port number : [1314]
1314

DBM ENV Server Port number : [2405]
2405

DB_ADDR IP Address : []
10.10.202.183

DB_ADDR Port number : [1521]
1521

DB_ADDR Database Name (SID) : [ORA112]
ora112

Enter Oracle maxgauge user: [maxgauge]
maxgauge
```

```
Oracle maxgauge pass:
*****

Default Tablespace for MaxGauge: [USERS]
USERS

Temporary Tablespace for MaxGauge: [TEMP]
TEMP

=====

Conf name ORA112RTS
IPC key 0xd3ac6c80
pmon name ora_pmon_ORA112
UDP port 2404
DataGather Address 192.168.0.10:1314
ENV Server Port 2405
DB Address 10.10.202.183.1521.ora112
Maxgauge user maxgauge

=====

Cfg directory created
Make conf files (common.conf, imx.prop, jspd.prop. ...)

Execute run_by_sys ...
Done.

Make env ...
/home/intermax/YU_RTS/intermax/util/db_setup/mke.sh
version: Linux 11.2.0.3.0 - 64bit
build: Mar  3 2015 11:15:59
sga_base_addr: 0x60000000
s: 0xa5f8      e: 0x9650      p: 0x9488
p: 0x0528      p: 0xcb38      h: 0xa680
s: 0xc9a0      u: 0x0020      d: 0x2b3f0
v: 0x1138      d: 0x25a28     s: 0x0000
f: 0x0170      n: 0x0010      t: 0x2ba08
s: 0xa5f8      e: 0x9648      e: 1152
db_version: 0xb200300]
Done.

Make list.conf ...
Done.
DBM Installation is complete.
```

## 5. 라이선스 파일 적용

`$INTERMAX_HOME/cfg` 디렉토리로 **License file** 을 이동한다.

...

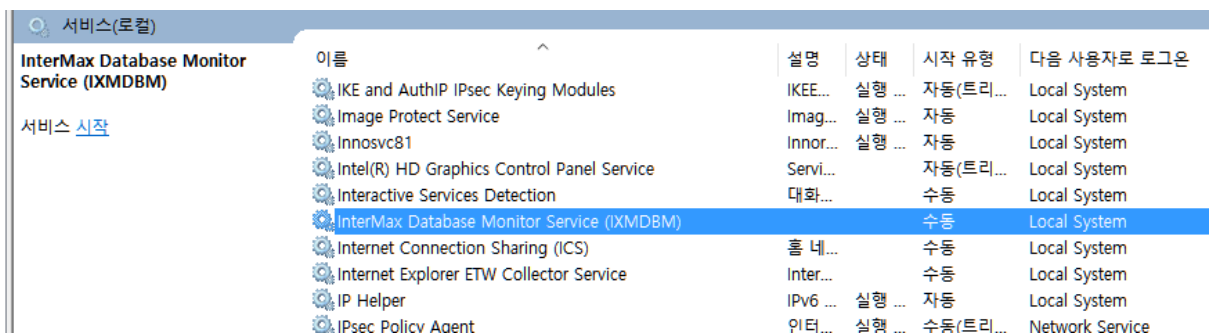
```
$ mv $HOME/License.key $INTERMAX_HOME/cfg
```

### 2.2.3. 기동 방법

#### Windows 환경

services.msc 목록을 통해 InterMax DB Agent 를 구동 시킨다.

InterMax DB Agent Set 은 Window Local Service 로 등록되며, Service(Local) 에서의 각각의 서비스를 실행하여 기동한다.



#### Unix / Linux 환경

IMXCTL Command 를 통해 InterMax DB Agent 를 구동 시킨다.

**IMXCTL** 은 **InterMax Agent Set** 을 제어하는 유틸리티로서, OS command line 에서 사용하는 Non Interactive Mode 방식과 **IMXCTL** 유틸리티 내에서 사용하는 Interactive Mode 방식이 있다. **IMXCTL** 유틸리티의 사용법은 다음과 같다.

```
#Non Interactive Mode Usage:
$ imxctl <start | stop | status | restart > {config_name}
$ imxctl version

#Interactive Mode Usage:
$ imxctl
RTSCTL> <start | stop | status | restart > {config_name}
RTSCTL> <version | quit | exit >
```

Operation	설 명
start	InterMax Agent Set 시작
stop	InterMax Agent Set 종료
status (stat)	InterMax Agent Set 상태 파악
restart	InterMax Agent Set 재 기동
version (ver)	InterMax Agent Set 버전 출력

**Note.** IMXCTL 유틸리티의 상세 설명과 사용 예제는 “[InterMax Administration Guide](#)”를 참고한다.

## 2.2.4. 예외 처리

### MakeConf Script Error (Windows 환경 해당 없음)

Install.sh 실행 시, Conf 파일이 생성되지 않을 경우에는 아래의 항목을 참조한다.

```
$ {InterMax Home Directory}/intermax/install
```

Script Name	설명
Makecommonconf	Oracle SGA 에 Direct Memory Access 를 위해 필요한 환경 파일을 생성 {ORACLE_SID} {IPC_KEY} {PMON_NAME}
Makertsconf	실시간 Data 및 Log Data 전송 환경 파일을 생성 {ORACLE_SID} {RTS_PORT} {DG_IP_ADDRESS} {DG_PORT}

#### Script 수행 방법 및 변수 기입방법

```
FILE_PATH: {InterMax Home Directory}/intermax/install
# START COMMON FILE CREATE
$. makecommonconf {ORACLE_SID} {IPC_KEY} {PMON_NAME}
ex) $. makecommonconf ORCL 0x992513cc ora_pmon_ORCL

# START RTS FILE CREATE
$. makertsconf {ORACLE_SID} {RTS_PORT} {DG_IP_ADDRESS} {DG_PORT}
ex) $. makertsconf ORCL 5080 192.168.0.10 7000
```

### Run by sys.sql Error

Install.sh 실행 시, maxgauge 유저생성 및 권한부여에 실패하면 아래의 항목을 참조한다.

```
$ sqlplus DBA or SYS User Login

# InterMax user Password, Default Tablespace, Temporary Tablespace
SQL>
CREATE USER maxgauge IDENTIFIED BY &password
DEFAULT TABLESPACE &default_ts
TEMPORARY TABLESPACE &temp_ts;

GRANT RESOURCE TO maxgauge ;
GRANT CREATE SESSION TO maxgauge;
GRANT CREATE DATABASE LINK TO maxgauge;
GRANT SELECT_CATALOG_ROLE TO maxgauge;
GRANT SELECT ANY TABLE TO maxgauge
GRANT CREATE ANY PROCEDURE TO maxgauge
GRANT EXECUTE ON SYS.DBMS_SESSION TO maxgauge
GRANT EXECUTE ON SYS.DBMS_SYSTEM TO maxgauge
```

```
GRANT ALTER SESSION TO maxgauge
GRANT ALTER SYSTEM TO maxgauge
GRANT SELECT ANY DICTIONARY TO maxgauge
```

## Env & List.conf Error

Install.sh 실행 시, Env 및 List.conf 파일 생성에 실패하면 \$INTERMAX\_HOME/util/db\_setup 에서 mke.sh , listconf3.sql 을 수행하여 수동으로 생성할 수 있다.

### 수행 예제

```
# Env Create
$. mke.sh
version: Linux 11.2.0.3.0 - 64bit
build: Mar  3 2015 11:15:59
sga_base_addr: 0x60000000
s: 0xa5f8      e: 0x9650      p: 0x9488
p: 0x0528      p: 0xcb38      h: 0xa680
s: 0xc9a0      u: 0x0020      d: 0x2b3f0
v: 0x1138      d: 0x25a28     s: 0x0000
f: 0x0170      n: 0x0010      t: 0x2ba08
s: 0xa5f8      e: 0x9648      e: 1152
db_version: 0xb200300]

# List.conf Create
$ sqlplus maxgauge/maxgauge
SQL> @listconf3.sql
```

---

**Note.** Env 파일과 list.conf 파일이 생성되면 해당 파일을 다음위치로복사한다.  
\$INTERMAX\_HOME/cfg/{SID}RTS

---

## 3. AP Server 및 Data Storage Layer 설치 및 구성

AP Server 및 Data Storage Layer 는 Platform.JS, Data Gatherer, Repository Database 세가지로 구성된다. 각 항목이 지원하는 OS Type 은 다음과 같다.

설치 및 구성 항목	상세 기능	지원 OS
Platform.JS	Client PC 를 통하여 모니터링 및 분석을 위한 UI View 단 모듈	Windows, Unix/Linux
Data Gatherer	성능 데이터를 수집/가공/분석하는 서버단 모듈	Windows, Unix/Linux
Repository Database	수집된 데이터를 저장하는 데이터베이스 저장소	PostgreSQL (Windows, Unix/Linux) Oracle (Windows, Unix/Linux)

### 3.1. 사전 준비 사항

#### 3.1.1. AP 서버 사양

InterMax 의 AP 서버 및 Data Storage 용 서버 사양은 모니터링 대상 시스템의 규모 및 수집하는 데이터의 양에 따라 사전에 고객사와 협의하여 준비하여야 하며, 일반적으로 10-node 이내, 50-instances 이내, 1,000TPS 이하의 트랜잭션 서비스를 기준으로 다음과 같은 사양을 필요로 한다.

Repository DB 는 가급적 분리 구성을 권장하며, 대용량 수집시 별도 Storage 서버 구성을 권장합니다.

3 항목	최소 사양	비고
지원 OS	Windows, Linux, HP, AIX, Solaris	
지원되는 JDK	JDK 1.8 지원	
CPU(Core)	2CPU(4Core) 이상(1.8GHz 이상)	
Memory	8GB 이상의 여유 메모리	16G 이상 권장
Hard Disk	설치공간-100GB 이내, 로그저장공간-200GB 이상	충분한 공간 확보

**Note.** InterMax 데이터 수집용 AP 서버 및 Data Storage 용 서버는 Java 데몬 형태의 프로그램으로 Java 가 설치된 대부분의 OS 에서 운영이 가능하며, 하드디스크 용량은 모니터링 대상 시스템규모와 수집 데이터 량과 범위에 따라 유동적으로 확대/축소 될 수 있다.

## 3.2. Windows 환경

### 3.2.1. 사전 준비사항

Java (JDK 1.8 이상)

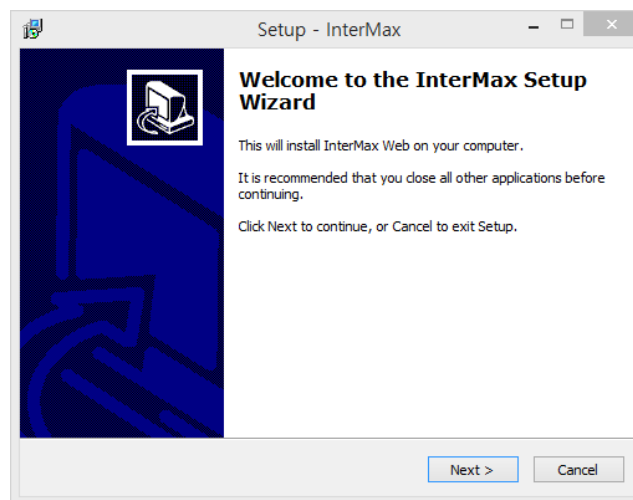
Java 는 Data Gatherer 와 Platfrom\_JS 와 같은 서버에 설치한다.

시스템 환경 변수에 JAVA\_HOME 설정한다.

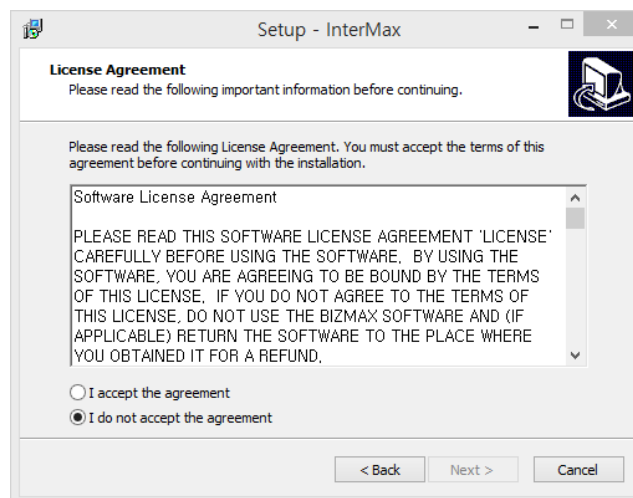
### 3.2.2. 설치 절차 (자동-Installer)

통합 Installer 를 통한 설치 방법은 다음과 같다.

1. Intermax 통합 설치 프로그램(**InterMax\_Installer\_버전명.exe**)을 실행한다.
2. 설치 마법사가 실행되면 **Next** 버튼을 클릭한다.

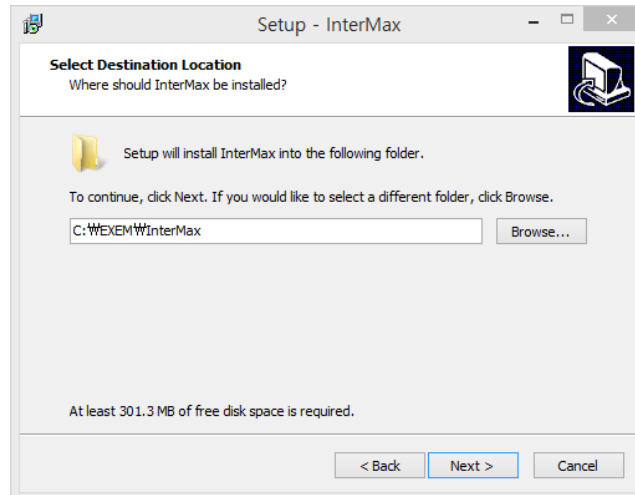


3. 라이선스 규정에 동의를 선택하고 **Next** 버튼을 클릭한다.

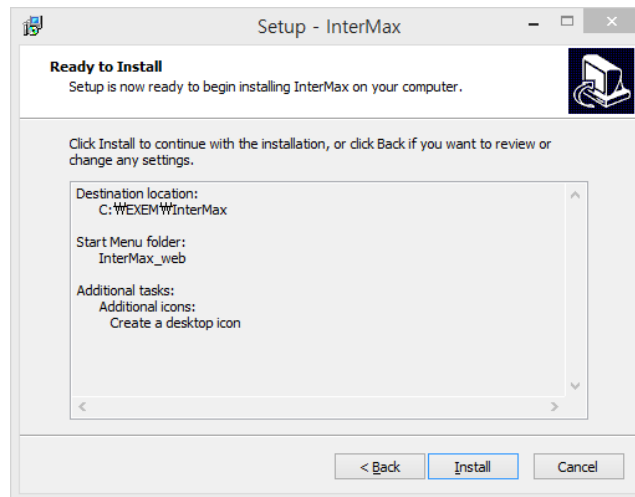


4. InterMax 설치 위치를 지정한다.





5. **Install** 버튼을 클릭하면 설치가 시작된다. 설치 시간은 대략 2 분 정도 소요된다.



6. 설치가 완료되면 [Finish] 버튼을 눌러 설치 마법사를 종료한다..



---

**Note.** Platform.JS, Data Gather, PostgreSQL 은 설치 후 모두 로컬 서비스로 자동 등록 된다.

---

### 3.2.3. 설치 절차 (수동)

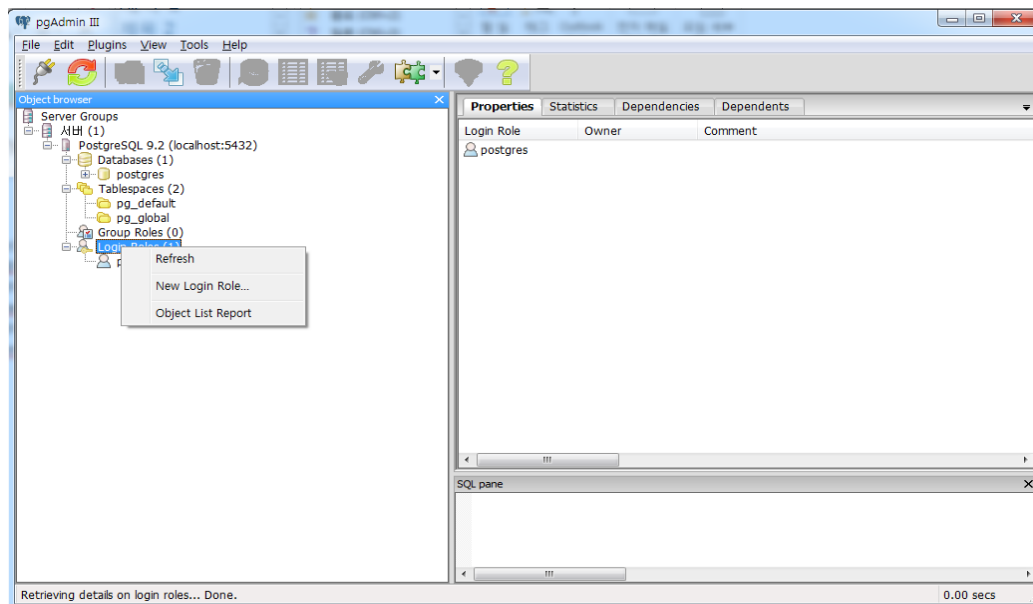
#### PostgreSQL 수동 설치

본 Install Guide에서는 PostgreSQL Database 설치에 대한 내용은 생략한다. 해당 Database 설치의 상세 설명은 PostgreSQL의 공식 Install Guide를 참고한다.

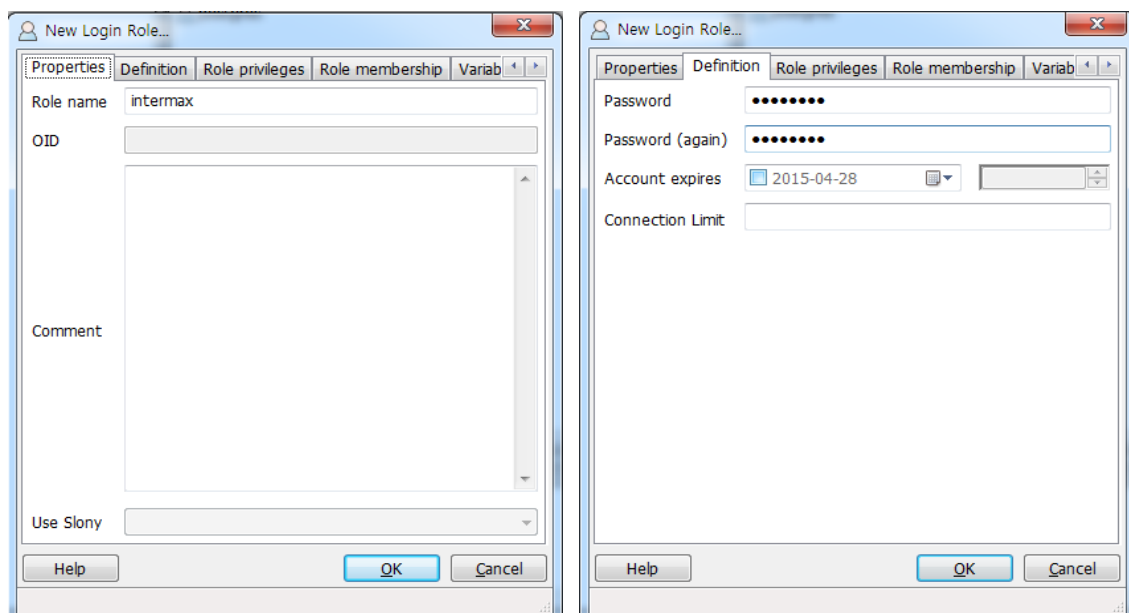
#### Repository User 생성 및 Database 설정

1. PostgreSQL에서 Repository User 및 Database를 생성하기 위해 pgAdmin3를 실행한다.

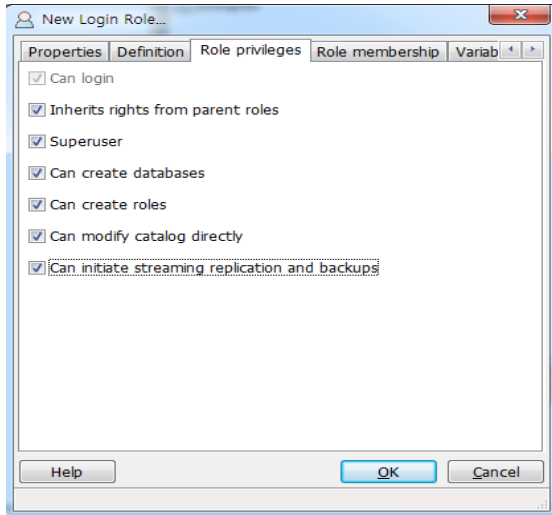
Object browser에서 Login Role에 오른쪽 클릭한 후 New Login Role을 클릭한다.



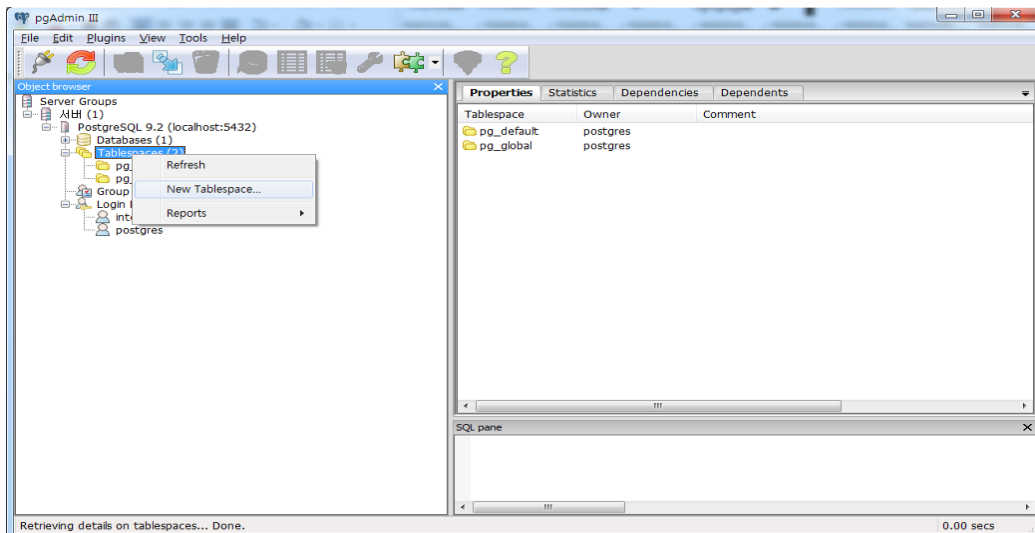
2. intermax 유저 정보를 기입한다. Properties 탭의 Role name에 intermax를 입력한다. Definition 탭의 Password에 적절한 Password를 기입한다.



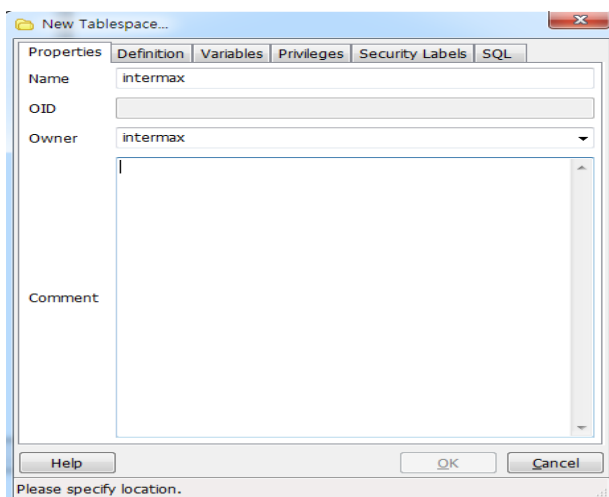
3. Role privileges 에서 모든 권한에 체크한 후 OK 를 클릭한다.



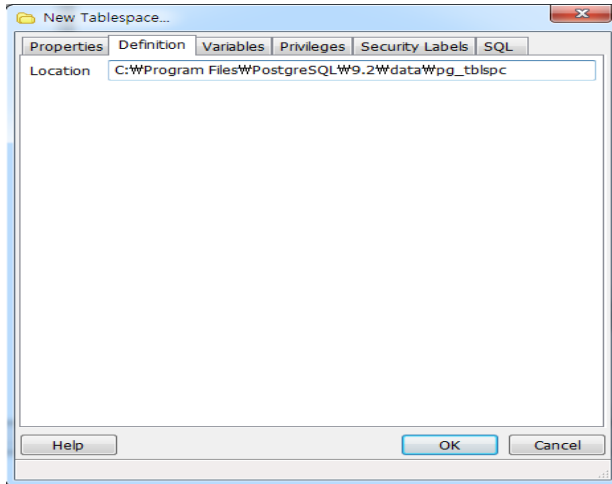
4. Tablespace 생성을 위하여 Object browser 에서 Tablespaces 에 우 클릭 후 New Tablespace 를 클릭한다.



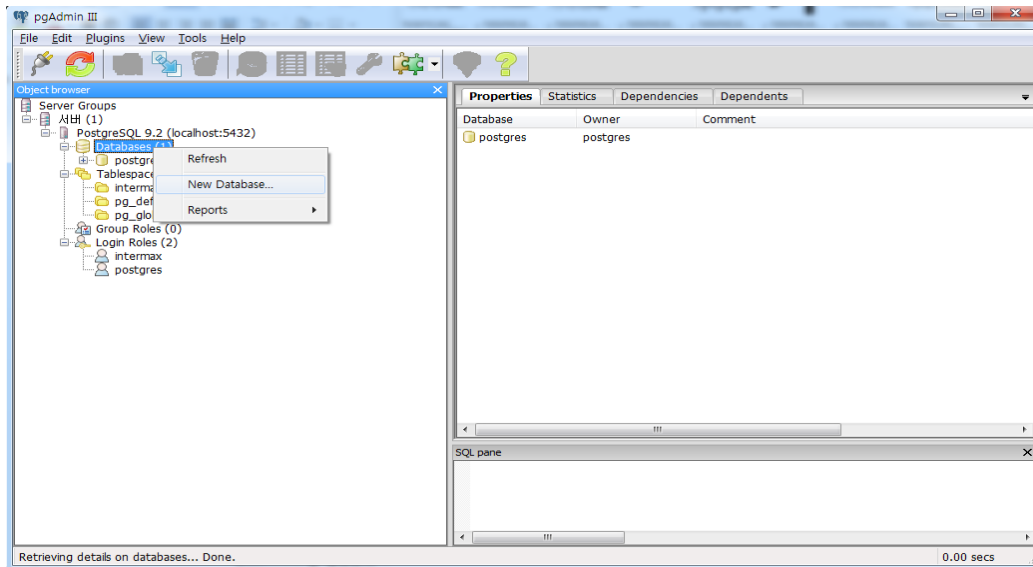
5. Properties 탭의 Name 에는 Tablespace 명으로 intermax 를 기입한다. Owner 는 Drop-Down List 에서 intermax 를 선택한다.



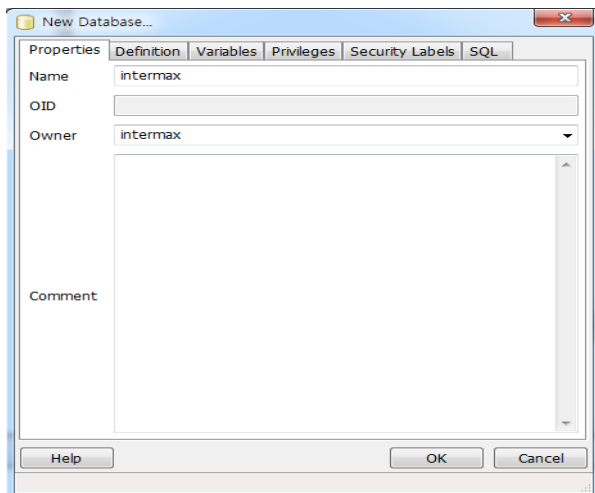
6. Definition 탭의 Location 에는 Tablespace 위치를 지정한 후 OK 를 클릭한다.



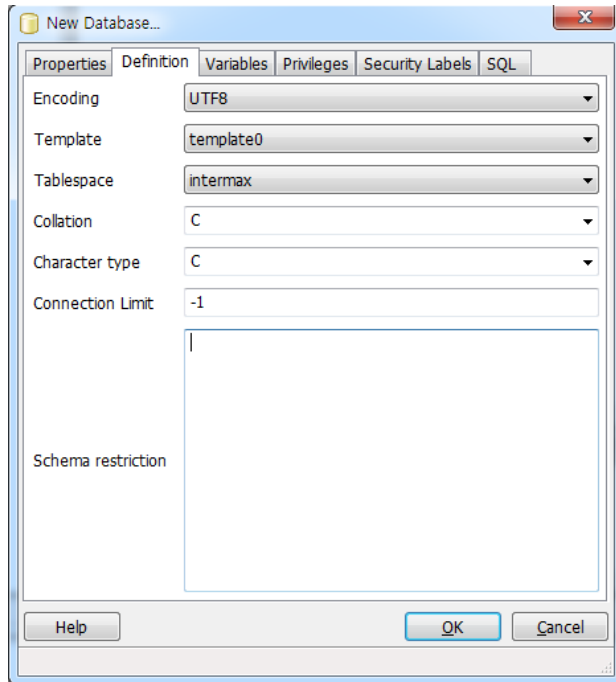
7. intermax Database 를 생성하기 위해 Obejct browser에서 Databases 에 오른쪽 클릭 후 New Database 를 클릭한다.



8. Properties 탭에서 Name 에 Database 명으로 intermax 입력 한다. Owner 는 intermax 로 선택한다.



9. Definition 탭에서 Encoding 은 UTF8, Template 은 template0 으로 선택한다. Tablespace 는 intermax 를 선택한다. Collation 와 Character type 각각 C 로 선택하고 OK 를 클릭한다.



### Repository Parameter Settings

설치한 PostgreSQL Database 의 Parameter 는 다음과 같이 설정한다.

파일 위치 예) D:\Program Files\PostgreSQL\9.4\data\postgresql.conf

Parameter 명	권장 설정 (Memory 16GB 기준)
shared_buffers	4GB
work_mem	512MB
effective_cache_size	1GB
enable_seqscan	off
logging_collector	off
default_transaction_isolation	read uncommitted
log_truncate_on_rotation	on
log_rotation_size	0
wal_sync_method	fsync_writethrough
constraint_exclusion	partition
autovacuum_vacuum_threshold	2147483647
autovacuum_analyze_threshold	2147483647
checkpoint_segments	32
track_counts	off
autovacuum	off

## Oracle 수동 설치

본 Install Guide 절에서는 Oracle Database 설치에 대한 내용으로 해당 Database 설치의 상세 설명은 Oracle의 공식 Install Guide를 참고하여 설치할 것을 권장한다.

### Repository User 생성 및 Database 설정

**Oracle에서 Repository User 및 Database를 생성하기 위해 SQL\*Plus를 실행한다.**

#### 1. 테이블스페이스 생성

**Intermax에서 사용할 테이블스페이스를 생성한다.**

##### 수행 예제

```
SQL>create tablespace [tablespace_name] datafile '위치' size[size]
SQL>extent management local
SQL>segment space management auto;
```

#### 2. 사용자 생성

**Intermax User 생성 후 권한을 부여한다.**

##### 수행 예제

```
# By sys or dba User
SQL> create user [user_name] identified by [password] default tablespace [tablespace_name] temporary tablespace temp;
SQL>GRANT RESOURCE TO intermax;
SQL>GRANT CONNECT TO intermax;
SQL>GRANT CREATE SESSION TO intermax;
SQL>GRANT CREATE DATABASE LINK TO intermax;
SQL>GRANT SELECT_CATALOG_ROLE TO intermax;
SQL>GRANT SELECT ANY TABLE TO intermax;
SQL>GRANT EXECUTE ON SYS.DBAS_SESSION TO intermax;
SQL>GRANT EXECUTE ON SYS.DBAS_SYSTEM TO intermax;
SQL>GRANT EXECUTE ON DBMS_LOCK TO intermax;
SQL>GRANT ALTER SESSION TO intermax;
SQL>GRANT ALTER SYSTEM TO intermax;
SQL>GRANT SELECT ANY DICTIONARY TO intermax;
SQL>GRANT CREATE VIEW TO intermax;
SQL>GRANT CREATE SEQUENCE TO intermax;
SQL>GRANT EXECUTE ON CTXSYS.CTX_DDL TO intermax;
SQL>GRANT SELECT ON DBA_TAB_PARTITIONS TO intermax;
```

## Data Gatherer 수동 설치

1. InterMax\_DG\_YYMMDD.tar 파일의 압축을 해제한다.

이름	수정한 날짜	유형	크기
DGServer_M	2017-01-26 오후...	파일 폴더	
DGServer_S1	2017-01-26 오후...	파일 폴더	
logs	2017-01-26 오후...	파일 폴더	
XmPing	2017-01-26 오후...	파일 폴더	

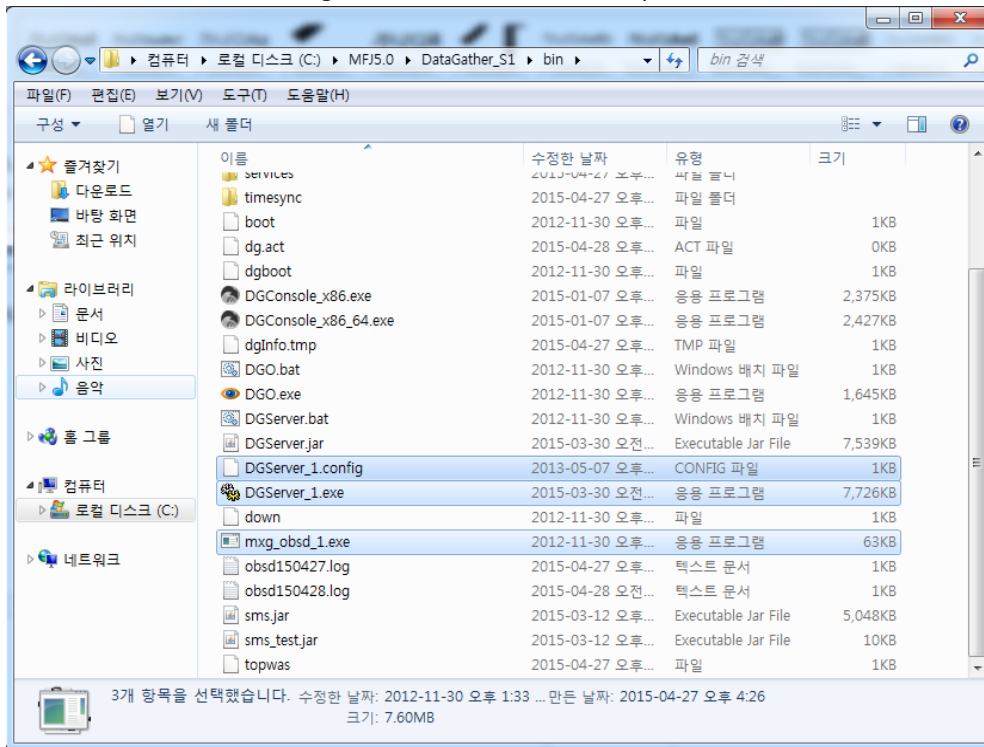
2. 먼저 Slave Data Gather 를 설치한다. DataGather\_S1\bin 에서 DGServer\_x86\_64.exe 와 DGServer\_x86\_64.config 파일을 확인한다. (32bit Windows 의 경우 DataGather\_S1\bin\services 에서 DGServer\_x86.exe 와 DGServer\_x86.config 파일을 복사하여 DataGather\_S1\bin 에 붙여넣는다.)

**Note.** 복사한 파일은 Data Gatherer 의 서비스 파일이다.

3. 두 파일의 이름을 DGServer\_1.exe 및 DGServer\_1.config 로 변경한다.
4. DataGather\_S1\bin\mxg\_obsd\win64\mxg\_obsd\_x64.exe 를 복사하여 DataGather\_S1\bin 에 붙여넣는다. (32bit Windows 의 경우 DataGather\_S1\bin\mxg\_obsd\win32\mxg\_obsd.exe)

**Note.** 복사한 파일은 Data Gatherer 의 옵저버 실행 파일이다.

5. 복사한 파일의 이름을 mxg\_obsd\_1.exe 로 바꾼다. Step 1~4 까지 수행 결과는 아래와 같다.



6. Slave Data Gatherer 설정을 위해 DataGather\_S1\conf\DGServer.xml 을 편집한다. 설정 사항은 아래 표와 같다.

```

<encryption>>false</encryption>
<master>>false</master>
<storage>>false</storage>
<dg_id>1</dg_id>
<dg_list></dg_list>
<dg_port>1314</dg_port>
<ClientPool>
  <thread_core_size>40</thread_core_size>
  <thread_max_size>80</thread_max_size>
</ClientPool>
<DBPool>
  <db_type>postgres</db_type>
  <conn_ip>127.0.0.1</conn_ip>
  <conn_port>5432</conn_port>
  <sid>postgres</sid>
  <user>postgres</user>
  <password>postgres</password>
  <conn_init_size>5</conn_init_size>
  <conn_max_size>10</conn_max_size>
  <partition>>true</partition>

```

Parameter 명	설명
master	Master 여부를 설정 <ul style="list-style-type: none"> <li>Slave Data Gatherer 인 경우 false</li> </ul>
dg_id	ID 설정 <ul style="list-style-type: none"> <li>Slave Data Gatherer 인 경우 1 이상</li> </ul>
dg_port	Slave Data Gatherer 의 통신 Port <ul style="list-style-type: none"> <li>Slave 의 Port 로는 1314 를 권장</li> </ul>
db_type	Repository 의 데이터베이스타입 설정 <ul style="list-style-type: none"> <li>PostgreSQL 의 경우 postgres</li> </ul>
conn_ip	JDBC 로 연결할 Repository 데이터베이스 IP 설정
conn_port	JDBC 로 연결할 Repository 데이터베이스 Listener Port 설정 <ul style="list-style-type: none"> <li>PostgreSQL 기본 값은 5430</li> </ul>
sid	Repository 데이터베이스 명 설정
user	Repository 데이터베이스의 User
password	Repository 데이터베이스 User 의 Password

7. 관리자 권한 커맨드 창에서 다음 명령을 수행하여 Slave Data Gatherer 를 서비스로 등록한다.

```
> sc create DGServer_1 binPath= "Absolute\path\to\DGServer_1.exe"
```

**Note.** 서비스 명은 DGServer\_1 외의 다른 것으로 지정해도 무방하다. 단, 본 매뉴얼에서는 DGServer\_1 로 지정한 것을 가정한다.

8. 옴저버 설정을 위해 DataGather\_S1\conf\DG\common.conf 파일의 설정을 편집한다. 설정 항목은 아래 표와 같다.



```
obs1=1
obs1_cmd=
obs1_keyword=
obs1_keyword2=
obs1_cpu_limit=80
obs1_mem_limit=3000000000
obs1_init_wait=20
obs1_status_file=dg.status
```

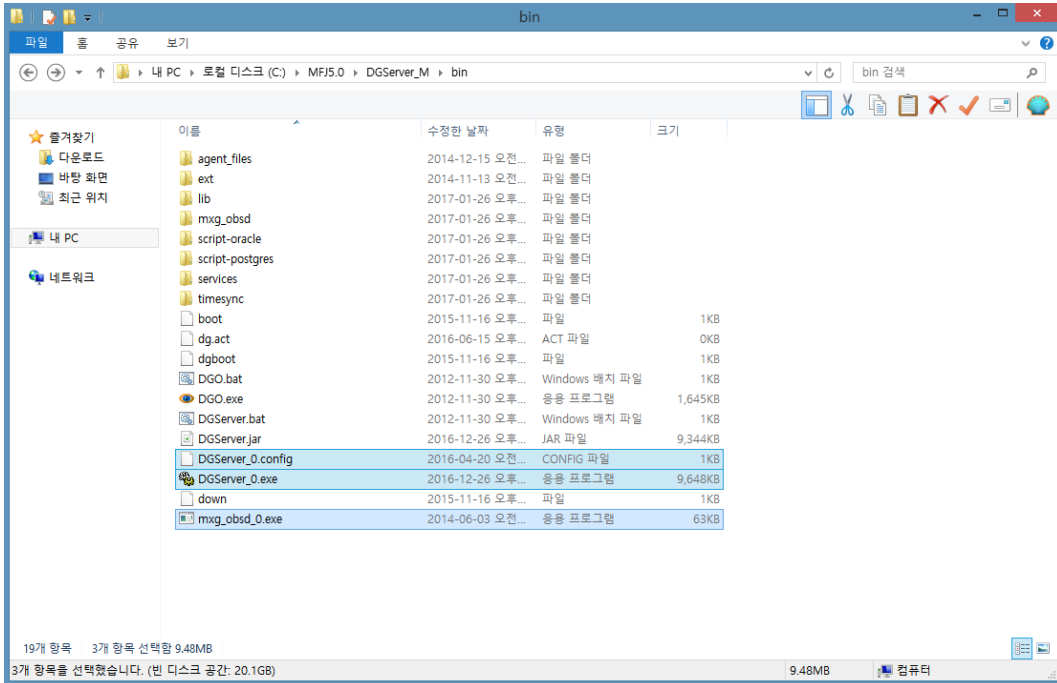
Parameter 명	설 명
obs1_cmd	Slave Data Gatherer 의 서비스 명 <ul style="list-style-type: none"> <li>● e.g.) DGServer_1</li> </ul>
obs1_keyword	Slave Data Gatherer 의 프로세스 명 일부 <ul style="list-style-type: none"> <li>● e.g.) DGServer</li> </ul>
obs1_keyword2	Slave Data Gatherer 의 프로세스 명 일부 <ul style="list-style-type: none"> <li>● e.g.) DGServer_1</li> </ul>

9. 관리자 권한 커맨드 창에서 다음 명령을 수행하여 Slave Data Gatherer 의 옵저버를 서비스로 등록한다.

```
> sc create DGServer_obsd_1 binPath= "Absolute\Path\to\mxg_obsd_1.exe -f Absolute\Path\to\common.conf -i 10 -D -OTHERD"
```

**Note.** 서비스 명은 DGServer\_obsd\_1 외의 다른 것으로 지정해도 무방하다. 단, 본 매뉴얼에서는 DGServer\_obsd\_1 로 지정한 것을 가정한다.

- 다음으로 **Master Data Gather** 를 설치한다. DataGather\_M\bin 에서 DGServer\_x86\_64.exe 와 DGServer\_x86\_64.config 파일을 확인한다. (32bit Windows 의 경우 DGServer\_x86.exe 와 DGServer\_x86.config 파일을 복사하여 DataGather\_M\bin 에 붙여넣는다.)
- 두 파일의 이름을 DGServer\_0.exe 및 DGServer\_0.config 로 변경한다.
- DataGather\_M\bin\mxg\_obsd\win64\mxg\_obsd\_x64.exe 를 복사하여 DataGather\_M\bin 에 붙여넣는다. (32bit Windows 의 경우 DataGather\_M\bin\mxg\_obsd\win32\mxg\_obsd.exe)
- 복사한 파일의 이름을 mxg\_obsd\_0.exe 로 바꾼다. Step 10~12 까지 수행 결과는 아래와 같다.



14. Master Data Gatherer 설정을 위해 DataGather\_M\conf\DGServer.xml 을 편집한다. 설정 사항은 아래 표와 같다.

```

<encryption>false</encryption>
<master>>true</master>
<storage>>false</storage>
<dg_id>0</dg_id>
<dg_list>127.0.0.1:1314</dg_list>
<dg_port>1313</dg_port>
<ClientPool>
  <thread_core_size>10</thread_core_size>
  <thread_max_size>20</thread_max_size>
</ClientPool>
<DBPool>
  <db_type>postgres</db_type>
  <conn_ip>127.0.0.1</conn_ip>
  <conn_port>5432</conn_port>
  <sid>postgres</sid>
  <user>postgres</user>
  <password>postgres</password>
  <conn_min_size>5</conn_min_size>
  <conn_max_size>10</conn_max_size>
  <autotune>true</autotune>
    
```

Parameter 명	설 명
master	Master 여부를 설정 <ul style="list-style-type: none"> <li>● Master Data Gatherer 인 경우 true</li> </ul>
dg_id	ID 설정 <ul style="list-style-type: none"> <li>● Master Data Gatherer 인 경우 0</li> </ul>
dg_port	Master Data Gatherer 의 통신 Port <ul style="list-style-type: none"> <li>● Master 의 Port 로는 1313 을 권장</li> </ul>
dg_list	이 Master Data Gatherer 에 속한 Slave Data Gatherer 들의 정보(IP:Port)를 “,”로 구분하여 입력. <ul style="list-style-type: none"> <li>● e.g.) 127.0.0.1:1314,127.0.0.1:1315, ...</li> </ul>
db_type	Repository 의 데이터베이스타입 설정 <ul style="list-style-type: none"> <li>● PostgreSQL 의 경우 postgres</li> </ul>

conn_ip	JDBC 로 연결할 Repository 데이터베이스 IP 설정
conn_port	JDBC 로 연결할 Repository 데이터베이스 Listener Port 설정 ● PostgreSQL 기본 값은 5430
sid	Repository 데이터베이스 명 설정
user	Repository 데이터베이스의 User
password	Repository 데이터베이스 User 의 Password

15. 관리자 권한 커맨드 창에서 다음 명령을 수행하여 Master Data Gatherer 를 서비스로 등록한다.

```
> sc create DGServer_0 binPath= "Absolute\path\to\DGServer_0.exe"
```

**Note.** 서비스 명은 DGServer\_0 외의 다른 것으로 지정해도 무방하다. 단, 본 매뉴얼에서는 DGServer\_0 으로 지정한 것을 가정한다.

16. 옵저버 설정을 위해 DataGather\_M\conf\DG\common.conf 파일의 설정을 편집한다. 설정 항목은 아래 표와 같다.

```
obs1=1
obs1_cmd=
obs1_keyword=
obs1_keyword2=
obs1_cpu_limit=80
obs1_mem_limit=300000000
obs1_init_wait=20
obs1_status_file=dg.status
```

Parameter 명	설명
obs1_cmd	Master Data Gatherer 의 서비스 명 ● e.g.) DGServer_0
obs1_keyword	Master Data Gatherer 의 프로세스 명 일부 ● e.g.) DGServer
obs1_keyword2	Master Data Gatherer 의 프로세스 명 일부 ● e.g.) DGServer_0

17. 관리자 권한 커맨드 창에서 다음 명령을 수행하여 Master Data Gatherer 의 옵저버를 서비스로 등록한다.

```
> sc create DGServer_obsd_0 binPath= "Absolute\Path\to\mxg_obsd_0.exe -f Absolute\Path\to\common.conf -i 10 -D -OTHERD"
```

**Note.** 서비스 명은 DGServer\_obsd\_0 외의 다른 것으로 지정해도 무방하다. 단, 본 매뉴얼에서는 DGServer\_obsd\_0 으로 지정한 것을 가정한다.

18. Slave 와 Master 의 설정이 완료되면 Repository 에 InterMax 를 위한 Repository 를 구성해야 한다. 구성을 위해 다음 명령을 실행한다.

```
> cd DataGather_M\bin
> java -jar DGServer.jar install
```

19. 아래와 같이 실행 되면 1. install Respository 를 선택한다. DGServer.xml 파일에 설정한 Database 에 Respository 가 구성된다.

```

InterMax DataGatherer TFT (Build 160826.01)
===== DataGatherer Install Menu =====
1. Install Repository
2. Remove Repository
9. Get Repository Script
0. Exit
SELECT>

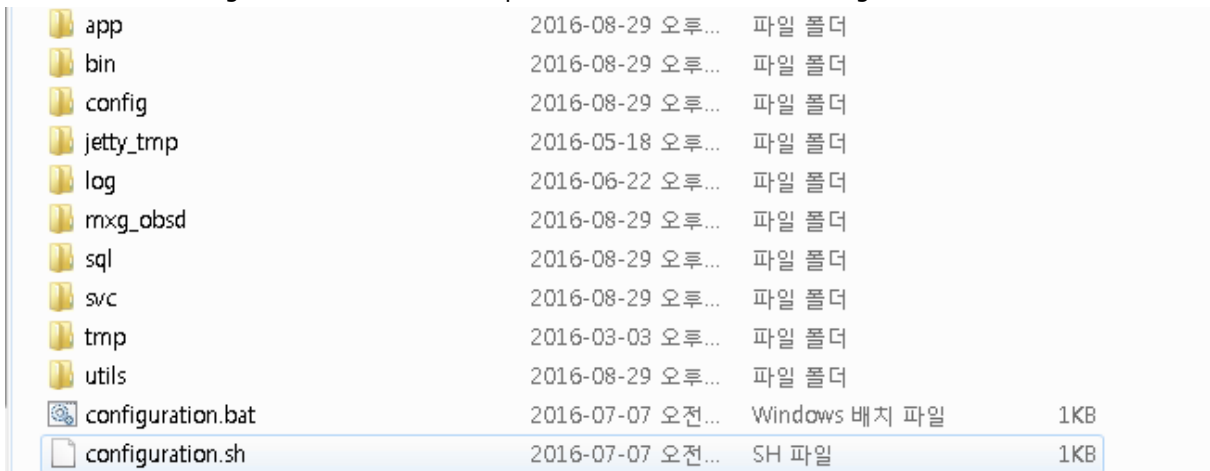
```

20. Repository 구성이 완료되면 0 을 입력하여 종료한다.
21. 여기까지 진행하면 **Data Gatherer** 기본 구성이 완료된다.

## Platform.JS 수동 설치

Windows 에 Platform.JS 설치 방법은 다음과 같다.

1. InterMax\_WEB\_YYMMDD.zip 압축해제 한다.
2. chmod +x configuration.sh 실행하여 scripts 실행 권한을 부여한 후 configuration.sh 을 실행한다.



## 수행 예제

```

1 : Configurations
2: SSL Settings ( Current state : Disabled )
0 : Exit

Select Number : 1

Step 1. DataGatherer IP [ Default : 127.0.0.1 ] ( BACK : 0 )
Input Text : 10.10.202.182

Step 2. DataGatherer Port [ Default : 1313 ] ( BACK : 0 )
Input Text : 1313

Step 3. Repository DB Type [ Default (1)PostgreSQL ] ( BACK : 0 )

.
1.PostgreSQL
2.Oracle

```

```

Select Number : 1

Step 4. Database Server [ Default : 127.0.0.1 ] ( BACK : 0 )
Input Text : 10.10.202.98

Step 5. Database Port [ Default : 5430 ] ( BACK : 0 )
Input Text : 5430

Step 6. Database Name [ Default : intermax ] ( BACK : 0 )
Input Text : intermax

Step 7. Database User [ Default : intermax ] ( BACK : 0 )
Input Text : intermax

Step 8. Database Password [ Default : intermax ] ( BACK : 0 )
Input Text : intermax

Step 9. Service Port [ Default : 8082 ] ( BACK : 0 )
Input Text : 8080

Do you want to save ? 1.Save 2.Cancel [ Default (1)Save ]
Select Number : 1
    
```

2. 환경구성이 끝나면 동일 폴더에 실행 파일이 추가된다.

```

tmp                2016-0
utils              2016-0
configuration.bat  2016-0
configuration.sh   2016-0
mxg_obsd_service_install.bat  2016-0
mxg_obsd_service_uninstall.bat  2016-0
platformjs.start.bat  2016-0
platformjs.stop.bat  2016-0
service_install.bat  2016-0
service_uninstall.bat  2016-0
    
```

3. 관리자권한으로 service\_install.bat 을 실행하여 서비스에 추가한다.

Services (Local)					
DCOM Server Process Launcher					
Name	Description	Status	Startup Type	Log On As	
DGServer_M	DataGather...		Manual	Local System	
DGServer_S1	DataGather...		Manual	Local System	
Platform.JS (8080) - ExEm Appli...		Running	Automatic	Local System	

### 3.2.4. 기동 및 접속 확인

#### InterMax Local Services

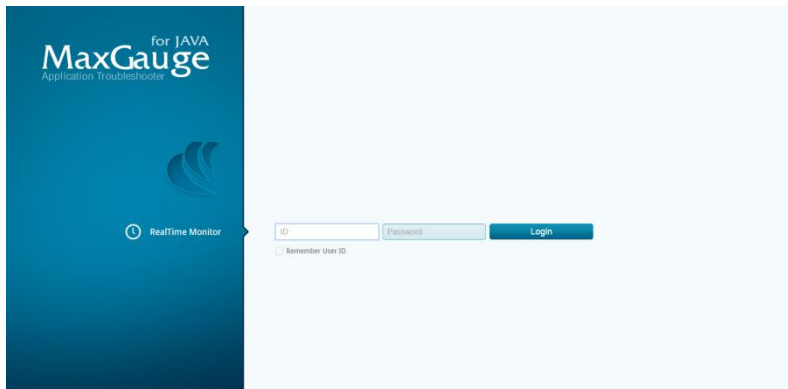
Platform.JS, Data Gatherer, Postgresql/Oracle Database 는 Windows Services(Local)에서 각각의 서비스를 실행하여 자동/수동으로 개별 기동한다.

Services (Local)						
DCOM Server Process Launcher		Name	Description	Status	Startup Type	Log On As
		DGServer_M	DataGather...		Manual	Local System
		DGServer_S1	DataGather...		Manual	Local System
		Platform.JS (8080) - ExEm Appli...		Running	Automatic	Local System

## InterMax 접속 확인

InterMax 를 사용하기 위해서는 크롬브라우저를 필요로 하며(크롬브라우저 설치 방법은 생략-최신버전 권장), 해당 브라우저를 통하여 다음과 같은 방법으로 진행한다.

1. 크롬 웹 브라우저에서 <http://127.0.0.1:8080/intermax/Config> 로 접속한다.

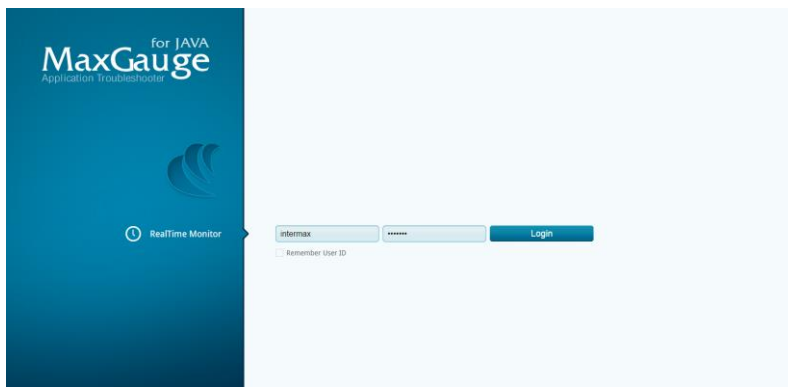



---

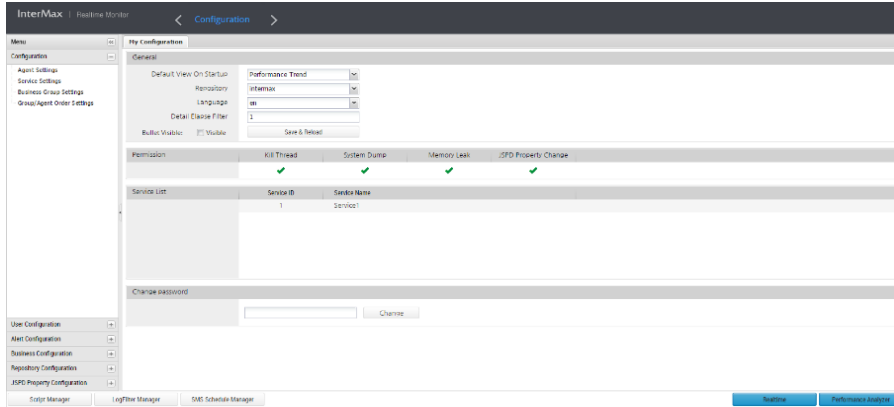
**Note.** IP 와 Port 는 Platform.JS 가 설치된 호스트 IP 와 서비스 Port 를 입력한다.

---

2. 기본적으로 제공되는 계정으로 접속한다. (ID: intermax / PW:manager)

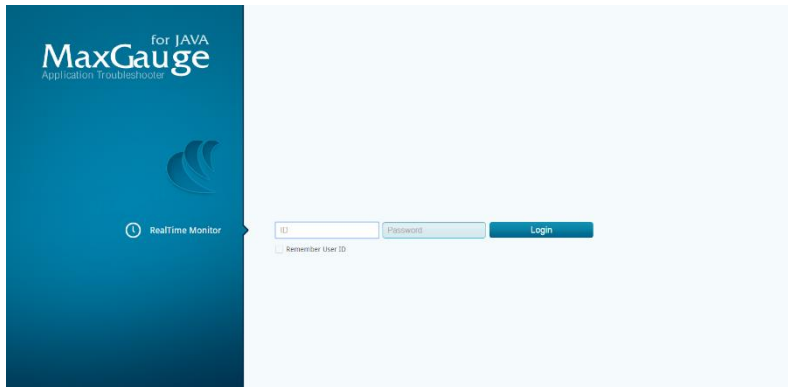


3. Configuration 을 설정한다. 최소 하나 이상의 **Service Group** 을 구성하고 접속 사용자에게 **Service Privileges** 를 부여해야 한다.

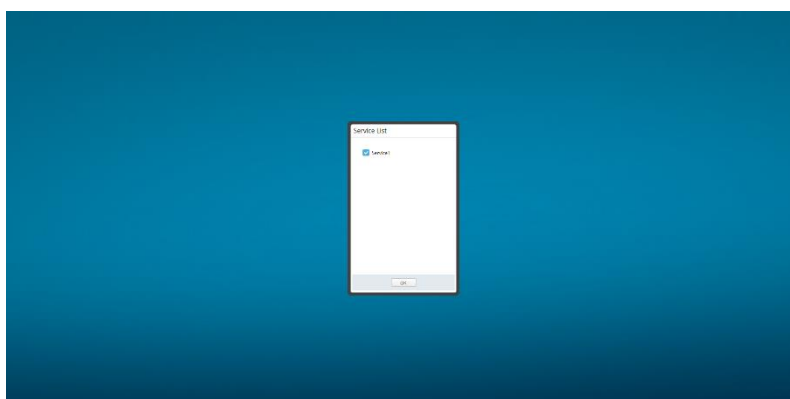


**Note.** InterMax Configuration 의 자세한 설정 방법은 “[InterMax Configuration Guide](#)”를 참고한다.

4. 크롬 웹 브라우저에서 <http://127.0.0.1:8080/intermax/RTM> 으로 접속하여 ID/Password 입력 후 로그인한다. (ID: intermax / PW:manager)



5. Configuration 과정에서 설정한 **Service Group** 목록이 나타나면, 모니터링 할 **Service Group** 을 선택 후 OK 를 누르면 실시간 모니터링 화면 view 가 로드된다.

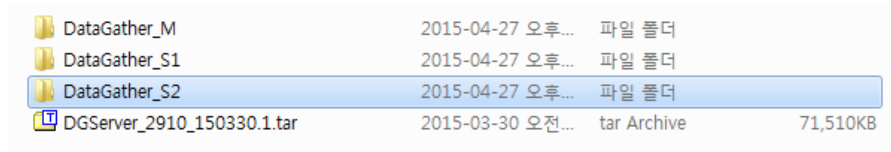


### 3.2.5. 사용자 정의 옵션

#### Slave Gatherer Process 추가

하나의 **Slave Data Gatherer** 가 너무 많은 **InterMax Agent Set** 과 통신한다면 부하가 발생할 수 있다. 이러한 경우에는 **Slave DG** 를 추가할 필요가 있다. **Slave DG** 를 추가하는 방법은 다음과 같다.

1. InterMax 가 설치된 경로에 **DataGather\_S1** 폴더를 복사하여 **DataGather\_S#** 폴더를 생성한다.



2. DataGather\_S#\bin\DGServer\_x86\_64\_1314.exe 와 DGServer\_x86\_64\_1314.config 파일의 이름을 DGServer\_x86\_64\_{other port#}로 변경한다.
3. DataGather\_S#\conf\DGServer.xml 파일을 편집하여 dg\_id 와 dg\_Port 를 변경한다.

```

<master>>false</master>
<storage>>false</storage>
<dg_id>2</dg_id>
<dg_list></dg_list>
<dg_port>1315</dg_port>
<ClientPool>
  <thread_core_size>40</thread_core_size>
  <thread_max_size>80</thread_max_size>

```

**Note.** 다른 **Slave Data Gatherer** 와 dg\_id 와 dg\_port 가 중복 되지 않도록 주의한다.

4. DataGather\_M\conf\DGServer.xml 파일을 편집하여 **Slave\_Gather\_List** 에 IP 주소와 Port 번호를 추가한다.

```

<master>>true</master>
<storage>>false</storage>
<dg_id>0</dg_id>
<dg_list>127.0.0.1:1314,127.0.0.1:1315</dg_list>
<dg_port>1313</dg_port>
<ClientPool>
  <thread_core_size>40</thread_core_size>
  <thread_max_size>80</thread_max_size>
</ClientPool>
<DBPool>

```

5. 관리자 권한 커맨드 창에서 **Slave DG #**를 서비스로 등록한다.

```
sc create {Service Name} binPath= "Absolute\path\to\Data Gather_S#\bin\DGServer_{bit}.exe"
```

```

c:\MFJ5.0\DataGather_S2\bin>sc create DGServer_1315 binPath= "C:\MFJ5.0\DataGather_S2\bin\DGServer_x86_64_1315.exe"
[SC] CreateService 성공
c:\MFJ5.0\DataGather_S2\bin>

```

**Note.** 권장 Slave Process 의 개수는 (Slave 1): (JVM instance 수 < 50 개)이다. 하지만 각각의 Slave process 에는 메모리 1G 가 할당되기 때문에, Free Memory 를 충분히 고려한 후 추가하여야 한다. DG 의 할당 메모리는 각 bin 폴더안에 DGServer\_\*.config 파일을 편집하여 변경이 가능하다.

## PostgreSQL Tablespace 설정

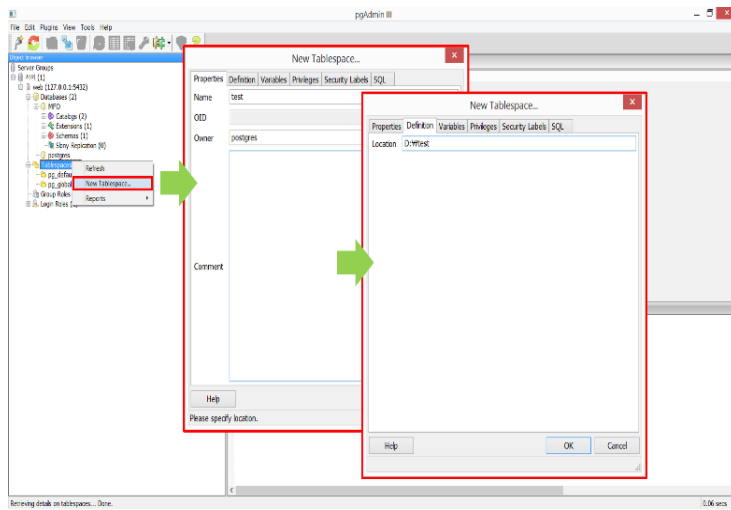


PostgreSQL Repository 에 저장하는 데이터 량이 늘어나면 디스크 용량 부족이 발생할 수 있다. 이러한 문제점은 개별 테이블에 별도의 테이블스페이스를 생성하여 분할 저장 함으로써 해결할 수 있다. Tablespace 할당 방법은 다음과 같다.

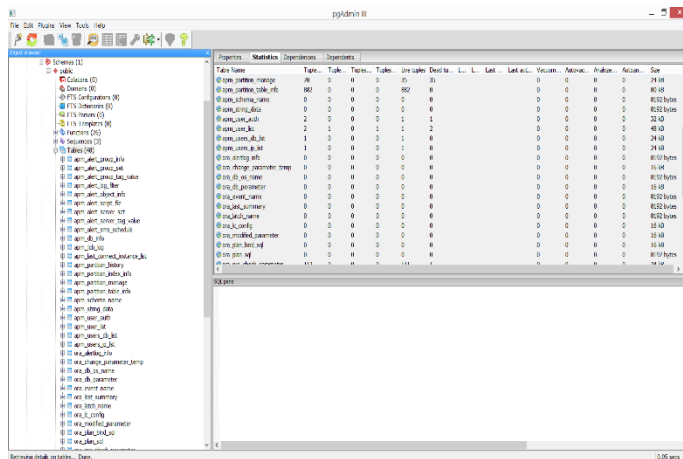
1. pgAdmin3 를 실행한다. ({InterMax Home Directory}/Database/bin/pgAdmin3)

pg_test_fsync.exe	2/18/2014 3:51 PM	Application	36 KB
pg_test_timing.exe	2/18/2014 3:51 PM	Application	24 KB
pg_upgrade.exe	2/18/2014 3:52 PM	Application	113 KB
<input checked="" type="checkbox"/> pgAdmin3.exe	2/18/2014 3:56 PM	Application	9,814 KB
pgbench.exe	2/18/2014 3:51 PM	Application	57 KB
postgres.exe	2/18/2014 3:49 PM	Application	5,403 KB
psql.exe	2/18/2014 3:50 PM	Application	405 KB

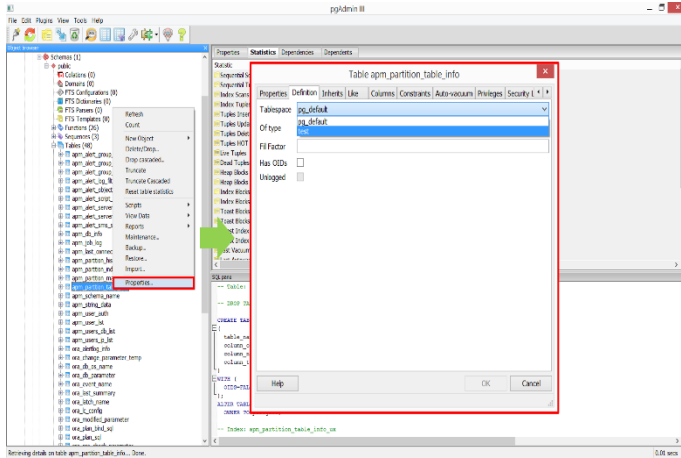
2. 새로운 테이블스페이스를 생성하고, 이름/소유자/경로를 입력한다.



3. 용량이 큰 테이블들을 확인한다.



4. 테이블에 생성한 테이블스페이스를 개별적으로 지정해준다.



**Note.** 데이터 저장주기는 InterMax Configuration 에서 변경할 수 있다. 상세 내용은 “InterMax Configuration Guide”를 참고한다.

## 3.3. Unix/Linux 환경

### 3.3.1. 사전 준비사항

Java (JDK 1.8 이상)

Java 는 Data Gatherer 와 Platfrom\_JS 와 같은 서버에 설치한다.

### 3.3.2. 설치 절차 (수동)

InterMax 의 Unix/Linux 용 Installer 를 통한 자동 설치는 고객사 시스템 환경(CDE)에 따라 유동적이고, 활용성이 떨어지므로 별도 제공하지 않고 수동 설치를 권장한다.

#### PostgreSQL 수동 설치

본 Install Guide 절에서는 PostgreSQL Database 설치에 대한 내용으로 Windows 수동 설치 내용과 유사하므로 참고하여 설치하기 바라며, 해당 Database 설치의 상세 설명은 PostgreSQL 의 공식 Install Guide 를 참고한다

##### 1. Repository User 생성 및 권한 설정

Repository user 및 database 를 생성하기 위해 psql 을 실행한다.

(설치 된 폴더/bin 에서 ./psql 실행)

수행 예제

```
Psql postgres

Postgres=#
CREATE USER intermax PASSWORD 'intermax';
ALTER USER intermax WITH SUPERUSER;
ALTER USER intermax WITH CREATEROLE;
ALTER USER intermax WITH REPLICATION;
ALTER UAER intermax WITH VALID UNTL 'infinity

Postgres=# \du
List of roles
Role name | Attributes | Member of
-----+-----+-----
intermax | Superuser, Create role, Create DB, Replication + | {}
          | Password valid until infinity |
postgres | Superuser, Create role, Create DB, Replication | {}
```

##### 2. 테이블스페이스 생성

## 수행 예제

```
Psql template1
```

```
Template1=#
```

```
CREATE TABLESPACE intermax OWNER intermax
LOCATION '/app/postgresql/pgsql/data/pg_tblspc';
```

```
CREATE DATABASE intermax
WITH OWNER = intermax
ENCODING = 'UTF8'
TEMPLATE = template0
TABLESPACE = intermax
LC_COLLATE = 'C'
LC_CTYPE = 'C'
CONNECTION LIMIT = -1;
```

```
template1=# \l
```

```
List of databases
```

Name	Owner	Encoding	Collate	Ctype	Access privileges
Intermax	intermax	UTF8	C	C	
Postgres	postgres	UTF8	ko_KR.utf8	ko_KR.utf8	
template0	postgres	UTF8	ko_KR.utf8	ko_KR.utf8	=c/postgres postgres=CtC/postgres
template1	postgres	UTF8	ko_KR.utf8	ko_KR.utf8	=c/postgres postgres=CtC/postgres

```
(4 rows)
```

```
template1=# \db
```

```
List of tablespaces
```

Name	Owner	Location
intermax	intermax	/app/postgresql/pgsql/data/pg_tblspc
pg_default	postgres	
pg_global	postgres	

```
(3 rows)
```

## Oracle 수동 설치

본 Install Guide 절에서는 Oracle Database 설치에 대한 내용으로 해당 Database 설치의 상세 설명은 Oracle의 공식 Install Guide를 참고하여 설치할 것을 권장한다.

### Repository User 생성 및 Database 설정

Oracle에서 Repository User 및 Database를 생성하기 위해 SQL\*Plus를 실행한다.

### 1. 테이블스페이스 생성

Intermax 에서 사용할 테이블스페이스를 생성한다.

#### 수행 예제

```
SQL>create tablespace [tablespace_name] datafile '위치' size[size]
SQL>extent management local
SQL>segment space management auto;
```

### 2. 사용자 생성

Intermax User 생성 후 권한을 부여한다.

#### 수행 예제

```
# By sys or dba User
SQL> create user [user_name] identified by [password] default tablespace [tablespace_name] temporary tablespace temp;
SQL>GRANT RESOURCE TO intermax;
SQL>GRANT CONNECT TO intermax;
SQL>GRANT CREATE SESSION TO intermax;
SQL>GRANT CREATE DATABASE LINK TO intermax;
SQL>GRANT SELECT_CATALOG_ROLE TO intermax;
SQL>GRANT SELECT ANY TABLE TO intermax;
SQL>GRANT EXECUTE ON SYS.DBMS_SESSION TO intermax;
SQL>GRANT EXECUTE ON SYS.DBMS_SYSTEM TO intermax;
SQL>GRANT EXECUTE ON DBMS_LOCK TO intermax;
SQL>GRANT ALTER SESSION TO intermax;
SQL>GRANT ALTER SYSTEM TO intermax;
SQL>GRANT SELECT ANY DICTIONARY TO intermax;
SQL>GRANT CREATE VIEW TO intermax;
SQL>GRANT CREATE SEQUENCE TO intermax;
SQL>GRANT EXECUTE ON CTXSYS.CTX_DDL TO intermax;
SQL>GRANT SELECT ON DBA_TAB_PARTITIONS TO intermax;
```

## Data Gatherer 수동 설치

1. InterMax\_DG\_YYMMDD.tar 압축을 해제한다.

#### 수행 예제

```
$ tar -xvf InterMax_DG_YYMMDD.tar
```

2. DGServer\_M/conf 폴더의 DGServer.xml 을 작성한다.

```
<?xml version="1.0" encoding="EUC-KR" ?>
<DataGather>
  <DefaultOptions>
    <encryption>>false</encryption>
    <master>>true</master>
    <dg_id>0</dg_id>
    <dg_list>127.0.0.1:1314</dg_list>
    <dg_port>1313</dg_port>
    <ClientPool>
      <thread_core_size>100</thread_core_size>
      <thread_max_size>200</thread_max_size>
    </ClientPool>
    <DBPool>
      <db_type>postgres</db_type>
      <conn_ip>10.10.202.215</conn_ip>
      <conn_port>5432</conn_port>
      <sid>intermax1</sid>
      <user>intermax</user>
      <password>intermax</password>
      <conn_init_size>50</conn_init_size>
      <conn_max_size>100</conn_max_size>
      <partition>>true</partition>
    </DBPool>
  </DefaultOptions>
  <Oracle>
```

Parameter 명	설 명
master	Master 여부를 설정 <ul style="list-style-type: none"> <li>● Master Data Gatherer 인 경우 true</li> </ul>
dg_id	ID 설정 <ul style="list-style-type: none"> <li>● Master Data Gatherer 인 경우 0</li> </ul>
dg_port	Master Data Gatherer 의 통신 Port <ul style="list-style-type: none"> <li>● Master 의 Port 로는 1313 을 권장</li> </ul>
dg_list	이 Master Data Gatherer 에 속한 Slave Data Gatherer 들의 IP:Port 를 “;”로 구분하여 입력. <ul style="list-style-type: none"> <li>● e.g.) 127.0.0.1:1314,127.0.0.1:1315, ...</li> </ul>
db_type	Repository 의 데이터베이스타입 설정 <ul style="list-style-type: none"> <li>● PostgreSQL 의 경우 postgres</li> <li>● Oracle 의 경우 oracle</li> </ul>
conn_ip	JDBC 로 연결할 Repository 데이터베이스 IP 설정
conn_port	JDBC 로 연결할 Repository 데이터베이스 Listener Port 설정 <ul style="list-style-type: none"> <li>● PostgreSQL 기본 값은 5430</li> <li>● Oracle 기본 값은 1521</li> </ul>
sid	Repository 데이터베이스 명 설정
user	Repository 데이터베이스의 User
password	Repository 데이터베이스 User 의 Password

3. DGServer\_S1/conf 폴더에 DGServer.xml 을 작성한다.

```

?xml version="1.0" encoding="EUC-KR"?
<DataGather>
  <DefaultOptions>
    <encryption>>false</encryption>
    <master>>true</master>
    <dg_id>0</dg_id>
    <dg_list>127.0.0.1:1314</dg_list>
    <dg_port>1313</dg_port>
    <ClientPool>
      <thread_core_size>100</thread_core_size>
      <thread_max_size>200</thread_max_size>
    </ClientPool>
    <DBPool>
      <db_type>postgres</db_type>
      <conn_ip>10.10.202.215</conn_ip>
      <conn_port>5432</conn_port>
      <sid>intermax1</sid>
      <user>intermax</user>
      <password>intermax</password>
      <conn_init_size>50</conn_init_size>
      <conn_max_size>100</conn_max_size>
      <partition>>true</partition>
    </DBPool>
  </DataGather>

```

Parameter 명	설명
master	Master 여부를 설정 <ul style="list-style-type: none"> <li>Slave Data Gatherer 인 경우 false</li> </ul>
dg_id	ID 설정 <ul style="list-style-type: none"> <li>Slave Data Gatherer 인 경우 1 이상</li> </ul>
dg_port	Slave Data Gatherer 의 통신 Port <ul style="list-style-type: none"> <li>Slave 의 Port 로는 1314 를 권장</li> </ul>
db_type	Repository 의 데이터베이스타입 설정 <ul style="list-style-type: none"> <li>PostgreSQL 의 경우 postgres</li> </ul>
conn_ip	JDBC 로 연결할 Repository 데이터베이스 IP 설정
conn_port	JDBC 로 연결할 Repository 데이터베이스 Listener Port 설정 <ul style="list-style-type: none"> <li>PostgreSQL 기본 값은 5430</li> </ul>
sid	Repository 데이터베이스 명 설정
user	Repository 데이터베이스의 User
password	Repository 데이터베이스 User 의 Password

4. Slave 와 Master 의 설정이 완료되면 Repository 에 InterMax 를 위한 Repository 를 구성해야 한다. 구성을 위해 다음 명령을 실행한다

```

> cd DataGather_M/bin
> java -jar DGServer.jar install

```

5. 아래와 같이 Install Menu 가 나타나며, Database 에 Repository 가 구성작업을 시작한다.

- 1) 1. install Respository 를 선택한다.

```
===== DataGather Install Menu =====
1. Install Repository
2. Remove Repository
9. Get Repository Script
0. Exit
SELECT>1
```

- 2) (Oracle 인 경우만 해당)최초 설치인 경우에는 "N" 을 입력하고, 재설치인 경우에는 기존 환경설정 정보를 유지하기 위해 "Y" 를 입력한다.

```
Do you ever have installed in this repository ? [Y/N]:N
Input Number of Database :2
Set Common Repository Tables :155
Append Oracle Repository Tables :165
```

Input Number of Databsse: 2 는 고객사 모니터링 대상 DB Instance 수를 의미하는 값으로 imxdbm 모듈을 통하여 모니터링 하고자 하는 대상 수를 입력한다.(imxdbm 모듈 미설치시에는 해당 값을 입력하지 않아도 된다).

해당 값이 필요한 이유는 이후 imxdbm 을 통하여 수집되는 성능 데이터에 대해 partition table 저장시 sub-partition 을 생성하기 위한 기준값으로 사용한다. (미 입력시 기본 partition table 만을 생성하게 된다)

- 3) (Oracle 인 경우만 해당)Enter Table Tablespace for InterMax : intermax\_ts (생성한 tablespace 명을 입력한다)

```
Enter table tablespace for InterMax : intermax_ts
Table Tablespace for InterMax : intermax_ts
Enter Index Tablespace for InterMax [intermax_ts] : intermax_ts
```

Enter Index Tablespace for InterMax [intermax\_ts] : intermax\_ts (index tablespace 를 분리하고자 할 경우 해당 tablespace 명을 입력한다. 기본 tablespace 를 같이 사용할 경우 동일하게 입력하면 된다.)

```
PROCEDURE RT_TXN_DETAIL created.
TRIGGER XAPM_RTM_SORT_KEY_WAS_TRIGGER created.
TRIGGER XAPM_RTM_SORT_KEY_DB_TRIGGER created.
TRIGGER XAPM_RTM_SORT_KEY_WS_TRIGGER created.
TRIGGER XAPM_RTM_SORT_KEY_BS_TRIGGER created.
FUNCTION GET_DB_ID_DBADDR created.
FUNCTION GET_ANY_ID created.
PROCEDURE INSERT_WAS_DB_INFO created.
PROCEDURE update_was_app_type created.
PROCEDURE GET_WAS_MONITOR_DAILY created.
FUNCTION SET_HOST_INFO created.
TRIGGER FUNCTION XAPM_WAS_VISITOR_TRIGGER_FUNC created.
PROCEDURE WAS_SERVICE_STAT created.
Insert into XAPM_USER_AUTH processed: 1 failed: 0
Insert into XAPM_USER_SERVICE_INFO processed: 1 failed: 0
Insert into XAPM_STATE_CODE processed: 21 failed: 0
Insert into XAPM_PARTITION_MANAGE processed: 76 failed: 0
Insert into XAPM_ALERT_GROUP_SET processed: 16 failed: 0
Insert into XAPM_ALERT_GROUP_TAG_VALUE processed: 46 failed: 0
Insert into XAPM_RTM_DUMMY processed: 2880 failed: 0
===== DataGather Install Menu =====
1. Install Repository
2. Remove Repository
9. Get Repository Script
0. Exit
SELECT>
```

- 4) 모든 입력이 완료되면 관련 테이블 생성 및 구성 작업이 완료되며 상기와 같이 메시지가 출력된다.

6. Repository 구성이 완료되면 0 을 입력하여 종료한다.



7. \$INTERMAX\_HOME/Bin 폴더에서 boot 를 실행하면 Data Gather 가 기동된다.

## Platform.JS 수동 설치

Platform.JS 설치 방법은 다음과 같다.

1. InterMax\_WEB\_YYMMDD.zip 압축해제.
2. chmod +x configuration.sh 실행하여 script 실행 권한을 부여한 후 configuration.sh 을 실행한다.
3. Configuration.sh 실행한다.

수행 예제

```
=====
PlatformJS Configuration
=====
1 : Configurations
2: SSL Settiions ( Current state : Disabled )
0 : Exit

Select Number : 1

Step 1. DataGather IP [ Default : 127.0.0.1 ] ( BACK : 0 )
Input Text : 10.10.10.100

Step 2. DataGather Port [ Default : 1313 ] ( BACK : 0 )
Input Text : 1313

Step 3. Repository DB Type [ Default (1)PostgreSQL ] ( BACK : 0 )
1 PostgreSQL
2 Oracle
Select Number : 1

Step 4. Database Server [ Default : 127.0.0.1 ] ( BACK : 0 )
Input Text : 10.10.10.100

Step 5. Database Port [ Default : 5430 ] ( BACK : 0 )
Input Text : 5430

Step 6. Database Name [ Default : intermax ] ( BACK : 0 )
Input Text : intermax

Step 7. Database User [ Default : intermax ] ( BACK : 0 )
Input Text : intermax

Step 8. Database Password [ Default : intermax ] ( BACK : 0 )
Input Text : intermax

Step 9. Service Port [ Default : 8082 ] ( BACK : 0 )
Input Text : 8080

Do you want to save ? 1.Save 2.Cancel [ Default (1)Save ]
```

Select Number : 1

4. 환경구성이 끝나면 동일 폴더에 실행 파일이 추가된다.
5. Platformjs.start.sh 실행하면 PlatformJS 기동 옵션이 표시된다.
  - 1선택하면 기본적으로 운영 레벨의 로그 출력과 함께 백그라운드 서비스로 해당 서비스가 기동된다(기본 선택은 1이다)
  - 2선택하면 콘솔모드로 디버그 레벨의 로그 출력과 함께 서비스가 기동된다.

```

PlatformJS
Select the operation mode you wish to perform.

1. Release Mode ( background execution )
2. Debug Mode ( Console execution )

Choose Mode (Enter Key. Default '1') :

```

### 3.3.3. 기동 방법

#### InterMax PlatformJS 기동

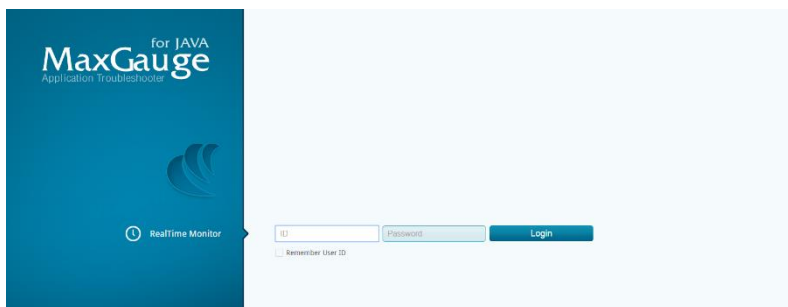
PlatformJS.start.sh를 실행하여 표시되는 옵션 중 하나를 선택하면 기동한다.

옵션	설명
Release Mode	● Background 모드로 PlatformJS 실행
Debug Mode	● Debug 모드로 PlatformJS 실행

#### InterMax 설정 및 확인 방법

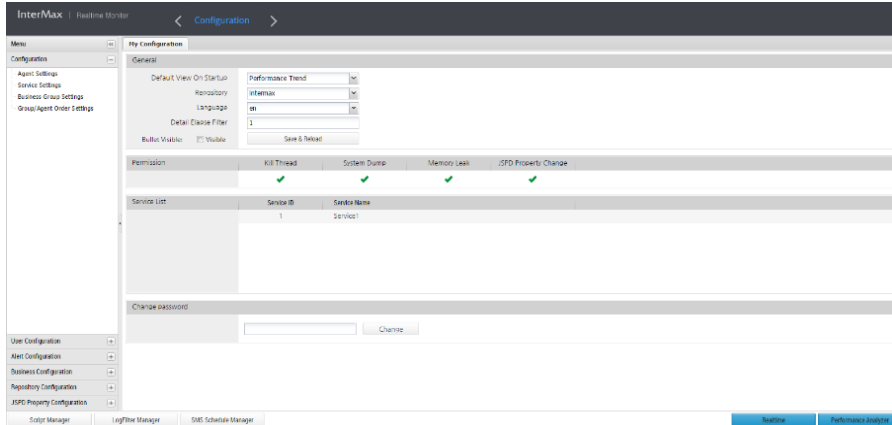
InterMax 를 사용하기 위해서는 크롬브라우저를 필요로 하며(크롬브라우저 설치 방법은 생략-최신버전 권장), 해당 브라우저를 통하여 다음과 같은 방법으로 진행한다.

1. 크롬 웹 브라우저에서 <http://127.0.0.1:8080/intermax/Config> 로 접속한다.



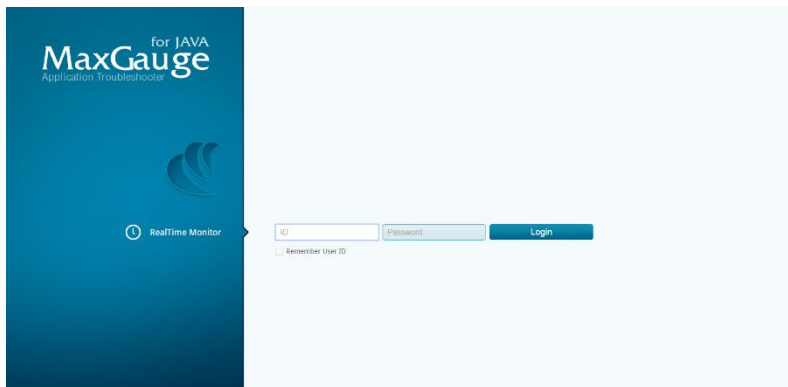
**Note.** IP 와 Port 는 Platform.JS 가 설치된 호스트 IP 와 서비스 Port 를 입력한다.

2. 기본적으로 제공되는 계정으로 접속한다. (ID: intermax / PW:manager)
3. Configuration 을 설정한다. 최소 하나 이상의 **Service Group** 을 구성하고 접속 사용자에게 **Service Privileges** 를 부여해야 한다.

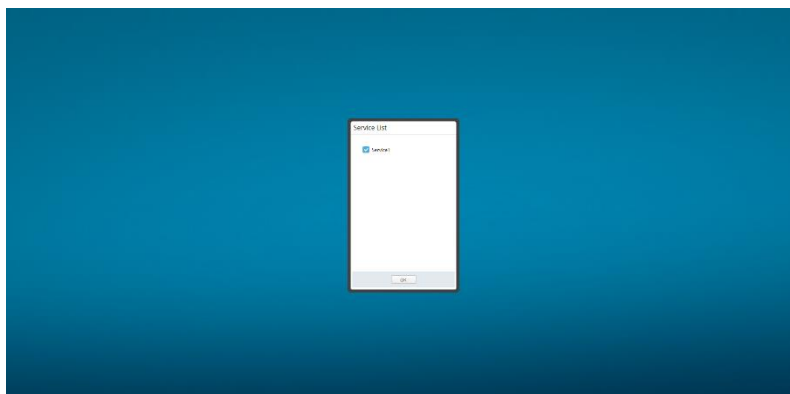


**Note.** InterMax Configuration 의 자세한 설정 방법은 “[InterMax Configuration Guide](#)”를 참고한다.

4. 크롬 웹 브라우저에서 <http://127.0.0.1:8080/intermax/RTM> 으로 접속하여 ID/Password 입력 후 로그인한다. (ID: intermax / PW:manager)



5. Configuration 과정에서 설정한 **Service Group** 목록이 나타나면, 모니터링 할 **Service Group** 을 선택 후 OK 를 누르면 모니터링 View 가 로드된다.



## 4. Appendix

### 4.1 WAS 벤더 별 InterMax Option 설정

2장에서 설명한 것과 같이 Java Version 에 따라 InterMax 옵션이 다르다.

#### Java Version 1.7 이상

```
-noverify -Djspd.wasid={WAS_ID} -javaagent:%JSPD_HOME%\lib\jspd.jar
```

#### Java Version 1.5 이상

```
-Djspd.wasid={WAS_ID} -javaagent:%JSPD_HOME%\lib\jspd.jar
```

#### Java Version 1.4

```
-Djspd.wasid={WAS_ID} -Xbootclasspath/p:%JSPD_HOME%\lib\jspd.jar
```

---

**Note.** 각 WAS 별 InterMax JSDP 옵션 설정은 이어지는 장을 참고한다.

---

#### 4.1.1. JEUS InterMax Option 설정

\$JEUS\_HOME/config/`hostname`/JEUSMain.xml 에 InterMax 옵션을 적용한다.

<command-option></command-option> 태그 사이에 삽입한다. 만약 기존 옵션이 적용되어 있다면 기존 옵션 뒤에 삽입 하도록 한다.

```
<?xml version="1.0" encoding="UTF-8"?>
<jeus-system version="6.0" xmlns="http://www.tmaxsoft.com/xml/ns/jeus"
  xmlns:ns2="http://java.sun.com/xml/ns/javaee" xmlns:ns3="http://java.sun.com
/xml/ns/persistence">
  <node>
    <name>InterMax1</name>
    <engine-container>
      <name>container1</name>
      <id>90</id>
      <base-port>9040</base-port>
      <command-option>
        -Djspd.wasid=1
        -javaagent:/home/dh/intermax/jspd/lib/jspd.jar
      </command-option>
      <engine-command>
        <type>ws</type>
        <name>engine1</name>
      </engine-command>
      <engine-command>
        <type>servlet</type>
        <name>engine1</name>
      </engine-command>
      <sequential-start>true</sequential-start>
```

---

**Note.** 문제가 발생하면 복원할 수 있도록 Script 수정 전 백업을 수행한다.

---

## 4.1.2. WebLogic InterMax Option 설정

{도메인 디렉터리(이하 \$DOMAIN\_HOME)}/bin/StartWeblogic.sh 에 InterMax 옵션을 적용한다.  
INTERMAX\_OPTION 을 Export 하고 JVM 수행 스크립트에 INTERMAX\_OPTION 을 넣어준다.

```

#####
export INTERMAX_OPT="-D:epd.wasid=60000 -javaagent:/home/intermax/intermax/jepd/lib/jepd.jar"
#####

# START WebLogic

echo "starting weblogic with Java version:"

[[JAVA_HOME]/bin/java ${JAVA_VM} -weblogic

if [ "${WLS_REDIRECT_LOG}" = "" ]; then
    echo "Starting WLS with line:"
    echo "${JAVA_HOME}/bin/java ${JAVA_VM} ${MEM_ARGS} -Dweblogic.Name=${SERVER_NAME} -Djava.security.poli
    cy=${WL_HOME}/server/lib/weblogic.policy ${INTERMAX_OPT} ${JAVA_OPTIONS} ${PROXY_SETTINGS} ${SERVER_CLASS}"
    [[JAVA_HOME]/bin/java ${JAVA_VM} ${MEM_ARGS} -Dweblogic.Name=${SERVER_NAME} -Djava.security.policy=${
    WL_HOME}/server/lib/weblogic.policy ${INTERMAX_OPT} ${JAVA_OPTIONS} ${PROXY_SETTINGS} ${SERVER_CLASS}
else
    echo "Redirecting output from WLS window to ${WLS_REDIRECT_LOG}"
    [[JAVA_HOME]/bin/java ${JAVA_VM} ${MEM_ARGS} -Dweblogic.Name=${SERVER_NAME} -Djava.security.policy=${
    WL_HOME}/server/lib/weblogic.policy ${INTERMAX_OPT} ${JAVA_OPTIONS} ${PROXY_SETTINGS} ${SERVER_CLASS} >"${W
    LS_REDIRECT_LOG}" 2>&
fi

```

---

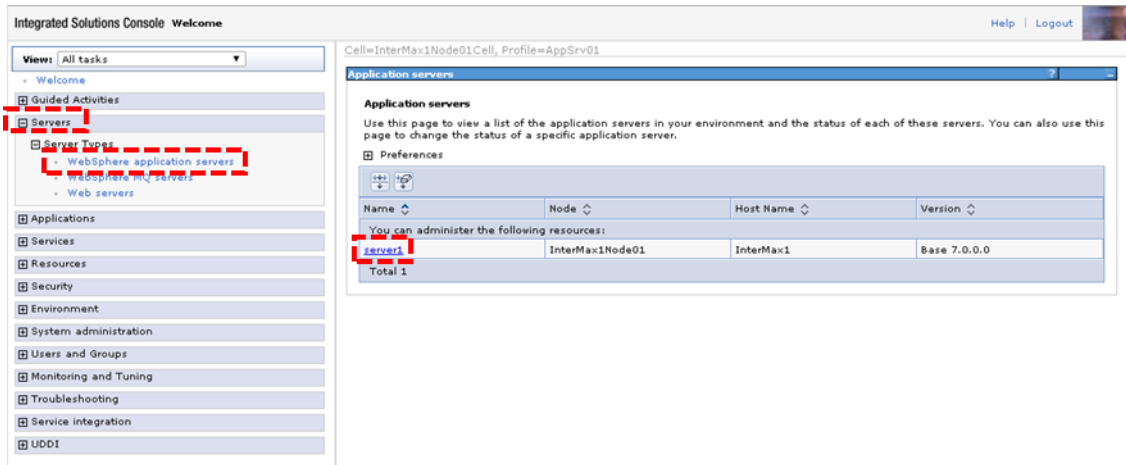
**Note1.** 문제가 발생하면 복원할 수 있도록 Scripts 수정 전 백업을 수행한다.

**Note2.** 벤더 버전 혹은 서버 구성에 따라 InterMax 옵션 넣는 위치는 조금씩 다를 수 있다.

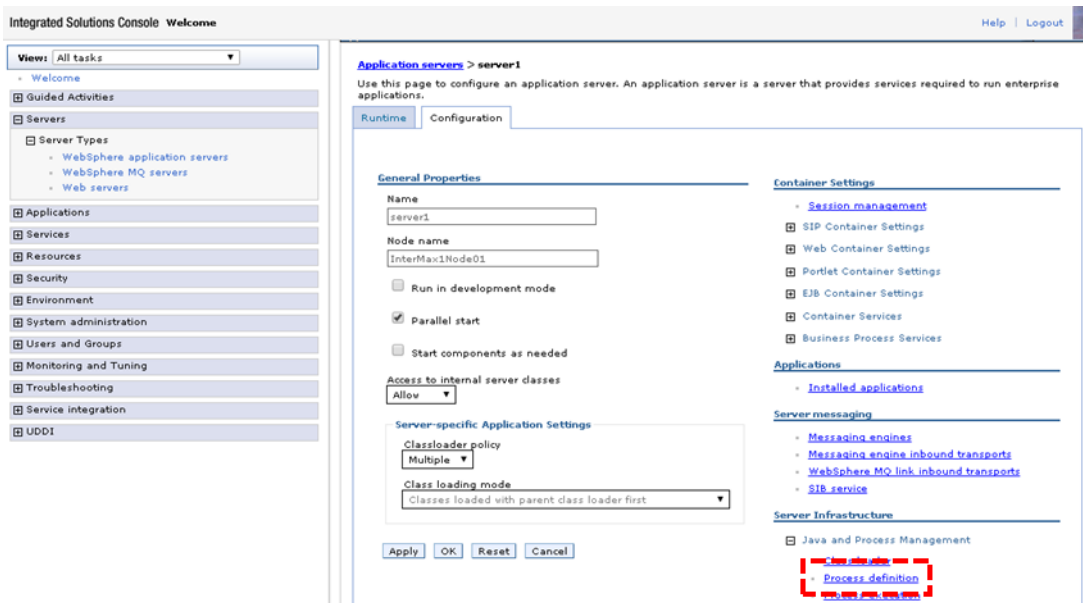
---

### 4.1.3. WebSphere InterMax Option 설정

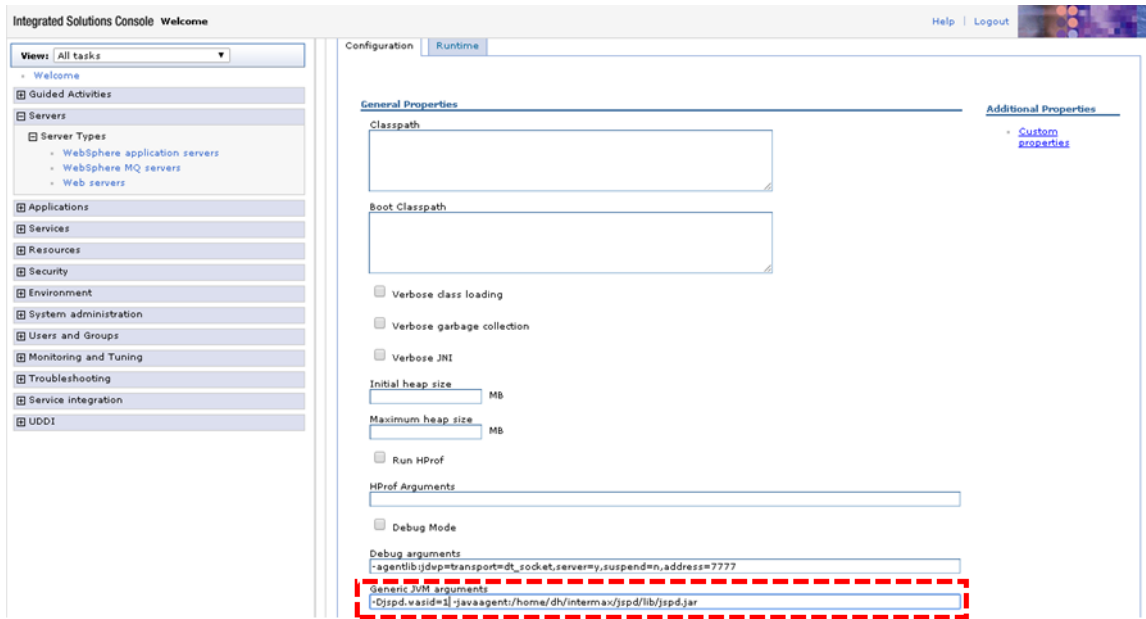
1. 웹스피어 웹콘솔에 접속한다.



2. Server -> Server Types -> WebSphere application server -> server1 클릭
3. Process Definition 클릭



4. Java Virtual Machine 클릭
5. Generic JVM arguments 부분에 InterMax 옵션 적용.



**Note.** 벤더 버전 혹은 구성된 구조에 따라 InterMax 옵션 넣는 위치는 조금씩 다를 수 있다.

#### 4.1.4. Tomcat InterMax Option 설정

`$CATALINA_HOME/bin/catalina.sh` 에서 InterMax 옵션을 적용한다.

```
JAVA_OPTS = "$JAVA_OPTS:$INTERMAX_OPTION"
```

```
##### multi_instance_wasid
JAVA_OPTS="-Djspd.wasid=3 -javaagent:/home1/jt/jy/multi/jspd_agent/jspd/lib/jspd.jar"
# OS specific support. $var _must_ be set to either true or false.
cygwin=false
darwin=false
os400=false
case "`uname`" in
```

**Note1.** 문제가

발생하면 복원할 수 있도록 Script 수정 전 백업을 수행한다.

**Note2.** 벤더 버전 혹은 구성된 구조에 따라 InterMax 옵션 넣는 위치는 조금씩 다를 수 있다.

#### 4.1.5. JBoss InterMax Option 설정

동작 방식에 따라 InterMax Option 넣는 위치가 달라진다.

```
$ Set up the module arguments
MODULE_OPTS=""
if [ "$SECmgr" = "true" ]; then
    MODULE_OPTS="$MODULE_OPTS -secmgr";
fi
JAVA_HOME="/usr/jdk1.7.0_79"
JAVA_OPTS="$JAVA_OPTS -noverify -Djspd.wasid=789 -javaagent:/home/baek/intermax0902/intermax/jspd/lib/jspd.jar"
```

Standalone 방식은 `$JBOSS_HOME/bin/standalone.sh` 에 InterMax Option 을 적용한다.

Multiple Instances 방식은 `$JBOSS_HOME/domain/configuration/host.xml` 에 InterMax Option 을 적용한다.

**Note.** JBoss7의 경우 OSGI 클래스 로더 구조로 standalone.conf 혹은 domain.conf에 아래와 같이 추가로 적용해야 한다. (-Djboss.modules.system.pkgs=org.jboss.byteman;com.exem)

```
# Specify options to pass to the Java VM.
#
if [ "x$JAVA_OPTS" = "x" ]; then
    JAVA_OPTS="-Xms64m -Xmx512m -XX:MaxPermSize=256m -Djava.net.preferIPv4Stack=true -Dorg.jboss.resolver.warning=true -Dsun.rmi.dgc.client.gcInterval=3600000 -Dsun.rmi.dgc.server.gcInterval=3600000"
    JAVA_OPTS="$JAVA_OPTS -Djboss.modules.system.pkgs=$JBASS_MODULES_SYSTEM_PKGS -Djava.awt.headless=true"
    JAVA_OPTS="$JAVA_OPTS -Djboss.domain.default.config=domain.xml -Djboss.host.d
    p.config.config=jboss.xml"
    JAVA_OPTS="$JAVA_OPTS -Djspd.wasid=123 -noverify -javaagent:/home/intermax/j
    pd/jboss_jspd/intermax/jspd/lib/jspd.jar"
else
    echo "JAVA_OPTS already set in environment; overriding default settings with
    values: $JAVA_OPTS"
fi
# Use JBoss Modules lockless mode
```

**Note1.** 문제가 발생하면 복원할 수 있도록 Script 수정 전 백업을 수행한다.

**Note2.** 벤더 버전 혹은 구성된 구조에 따라 InterMax 옵션 넣는 위치는 조금씩 다를 수 있다.

#### 4.1.6. Resin InterMax Option 설정

Resin 버전에 따라 InterMax Option 을 적용하는 위치가 다르다.

Resin 2.x/3.0.x 에서는 \$RESIN\_HOME/bin/httpd.sh 에 JAVA\_OPTIONS 관련 태그에 InterMax 옵션을 넣어주는데 앞에 -J가 붙게 된다.

예를 들어 -Djspd.wasid={WAS ID}-Jjavaagent:\$JSPD\_HOME/lib/jspd.jar 으로 써준다.

Resin 3.1.x 에서는 \$RESIN\_HOME/conf/resion.conf 에 Server 태그 안에 jvm-arg 태그에 InterMax Option 을 적용한다.

예를 들어

```
<jvm-arg>Djspd.wasid={WAS ID}</jvm-arg>
```

```
<jvm-arg>javaagent:$JSPD_HOME/lib\spd.jar</jvm-arg>
```

으로 써준다.

Resin 4.x 에서는 \$RESIN\_HOME/conf/resion.xml 에 jvm-arg 태그에 InterMax Option 을 적용한다.

```
<cluster id="app">
  <!-- define the servers in the cluster -->
  <server-default>
    <jvm-arg>-Xms258m</jvm-arg>
    <jvm-arg>-Xmx258m</jvm-arg>
    <jvm-arg>-Djspd.wasid=1</jvm-arg>
    <jvm-arg>-javaagent:/home/park/intermax/jspd/lib/jspd.jar</jvm-arg>
  </server-default>

  <server-multi id-prefix="app-" address-list="{app_servers}" port="6800"/>
  <host-default>
```

**Note1.** 문제가 발생하면 복원할 수 있도록 Script 수정 전 백업을 수행한다.

**Note2.** 벤더 버전 혹은 구성된 구조에 따라 InterMax 옵션 넣는 위치는 조금씩 다를 수 있다.

#### 4.1.7. OC4J(Oracle Containers for J2EE) InterMax Option 설정



동작 방식과 기동방식 따라 InterMax Option 넣는 위치가 달라진다.

Standalone 방식은 oc4j.jar 을 기동시키는 기동 스크립트에 InterMax Option 을 적용한다.

```
INTERMAX_OPTIONS="-Djspd.wasid=2-javaagent:/home/park/intermax/jspd/lib/jspd.jar"
check_oc4j()
{
    EXIT=0
    if [ "$JAVA_HOME" = "" ]
    then
        if [ "$VERBOSE" = "on" ]
        then
            echo "Executing: $JAVA_HOME/bin/java $JVMARGS -jar $OC4J_JAR $CMDARGS"
        fi
        $JAVA_HOME/bin/java $JVMARGS -jar ${INTERMAX_OPTIONS} $OC4J_JAR $CMDARGS
        $JAVA_HOME/bin/java $JVMARGS -jar $OC4J_JAR $CMDARGS
    fi
}
```

Multiple Instances 방식은 \$OC4J\_HOME/opmn/conf/opmn.xml <data id="java-options" value="> 태그 에 InterMax Option 을 적용한다.

```
<category id="start-parameters">
  <data id="java-options"
    value="-Xrs -server -XX:MaxPermSize=128M
          -ms512M -mx1024M -XX:AppendRatio=3
          -Djava.awt.headless=true
          -Dhttp.webdir.enable=false
          -Djspd.wasid=2
          -javaagent:/home/park/intermax/jspd/lib/jspd.jar"/>
</category>
```

**Note1.** 문제가 발생하면 복원할 수 있도록 Script 수정 전 백업을 수행한다.

**Note2.** 벤더 버전 혹은 구성된 구조에 따라 InterMax 옵션 넣는 위치는 조금씩 다를 수 있다.

## 4.1.8. GlassFish InterMax Option 설정

GlassFish 관리 콘솔 혹은 GlassFish\_HOME/domains/domain1/config/domain.xml 에 InterMax Option 을 적용한다.

```
<jvm-options>-Dfelix.fileinstall.disableConfigSave=false</jvm-options>
<jvm-options>-XX:NewRatio=2</jvm-options>
<jvm-options>-Djspd.wasid=1</jvm-options>
<jvm-options>-javaagent:/home/park/intermax/jspd/lib/jspd.jar</jvm-options>
</java-config>
</feature-config>
```

```
# 4. NetBeans profiler packages exist in parent class loader (see issue #6612)
# 5. BTrace exists in bootclasspath.
org.osgi.framework.bootdelegation=${eclipseLink.bootdelegation}, \
    com.sun.btrace, com.sun.btrace.*, \
    org.netbeans.lib.profiler, org.netbeans.lib.profiler.* \
    com.sun.*
# The OSGI R4.2 spec says boot delegation uses the boot class loader by default. We need
# to configure it to use the framework class loader because that class loader is
# configured with extra classes like idk tools.jar, derby jars, etc. that must be
```

\*GlassFish 의 경우 OSGI 클래스 로더 구조로 osgi.properties 에 아래와 같이 추가로 적용해야 한다.

**Note1.** 문제가 발생하면 복원할 수 있도록 Script 수정 전 백업을 수행한다.

**Note2.** 벤더 버전 혹은 구성된 구조에 따라 InterMax Option 넣는 위치는 조금씩 다를 수 있다.

[www.ex-em.com](http://www.ex-em.com)

서울 강서구 양천로 583, A동 1208호 ( 염창동, 우림 비즈니스센터 )

TEL : 02 6203 6300