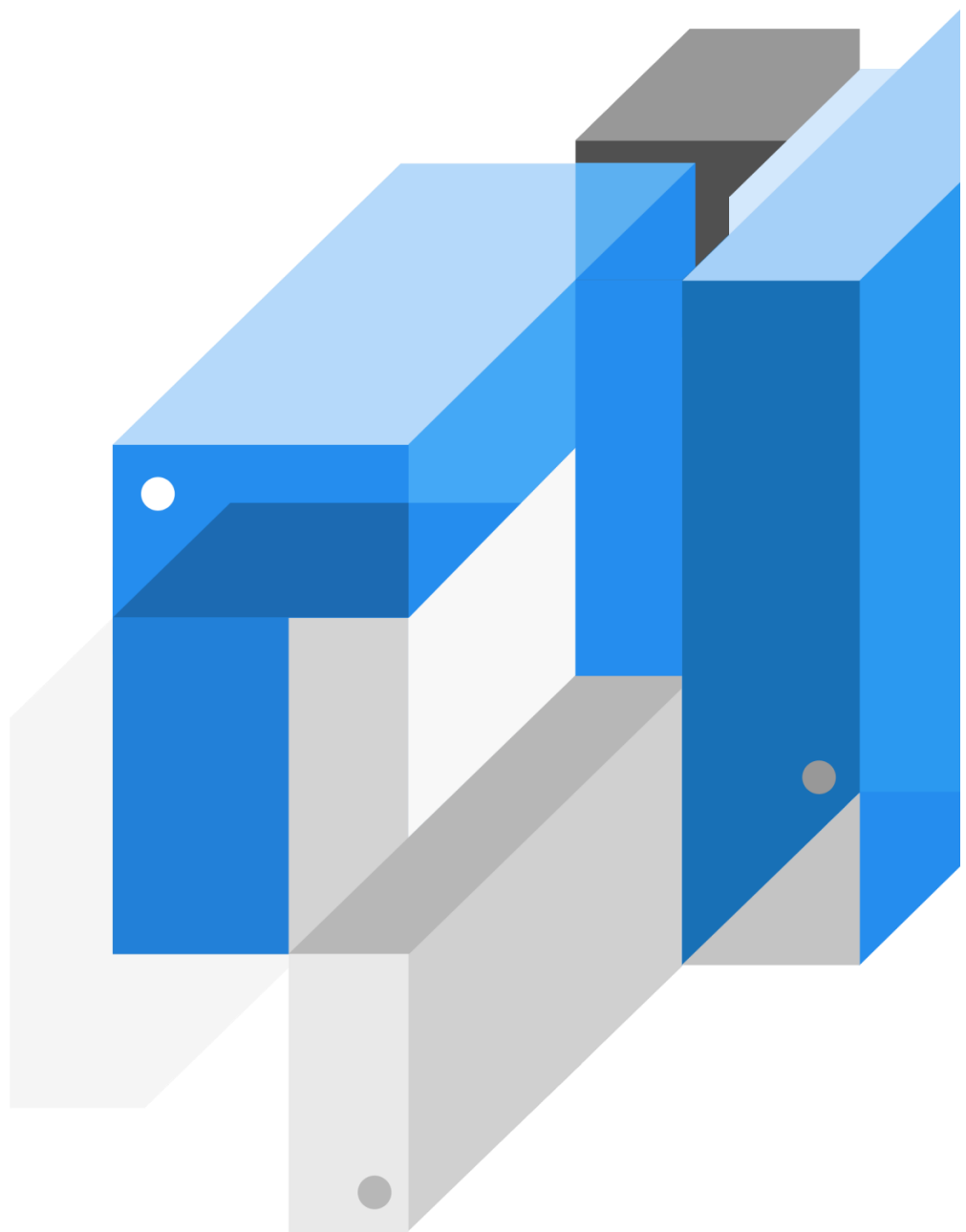


# MaxGauge for JAVA

Installation & Architecture Guide



MaxGauge

# Table of Contents

1. InterMax Architecture.....	7
1.1. InterMax Network Connection.....	9
Data Collection Layer.....	9
InterMax AP Server Layer.....	9
服务端口.....	10
1.2 Compatibility .....	10
JAVA 环境 .....	11
.Net 环境 .....	11
TP环境.....	11
InterMax产品的支持现状 .....	11
1.3. InterMax License .....	13
1.3.1. Trial License Key .....	13
1.3.2. Formal License Key .....	13
2. Data Collection Layer安装及构成.....	14
2.1 InterMax WAS Agent Set (JSPD) .....	14
2.1.1 事前准备事项 .....	14
Network Port (Windows) .....	14
Network Port (Unix / Linux) .....	14
WAS Agent Set构成 .....	15
2.1.2. 安装步骤 .....	15
Windows 环境 .....	15
Unix / Linux 环境 .....	16
DB监控联动时追加设置 .....	17

2.1.3. 各OS的追加设置 .....	17
AIX.....	17
Sun Solaris.....	18
2.1.4 启动方法 .....	19
2.1.5. InterMax WAS Agent 启动 .....	19
2.2 InterMax DB Agent Set (IMXDBM).....	19
2.2.1 事前准备事项 .....	19
OS用户权限 .....	19
InterMax Profile Setting (在Windows环境中可省略) .....	19
Oracle Version .....	20
Oracle Instance .....	20
Oracle Numa Segment .....	20
Shared Memory IPC key.....	20
Oracle PMON.....	21
Network Port .....	21
2.2.2. 安装步骤 .....	22
Windows 环境 .....	22
Unix / Linux 环境 .....	24
2.2.3. 启动方法 .....	27
Windows 环境 .....	28
Unix / Linux 环境 .....	28
2.2.4. 例外处理 .....	28
MakeConf Script Error (无相应Windows环境) .....	28
Run by sys.sql Error .....	29
Env & List.conf Error.....	29

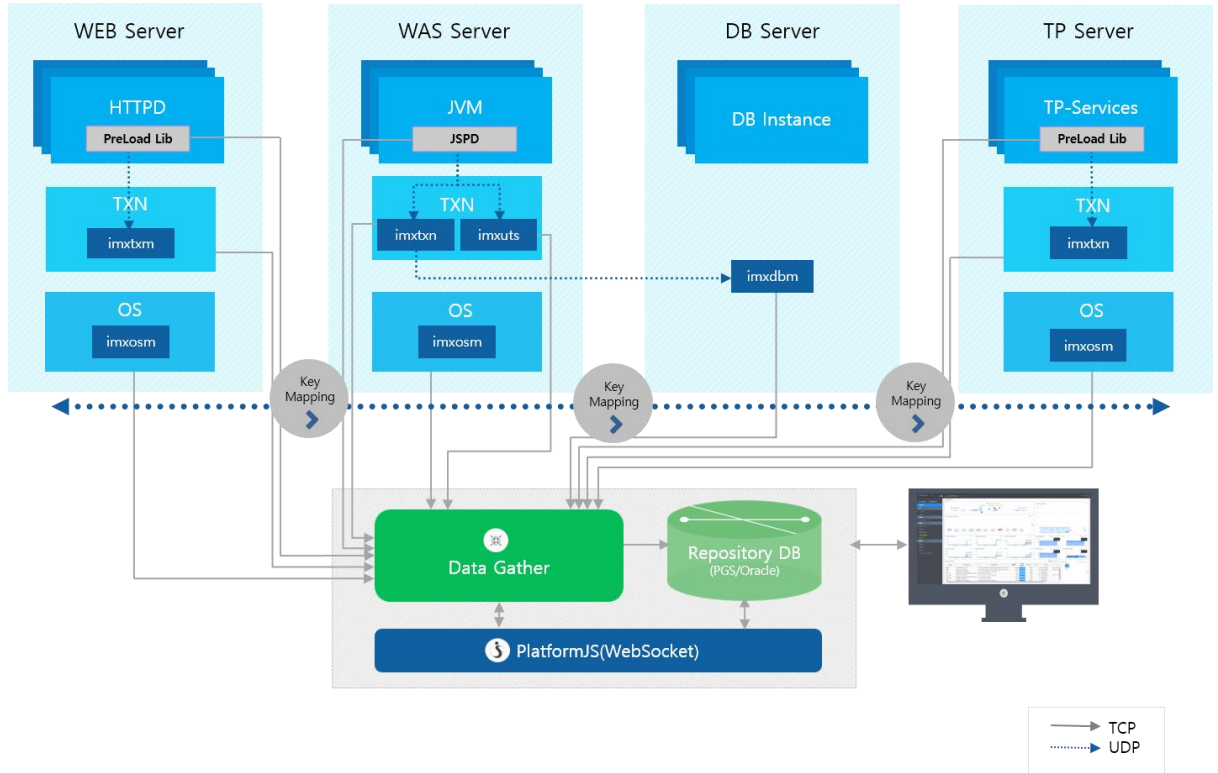
3. AP Server、Data Storage Layer的安装及构成 .....	31
3.1. 事前准备事项 .....	31
3.1.1. AP服务器配置 .....	31
3.2. Windows 环境 .....	31
3.2.1. 事前准备事项 .....	31
3.2.2. 安装步骤 ( 自动-Installer ) .....	32
3.2.3. 安装步骤 ( 手动 ) .....	33
PostgreSQL 手动安装 .....	34
Oracle 手动安装 .....	38
手动安装Data Gatherer .....	38
Platform. JS 手动安装 .....	43
3.2.4. 查看启动及连接 .....	44
InterMax Local Services .....	44
查看InterMax连接 .....	45
3.2.5. 用户定义选项 .....	46
添加Slave Gatherer Process .....	46
设置PostgreSQL Tablespace .....	47
3.3. Unix/Linux 环境 .....	49
3.3.1. 事前准备事项 .....	49
3.3.2. 安装步骤 ( 手动 ) .....	49
PostgreSQL 手动安装 .....	49
Oracle 手动安装 .....	50
手动安装Data Gatherer .....	51
Platform. JS 手动安装 .....	54

3.3.3. 启动方法 .....	56
启动InterMax PlatformJS .....	56
设置及查看InterMax的方法.....	56
4. Appendix.....	58
4.1 各WAS芯片供应商的InterMax Option设置 .....	58
4.1.1. JEUS InterMax Option 设置 .....	58
4.1.2. WebLogic InterMax Option 设置.....	59
4.1.3. WebSphere InterMax Option 设置 .....	60
4.1.4. Tomcat InterMax Option 设置.....	61
4.1.5. JBoss InterMax Option 设置.....	61
4.1.6. Resin InterMax Option 设置.....	62
4.1.7. OC4J(Oracle Containers for J2EE) InterMax Option 设置 .....	62
4.1.8. GlassFish InterMax Option 设置 .....	63



## 1. InterMax Architecture

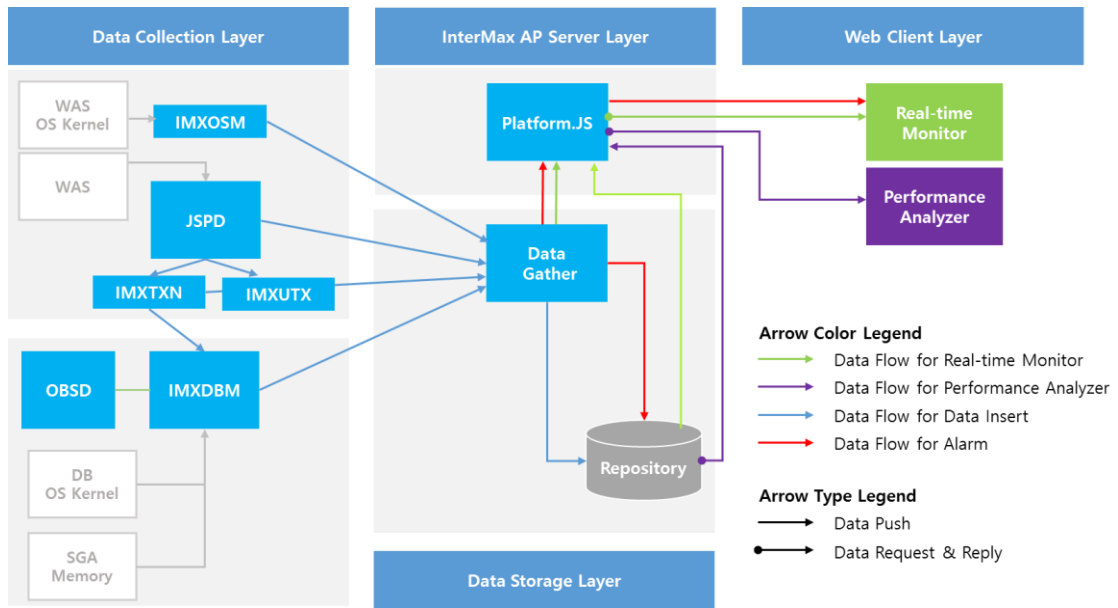
InterMax 不仅可以监控普通的 WEB^WAS^DB 环境的监控结构，还可同时从 End-To-End 角度监控用户终端到企业核心后端（Non-Java 环境）的全部区间。



企业的应用监控对象系统已安装相应 Agent，起到传输性能数据的作用。必须额外安装用于收集数据的服务器以及用于储存的数据库。

按照 WAS (JVM) 监控标准，InterMax 内部引擎模块分为以下 4 个基本 Layer。

- 1 **Data Collection Layer**:是直接监控企业的应用的 Agent 区域，收集多种性能数据并将其发送至服务器。
- 2 **InterMax Application Server Layer**:是 InterMax 服务器区域，是对从 Agent 获取的性能数据进行收集/分析/加工后保存至数据库并管理构成信息的专用 Web Daemon Server。
- 3 **Data Storage Layer**:是 InterMax 服务器加工过的性能数据的存储区域，用于保存并分析各种性能信息。
- 4 **Web Client Layer**:为了对收集到的性能数据进行实时监控及性能分析而提供网络 base 用户接口的区域。



InterMax Agent (JSPD) 及独立进程 (IMXTXN, IMXUTX) 收集多种性能数据并将其传送至收集服务器 (Data Gather)。收集服务器将对获取的性能信息进行分析及加工后保存至数据存储库 (Repository DB)，并管理各种构成信息等。保存的数据通过网络客户端的用户接口 (HTML5) 提供各种实时监控及性能指标、统计分析等接口。

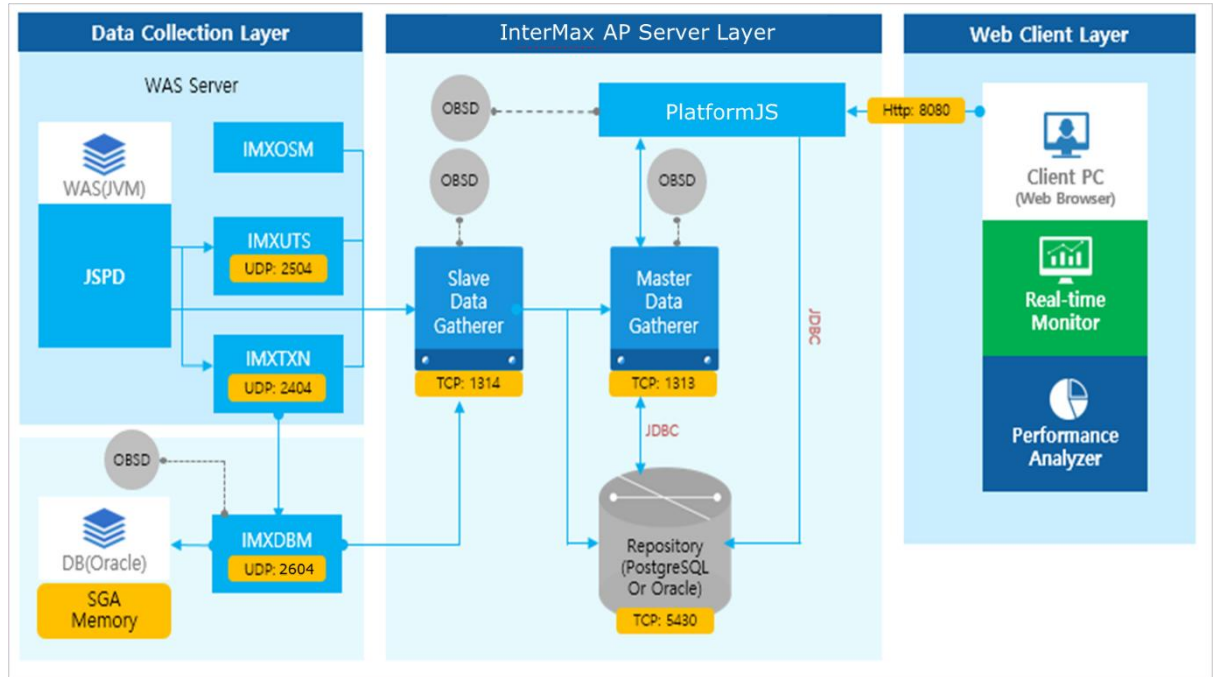
---

**Note.** InterMax AP Server Layer 及 Data Storage Layer 是 Logical 的区分。可在同一个服务器内构成两个 Layer。InterMax 体系结构的详细说明请参考 “[InterMax Administration Guide](#)”。



# 1.1. InterMax Network Connection

对处理 InterMax 各 Layer 间的网络通信所需的网络端口的构成进行说明。安装 InterMax 所需的网络端口如下。主要是关于端口号码的设置，默认端口号码已经被其他应用使用时，需变更端口号码。此外，在同一个硬件上安装多个 InterMax 环境的情况下，设置时要注意不要使相应端口号码重复。



## Data Collection Layer

范畴	说明
JSPD	通过 JVM 内部的 thread 驱动，并收集大部分与 WAS 相关的主要性能数据
IMXOSM	收集 OS 的系统资源(Memory、CPUetc)及统计信息、检查 WAS container。
IMXTXN	收集 SQL Query 信息
IMXDBM	以 SQL OWI 基准做统计&收集 Event 数据
IMXUTS	收集 Remote 信息(EtoE)
OBSD	内部 process 监控功能(以 30 秒为周期进行监控，down 时有重启功能)

## InterMax AP Server Layer

范畴	说明
Slave DataGatherer	是收集并处理从 Data Collection Layer 获取的数据模块，可根据收集对象数及 부하량 扩大为多个进行构成 建议每 50 个 instances 监控组成一个 Slave。(Slave 数=instance 数/50)
Master DataGatherer	通过 Slave DataGatherer 管理作用及 PlatformJS 提供申请相关信息
PlatformJS	通过用户的网络浏览器提供实时监控信息及分析信息
OBSD	内部 process 监控功能(以 30 秒为周期进行监控，down 时驱动)

## 服务端口

Source	Target	Port	Protocol	说明
JSPD	IMXTXN	2404	UDP	传送 SQL 相关信息
JSPD	IMXUTS	2504	UDP	传送 Remote 相关信息(EtoE)
IMXTXN	IMXDBM	2404	UDP	传送 DB 链接相关信息
JSPD	Slave Data Gather	1314	TCP	传送 JVM 主要性能信息
IMXOSM				传送 OS Resource 信息
IMXTXN				传送 SQL 相关信息
IMXUTS				传送 Remote 相关信息(EtoE)
IMXDBM				传送 DB 链接相关信息
Slave Data Gather	Master Data Gather	1313	UDP	传送 Server 及 DB Agent 信息
Slave Data Gather	Repository	5430	TCP	保存 Server 及 DB Agent 信息
Master Data Gather				保存统计性信息
PlatformJS	Web Client	8080	TCP	传送浏览器显示的信息

## 1.2 Compatibility

InterMax 产品的支持范围及可互换版本如下。

## JAVA 环境

操作系统 (OS)	应用服务器 (WAS)	支持 DB
AIX 5. x 以上 (32/64bit)	WebLogic 10. x 以上版本	Oracle
HP-UX IA64	WebSphere 6.1 以上版本	DB2
Linux (32/64bit)	JEUS 5. x 以上版本	MS SQL Server
Solaris SPARC (32/64bit)	Tomcat 5. x 以上版本	Mysql
Solaris (x86/x64)	Oracle Application Server (OC4J)	Postgres
Windows Server 2003 以上版本 (x86/x64)	Resin 3. x 以上版本	Sybase
	Jboss 5. x 以上版本	Tibero
	GlassFish 2. x 以上版本 (JDK 1.5 以上版本)	

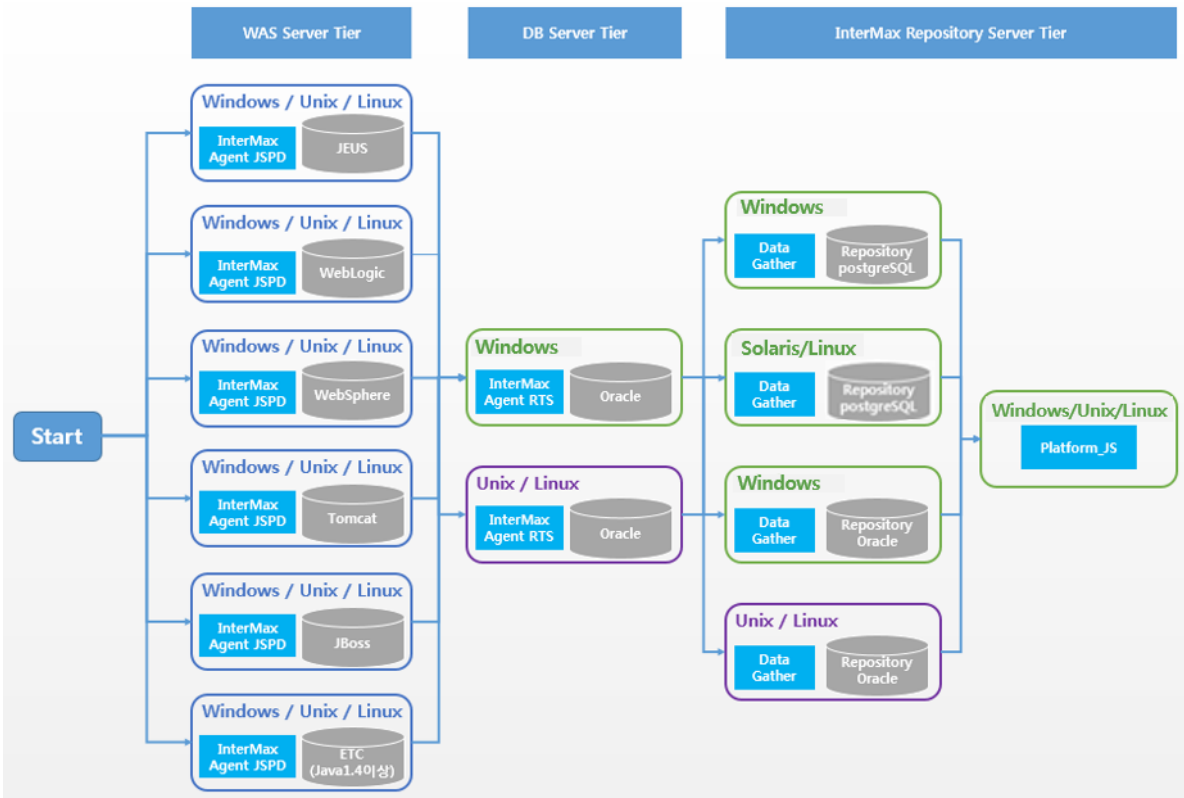
## .Net 环境

操作系统 (OS)	Web 服务器	应用服务器哦	支持 DB
Windows Server 2003 以上版本 (x86/x64 )	IIS 6.0 以上版本	.NET Framework 2.0 以上版本	MS SQL Server 2008 以上版本

## TP环境

操作系统 (OS)	应用服务器 (WAS)	支持 DB
AIX 5. x 以上版本 (32/64bit)	TMAX 5. x 以上版本	Oracle
HP-UX IA64	TUXEDO 10. x 以上版本	DB2
Linux (32/64bit)	TIBCO 5. x 以上版本	MS SQL Server
Solaris SPARC (32/64bit)		Mysql
Solaris (x86/x64)		Postgres
		Sybase
		Tibero

## InterMax产品的支持现状



## 1.3. InterMax License

为驱动 InterMax Agent Set 需要 InterMax License Key。

### 1.3.1. Trial License Key

Trial License Key 用于 Test，使用时间有限。

### 1.3.2. Formal License Key

Formal License Key 将在产品签约后发放，申请 License Key 时需提供以下信息。

范畴	项目
工作名称	工作名称
OS 信息	Unix Type Unix Version Unix Bit Level
Database 信息 (Oracle 时)	Oracle Version Oracle Bit Level Oracle SID
Host Server 信息	IP Address Host ID Real CPU Dual Core Count

---

**Note.** InterMax License 政策是 CPU Core 单位，通过 Server 的 Host ID 及 CPU Core 数检查 Formal License Key 的有效性。获取的 License Key 仅限在对应服务器使用，对应服务器的 CPU Core 增加时，Formal License Key Validation 检查将发生错误，且 InterMax Agent Set 无法正常运行。因此，CPU Core 数增加时，必须提前重新申请 Formal License Key。（根据情况，可能需要重签合同）

---

## 2. Data Collection Layer 安装及构成

### 2.1 InterMax WAS Agent Set (JSPD)

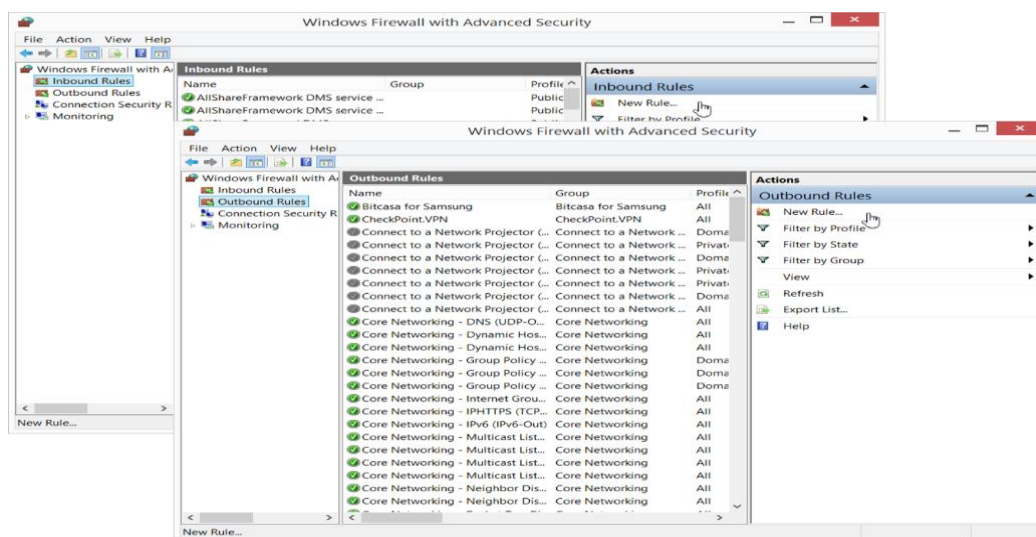
#### 2.1.1 事前准备事项

项目	配置（建议）标准
WAS Type	J2EE base
Java Version	支持 Java 1.4 以上 ~ 1.8 版本
OS Disk Size	Agent Set Size : 100MB

**Note.** 可监控 JAVA base 守护进程 (Deamon)

#### Network Port (Windows)

WAS Agent 通过 1314 TCP 端口与 Slave Data Gatherer 通信。在控制面板允许 1314 TCP 端口的 Inbound/Outbound。



#### Network Port (Unix / Linux)

JSPD 通过 1314 TCP 端口与 Slave Data Gatherer 通信。查看是否使用端口的方法如下。

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

**Note.** 必须打开 DB Server 对应的 2404(Default) UDP 端口。

## WAS Agent Set构成

除添加到 WAS (JVM) 守护进程的 JSPD 模块以外，安装在相应服务器中的 agent 进程按功能可分为以下三种 agent set。

- **IMXOSM**：收集相应 OS 的 Memory 或 CPU 等 OS Resources 相关信息。
- **IMXTXN**：收集 SQL 相关信息。
- **IMXUTS**：收集 Remote Data 相关信息。

## 2.1.2. 安装步骤

### Windows 环境

- 1 安装 WAS Agent 需要如下安装文件。将以下文件上传至 WAS 服务器。

文件名	说明
InterMax_Agent_YMMDD.tar	WAS Agent 安装文件
License_.key	许可证文件

- 2 在 WAS Os User Home Directory 中解压已上传的文件。

{解压路径}\intermax\jspd 目录，以下称%JSPD\_HOME%。

**Note.** 可变更解压位置。

- 3 为输入进程信息及 Data Gatherer 信息，

修改%JSPD\_HOME%\cfg\agent\jspd.prop 文件。默认设置参数如下。

项目	说明
WR_ADDR	输入 <b>Slave Data Gatherer</b> 的 IP:Port 信息。
TXN_ADDR	输入 IMXTXN port 信息。
UTS_ADDR	输入 IMXUTS port 信息。

### 执行示例

```
# WR_ADDR  
WR_ADDR=192.168.123.52:1314  
  
# ${UDP_PORT}|UDP_PORT}  
TXN_ADDR=2404  
  
# ${UDP_PORT}|UDP_PORT}  
UTS_ADDR=2504
```

- 4 为启动 InterMax WAS Agent 需将 InterMax 选项适用到各 WAS 的 Start Batch 文件。InterMax 选项如下。

### Java Version 1.7 以上版本

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

### Java Version 1.5 以上版本

```
-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 用于映像各 WAS 及各 Agent，可赋予 1 至 65535。注意不要重复设置同一个号码。

**Note2.** 在 Java Version 1.4 以下的环境下，需移动至 %JSPD\_HOME%\build-jdk 文件夹运行 build.bat jdk。

- 5 详情请参考各“Appendix.WAS vendor 提供的 InterMax Option 设置”。
- 6 为适用 license，将 License file 复制到 %JSPD\_HOME%\cfg\ 目录。

**Unix / Linux 环境**

- 1 安装 WAS Agent 需要如下安装文件。以 Binary 形式上传相应文件。

文件名	说明
InterMax_Agent_YYMMDD.tar	WAS Agent 安装文件
License_.key	许可证文件

- 1 在 WAS OS User Home Directory 中解压已上传的文件。解压方法如下。{解压路径}/Jspd 目录，以下称 %JSPD\_HOME%。

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

- 2 为输入内部进程信息及 Data Gatherer 信息，修改 %JSPD\_HOME%/cfg/agent/jspd.prop 文件。基本设置参数如下。

项目	说明
WR_ADDR	输入 Slave Data Gatherer 的 IP:Port 信息。
TXN_ADDR	输入 IMXTXN port 信息。
UTS_ADDR	输入 IMXUTS port 信息。

**3 执行示例**

```
# WR_ADDR
WR_ADDR=192.168.123.52:1314

# ${UDP_PORT}|UDP_PORT}
TXN_ADDR=2404

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

- 4 为启动 InterMax WAS Agent 需将 InterMax 选项适用到各 WAS 的 Start script 文件。InterMax 选项如下。

**Java Version 1.7 以上版本**

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

**Java Version 1.5 以上版本**

```
-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 用于映像各 WAS 及各 Agent，可赋予 1 至 65536。注意不要设置重复号码。

**Note2.** 在 Java Version 1.4 以下的环境下，需移动至 \$JSPD\_HOME/build-jdk 文件夹运行 build.sh jdk。

---

5 为适用 license，将 **License file** 复制到 \$JSPD\_HOME/cfg/ 目录。

## DB监控联动时追加设置

为 DB 的监控修改 \$JSPD\_HOME/cfg/{sid}/imx.prop 文件。

项目	说明
IMX_ADDR	IMXDBM 地址及 UDP PORT
DB_ADDR	监控目标 DB 地址 (IP)。端口。SID 信息

### 执行示例

```
# IMX_ADDR = DB_ADDR
10.10.202.183:2404=10.10.202.183.1521.ora112
```

---

**Note1.** DB IP 需输入实际 IP，而非 Virtual IP。

**Note2.** SID 小写输入。

**Note3.** TXN\_ADDR 端口是在 IMXDBM 的 jspd.prop 中设置的 UDP\_PORT

---

## 2.1.3. 各OS的追加设置

各 OS 分别需要如下追加设置。

### AIX

收集 GC 相关数据 (执行时间、执行次数) 时，需进行以下设置。

项目	说明
JAVA 1.4 以下版本	在 \$JAVA_HOME 执行 find . -name *.so command。 在子目录中查找 *.so 文件集中的位置。 JAVA 32 bit 的情况下，将 libXmJvmpiSvc_32.so 文件复制到 ppc 的子目录中 *.so 文件集中的位置。 JAVA 64 bit 的情况下，将 libXmJvmpiSvc_64.so 文件复制到 ppc64 的子目录中 *.so 文件集中的位置。
JAVA 1.5 以上版本	从 \$JAVA_HOME 执行 find . -name *.so command。 在子目录中查找 *.so 文件集中的位置。 JAVA 32 bit 的情况下，将 libXmJvmtiSvc_32.so 文件复制到 ppc 的子目录中 *.so 文件集中的位置。 JAVA 64 bit 的情况下，将 libXmJvmtiSvc_64.so 文件复制到 ppc64 的子目录中 *.so 文件集中的位置。

### 执行示例

Java Version 1.4

```

$JSPD_HOME/lib/jni/libXmJvmpiSvc.so
32 bit 时
$ cp $JSPD_HOME/lib/jni/libXmJvmpiSvc.so ${JAVA_HOME}/../ppc/
64bit 时
$ cp $JSPD_HOME/lib/jni/libXmJvmpiSvc.so ${JAVA_HOME}/../ppc64/
复制文件后就爱那个用户权限与相应目录权限一同进行更改。
$ cd ${JAVA_HOME}/../ppc{_64}
$ chown root:root libXmJvmpiSvc.so

```

### Java Version 1.5 以上版本

```

$JSPD_HOME/lib/jni/libXmJvmtiSvc.so
32 bit 时
$ cp $JSPD_HOME/lib/jni/libXmJvmtiSvc.so ${JAVA_HOME}/../ppc/
64bit 时
$ cp $JSPD_HOME/lib/jni/libXmJvmtiSvc.so ${JAVA_HOME}/../ppc64/
复制文件后就爱那个用户权限与相应目录权限一同进行更改。
$ cd ${JAVA_HOME}/../ppc{_64}
$ chown root:root libXmJvmtiSvc.so

```

---

**Note1.** 安装 Java 的 User 为 root 时，需要 root 权限。

**Note2.** 上述的 \$JAVA\_HOME 是指实际 WAS 使用的 JAVA。

---

## Sun Solaris

OS 为 Sun 时，需要以下附加设置。

项目	说明
Sun	<p>在 \$JAVA_HOME 执行 <code>find . -name *.so command</code>。</p> <p>在子目录中查找 *.so 文件集中的位置。</p> <p>JAVA 32 bit 的情况下，将 <code>libgcc_s.so.1_32</code> 文件复制到 <code>sparc</code> 的子目录中 *.so 文件集中的位置。</p> <p>JAVA 64 bit 的情况下，将 <code>libgcc_s.so.1_64</code> 文件复制到 <code>sparc9</code> 的子目录中 *.so 文件集中的位置。</p>

### 执行示例

```

$JSPD_HOME/lib/jni/libgcc_s.so.1
32 bit 时
$ cp $JSPD_HOME/lib/jni/ libgcc_s.so.1 ${WAS_JAVA}/../sparc/
64bit 时
$ cp $JSPD_HOME/lib/jni/ libgcc_s.so.1 ${WAS_JAVA}/../sparc9v/
复制文件后就爱那个用户权限与相应目录权限一同进行更改。
$ cd ${WAS_JAVA}/../sparc{9v}
$ chown root:root libgcc_s.so.1

```

---

**Note1.** 安装 Java 的 User 为 root 时，需要 root 权限。

**Note2.** 上述的 \$JAVA\_HOME 是指实际 WAS 使用的 JAVA。

---

## 2.1.4 启动方法

各 OS 分别进行设置后重启 WAS。InterMax Agent 无特殊的管理 Point。因此，启动方法同 WAS 启动。

## 2.1.5. InterMax WAS Agent 启动

1. JSPD 启动，则启动 IMXOSM。(JSPD 在 WAS (JVM) 启动时同时启动)
2. IMXOSM 启动，则启动 IMXTXN。
3. IMXOSM 启动，则启动 IMXUTS。

## 2.2 InterMax DB Agent Set (IMXDBM)

### 2.2.1 事前准备事项

项目	配置 (建议) 标准
Oracle Version	Oracle 9i 以上版本
OS Disk Size	Agent Set Size : 10MB

### OS用户权限

安装 **Oracle** 时，生成拥有相同权限的用户或所属于 **DBA** group 的用户并安装 **DB Agent**。Linux 系列和 Unix 系列分别使用 Bash 和 Ksh。生成方法如下。

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

**Note1.** 已安装 Maxgauge 时，无需生成 OS User，可直接通过 MaxGauge 的 OS User 进行安装。

**Note2.** Windows 环境下的用户生成可在控制面板 > 用户账号进行。

### InterMax Profile Setting (在Windows环境中可省略)

为连接 DBMS，在 Oracle 用户的 .profile 中将 ORACLE\_HOME、ORACLE\_BASE、ORACLE\_SID、PATH 添加至 intermax 用户的 .profile。

```
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

查看相应 Instance 的 Oracle Version 信息。方法如下。

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

### 执行示例

```
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

查看相应 Instance 名称。查看方法如下。

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

### 执行示例

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

## Oracle Numa Segment

InterMax 同时支持 Uniform Memory Access 及 Non-uniform Memory Access (NUMA) 两种方式，因此需确认服务器的 NUMA 与否。通过 SID 排列确认 NUMA 的方法如下。

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

### 执行示例

```
SID
-----
21
22
126      <- SID 排列存在增加的部分时使用 Numa Segment
127
128
```

---

**Note1.** NUMA 结构的服务器使用分散 segment，因此 SID 以 10~100 为单位增加。Oracle 11g 以上版本大部分使用 NUMA segment。

**Note2.** 这里所提到的 NUMA 不是 NUMA 体系结构。是根据 Oracle Session Structure Array 位于连续的内存空间还是分散在 2 个以上的内存空间而称其为 UMA、NUMA。

---

## Shared Memory IPC key

InterMax 的 DB Agent 通过 Shared Memory 的 IPC key Address 执行 SGA Direct Access。查看相应 Instance 的 IPC Key 的方法如下。(Windows 环境下用 SID 代替)

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

### 执行示例

```
----- 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.** 一台 Instance 中存在 2 个以上的 IPC Key 值时，利用 **Oradebug** 查看准确的 IPC Key 值。

---

通过 Oradebug 查看 IPC Key 的方法如下。

```
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
```

### 执行示例

```
...
Area #5 `skgm overhead' containing Subareas 5-5
Total size 0000000000003000 Minimum Subarea size 00000000
Area Subarea Shmid Stable Addr Actual Addr
  5      5 4784147 0x00000092000000 0x00000092000000
...
```

---

**Note.** 查看 'skgm overhead' 区间的 shmid 值，并通过 ipcs command 查看相应 shmid 的 IPC key 值。

---

## Oracle PMON

查看相应 Instance 的 Oracle PMON 名称及 User。  
查看方法如下。(Windows 环境下用 Oracle Process 名称代替)

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

### 执行示例

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

## Network Port

为从 WAS Agent 获取事务信息，需使用 2404 UDP 端口。是否使用端口的查看方法如下。

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

---

**Note.** TCP 可通过 netstat command 查看，而 UDP 无法通过 netstat command 查看。

---

## 2.2.2. 安装步骤

### Windows 环境

#### 1. 上传安装文件

InterMax 需要如下安装文件，需将相应文件复制到服务器中。

文件名	说明
InterMax_DBM_[OS Ver]_[Oracle Ver].tar	InterMax DB Agent Set 安装文件
License_.key	许可证文件

#### 2. 解压安装文件

解压已上传的文件。{解压路径}\intermax\，以下称%IMX\_HOME%。

---

**Note.** 可变更解压位置。

---

当前的 Window 版本不支持自动安装。手动执行安装。

#### 执行示例

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

#### 3. 设置环境文件

为设置 DB Agent，修改位于%IMX\_HOME%\cfg\{SID}RTS\ 的环境设置文件。

#### Jspd.prop

默认设置参数如下。

项目	说明
WR_ADDR	输入 Data Gatherer 的 IP 信息及端口。
TXN_ADDR	输入 IMXDBM 连接 Port 信息。

#### 执行示例

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

# ${UDP_PORT|UDP_PORT}
TXN_ADDR=2404
```

#### Imx.prop

基本设置参数如下。

项目	说明
----	----

DB_ADDR	DB IP.LISTENER PORT.sid
---------	-------------------------

### 执行示例

```
# 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 需输入实际 IP，而非 Virtual IP。  
**Note2.** PORT 是 Oracle 的 LISTENER PORT。  
**Note3.** SID 小写输入。

### common.conf

默认设置参数如下。

项目	说明
ipc_key	输入 SID 名称。
pmon_name	输入 Oracle 进程名称。

### 执行示例

```
# Oracle shared memory key
ipc_key=ora112

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

**Note1.** InterMax common.conf 设置相关的详细说明请参考“[InterMax Administration Guide](#)”。

## 4. 生成环境文件

Windows 下需手动生成 IMXDBM 使用的环境文件。

前往%INTERMAX\_HOME%\util\db\_setup 文件夹。

### 生成 Maxgauge User

生成 Intermax 中使用的 DB User 并赋予权限。

通过 SYS User 连接到 SQL\*PLUS 并执行 **run\_by\_sys.sql**。

### 执行示例

```
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 生成**

5. 生成用于 Intermax 中使用的 DB 的 Stat 及 Event 信息收集的环境文件。

通过 maxgauge User 连接到 SQL\*PLUS 并执行 **listconf3.sql**。

执行示例

```
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 生成**

6. 生成用于 Intermax 的环境文件。执行 mkenv.exe。

执行示例

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

---

**Note.** Env 文件及 list.conf 文件生成后，将其复制到以下位置。

%INTERMAX\_HOME%\cfg\{SID}RTS

---

7. Service 注册及删除

Service 注册指令如下。必须在拥有管理员权限的输入窗口中执行。

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

**执行示例**

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

Service 删除指令如下。

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

**执行示例**

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

8. 适用许可证文件

将 License file 移动至 %IMX\_HOME%\cfg。

**Unix / Linux 环境**

1. 上传安装文件

InterMax 需要如下安装文件，需将相应文件以 Binary 形式上传。

文件名	说明
InterMax_DBM_[OS Ver]_[Oracle Ver].tar	InterMax DB Agent Set 安装文件
License_.key	许可证文件

**执行示例**

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. 解压安装文件

在 maxgauge 用户 Home Directory 中解压已上传的文件。解压方法如下。

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

### 执行示例

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

## 3. 运行 InterMax 环境文件

前往 InterMax Home 并执行环境变数(.mxgrc) 文件。

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

```
.$ ..mxgrc.
```

## 4. 运行 Install Script

通过 Install 文件夹内的 **install.sh** 执行自动安装。

项目	说明
DBM setup Type	监控目标 DB Type
Database owner	驱动 Oracle Instance 的 OSUser
Conf name	[ORACLE_SID]RTS ORACLE_SID 大写输入
IPC Key	安装时必须检查事项的 Oracle Shared Memory Key
PMON process	安装时必须检查事项的 Oracle PMON Name
DBM UDP port	WAS Agent 及通信端口(Default 2404)
Data Gather IP address	安装 DG Slave 的 IP 地址
Data Gather Port	DG Slave 及通信端口(Default 1314)
DBM EVV Server port	DBM 内部通信端口(Default 2405)
DB_ADDR IP ADDRESS	Database 安装地址
DB_ADDR PORT	Database 的 LISTENER PORT
DB_ADDR Database Name	Database 的 SID
Oracle Database user	生成 MaxaGauge DB User
Oracle Database Password	MaxGauge DB User 密码
Default Tablespace	MaxGauge User 的 Default Tablespace
Temporary Tablespace	MaxGauge User 的 Temporary Tablespace
Conf file	生成服务器 agent Configuration 文件
Run_by_sys	MaxGauge DB User 的生成及权限的赋予
Env	生成 Agent 所需的环境 File
List.conf	生成 Agent 所需的环境 File

### 执行示例

```
$ 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 : [2404]
2404

DataGather IP Address : []
192.168.0.10

DataGather Port number : [1314]
1314

DBM ENV Server Port numbe : [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. 适用许可证文件

将 **License file** 移动至 `$INTERMAX_HOME/cfg` 目录。

...

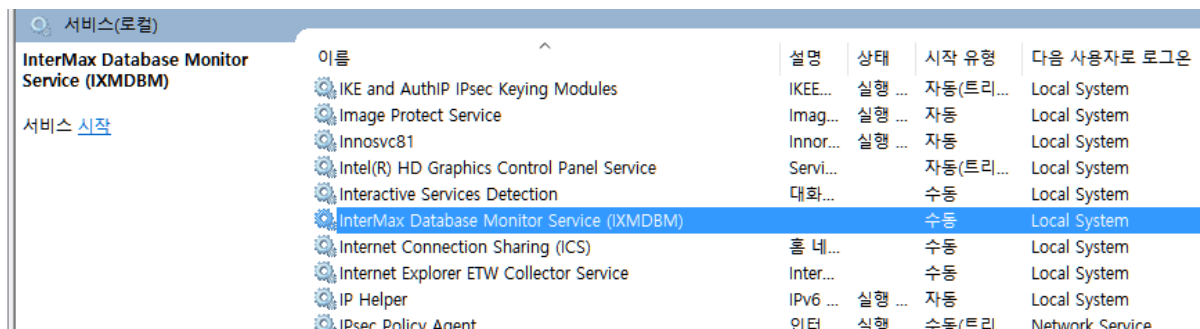
```
$ mv $HOME/License.key $INXTERMAX_HOME/cfg
```

## 2.2.3. 启动方法

## Windows 环境

通过 services.msc 目录驱动 InterMax DB Agent。

InterMax DB Agent Set 通过 Window Local Service 注册，执行 Service(Local) 中的各项服务并启动。



## Unix / Linux 环境

通过 IMXCTL Command 驱动 InterMax DB Agent。

**IMXCTL** 是控制 **InterMax Agent Set** 的 Utility，有在 OS command line 中使用的 Non Interactive Mode 方式和 **IMXCTL** Utility 中使用的 Interactive Mode 方式。**IMXCTL** Utility 使用方法如下。

#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	说明
start	InterMax Agent Set 开始
stop	InterMax Agent Set 结束
status (stat)	InterMax Agent Set 情况的掌握
restart	InterMax Agent Set 重启
version (ver)	输出 InterMax Agent Set 版本

Note. IMXCTL Utility 的详细说明及使用示例请参考“[InterMax Administration Guide](#)”。

### 2.2.4. 例外处理

#### MakeConf Script Error (无相应Windows环境)

执行 Install.sh 时，不生成 Conf 文件时，请参考以下项目。

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

Script Name	说明
-------------	----

Makecommonconf	为 Direct Memory Access 在 Oracle SGA 生成所需的环境文件 {ORACLE_SID} {IPC_KEY} {PMON_NAME}
Makertsconf	生成实时 Data 及 Log Data 传送的环境文件 {ORACLE_SID} {RTS_PORT} {DG_IP_ADDRESS} {DG_PORT}

### Script 的执行方法及变量的写入方法

```
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

执行 Install.sh 时，生成 maxgauge User 及权限赋予失败，则参考以下项目。

```
$ 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 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

执行 Install.sh 时，生成 Env 及 List.conf 文件失败，则可在 \$INTERMAX\_HOME/util/db\_setup 执行 mke.sh，listconf3.sql 并手动生成。

#### 执行示例

```
# 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 文件及 list.conf 文件生成后, 将其复制到以下位置。  
\$INTERMAX\_HOME/cfg/{SID}RTS

---

## 3. AP Server、Data Storage Layer的安装及构成

AP Server 及 Data Storage Layer 由 Platform JS、Data Gatherer、Repository Database 三种组成。各项支持的 OS Type 如下。

安装及构成项目	具体功能	支持的 OS
Platform JS	通过 Client PC 进行监控及分析用的 UI View 端模块	Windows, Unix/Linux
Data Gatherer	收集/加工/分析性能数据的服务器端模块	Windows, Unix/Linux
Repository Database	保存已收集数据的数据库存储器	PostgreSQL (Windows, Unix/Linux) Oracle (Windows, Unix/Linux)

### 3. 1. 事前准备事项

#### 3. 1. 1. AP服务器配置

InterMax 的 AP 服务器及用于 Data Storage 的服务器配置根据监控目标系统的规模及所收集的量，要事先与客户公司协商准备。一般来说，10-node 以内、50-instances 以内、1,000TPS 以下的事务服务需如下配置。

Repository DB 建议尽量分离构成，收集大容量时建议另行构成 Storage 服务器。

3 项目	最低配置	备注
支持的 OS	Windows, Linux, HP, AIX, Solaris	
支持的 JDK	支持 JDK 1.8	
CPU(Core)	2CPU(4Core) 以上(1.8GHz 以上)	
Memory	8GB 以上闲置内存	建议 16G 以上
Hard Disk	安装空间-100GB 以下, 日志存储空间-200GB 以上	确保充分的空间

**Note.** 用于 InterMax 数据收集的 AP 服务器及用于 Data Storage 的服务器为 Java 守护进程形态的程序，可在大部分安装有 Java 的 OS 进行操作。硬盘容量根据系统规模及收集的数据量及范围也可扩大/缩小。

### 3. 2. Windows 环境

#### 3. 2. 1. 事前准备事项

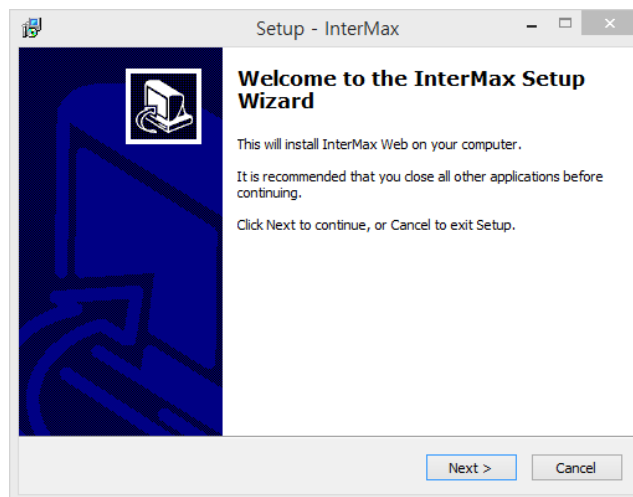
Java (JDK 1.8 以上版本)

Java 与 Data Gatherer 和 Platfrom\_JS 安装至同一个服务器。  
在系统环境变量中设置 JAVA\_HOME。

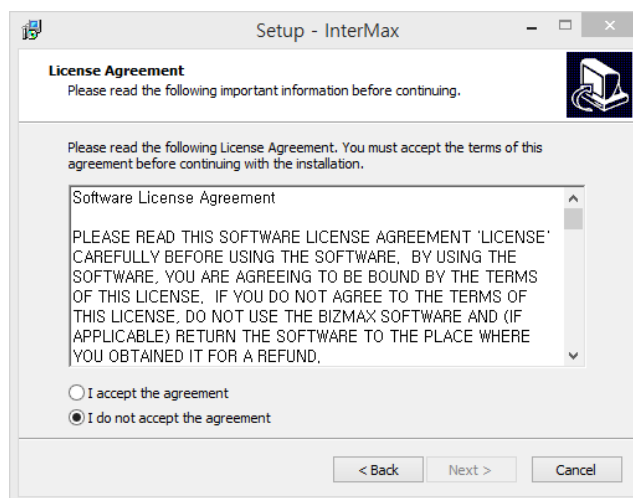
### 3.2.2. 安装步骤 ( 自动-Installer )

通过整合 Installer 安装的方法如下。

1. 执行 Intermax 整合安装程序(InterMax\_Installer\_版本名.exe)。
2. 执行安装魔法师后点击Next 按钮。

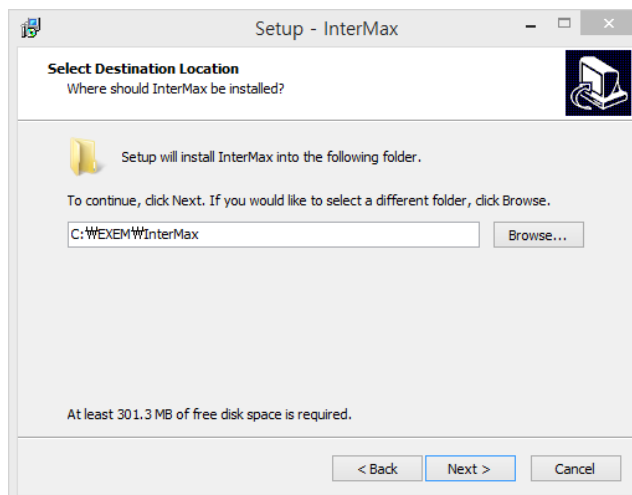


3. 选择同意许可证规定后点击Next 按钮。

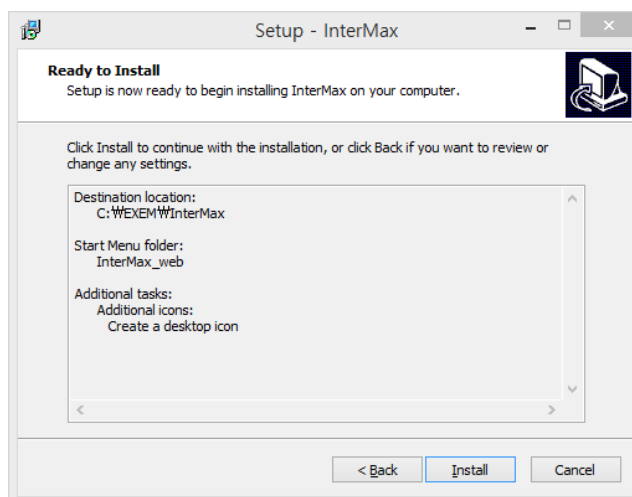


4. 指定 InterMax 的安装位置。

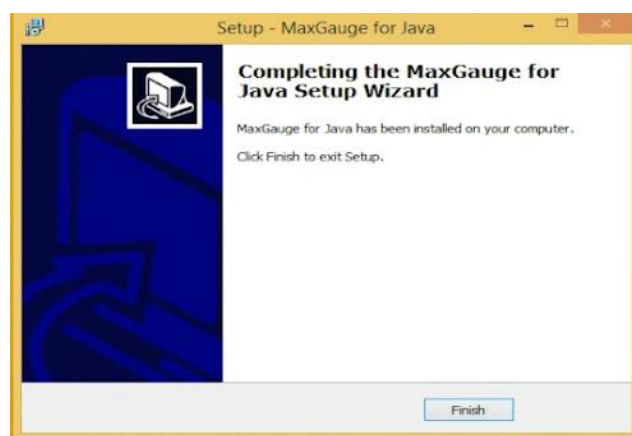




5. 点击 **Install** 按钮，即开始安装。安装大约需要 2 分钟。



6. 安装完成后点击[Finish]按钮，结束魔法师。



---

**Note.** Platform、JS、Data Gather、PostgreSQL 在安装后通过 Local 服务自动注册。

---

### 3. 2. 3. 安装步骤（手动）

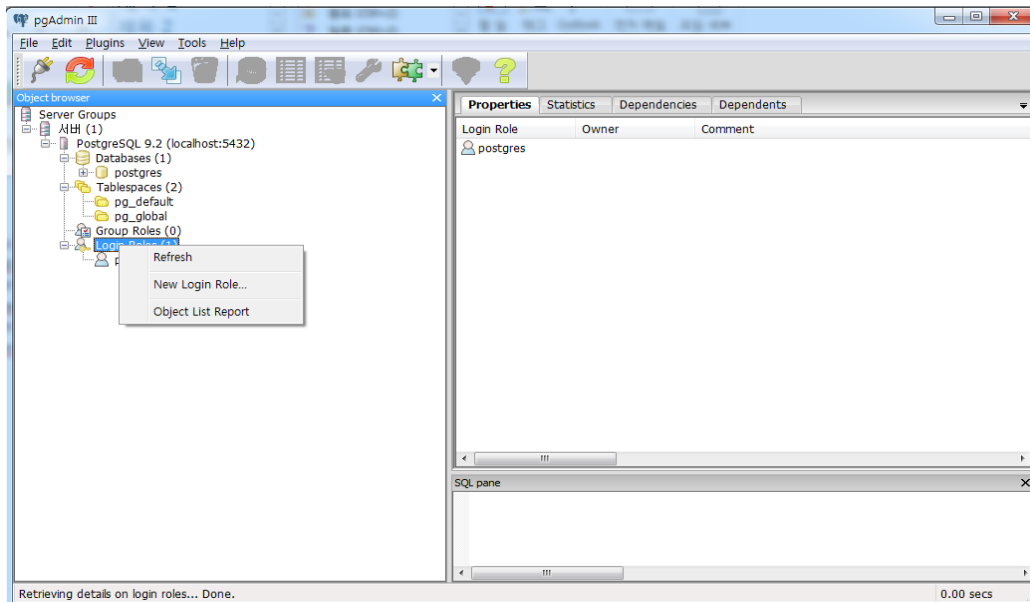
## PostgreSQL 手动安装

本 Install Guide 省略 PostgreSQL Database 安装相关的内容。相应 Database 安装相关的详细说明请参考 PostgreSQL 的官方 Install Guide。

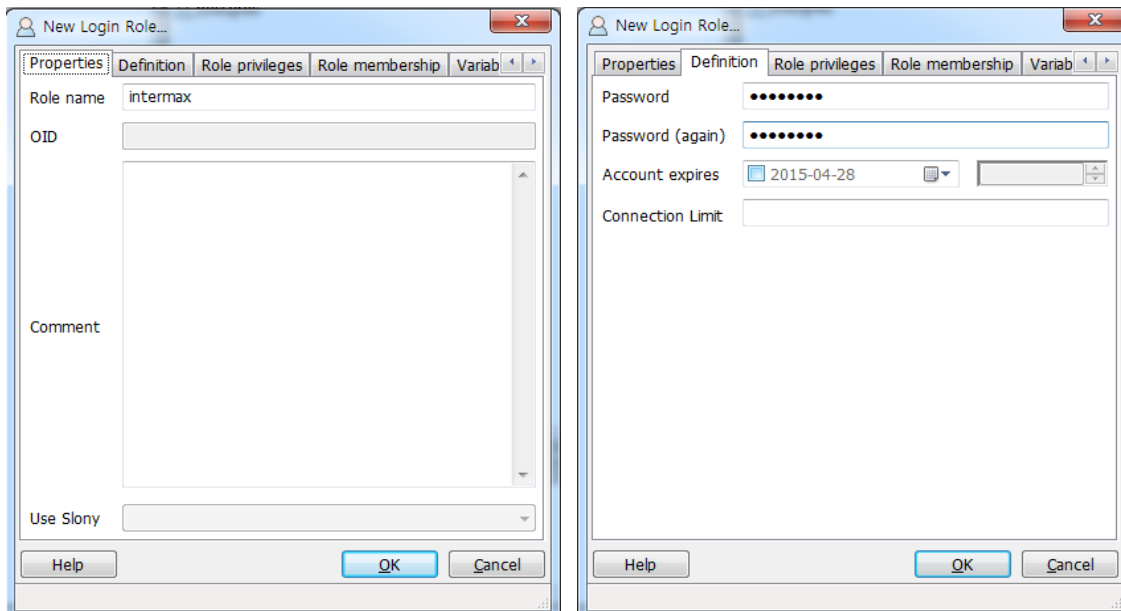
### Repository User 的生成及 Database 的设置

1. 为在 PostgreSQL 生成 Repository User 及 Database, 执行 pgAdmin3。

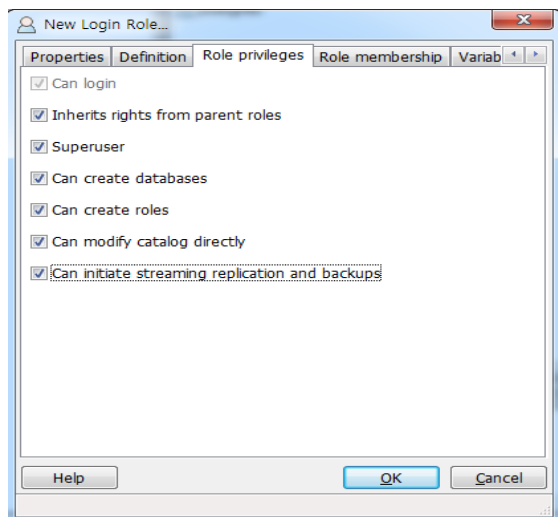
在 Object browser 中右击 Login Role 后点击 New Login Role。



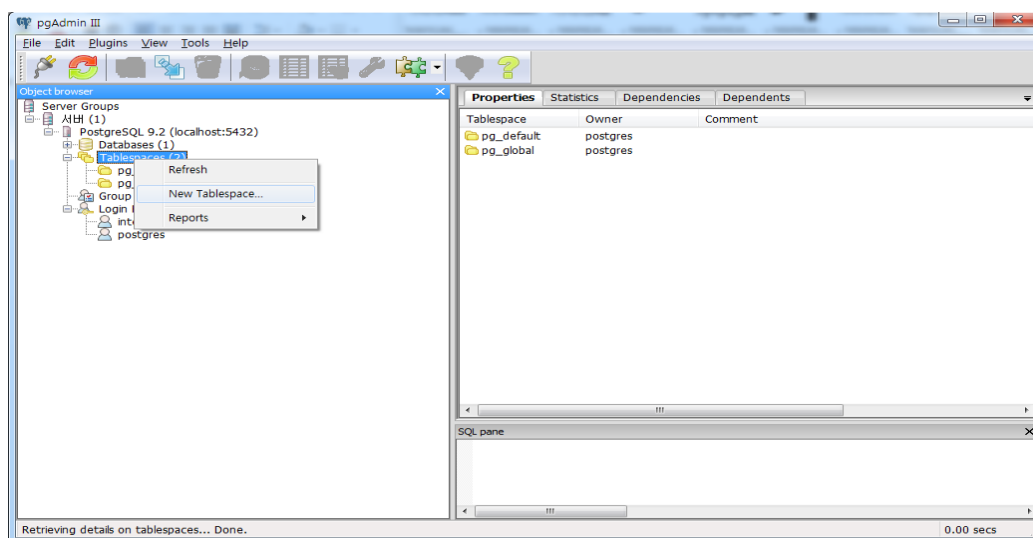
2. 写入 intermax User 信息。在 Properties Tab 的 Role name 中输入 intermax。在 Definition Tab 的 Password 中写入适当的 Password。



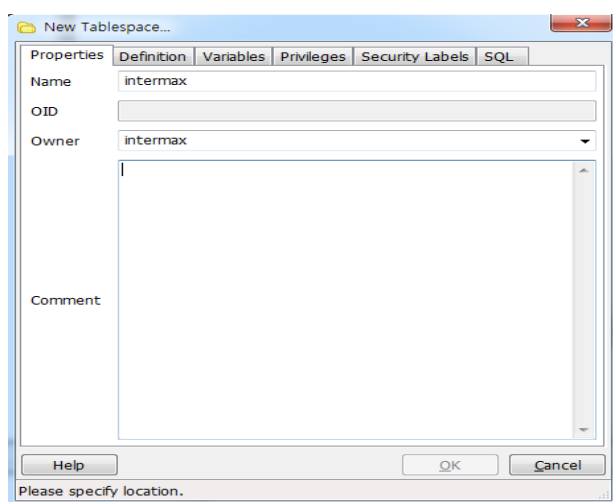
3. 勾选 Role privileges 中的所有权限后, 点击 OK。



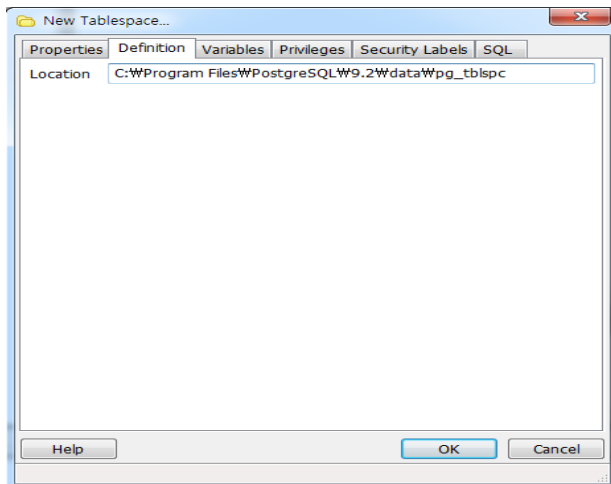
4. 为生成 Tablespace，在 Object browser 中右击 Tablespace 后点击 New Tablespace。



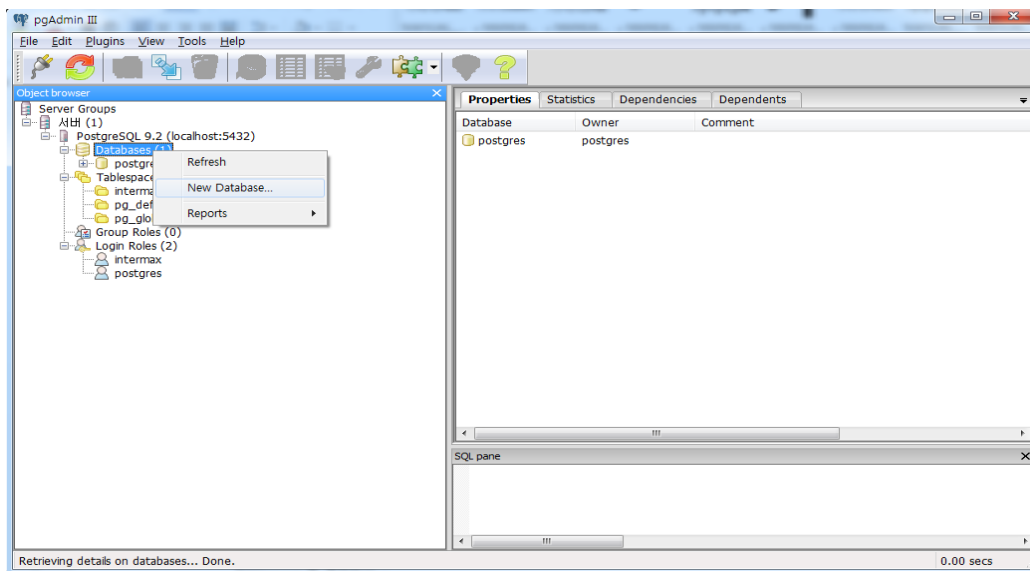
5. 在 Properties Tab 的 Name 中以 Tablespace 名称写入 intermax。Owner 在 Drop-Down List 中选择 intermax。



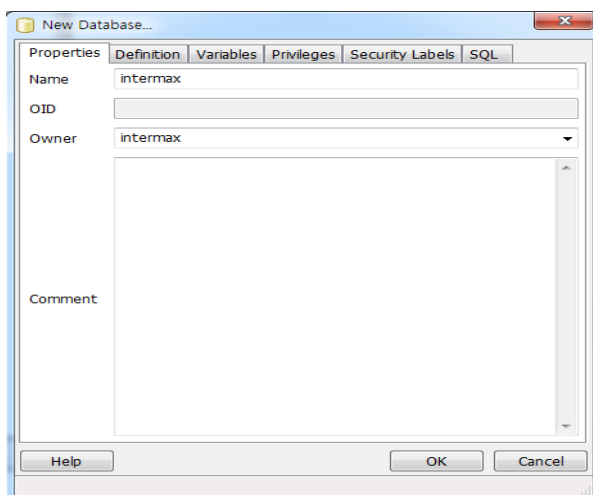
6. 在 Definition Tab 的 Location 中指定 Tablespace 位置后点击 OK。



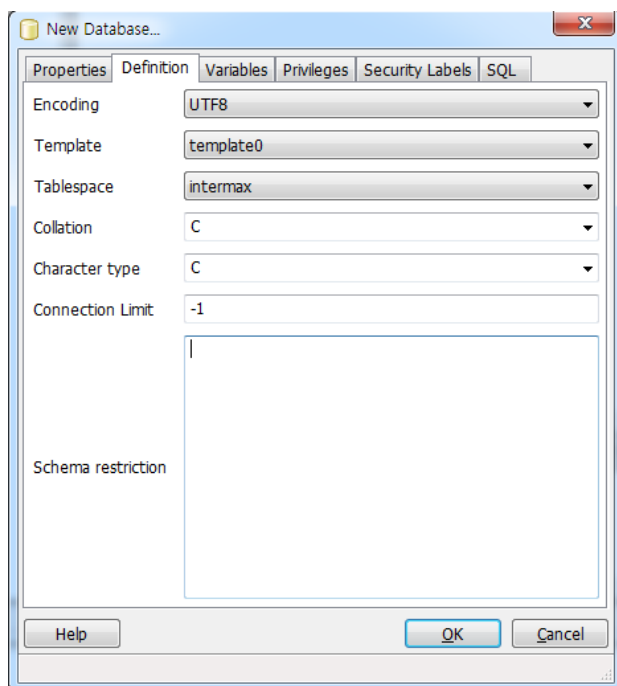
7. 为生成 intermax Database 在 Object browser 中右击 Databases 后点击 New Database。



8. 在 Properties Tab 的 Name 中以 Database 名称输入 intermax。Owner 选择 intermax。



9. Definition 中的 Encoding 选择 UTF8, Template 则选择 template0。Tablespace 选择 intermax。Collation 及 Character type 分别选择 C, 并点击 OK。



### Repository Parameter Settings

将已安装的PostgreSQL Database 的Parameter 设置如下。

文件位置: 如) D:\Program Files\PostgreSQL\9.4\data\postgresql.conf

Parameter 名称	建议安装(Memory 16GB)
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 手动安装

本 Install Guide 部分是 Oracle Database 安装相关内容，相应 Database 安装相关的详细内容请参考 Oracle 的官方 Install Guide。

### Repository User 的生成及 Database 的设置

为在 Oracle 生成 Repository User 及 Database，执行 SQL\*Plus。

#### 1. 生成 Table Space

生成共 Intermax 中使用的 Table Space。

##### 执行示例

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

#### 2. 生成用户

生成 Intermax User 并赋予权限。

##### 执行示例

```
# By sys or dba User
SQL> create user [user_name] identified by [password] default tablespace [tablespace_name] temporary ta
blespace 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 FROM intermax;
SQL>GRANT SELECT ON DBA_TAB_PARTITIONS TO intermax;
```

## 手动安装Data Gatherer

1. 解压 InterMax\_DG\_YYMMDD.tar 文件。

이름	수정한 날짜	유형	크기
DGServer_M	2016-07-07 오전...	파일 폴더	
DGServer_S1	2016-08-23 오후...	파일 폴더	

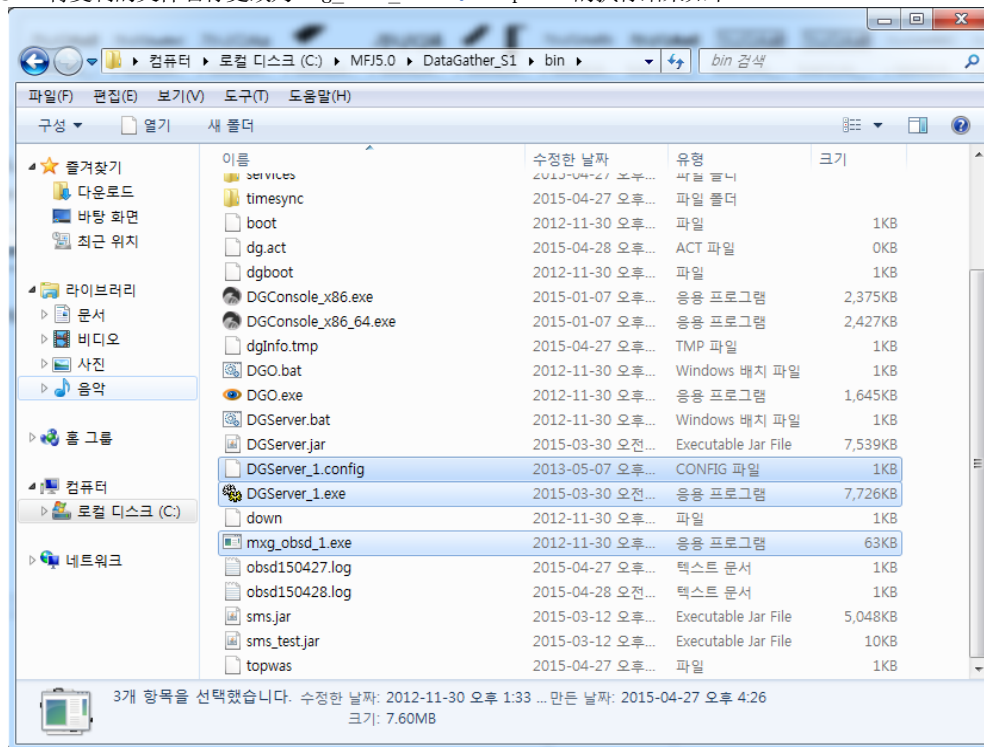
2. 首先安装Slave Data Gatherer. 在DataGather\_S1\bin\services 中将DGServer\_x86\_64\_1314.exe 和 DGServer\_x86\_64\_1314.config 文件复制到DataGather\_S1\bin. (32bit Windows 时为DGServer\_x86\_1314.exe和DGServer\_x86\_1314.config)

**Note.** 复制的文件是Data Gatherer 的服务文件。

3. 将复制的两个文件名称更改为 DGServer\_1.exe 和 DGServer\_1.config。
4. 将DataGather\_S1\bin\mxg\_obsd\win64\mxg\_obsd\_x64.exe 复制到DataGather\_S1\bin 中。(32bit Windows 时DataGather\_S1\bin\mxg\_obsd\win32\mxg\_obsd.exe)

**Note.** 复制的文件是Data Gatherer 的Observer 可执行文件。

5. 将复制的文件名称更改为 mxg\_obsd\_1.exe。Step 1~4 的执行结果如下。



6. 为设置Slave Data Gatherer, 编辑DataGather\_S1\conf\DGServer.xml. 设置事项如下表。

```

<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 名称	说明
master	设置 Master 与否 ● Slave Data Gatherer 时 false
dg_id	设置 ID ● Slave Data Gatherer 时为 1 以上
dg_port	Slave Data Gatherer 的通信 Port ● Slave 的 Port 建议 1314
db_type	设置 Repository 的数据库类型 ● PostgreSQL 时 postgres
conn_ip	设置供连接 JDBC 的 Repository 数据库 IP
conn_port	设置供链接 JDBC 的 Repository 数据库 Listener Port ● PostgreSQL 的默认值为 5432
sid	设置 Repository 数据库名称
user	Repository 数据库 User
password	Repository 数据库 User 密码

7. 在管理员权限命令窗口执行如下命令并将 Slave Data Gatherer 注册为服务。

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

**Note.** 服务名称可指定为 DGServer\_1 以外的其他名称。本指南中假设将服务名称指定为 DGServer\_1。

8. 为设置 Observer，编辑 DataGather\_S1\conf\DG\common.conf 文件的设置。设置项目如下表。

```

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 名称	说明
--------------	----



obs1_cmd	Slave Data Gatherer 的服务名称 ● e. g. )DGServer_1
obs1_keyword	Slave Data Gatherer 的部分进程名称 ● e. g. )DGServer
obs1_keyword2	Slave Data Gatherer 的部分进程名称 ● e. g. )DGServer_1

9. 在管理员权限命令窗口执行如下命令并将 Slave Data Gatherer 的 Observer 注册为服务。

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

**Note.** 服务名称可指定为 DGServer\_obsd\_1 以外的其他名称。本指南中以指定为 DGServer\_obsd\_1 为例。

10. 接下来安装 **Master Data Gatherer**。在 DataGather\_M\bin\services 中将 DGServer\_x86\_64\_1313.exe 和 DGServer\_x86\_64\_1313.config 文件复制到 DataGather\_M\bin。(32bit Windows 时 DGServer\_x86\_1313.exe 及 DGServer\_x86\_1313.config)
11. 将复制的两个文件名称更改为 DGServer\_0.exe 和 DGServer\_0.config。
12. 将 DataGather\_M\bin\mxg\_obsd\win64\mxg\_obsd\_x64.exe 复制到 DataGather\_M\bin 中。(32bit Windows 时 DataGather\_M\bin\mxg\_obsd\win32\mxg\_obsd.exe)
13. 将复制的文件名称更改为 mxg\_obsd\_0.exe。Step 10~12 的执行结果如下。
14. 为设置 Master Data Gatherer, 编辑 DataGather\_M\conf\DGServer.xml。设置事项如下表。

```
<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_init_size>75</conn_init_size>
  <conn_max_size>10</conn_max_size>
  <repository>true</repository>
```

Parameter 名称	说明
master	设置是否 Master ● Master Data Gatherer 时为 true
dg_id	设置 ID ● Master Data Gatherer 时为 0
dg_port	Master Data Gatherer 的通信 Port ● Master 的 Port 建议 1313
dg_list	用“,”区分输入所属于该 Master Data Gatherer 的 Slave Data Gatherer 信息(IP:Port)。 ● e. g. )127.0.0.1:1314, 127.0.0.1:1315, ...
db_type	设置 Repository 的数据库类型

	<ul style="list-style-type: none"> <li>● PostgreSQL 时 postgres</li> </ul>
conn_ip	设置供连接 JDBC 的 Repository 数据库 IP
conn_port	设置供连接 JDBC 的 Repository 数据库 Listener Port <ul style="list-style-type: none"> <li>● PostgreSQL 的默认值为 5432</li> </ul>
sid	设置 Repository 数据库名称
user	Repository 数据库 User
password	Repository 数据库 User 密码

15. 在管理员权限命令窗口执行如下命令并将 Master Data Gatherer 注册为服务。

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

**Note.** 服务名称可指定为 DGServer\_0 以外的其他名称。本指南中以将服务名称指定为 DGServer\_0 为例。

16. 为设置 Observer，编辑 DataGather\_M\conf\DG\common.conf 文件的设置。设置项目如下表。

```
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 名称	说明
obs1_cmd	Master Data Gatherer 的服务名称 <ul style="list-style-type: none"> <li>● e. g.) DGServer_0</li> </ul>
obs1_keyword	Master Data Gatherer 的部分进程名称 <ul style="list-style-type: none"> <li>● e. g.) DGServer</li> </ul>
obs1_keyword2	Master Data Gatherer 的部分进程名称 <ul style="list-style-type: none"> <li>● e. g.) DGServer_0</li> </ul>

17. 在管理员权限命令窗口执行如下命令并将 Master DataGatherer 的 Observer 注册为服务。

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

**Note.** 服务名称可指定为 DGServer\_obsd\_0 以外的其他名称。本指南将以将服务名称指定为 DGServer\_obsd\_0 为例。

18. Slave 及 Master 设置完成后，需在 Repository 中构成用于 InterMaxRepository。为完成此构成需执行以下命令。

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

19. 执行如下则选择 1.intall Respository。在 DGServer.xml 文件中设置的 Database 内构成 Repository。

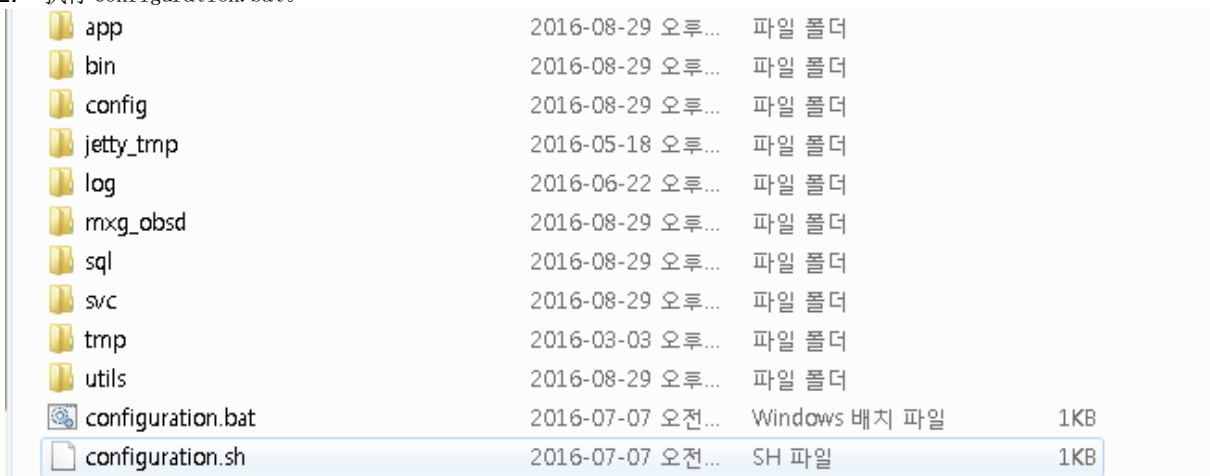
```
InterMax DataGather TFT (Build 160826.01)
===== DataGather Install Menu =====
1. Install Repository
2. Remove Repository
9. Get Repository Script
0. Exit
SELECT>
```

- 20. Repository 构成完成，则输入0。
- 21. 进行到此步骤，Data Gatherer 的基本构成已经完成。

### Platform. JS 手动安装

在 Windows 中安装 Platform. JS 的方法如下。

1. 解压 InterMax\_WEB\_YYMDD.zip。
2. 执行 Configuration.bat。



### 执行示例

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

Select Number : 1

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

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 : 10.10.202.98

Step 5.Database Port [ Default : 5432 ] ( BACK : 0 )

Input Text : 5432

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 : 8899

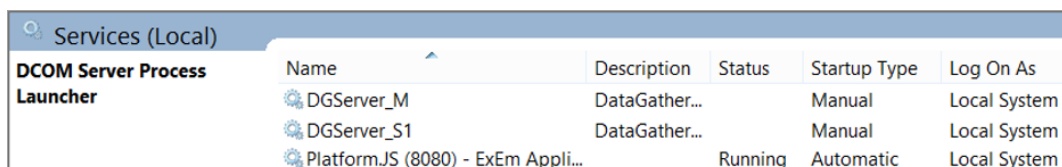
Do you want to save ?1.Save 2.Cancel [ Default (1)Save ]

Select Number : 1

3. 完成环境构成后，可执行文件将被添加到同一个文件夹中。

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

4. 利用管理员权限执行 service\_install.bat 并添加至服务。



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	
PlatformJS (8080) - ExEm Appli...		Running	Automatic	Local System	

### 3.2.4. 查看启动及连接

#### InterMax Local Services

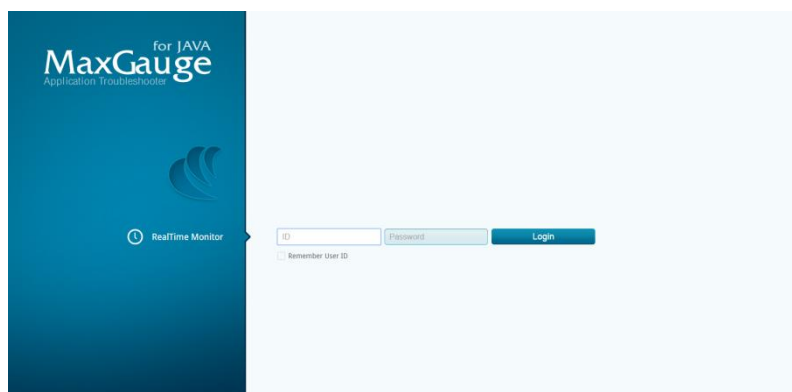
Platform.JS, Data Gatherer、Postgresql/Oracle Database 在 Windows Services (Local) 中分别执行各自的服务并分自动/手动启动。

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	

## 查看InterMax连接

为使用 InterMax，需要 Chrome 浏览器(Chrome 浏览器的安装方法省略 - 建议使用最新版本)，通过相应浏览器按照如下方法进行。

1. 通过 Chrome 网络浏览器访问 <http://127.0.0.1:8080/intermax/Config>。

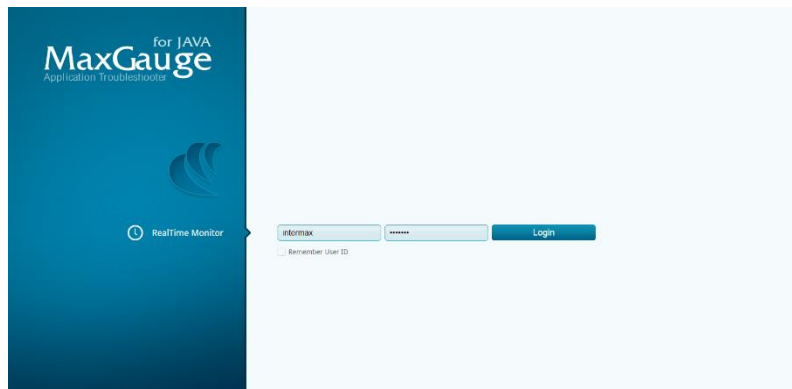



---

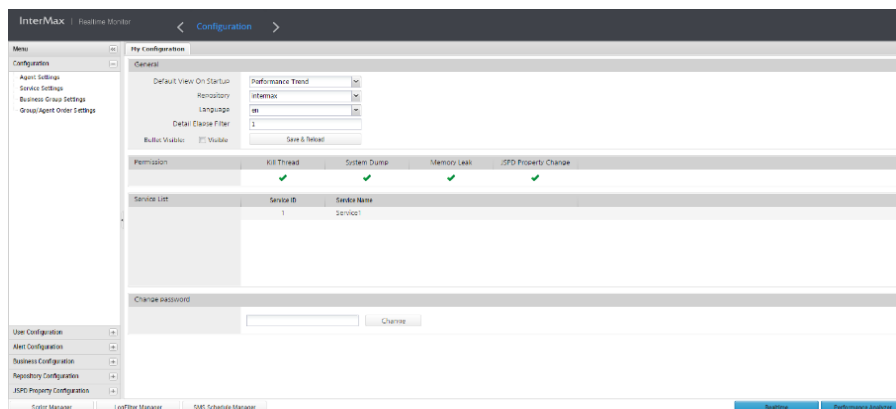
**Note.** IP 及 Port 分别输入安装有 Platform.JS 的 Host IP 及服务 Port。

---

2. 用所提供的默认账号访问。(ID:intermax / PW:manager)

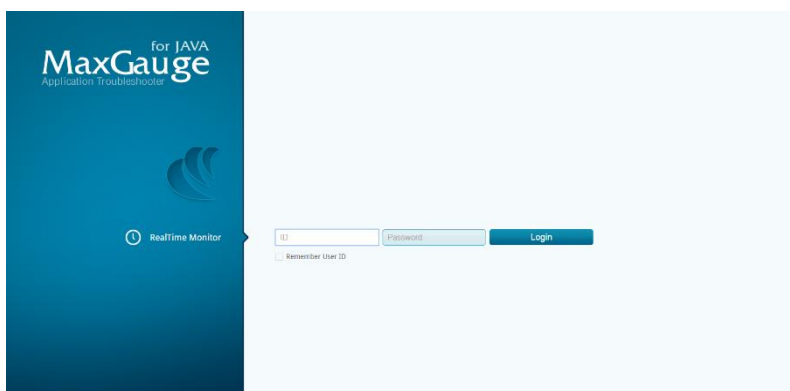


3. 设置 Configuration。必须构成至少一个以上的 **Service Group** 并赋予用户 **Service Privileges**。

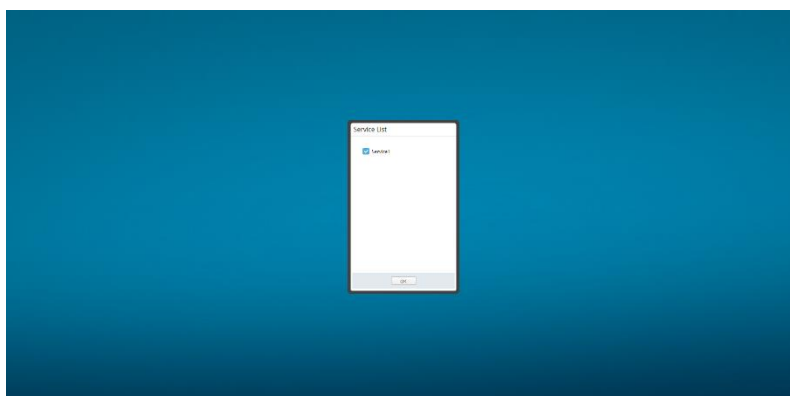


**Note.** InterMax Configuration 的具体安装方法请参考[“InterMax Configuration Guide”](#)。

- 通过 Chrome 网络浏览器访问 <http://127.0.0.1:8080/intermax/RTM>，输入 ID/Password 登录。(ID: intermax / PW: manager)



- 出现 Configuration 过程中设置的 **Service Group** 目录后，选择需进行监控的 **Service Group** 并点击 OK，即 Load 将实时监控的界面 view。



### 3.2.5. 用户定义选项

#### 添加 Slave Gatherer Process

一个 **Slave Data Gatherer** 与过多的 **InterMax Agent Set** 通信，将导致发生负荷。这种情况下，需添加 **Slave DG**。添加 **Slave DG** 的方法如下。

- 在 InterMax 安装的路径中复制 **DataGather\_S1** 文件夹并生成 **DataGather\_S#** 文件夹。

DataGather_M	2015-04-27 오후...	파일 폴더	
DataGather_S1	2015-04-27 오후...	파일 폴더	
DataGather_S2	2015-04-27 오후...	파일 폴더	
DGServer_2910_150330.1.tar	2015-03-30 오전...	tar Archive	71,510KB

- 将 DataGather\_S#\bin\DGServer\_x86\_64\_1314.exe 为 DGServer\_x86\_64\_1314.config 文件名称更改为 DGServer\_x86\_64\_{other port#}。
- 编辑 DataGather\_S#\conf\DGServer.xml 文件，更改 dg\_id 及 dg\_Port。

```

<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.** 注意不要使其他 Slave Data Gatherer 与 dg\_id、dg\_port 重复。

- 编辑 DataGather\_M\conf\DGServer.xml 文件，在 Slave\_Gather\_List 中添加 IP 地址及 Port 号码。

```

<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>

```

- 在管理员权限命令窗口注册 Slave DG # 服务。

```
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.** Slave Process 的建议个数为(Slave 1): (JVM instance 数 < 50)。但每一个 Slave process 将被分配 1 G 内存，因此要充分考虑 Free Memory 后再进行添加。DG 的分配内存可通过编辑各 bin 文件夹中的 DGServer\_\*.config 文件进行更改。

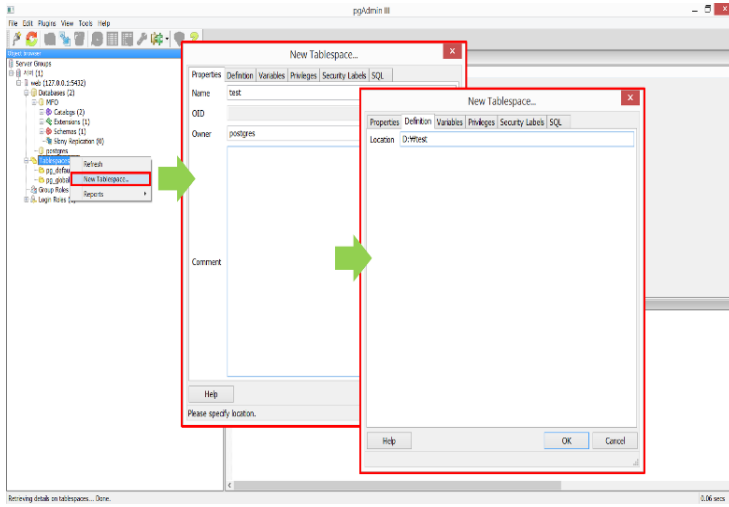
## 设置 PostgreSQL Tablespace

PostgreSQL Repository 中保存的数据量的增加将导致 Disk 容量不足。这个问题可通过在个别 Table 中另行生成 Tablespace 解决。Tablespace 的分配方法如下。

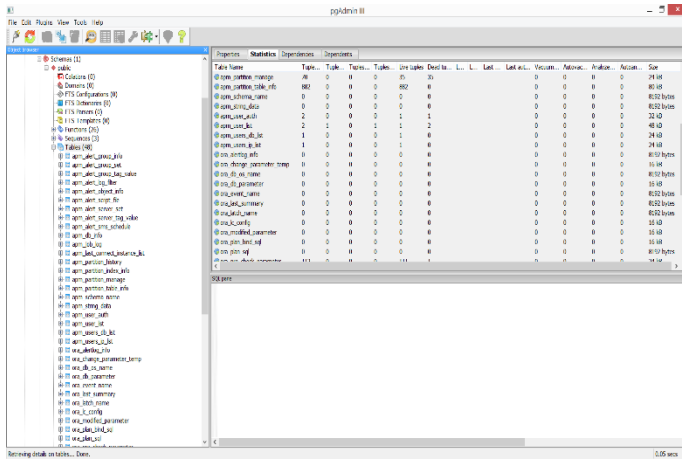
- 执行 pgAdmin3。({InterMax Home Directory}/Database/bin/pgAdmin3)

pg_test_fsnc.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
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

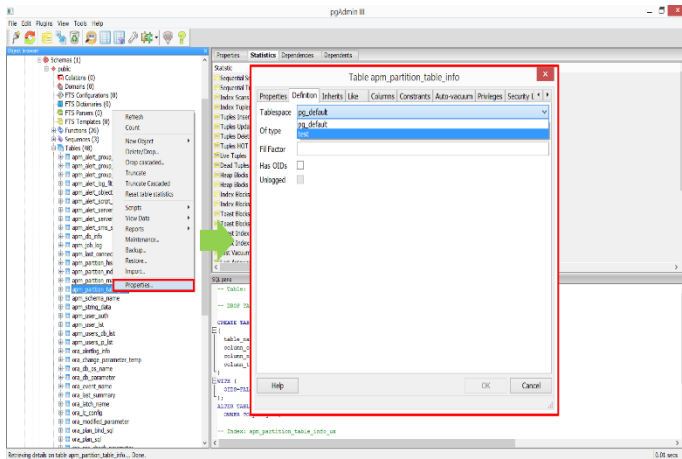
2. 生成新的 Table Space, 并输入名称/User/路径。



3. 查看大容量的 Table。



4. 分别指定 Table 生成的 Tablespace。



**Note.** 数据保存周期可在 InterMax Configuration 进行更改。详细内容请参考“[InterMax Configuration Guide](#)”



## 3.3. Unix/Linux 环境

### 3.3.1. 事前准备事项

Java (JDK 1.8 以上版本)  
Java 与 Data Gatherer 和 Platform\_JS 安装同至一个服务器。

### 3.3.2. 安装步骤 (手动)

通告 InterMax 的 Unix/Linux 用 Installer 的自动安装根据客户公司的系统环境 (CDE) 具有流动性, 因自动安装的可操作性差, 建议手动安装。

#### PostgreSQL 手动安装

Install Guide 部分内容为 PostgreSQL Database 安装相关的内容, 与 Windows 手动安装内容类似, 请参考。相应 Database 安装相关的详细说明请参考 PostgreSQL 的官方 Install Guide。

#### 1. Repository User 的生成及权限的设置

为生成 Repository user 及 Database, 执行 psql。  
(在已安装的文件夹/bin 中执行 ./psql)

执行示例

```
Psql postgres

Postgres=#
CREATE USER intermax PASSWORD 'intermax' ;
ALTER USER intermax WITH SUPERUSER;
ALTER USER intermax WITH CREATEROLE;
ALTER USER intermax WITH REPLICATION;
ALTER USER intermax WITH VALID UNTIL '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. 生成 Table Space

执行示例

```
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  | Encodiong | 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 手动安装

本 Install Guide 部分是 Oracle Database 安装相关内容，相应 Database 安装相关的详细内容请参考 Oracle 的官方 Install Guide。

### Repository User 的生成及 Database 的设置

为在 Oracle 生成 Repository User 及 Database，执行 SQL\*Plus。

#### 1. 生成 Table Space

生成供 Intermax 使用的 Table Space。

#### 执行示例

```

SQL>create tablespace [tablespace_name] datafile '位置' size[size]
SQL>extent management local

```

```
SQL>segment space management auto;
```

## 2. 生成用户

生成 Intermax User 并赋予权限。

### 执行示例

```
# By sys or dba User
SQL> create user [user_name] identified by [password] default tablespace [tablespace_name] temporary ta
blespace 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;
```

## 手动安装Data Gatherer

1. 解压 InterMax\_DG\_YYMMDD.tar。

### 执行示例

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

## 2. 编制 DGServer\_M/conf 文件夹的 DGServer.xml。

```

<?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 名称	说明
master	设置是否 Master <ul style="list-style-type: none"> <li>● Master Data Gatherer 时为 true</li> </ul>
dg_id	设置 ID <ul style="list-style-type: none"> <li>● Master Data Gatherer 时为 0</li> </ul>
dg_port	Master Data Gatherer 的通信 Port <ul style="list-style-type: none"> <li>● Master 的 Port 建议 1313</li> </ul>
dg_list	将所属于该 Master Data Gatherer 的 Slave Data Gatherer 的 IP:Port", "区分输入。 <ul style="list-style-type: none"> <li>● e. g.) 127.0.0.1:1314, 127.0.0.1:1315, ...</li> </ul>
db_type	设置 Repository 的数据库类型 <ul style="list-style-type: none"> <li>● PostgreSQL 时 postgres</li> <li>● Oracle 时 oracle</li> </ul>
conn_ip	设置用于连接 JDBC 的 Repository 数据库 IP
conn_port	设置用于连接 JDBC 的 Repository 数据库 Listener Port <ul style="list-style-type: none"> <li>● PostgreSQL 的默认值为 5432</li> <li>● Oracle 的默认值为 1521</li> </ul>
sid	设置 Repository 数据库名称
user	Repository 数据库 User
password	Repository 数据库 User 密码

3. 在 DGServer\_S1/conf 文件夹中编制 DGServer.xml。

```

<?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>
</DataGather>

```

Parameter 名称	说明
master	设置 Master 与否 ● Slave Data Gatherer 时 false
dg_id	设置 ID ● Slave Data Gatherer 时为 1 以上
dg_port	Slave Data Gatherer 的通信 Port ● Slave 的 Port 建议 1314
db_type	设置 Repository 的数据库类型 ● PostgreSQL 时 postgres
conn_ip	设置用于连接 JDBC 的 Repository 数据库 IP
conn_port	设置用于连接 JDBC 的 Repository 数据库 Listener Port ● PostgreSQL 的默认值为 5432
sid	设置 Repository 数据库名称
user	Repository 数据库 User
password	Repository 数据库 User 密码

4. 完成 Slave 及 Master 的安装后需在 Repository 构成用于 InterMax 的 Repository。为完成此构成需执行以下命令

```

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

```

5. 出现如下 Install Menu，并开始 Database 的 Repository 构成工作。

- 1) 选择 install Respository。

```

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

```

- 2) 初次安装时输入“N”，重新安装时则为保留现有的环境设置信息，应输入“Y”。

Input Number of Databsse:2 是代表客户公司的监控目标 DB Instance 数的值，通过 imxdbm 模块输入需监控的目标数。(未安装 imxdbm 模块时，无需输入相应值)。

```

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

```

相应值后续将被用为通过 imxdbm 收集到的性能数据保存到 partition table 时生成 sub-partition 所需的基准值。(未输入时只生成默认 partition table)

- 3) Enter Table Tablespace for InterMax : intermax\_ts (输入生成的 tablespace 名称)

```

Enter Table Tablespace for InterMax : intermax_ts
Table Tablespace for InterMax : intermax_ts
Enter Index Tablespace for InterMax [intermax_ts] : intermax_ts

```

需分离 Enter Index Tablespace for InterMax [intermax\_ts] : intermax\_ts (index tablespace 时输入相应的 tablespace 名称。若共用默认 tablespace，输入同样的名称即可。)

```

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) 输入完成即完成 Table 生产及构成工作，将显示如上信息。

6. Repository 构成完成，则输入 0。

7. 在 \$INTERMAX\_HOME/Bin 文件夹中执行 boot，即启动 Data Gather。

## Platform. JS 手动安装

Platform. JS 安装方法如下。

1. 解压 InterMax\_WEB\_YYMMDD.zip。
2. 执行 Configuration.sh。

执行示例

```
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 : 5432 ] ( BACK : 0 )
Input Text : 5432

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 : 8899

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

3. 完成环境构成后，可执行文件将被添加到同一个文件夹中。
4. 执行Platformjs.start.sh则显示PlatformJS的启动选项。
  - 选择1，则输出操作Level的日志，同时通过背景服务器启动相应服务（默认选择为1）
  - 选择2，则输出console模式的调节Level日志，同时启动服务。

```

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. 启动方法

#### 启动InterMax PlatformJS

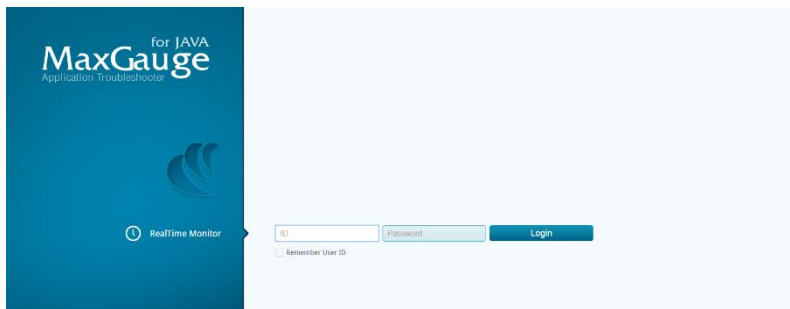
运行PlatformJS. start. sh中的一个选择即启动。

选项	说明
Release Mode	● 用 Background 模式执行 PlatformJS
Debug Mode	● 用 Debug 模式执行 PlatformJS

#### 设置及查看InterMax的方法

为使用 InterMax，需要 Chrome 浏览器(Chrome 浏览器的安装方法省略 - 建议使用最新版本)，通过相应浏览器按照如下方法进行。

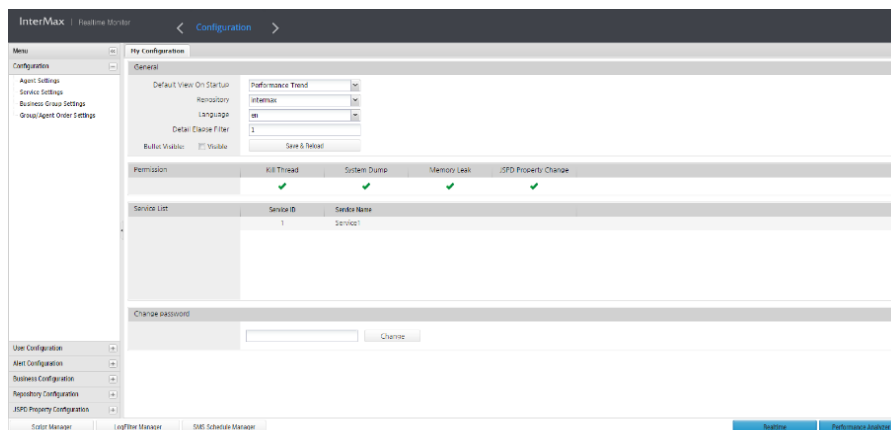
1. 通过 Chrome 网络浏览器访问 <http://127.0.0.1:8080/intermax/Config>。



**Note.** IP 及 Port 分别输入安装有 Platform. JS 的 Host IP 级服务 Port。

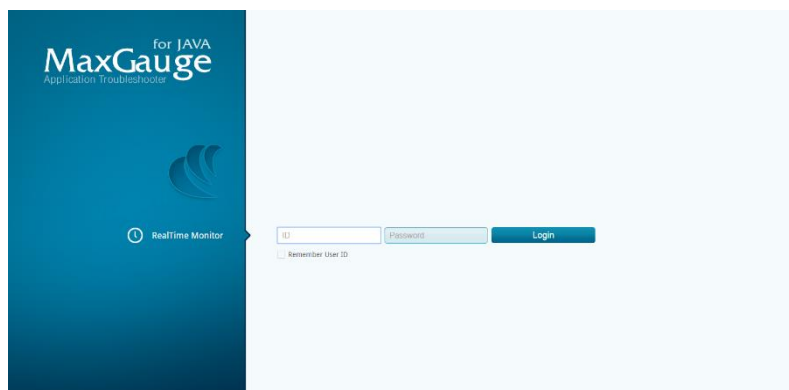
2. 用所提供的默认账号访问。(ID:intermax / PW:manager)
3. 设置 Configuration。必须构成至少一个以上的 **Service Group** 并赋予用户 **Service Privileges**。



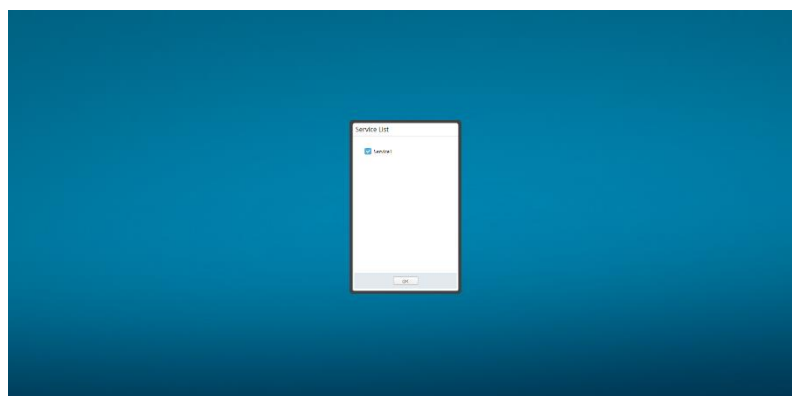


**Note.** InterMax Configuration 的具体安装方法请参考“[InterMax Configuration Guide](#)”。

4. 通过 Chrome 网络浏览器访问 <http://127.0.0.1:8080/intermax/RTM>，输入 ID/Password 登录。(ID:intermax / PW:manager)



5. 出现 Configuration 过程中设置的 **Service Group** 目录后，选择需进行监控的 **Service Group** 点击 OK，即 Load 监控 view。



## 4. Appendix

### 4.1 各WAS芯片供应商的InterMax Option设置

如在第2章所说，InterMax 选项根据 Jara Version 不同而存在差异。

#### Java Version 1.7 以上版本

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

#### Java Version 1.5 以上版本

```
-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.** 有关各 WAS 的 InterMax JSDP 选项的设置请参考接续章。

---

#### 4.1.1. JEUS InterMax Option 设置

采用`$(JEUS_HOME)/config/hostname`/JEUSMain.xml`为InterMax 选项。

插入到`<command-option></command-option>` 标签中。若已经采用了现有的选项，则插入到现有选项的后面。

```
?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.** 修改 Script 前需执行备份，以便发生问题时能够及时复原。

---

## 4.1.2. WebLogic InterMax Option 设置

{域名目录(以下: \$DOMAIN\_HOME)}/bin/StartWeblogic.sh 适用 InterMax 选项。  
输出 INTERMAX\_OPTION, 并将 INTERMAX\_OPTION 放入 JVM 执行脚本。

```

11111 *****
export INTERMAX_OPT="-Djpd.wesid=60000 -Djavaagent:/home/intermax/intermax/jpd/lib/jpd.jar"
11111 *****

# Start WebLogic

echo "starting weblogic with Java version."

${JAVA_HOME}/bin/java ${JAVA_VM} -wsl

if [ "${WLS_REDIRECT_LOG}" = "" ]; then
    echo "Starting WLS with link."
    echo "${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>&1
fi

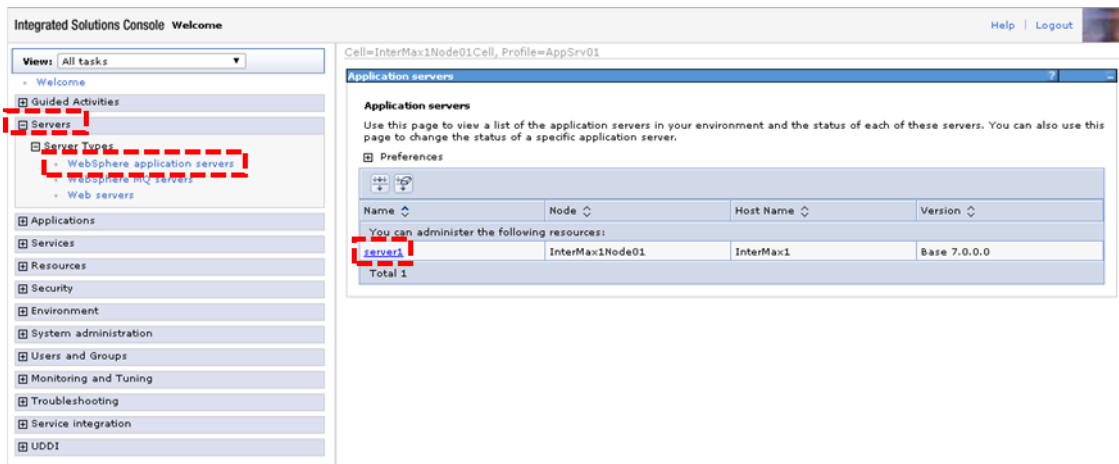
```

**Note1.** 修改 Scripts 前需执行备份, 以便发生问题时能够及时复原。

**Note2.** 根据版本或服务器构成的不同, InterMax Option 的位置也有所不同。

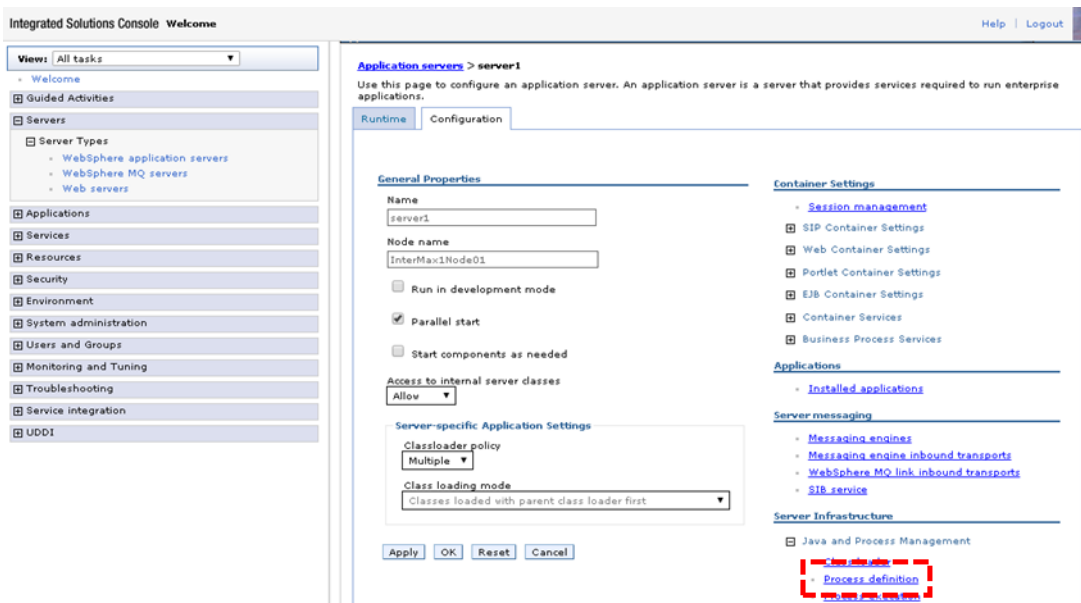
### 4.1.3. WebSphere InterMax Option 设置

1. 访问 WebSphere 网络控制台。



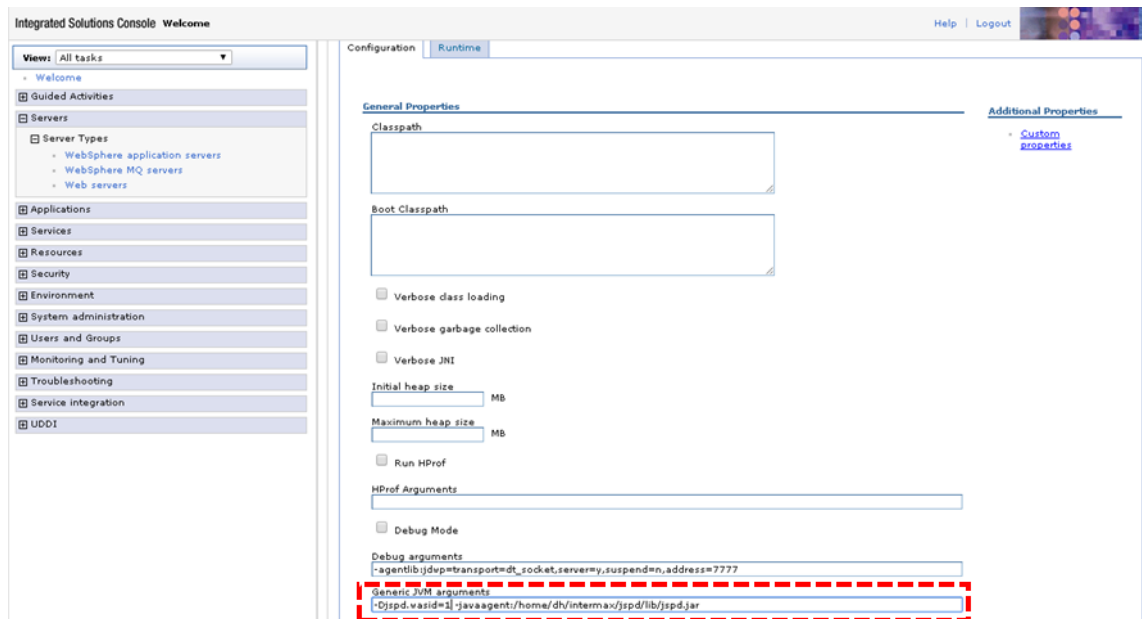
2. 点击 Server -> Server Types -> WebSphere application server -> server1

3. 点击 Process Definition



4. 点击 Java Virtual Machine

5. 在 Generic JVM arguments 部分适用 InterMax 选项。



Note. 根据版本或构成结构的不同，InterMax Option 的位置也有所不同。

#### 4.1.4. Tomcat InterMax Option 设置

\$CATALINA\_HOME/bin/catalina.sh 中适用 InterMax 选项。

JAVA\_OPTS = "\$JAVA\_OPTS:\$INTERMAX\_OPTION"

```
##### multi instance wasid
JAVA_OPTS="-Djspd.wasid=3 -javaagent:/home1/jt/jy/multi/jsp_agent/jsp/lib/jsp.jar"

# OS specific support. $var _must_ be set to either true or false.
cygwin=false
darwin=false
os400=false
case "`uname`" in
```

Note1. 修改 Script 前需执行备份，以便发生问题时能够及时复原。

Note2. 根据版本或构成结构的不同，InterMax Option 的位置也有所不同。

#### 4.1.5. JBoss InterMax Option 设置

根据运行方式不同，InterMax Option 的位置也不同。

Standalone 方式在 \$JBOSS\_HOME/bin/standalone.sh 中适用 InterMax Option。

```
JBOSS_BASE_DIR= cygpath --path --windows "$JBOSS_BASE_DIR"
JBOSS_LOG_DIR= cygpath --path --windows "$JBOSS_LOG_DIR"
JBOSS_CONFIG_DIR= cygpath --path --windows "$JBOSS_CONFIG_DIR"
fi

JAVA_OPTS="$JAVA_OPTS -Djspd.home=/home/park/intermax/jsp -Djspd.wasid=1 -javaagent:/home/park/intermax/jsp/lib/jsp.jar"
# Display our environment
echo "-----"
echo ""
echo " JBoss Bootstrap Environment"
```

Multiple Instances 方式在 \$JBOSS\_HOME/domain/configuration/host.xml 中适用 InterMax Option。

Note. JBoss7 的情况下，要以 OSGI Class loader 结构或在 standalone.conf domain.conf 中按照如下步骤追加适用。(Djboss.modules.system.pkgs=org.jboss.byteman;com.exem)

```

# Specify options to pass to the Java VM.
#
if [ "$JAVA_OPTS" = "x" ]; then
  JAVA_OPTS="-Xms64m -Xmx512m -XX:MaxPermSize=256m -Djava.net.preferIPv4Stack=true -Dorg.jboss.resolver.warning=true -Dsun.rmi.dgc.client.gcInterval=3600000 -Dsun.rmi.dgc.server.gcInterval=3600000"
  JAVA_OPTS="$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.default.config=host.xml"
  JAVA_OPTS="$JAVA_OPTS -Djspd.wasid=123 -noverify -javaagent:/home/intermax/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. 修改 Script 前需执行备份, 以便发生问题时能够及时复原。

Note2. 根据版本或构成结构的不同, InterMax Option 的位置也有所不同。

#### 4.1.6. Resin InterMax Option 设置

根据 Resin 版本不同 InterMax Option 所适用的位置也不同。

Resin 2. x/3. 0. x 的情况下, 在\$RESIN\_HOME/bin/httpd.sh 的 JAVA\_OPTIONS 相关标签中插入 InterMax 选项, 前面要加-J。

例如写入-Djspd.wasid={WAS ID} -J-javaagent:\$JSPD\_HOME/lib/jspd.jar。

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

例如

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

<jvm-arg>-javaagent:\$JSPD\_HOME/lib\spd.jar</jvm-arg>

。

Resin 4. x 在\$RESIN\_HOME/conf/resin.xml 的 jvm-arg 标签中适用 InterMax Option。

```

<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. 以便发生问题时能够及时复原, 修改 Script 前需执行备份。

Note2. 根据版本或构成结构的不同, InterMax Option 的位置也有所不同。

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

根据运行与启动方式不同, InterMax Option 的位置也不同。

Standalone 方式为在启动 oc4j.jar 启动脚本上采用 InterMax Option。

```

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
    fi
}

```

Multiple Instances 方式在\$OC4J\_HOME/opmn/conf/opmn.xml <data id="java-options" value="" > Tag 中适用 InterMax Option。

```

<category id="start-parameters">
  <data id="java-options"
    value="-Xrs -server -XX:MaxPermSize=128M
          -ms512M -mx1024M -XX:AppendRatio=3
          -Djava.awt.headless=true
          -Dhttp.wabdir.enable=false
          -Djspd.wasid=1
          -javaagent:/home/park/intermax/jspd/lib/jspd.jar"/>
</category>

```

Note1. 修改 Script 前需执行备份，以便发生问题时能够及时复原。

Note2. 根据版本或构成结构的不同，InterMax Option 的位置也有所不同。

#### 4.1.8. GlassFish InterMax Option 设置

在 GlassFish 管理 console 或 GlassFish\_HOME/domains/domain1/config/domain.xml 中适用 InterMax Option。

```

<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>
</testcase-config>

```

```

# 4. Netbeans profiler package 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.gxem.*

```

\*GlassFish 的情况下，要以 OSGI Class loader 结构或在 osgi.properties 中按照如下步骤追加适用。

Note1. 修改 Script 前需执行备份，以便发生问题时能够及时复原。

Note2. 根据版本或构成结构的不同，InterMax Option 的位置也有所不同。

