

应用服务器JVM参数配置

- JVM参数
- 配置JVM参数
 - Tomcat
 - Webshpere
 - Weblogic
 - Jboss

在Smartbi配置时，需要为Smartbi配置一定的JVM参数。 比如Smartbi在运行过程中，需要占用一定JVM内存，若需要的JVM内存等分配不够，则容易产生内存溢出（outofmemory）等问题。

JVM相关设置项的配置是通过应用服务器进行配置的。

JVM参数

为Smartbi配置的JVM参数有哪些呢？可以通过下表查看。一般只涉及到设置 -Xms、-Xmx、-XX: PermSize、-XX: MaxPermSize、 -XX: +HeapDumpOnOutOfMemoryError五个参数。各参数详情如下：

参数名称	建议值	描述
-Xms	512m以上	初始化“堆”大小
-Xmx	16384m及以上 当JVM运行过程中出现java.lang.OutOfMemoryError:Heap space时，应当增大此值，或者使用集群配置减轻并发压力	“堆”最大允许大小，指定Java程序中允许使用的最大内存数
-Xloggc	可以在跟踪内存溢出时打开	生成JVM垃圾回收日志，打开这个选项可以跟踪服务器是否存在内存泄漏。IBM JDK（例如WebSphere）生成的文件需要2.2.6中提及的IBM分析工具
-XX:PermSize	64m或以上	JVM为永久生成对象，包括类、方法定义等（Permanate generation）的内存区大小
-XX:MaxPermSize	128m或以上 当JVM运行过程中出现java.lang.OutOfMemoryError:PermGen space时应当增大此值	
-XX:+HeapDumpOnOutOfMemoryError	无需设置具体数值，但必须预留该参数	该参数必须添加 ，当发生内存溢出时，该参数用于存储堆信息，可以用来分析内存溢出的原因； 若部署时未进行添加，可参考方法“ 添加-XX:+HeapDumpOnOutOfMemoryError参数 ”
-Xss	一般不需要设置该参数，只有当递归层次过多时应该设置为128k或以上	为线程保留的堆栈大小
-Dfile.encoding	GBK或UTF-8	设置JVM运行时的默认字符集 在Unix环境下运行Smartbi时通常需要设置为GBK或UTF-8（操作系统必须预先安装该语言区域）才可以正常的显示中文
-Duser.timezone=Asia/Chongqing	一般不需要设置，但是当出现时间转换字符串后相差八小时等时，设置为Asia/Chongqing	设置JVM的默认时区 在某些服务器上时间转换为字符串输出（例如灵活分析中查询的数据）可能出现相差了八个小时，这是因为JVM无法正确的判断服务器的时区信息
-Djava.awt.headless=true	建议增加	启用headless模式

当报错信息是【outofmemory: permGen space】 时，需要增大-XX:MaxPermSize 的值

当报错信息是【outofmemory: java heap space】时，需要增大 -Xmx 的值

当报错信息是【JVM没有设置-XX:HeapDumpOnOutOfMemoryError，将导致系统宕机后无法跟踪解决】时，需要添加参数“-XX: +HeapDumpOnOutOfMemoryError”

当报错信息是【JVM（XMX）设置超出服务器内存：xxx，当前值为XXX。建议修改设置项的值】时，需要减少该设置项的值

当报错信息是【JVM没有配置headless参数】时，建议修改JVM参数配置，增加-Djava.awt.headless=true

配置JVM参数

不同引用服务器、不同操作系统、甚至不同的应用服务器版本，配置JVM参数的文件、方式都可能不同，具体可以通过下面的内容进行学习。

由于一般配置JVM参数只需要设置 `-Xms` 、`-Xmx` 、 和`-XX: MaxPermSize`三个参数，有时候还包含 `-XX:PermSize`，因此以下内容只对基本的这几个设置项进行设置

注意：此文档只是说明了如何去配置这几个参数，对于部署smartbi来说，在部署的时候对于JVM是有一些必要的参数要配置的。请参考：[在各种应用服务器上手工部署](#)

Tomcat

一、Windows系统

A. 如果Tomcat应用服务器是使用Smartbi exe安装包进行安装的，那么这个Tomcat是smartbi自带的，是经过改造的。设置JVM参数的步骤如下：

1. 关闭应用服务器
2. 进入Tomcat\bin目录，使用编辑方式打开startup.cmd（可以使用notepad++、UltraEdit等编辑工具）
3. 在startup.cmd中最下端找到以下内容

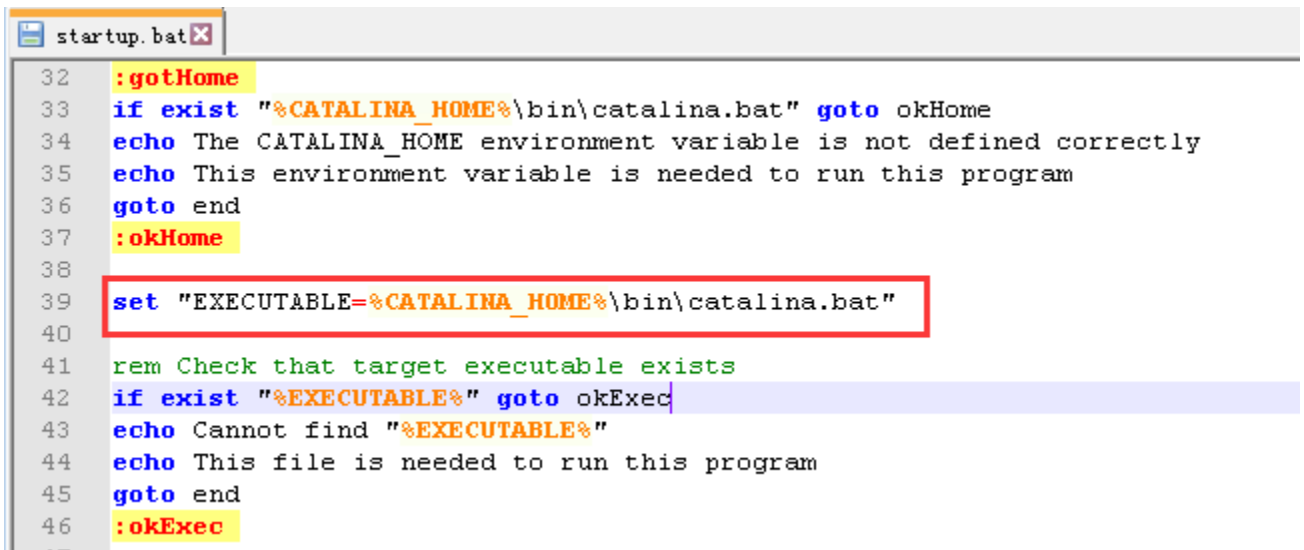
```
24 echo start Smartbi server
25 set JAVA_OPTS=-Xms128m -Xmx6102m
26 "%JAVA_HOME%\bin\java.exe" %JAVA_OPTS% -server -XX:MaxNewSize=256m -XX:MaxPermSize=256m
27 if "%errorlevel%"=="1" pause
28
```

修改其中的`-Xms`，`-Xmx`，`-XX:MaxPermSize`，比如将`Xms`设置为512m，`Xmx`修改为 10240m，`MaxPermSize`修改为512m。修改后如下：

```
24 echo start Smartbi server
25 set JAVA_OPTS=-Xms512m -Xmx10240m
26 "%JAVA_HOME%\bin\java.exe" %JAVA_OPTS% -server -XX:MaxNewSize=256m -XX:MaxPermSize=512m -Dmail.mime.splitlongparameters=f
27 if "%errorlevel%"=="1" pause
28
```

B. 如果Smartbi是使用war包部署的，也就是说Tomcat应用服务器是自行安装的，那么设置步骤如下：

1. 关闭应用服务器
2. 进入Tomcat\bin目录。在windows上tomcat是通过startup.bat进行启动的，启动时需要使用到tomcat的一些启动信息。可以用编辑方式打开startup.bat，找到set "EXECUTABLE" 设置项，看看该设置项是指向哪个文件，一般是catalina.bat，那么tomcat的启动信息就是在catalina.bat中定义的。



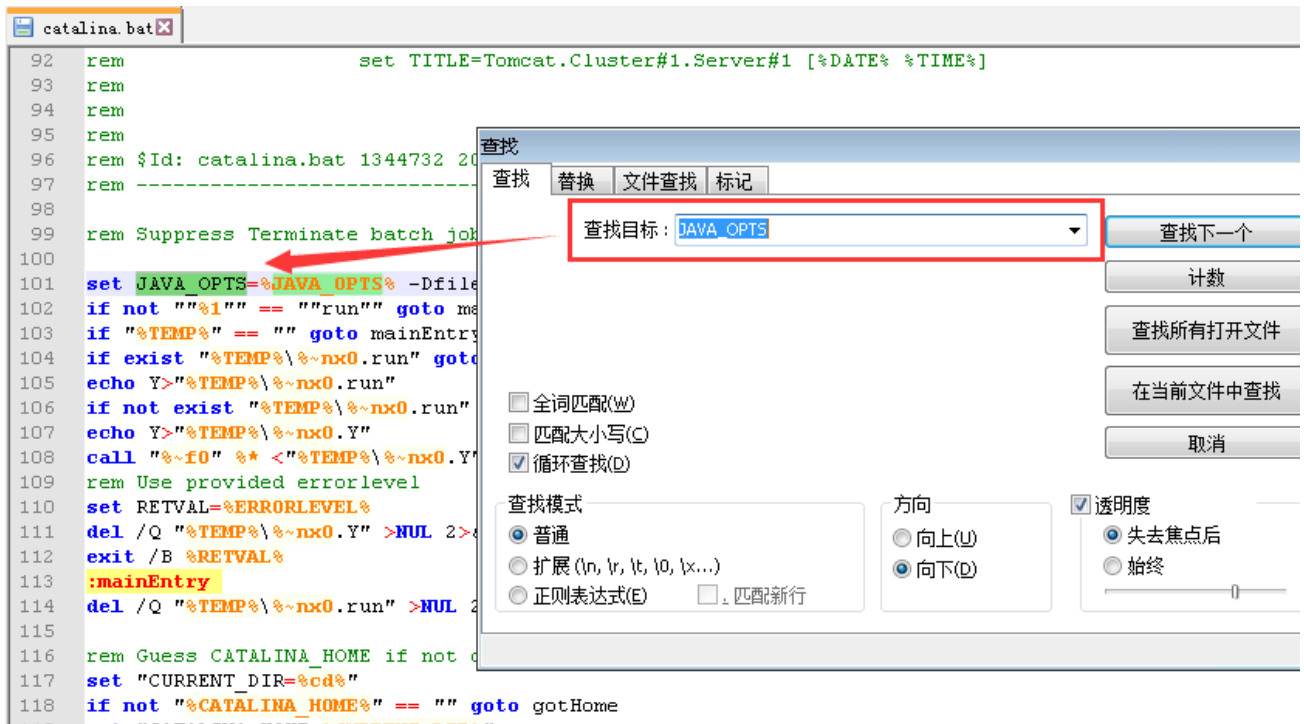
```
32 :gotHome
33 if exist "%CATALINA_HOME%\bin\catalina.bat" goto okHome
34 echo The CATALINA_HOME environment variable is not defined correctly
35 echo This environment variable is needed to run this program
36 goto end
37 :okHome
38
39 set "EXECUTABLE=%CATALINA_HOME%\bin\catalina.bat"
40
41 rem Check that target executable exists
42 if exist "%EXECUTABLE%" goto okExec
43 echo Cannot find "%EXECUTABLE%"
44 echo This file is needed to run this program
45 goto end
46 :okExec
```

3. 找到步骤2中所指向的文件，一般是catalina.bat。用编辑方式打开catalina.bat

在catalina.bat中，一般前面会有一些注释，这些注释列出了此脚本中的一些设置项。其中JAVA_OPTS就是我们接下来即将进行设置的设置项。（如果没有JAVA_OPTS，而只有JAVA_OPTIONS，那么设置项就是JAVA_OPTIONS，下面的内容中的JAVA_OPTS都换成JAVA_OPTIONS）

```
41 rem                                     %CATALINA_BASE%\temp.
42 rem
43 rem   JAVA_HOME       Must point at your Java Development Kit installation.
44 rem                   Required to run the with the "debug" argument.
45 rem
46 rem   JRE_HOME        Must point at your Java Runtime installation.
47 rem                   Defaults to JAVA_HOME if empty. If JRE_HOME and JAVA_HOME
48 rem                   are both set, JRE_HOME is used.
49 rem
50 rem   JAVA_OPTS        (Optional) Java runtime options used when any command
51 rem                   is executed.
52 rem                   Include here and not in CATALINA_OPTS all options, that
53 rem                   should be used by Tomcat and also by the stop process,
54 rem                   the version command etc.
55 rem                   Most options should go into CATALINA_OPTS.
56 rem
57 rem   JAVA_ENDORSED_DIRS (Optional) Lists of of semi-colon separated directories
58 rem                   containing some jars in order to allow replacement of APIs
59 rem                   created outside of the JCP (i.e. DOM and SAX from W3C).
60 rem                   It can also be used to update the XML parser implementation.
61 rem                   Defaults to $CATALINA_HOME/endorsed.
62 rem
63 rem   JPDA_TRANSPORT   (Optional) JPDA transport used when the "jpda start"
```

4. 首先在catalina.bat中，搜索“JAVA_OPTS”，检查一下该文件中是否已经对该设置项进行了设置。



若文件中已经有对该设置项进行设置，检查一下设置项的内容中是否包含 -Xms ， -Xmx， -XX:PermSize 和 -XX: MaxPermSize。如果已经存在，则可以直接修改对应项的值。

如下：已存在

```

100
101 set JAVA_OPTS=%JAVA_OPTS% -Dfile.encoding=GBK -Xms512m -Xmx2048m -XX:MaxPermSize=512m
102 if not "%1" == "run" goto mainEntry
103 if "%TEMP%" == "" goto mainEntry
104 if exist "%TEMP%\%~nx0.run" goto mainEntry
105 echo Y>"%TEMP%\%~nx0.run"
106 if not exist "%TEMP%\%~nx0.run" goto mainEntry
107 echo Y>"%TEMP%\%~nx0.Y"

```

对其直接修改值，比如设置-Xms为256m，-Xmx为10240m，-XX:MaxPermSize为1024m

```

100
101 set JAVA_OPTS=%JAVA_OPTS% -Dfile.encoding=GBK -Xms256m -Xmx10240m -XX:MaxPermSize=1024m
102 if not "%1" == "run" goto mainEntry
103 if "%TEMP%" == "" goto mainEntry
104 if exist "%TEMP%\%~nx0.run" goto mainEntry
105 echo Y>"%TEMP%\%~nx0.run"
106 if not exist "%TEMP%\%~nx0.run" goto mainEntry
107 echo Y>"%TEMP%\%~nx0.Y"

```

5. 如果检查过整个catalina.bat，发现没有对相应的设置项有进行过设置。那么可以在catalina.bat中注释下方的第一行，添加如下设置（该行添加的位置不要求必须在这个位置）

```
set JAVA_OPTS = %JAVA_OPTS% -Xms256m -Xmx10240m -XX:MaxPermSize=1024m
```

如果是tomcat6.0以上，也可使用

```
set JAVA_OPTS=%JAVA_OPTS% -Dfile.encoding=GBK -Duser.region=CN -Duser.language=zh -Djava.awt.headless=true -Xms8192m -Xmx8192m -XX:MaxPermSize=512m
```

```

91 rem                                     Example (all one line)
92 rem                                     set TITLE=Tomcat.Cluster#1.Server#1 [%DATE% %TIME%]
93 rem
94 rem
95 rem
96 rem $Id: catalina.bat 1344732 2012-05-31 14:08:02Z kkolinko $
97 rem -----
98
99 rem Suppress Terminate batch job on CTRL+C
100
101 set JAVA_OPTS=%JAVA_OPTS% -Xms256m -Xmx10240m -XX:MaxPermSize=1024m
102
103 if not "%1" == "run" goto mainEntry
104 if "%TEMP%" == "" goto mainEntry
105 if exist "%TEMP%\%~nx0.run" goto mainEntry
106 echo Y>"%TEMP%\%~nx0.run"

```

二、Linux系统

Linux操作系统上部署Smartbi只能使用war包部署，应用服务器都是自行部署的。设置JVM参数步骤如下：

1. 关闭应用服务器
2. 进入Tomcat\bin目录

在linux上tomcat是通过startup.sh进行启动的，启动时需要使用到tomcat的一些启动信息。可以用编辑方式打开startup.sh，找到 EXECUTABLE设置项，看看该设置项是指向哪个文件，一般是catalina.sh，那么tomcat的x相关启动信息就是在catalina.sh中定义的。

```

..
done

```

```

PRGDIR=`dirname "$PRG"`
EXECUTABLE=catalina.sh

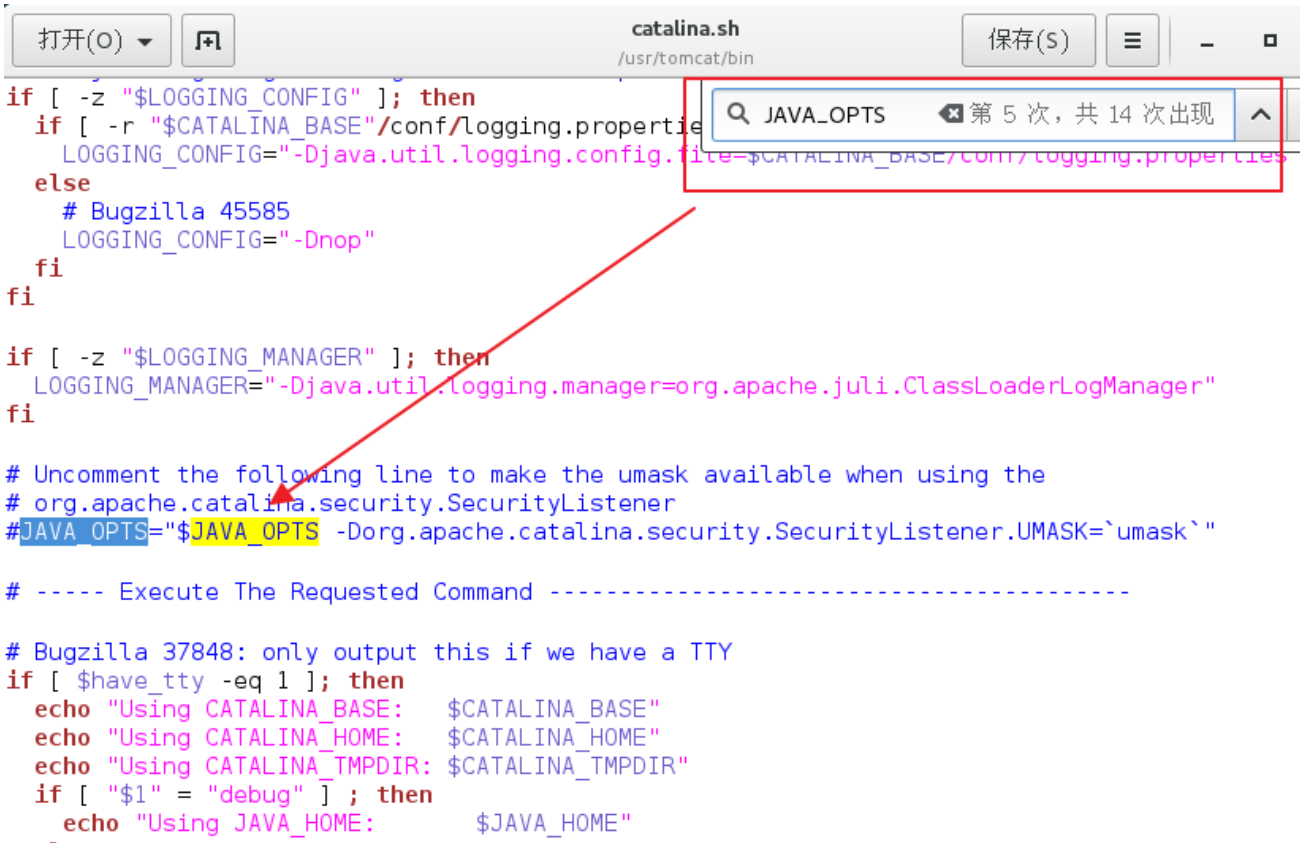
```

3. 找到步骤2中所指向的文件，一般是catalina.sh。用编辑方式打开catalina.sh

在catalina.sh中，一般前面会有一些注释，这些注释列出了此脚本中的一些设置项。其中JAVA_OPTS就是我们接下来即将进行设置的设置项。（如果没有JAVA_OPTS，而只有JAVA_OPTIONS，那么设置项就是JAVA_OPTIONS，下面的内容中的JAVA_OPTS都换成JAVA_OPTIONS）

```
# CATALINA_TMPDIR (Optional) Directory path location of temporary directory
# the JVM should use (java.io.tmpdir). Defaults to
# $CATALINA_BASE/temp.
#
# JAVA_HOME      Must point at your Java Development Kit installation.
#                Required to run the with the "debug" argument.
#
# JRE_HOME       Must point at your Java Runtime installation.
#                Defaults to JAVA_HOME if empty. If JRE_HOME and JAVA_HOME
#                are both set, JRE_HOME is used.
#
# JAVA_OPTS      (Optional) Java runtime options used when any command
#                is executed.
#                Include here and not in CATALINA_OPTS all options, that
#                should be used by Tomcat and also by the stop process,
#                the version command etc.
#                Most options should go into CATALINA_OPTS.
#
```

4. 首先在catalina.sh中，搜索“JAVA_OPTS”，检查一下该文件中是否已经对该设置项进行了设置。



```
if [ -z "$LOGGING_CONFIG" ]; then
if [ -r "$CATALINA_BASE"/conf/logging.properties
LOGGING_CONFIG="-Djava.util.logging.config.file=$CATALINA_BASE/conf/logging.properties
else
# Bugzilla 45585
LOGGING_CONFIG="-Dnop"
fi
fi

if [ -z "$LOGGING_MANAGER" ]; then
LOGGING_MANAGER="-Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager"
fi

# Uncomment the following line to make the umask available when using the
# org.apache.catalina.security.SecurityListener
#JAVA_OPTS="$JAVA_OPTS -Dorg.apache.catalina.security.SecurityListener.UMASK=umask"

# ----- Execute The Requested Command -----

# Bugzilla 37848: only output this if we have a TTY
if [ $have_tty -eq 1 ]; then
echo "Using CATALINA_BASE:  $CATALINA_BASE"
echo "Using CATALINA_HOME:  $CATALINA_HOME"
echo "Using CATALINA_TMPDIR: $CATALINA_TMPDIR"
if [ "$1" = "debug" ]; then
echo "Using JAVA_HOME:       $JAVA_HOME"
```

若文件中已经有对该设置项进行设置，检查一下设置项的内容中是否包含 -Xms，-Xmx，-XX:PermSize 和 -XX:MaxPermSize。如果已经存在，则可以直接修改对应项的值。

如下：已存在

```
# OS specific support. $var _must_ be set to either true or false.
```

```
JAVA_HOME="/usr/jdk1.7.0_45"  
JAVA_OPTS="$JAVA_OPTS -Xms256m -Xmx2048m -XX:MaxPermSize=512m"
```

对其直接修改值，比如设置-Xms为256m，-Xmx为10240m，-XX:MaxPermSize为1024m

```
JAVA_HOME="/usr/jdk1.7.0_45"  
JAVA_OPTS="$JAVA_OPTS -Xms256m -Xmx10240m -XX:MaxPermSize=1024m"
```

5. 如果检查过整个catalina.sh，发现没有对相应的设置项有进行过设置。那么可以在catalina.sh中注释下方的第一行，添加如下设置（该行添加的位置不要求必须在这个位置）

```
JAVA_OPTS = "$JAVA_OPTS -Xms256m -Xmx10240m -XX:MaxPermSize=1024m" （注意有没有双引号）
```

```
Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager"  
#  
# $Id: catalina.sh 1515929 2013-08-20 19:11:24Z markt $  
# -----
```

```
# OS specific support. $var _must_ be set to either true or false.
```

```
JAVA_OPTS="$JAVA_OPTS -Xms256m -Xmx2048m -XX:MaxPermSize=512m"
```

```
JAVA_HOME="/usr/jdk1.7.0_45"
```

另外，Tomcat并不建议JVM参数配置在catalina.sh文件中，通常是配置在tomcat\bin\setenv.sh这个文件中。若是catalina.sh文件中没有，可以找一下这个文件

注：这里有人会问需不需要配置JAVA_HOME，这个是tomcat自身运行需要的参数。如果电脑上有进行JAVA环境设置，这里不设置的话就是使用系统的JAVA环境变量；我们也可以自己通过配置JAVA_HOME指定Tomcat使用的jdk。总得来说，和JAVA_OPTS有一定的关联，但是不包含在配置JVM参数这一项中。

Webshpere

WebSphere中设置JVM参数可以在控制台中进行控制。（下面以Websphere8为例）

1. 在浏览器中输入websphere控制台地址，进入控制台界面。

Integrated Solutions Console Welcome

View: All tasks

- Welcome
- Guided Activities
- Servers
 - Server Types
 - WebSphere application servers
 - WebSphere MQ servers
 - Web servers
- Applications
 - New Application
 - Application Types
 - WebSphere enterprise applications
 - Business-level applications
 - Assets
- Services
- Resources
- Security
- Environment
- System administration
- Users and Groups
- Monitoring and Tuning
- Troubleshooting
- Service integration
- UDDI

Welcome

Welcome

Integrated Solutions Console provides a common administrative console for multiple products. The table lists the product suites that can be administered through this installation. Select a product suite to view more information.

Suite Name	Version
WebSphere Application Server	7.0.0.35

2. 点击左侧的Servers -> Server Types -> WebSphere application servers。

如果是中文版的控制台，点击左侧的 服务器 -> 应用程序服务器

View: All tasks

- Welcome
- Guided Activities
- Servers
 - Server Types
 - WebSphere application servers
 - WebSphere MQ servers
 - Web servers
- Applications
 - New Application
 - Application Types
 - WebSphere enterprise applications
 - Business-level applications
 - Assets
- Services
- Resources
- Security
- Environment

Cell=localhostNode01 Cell, Profile=xhns

Application servers

Application servers

Use this page to view a list of the application servers in your environment and the status of each of these servers. You can use this page to change the status of a specific application server.

Preferences

Name	Node	Host Name	Version
You can administer the following resources:			
server1	localhostNode01	localhost	Base 7.0.0.35
Total 1			

3. 在右侧点击对应的server，如server1

View: All tasks

Welcome

Guided Activities

Servers

Server Types

WebSphere application servers

WebSphere MQ servers

Web servers

Applications

New Application

Application Types

WebSphere enterprise applications

Business-level applications

Assets

Services

Resources

Security

Cell=localhostNode01Cell, Profile=xhns

Application servers

Application servers

Use this page to view a list of the application servers in your environment and the status of each of these servers. You can also use this page to change the status of a specific application server.

Preferences

Name	Node	Host Name	Version
You can administer the following resources:			
server1	localhostNode01	localhost	Base 7.0.0.35
Total 1			

进入设置界面

View: All tasks

Welcome

Guided Activities

Servers

Server Types

WebSphere application servers

WebSphere MQ servers

Web servers

Applications

New Application

Application Types

WebSphere enterprise applications

Business-level applications

Assets

Services

Resources

Security

Environment

System administration

Users and Groups

Monitoring and Tuning

Troubleshooting

Service integration

UDDI

Cell=localhostNode01Cell, Profile=xhns

Application servers

Application servers > server1

Use this page to configure an application server. An application server is a server that provides services required to run enterprise applications.

RuntimeConfiguration

General Properties

Name

server1

Node name

localhostNode01

☐ Run in development mode

☒ Parallel start

☐ Start components as needed

Access to internal server classes

Allow

Server-specific Application Settings

Classloader policy

Multiple

Class loading mode

Container Settings

Session management

SIP Container Settings

Web Container Settings

Portlet Container Settings

EJB Container Settings

Container Services

Business Process Services

Applications

Installed applications

Server messaging

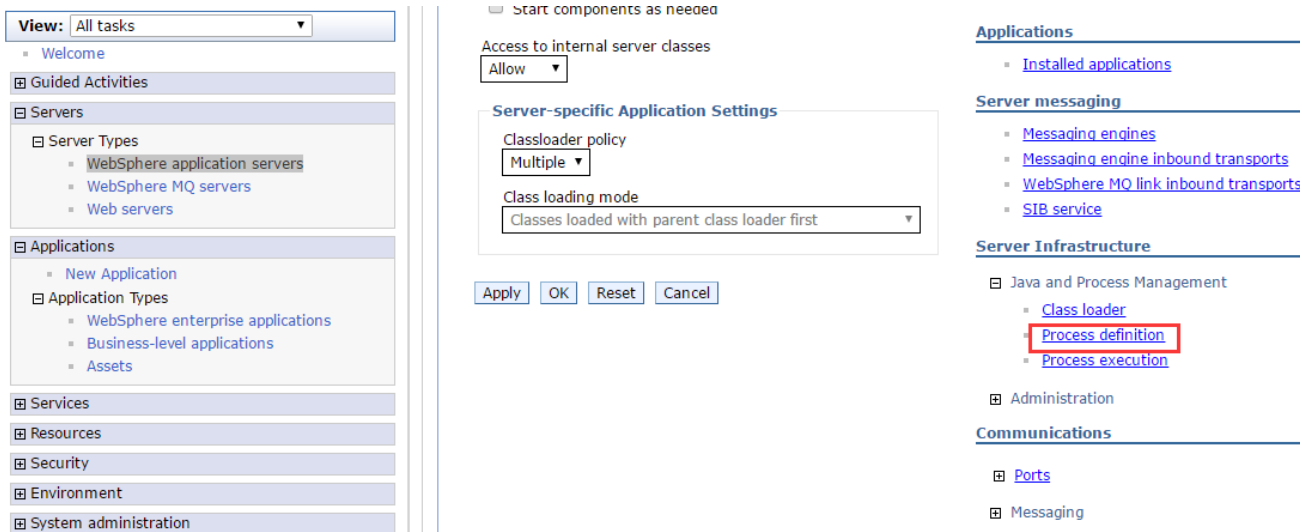
Messaging engines

Messaging engine inbound transports

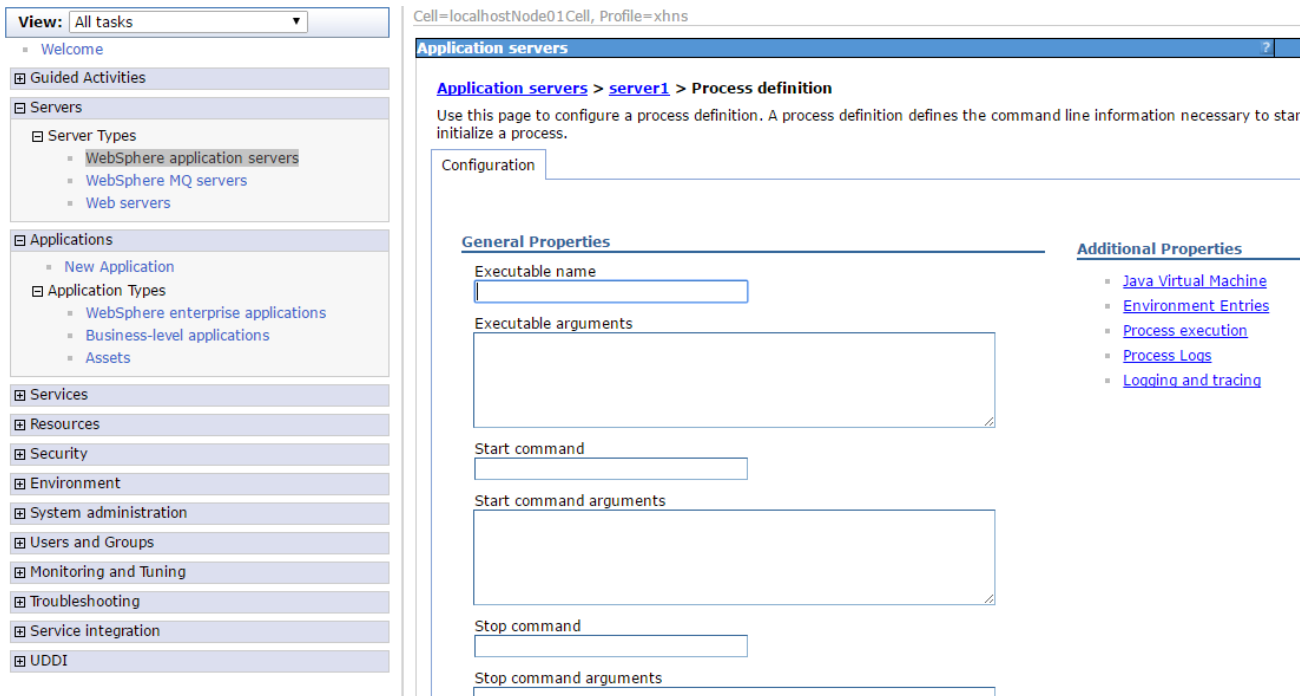
WebSphere MQ link inbound transports

4. 展开右下角的Server Infrastructure -> Java and Process Management, 点击Process definition

在中文版控制台界面中, 展开右下角的 服务器基础结构 -> Java和进程管理, 点击 进程定义

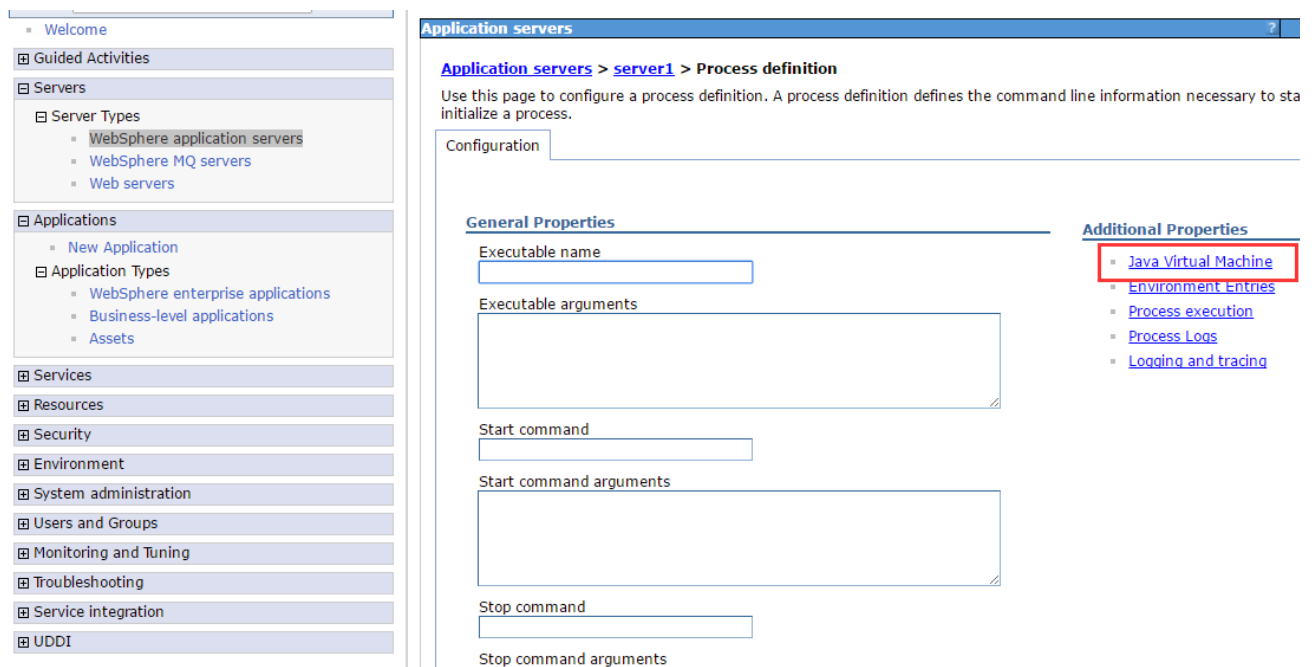


进入进程定义界面

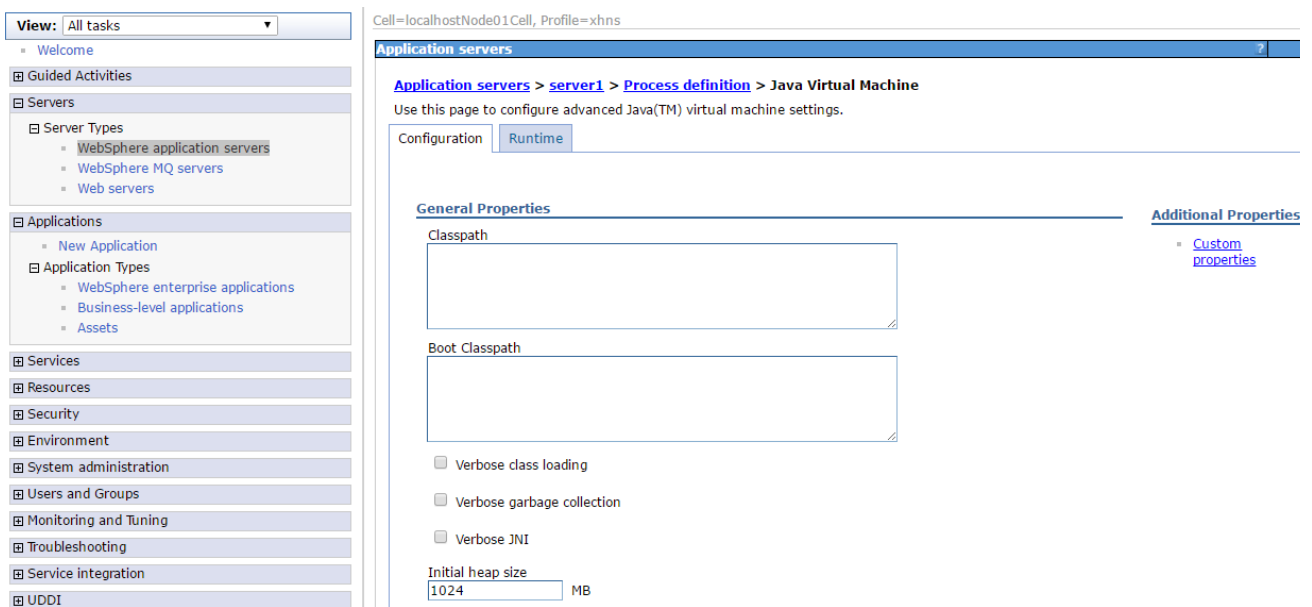


5. 在进程定义界面，点击右侧的 Java Virtual Machine

在中文版控制台界面中，点击右侧的 Java 虚拟机



进入JAVA虚拟机设置界面



6. 在JAVA虚拟机设置界面中，有三个设置项

- (1) Initial heap size，即JVM参数中的Xms，直接输入数值即可
- (2) Maximum heap size，即JVM参数中的Xmx，直接输入数值即可
- (3) Generic JVM arguments，即JVM参数中的其他参数设置。在下方的输入框中输入 `-XX:PermSize=256M -XX:MaxPermSize=512m`，如下：

View: All tasks ▼

- Welcome
- Guided Activities
- Servers
 - Server Types
 - WebSphere application servers
 - WebSphere MQ servers
 - Web servers
- Applications
 - New Application
 - Application Types
 - WebSphere enterprise applications
 - Business-level applications
 - Assets
- Services
- Resources
- Security
- Environment
- System administration
- Users and Groups
- Monitoring and Tuning
- Troubleshooting
- Service integration

☐ Verbose garbage collection

☐ Verbose JNI

Initial heap size
1024 MB

Maximum heap size
2048 MB

☐ Run HProf

HProf Arguments

☐ Debug Mode

Debug arguments
-agentlib:jdwp=transport=dt_socket,server=y,suspend=n,address=7778

Generic JVM arguments
-XX:PermSize=256M -XX:MaxPermSize=512m

Executable JAR file name

☐ Disable JIT

Operating system name
linux

Apply OK Reset Cancel

以上三个设置项在中文版控制台界面中对应的设置项分别是

- (1) 初始堆大小
- (2) 最大堆大小
- (3) 通用JVM实参

7. 保存设置项，重新启动应用即可。

Weblogic

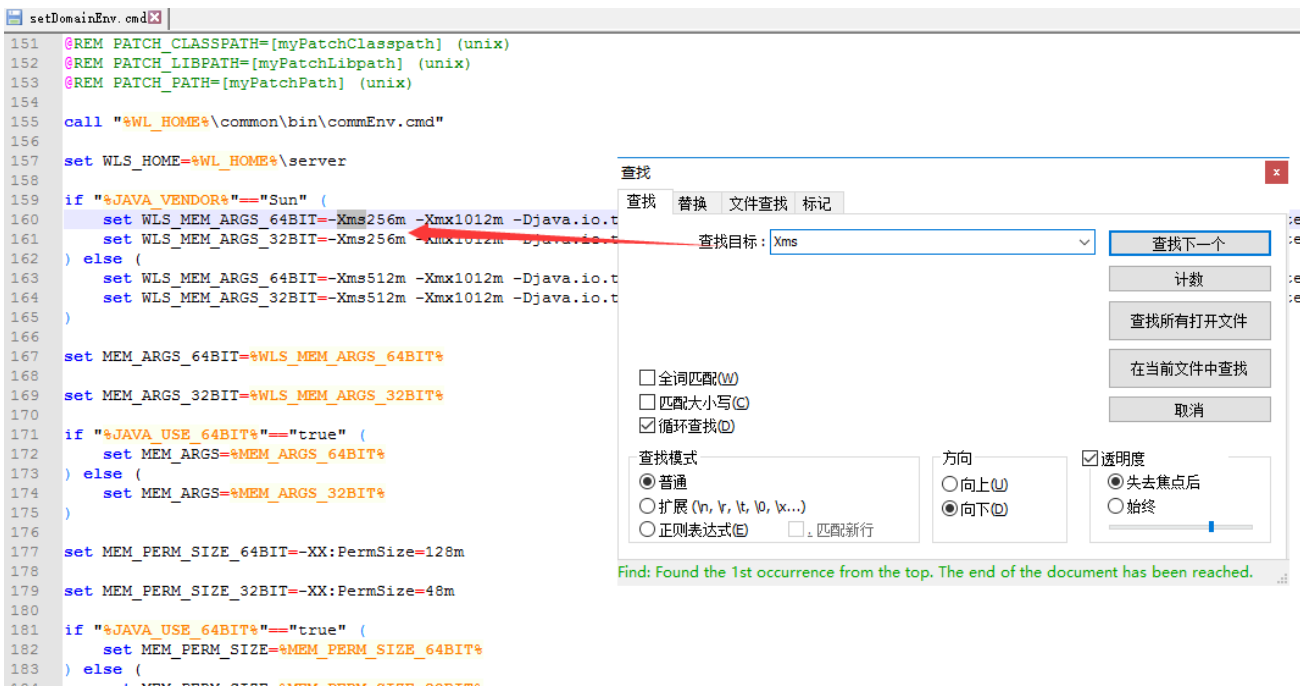
一、Windows系统

1. 关闭应用服务器

2. 进入Weblogic目录下对应 domain的启动目录，一般是 <domain>/bin目录（注意：这里的<domain>在实际上应该是对应的domain名称）

3. 在Windows系统上，对应的配置项是在setDomainEnv.cmd中进行配置的。用编辑方式打开setDomainEnv.cmd

4. 在setDomainEnv.cmd中依次搜索，找到Xms、Xmx、XX:PermSize、XX:MaxPermSize，然后修改其值



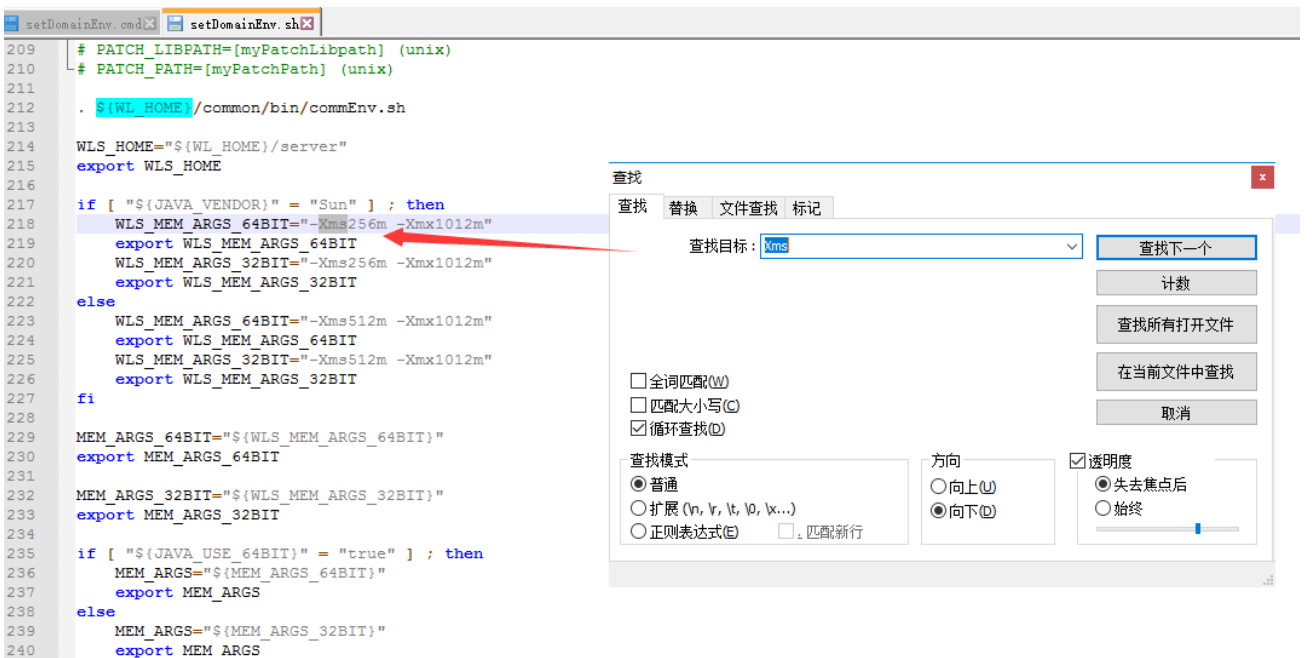
二、Linux系统

1. 关闭应用服务器

2. 进入Weblogic目录下对应 domain的启动目录，一般是 <domain>/bin目录（注意：这里的<domain>在实际上应该是对应的domain名称）

3. 在Windows系统上，对应的配置项是在setDomainEnv.sh中进行配置的。用编辑方式打开setDomainEnv.sh

4. 在setDomainEnv.sh中依次搜索，找到Xms、Xmx、XX:PermSize、XX:MaxPermSize，然后修改其值



Jboss

1. 关闭应用服务器；

2. 用编辑方式打开run.sh/cmd文件；

3. 在JAVA_OPTIONS变量中增加 `-Xms8192m -Xmx8192m -XX:PermSize=256M -XX:MaxPermSize=512m`

4. 重启应用服务器。

