BEATSAM™

TSAM Deployment Guide

Contents

BEA TSAM Deployment Guide

BEA TSAM Deployment Guide

This topic contains the following sections:

- Deploying BEA TSAM Agent

- Deploying BEA TSAM Manager

Deploying BEA TSAM Agent

The LMS (Local Monitor Server) is a Tuxedo system server provided by the BEA TSAM Agent.

To properly deploy the BEA TSAM Agent, you must add the LMS to each Tuxedo machine section of the UBBCONFIG file. Listing 1 shows an example of how to add the LMS to the UBBCONFIG file.

Listing 1 LMS Added to UBBCONFIG File

```
...
*MACHINES
MACHINEA
...
*GROUPS
...
LMSGRP LMID=MACHINEA
...
*SERVERS
```

```
LMS=LMSGRP SRVID=1 MINDISPATCHTHREADS=1 MAXDISPATCHTHREADS=5 CLOPT="-A --
-l tsamweb.abc.com:8080/tsam/dataserver"
...
```

The LMS is multi-threaded so that you can configure the service dispatching thread number based on the Tuxedo application work load. The "-l" option specifies the BEA TSAM Data Server address which is configured in the BEA TSAM Manager. For more LMS information, see the BEA TSAM Reference Guide. The BEA TSAM Data Server uses the same port number as the BEA TSAM Manager Console.

Note: LMS can also be added to a running Tuxedo application using TMIB.

Deploying BEA TSAM Manager

- Overview
- Quick Start
- BEA TSAM Environment Parameters
- Deploying the Database
- Deploying Apache Tomcat

Overview

The BEA TSAM Manager functions as a Web application in a Servlet/JSP container. It uses the DBMS (Database Management System) to store persistent Tuxedo performance and monitoring data.

Notes: Apache Tomcat 5.5.17 Servlet/JSP container is installed with BEA TSAM Manager.

An evaluation copy of Apache Derby 10.2.2.0 DBMS is installed with BEA TSAM Manager.

Quick Start

If you run BEA TSAM Manager with Apache Derby and Apache Tomcat on the same machine, you can start and shut down BEA TSAM Manager using the following steps:

1. Start BEA TSAM Manager

   UNIX:

   a. cd `<TSAMDIR>`

   b. . ./tsam.env

   c. cd `<TSAMDIR>/bin`

   d. ./startup.sh

   Windows:

   a. cd `<TSAMDIR>\bin`

   b. startup.cmd

2. Shut down BEA TSAM Manager

   UNIX:

   a. cd `<TSAMDIR>/bin`

   b. ./shutdown.sh

   Windows:

   a. cd `<TSAMDIR>\bin`

   b. shutdown.cmd

BEA TSAM Environment Parameters

Table 1 lists the BEA TSAM environment parameters.

Table 1 Environment Parameters

Name	Description
TSAMDIR	The BEA TSAM installed directory.
JAVA_HOME or JRE_HOME	JRE or JDK directory installed. BEA TSAM installs a JRE by default.
JAVA_OPTS	Set recommended JVM memory. Minimum: 512M; maximum: 1024M.

Table 1 Environment Parameters

Name	Description
DERBY_HOME	Apache Derby installed directory
CATALINA_HOME	Apache Tomcat installed directory

The BEA TSAM environment parameters are configured automatically by the BEA TSAM Manager installer. On UNIX platforms, the BEA TSAM environment parameters are located in the `tsam.env` file; on Windows platforms, they are located in the `startup.cmd` file.

Deploying the Database

The BEA TSAM Manager supports the following DBMS:

- Apache Derby
- Oracle

Deploying an Apache Derby Database

No other actions are required if you use the Apache Derby provided in the BEA TSAM installer. The following information is for reference.

To successfully deploy a Apache Derby database, you must do the following steps:

1. Setup a Apache Derby Environment
2. Start Up and Shut Down
3. Create BEA TSAM Manager Database
4. Configure the BEA TSAM Manager for an Apache Derby

Setup a Apache Derby Environment

Use the following commands to set up a Apache Derby environment on UNIX and Windows platforms:

UNIX:

```
export JAVA_HOME=<JDK or JRE directory>
export DERBY_HOME=<TSAMDIR>/db-derby-10.2.2.0-bin
cd <TSAMDIR>
```

Windows:

```
set JAVA_HOME=<JDK or JRE directory>
set DERBY_HOME=<TSAMDIR>\db-derby-10.2.2.0-bin
cd <TSAMDIR>
```

Start Up and Shut Down

Use the following commands to start up and shutdown a Derby database on UNIX and Windows platforms

UNIX:

```
cd <TSAMDIR>
startup: $DERBY_HOME/bin/startNetworkServer
shutdown: $DERBY_HOME/bin/stopNetworkServer
```

Note: If Apache Tomcat and Apache Derby are not run on the same machine, start Apache Derby using the following command:

```
NetworkServerControl start [-h <host>] [-p <portnumber>]
```

Windows:

```
cd <TSAMDIR>
startup: %DERBY_HOME%\bin\startNetworkServer.bat
shutdown:%DERBY_HOME%\bin\stopNetworkServer.bat
```

For more information, see the Apache Derby documentation (located in
`<TSAMDIR>/db-derby-10.2.2.0-bin/docs` or the Apache Derby Web site).

Create BEA TSAM Manager Database

A pre-built Apache Derby database instance is provided; however, you can also create one using the provided SQL script `sql/createdb.sql`. The following example creates a BEA TSAM Manager database using the Apache Derby `ij` utility:

```
ij version 10.2
ij> connect 'jdbc:derby://localhost/tmonitordb;create=true';
ij> run 'createdb.sql';
```

Configure the BEA TSAM Manager for an Apache Derby

Table 2 lists the BEA TSAM Manager configuration file parameters you need to edit to configure an Apache Derby database:

Note: The BEA TSAM Manager configuration file is located at:
`<TSAMDIR>/apache-tomcat-5.5.17/webapps/tsam/WEB-INF/web.xml:`

Table 2 BEA TSAM Manager Database Configuration Parameters

Parameter	Description
`tsam.jdbc.url`	JDBC connection string for BEA TSAM Manager database. For example: `jdbc:derby://localhost:1527/tmonitordb`
`tsam.jdbc.userna me`	BEA TSAM Manager database connection user name.
`tsam.jdbc.passwo rd`	BEA TSAM Manager database connection password.
`tsam.jdbc.classn ame`	JDBC driver class name.

Deploying an Oracle Database

To successfully deploy an Oracle database, you must do the following steps:

1. Create an Oracle Database

2. Install Oracle JDBC Drivers

3. Configure BEA TSAM Manager for Oracle DBMS

Create an Oracle Database

An SQL script, `sql/createdb_oracle.sql`, is provided to create a BEA TSAM Manager database for Oracle.

Notes: Before using this script, you must have already installed and configured and Oracle database.

If you use an Oracle database, it is recommended that you comment out or remove the lines in the BEA TSAM start/shutdown scripts that start/shutdown the pre-built Apache Derby database instance.

Install Oracle JDBC Drivers

To use BEA TSAM Manager with Oracle, you must download the Oracle JDBC drivers and put them into a specified directory. Do the following steps:

1. Download JDBC drivers from the Oracle Web site
   (`http://www.oracle.com/technology/software/tech/java/sqlj_jdbc/index.html`)

   Note: `ojdbc14.jar` for Oracle 10g Release 2 is recommended. You may choose different JDBC driver version according to your Oracle release.

2. Copy the downloaded jar files to the following directory:
   `apache-tomcat-5.5.17/webapps/tsam/WEB-INF/lib`.

   Note: Apache Tomcat typically creates this directory the first time the BEA TSAM Manager runs. If you do not find this directory, you can manually create it using the following command:

   ```
   cd apache-tomcat-5.5.17/webapps
   mkdir tsam
   cd tsam
   jar -xf ../tsam.war
   ```

For more information, contact your DBA or go to the Oracle website.

Configure BEA TSAM Manager for Oracle DBMS

Table 3 lists the BEA TSAM Manager configuration file parameters you need to edit to configure an Oracle database:

Note: The BEA TSAM Manager configuration file is located
at`<TSAMDIR>/apache-tomcat-5.5.17/webapps/tsam/WEB-INF/web.xml`:

Table 3 BEA TSAM Manager Database Configuration Parameters

Parameter	Description
`tsam.jdbc.url`	JDBC connection string for TSAM Manager database. For example: `jdbc:oracle:thin:@localhost:1521:orcl`
`tsam.jdbc.userna me`	BEA TSAM Manager database connection user name.

`tsam.jdbc.passwo` `rd`	BEA TSAM Manager database connection password.
`tsam.jdbc.classn` `ame`	JDBC driver class name.

Deploying Apache Tomcat

Apache Tomcat is automatically configured when the BEA TSAM Manager is installed.Configuration parameters for Apache Tomcat are located in the following files:

- `<TSAMDIR>/apache-tomcat-5.5.17/conf/server.xml`

- `<TSAMDIR>/apache-tomcat-5.5.17/conf/web.xml`

You can directly run the BEA TSAM Manager without changing any Apache Tomcat configuration parameters.

If required, you can change the default for the following parameters:

- Configure Apache Tomcat Listening Port

- Configure Session Timeout

- Configure HTTP KeepAlive

- Configure POST Maximum Byte Size

- Configure Maximum Thread Simultaneous Processing

- Configure Maximum Incoming Connection Requests

Configure Apache Tomcat Listening Port

The default Apache Tomcat port number is 8080. You can change the default based on your environment. The port is defined in `<TSAMDIR>/apache-tomcat-5.5.17/conf/server.xml` file. For example, you could change to port 909 by doing the following:

```
<!-- Define a non-SSL HTTP/1.1 Connector on port 8080 -->

    <Connector port="9090" acceptCount="100" connectionTimeout="20000"
disableUploadTimeout="true" enableLookups="false" maxHttpHeaderSize="8192"
maxSpareThreads="75" maxThreads="150" minSpareThreads="25"
redirectPort="8443"/>
```

The BEA TSAM Manager console and data server URLs depend on the Apache Tomcat port setting. The following example assumes the default port "8080" used:

- BEA TSAM Manager console URL

    http://host:8080/tsam

- BEA TSAM Manager data server URL

    http://host:8080/tsam/dataserver

The "host" is the full domain name or IP address where the BEA TSAM Manager is installed. The BEA TSAM Agent LMS must be set correctly using the BEA TSAM Manager data server URL.

Configure Session Timeout

The session timeout can be configured by modifying the `<session-config>` element in `<TSAMDIR>/apache-tomcat-5.5.17/conf/web.xml` file . For example, the following configuration invalidates an inactive session after 30 minutes.

```
<session-config>
<session-timeout>30</session-timeout>
</session-config>
```

Configure HTTP KeepAlive

This is the maximum number of HTTP requests that can be pipelined until the connection is closed by the server. Apache Tomcat provides an attribute `maxKeepAliveRequests` in the `<TSAMDIR>/apache-tomcat-5.5.17/conf/server.xml` file for custom control . Setting this attribute to `1` disables HTTP/1.0 keep-alive, as well as HTTP/1.1 keep-alive and pipelining. Setting this to `-1` allows an unlimited pipeline amount or keep-alive HTTP requests. If not specified, this attribute is set to 100.

For example:

```
<Connector port="8080" ...  maxKeepAliveRequests="-1"/>
```

Configure POST Maximum Byte Size

You must configure the maximum size in bytes of the POST, `maxPostSize`, that is handled by the container `FORM` `URL` parameter parsing. The limit can be disabled by setting this attribute to a value less than or equal to 0. If not specified, this attribute is set to 2097152 (2 megabytes). `maxPostSize` is modified in the `<TSAMDIR>/apache-tomcat-5.5.17/conf/server.xml` file.

For example:

```
<!-- Define a non-SSL HTTP/1.1 Connector on port 8080 -->

    <Connector maxPostSize="0" acceptCount="100" connectionTimeout="20000"
disableUploadTimeout="true" enableLookups="false" maxHttpHeaderSize="8192"
maxSpareThreads="75" maxThreads="150" minSpareThreads="25" port="8080"
redirectPort="8443"/>
```

If the BEA TSAM has huge amount monitoring data in the database, and `maxPostSize` is not adequate, an exception message, "javax.servlet.ServletException: Post too large", is given when any operations fail. It is recommended that `maxPostSize` is set to `0`.

Configure Maximum Thread Simultaneous Processing

This is the maximum number of request processing threads created by Apache Tomcat, `maxThreads`, which determines the maximum number of simultaneous requests that can be handled by Apache Tomcat. If not specified, this attribute is set to 200. `maxThreads` is modified in the Apache Tomcat `server.xml` file. BEA TSAM keeps a thread-level JDBC connection if the thread has database access. When you configure this attribute, you need to consider the DBMS capability.

Configure Maximum Incoming Connection Requests

This is the maximum queue length, `acceptCount`, for incoming connection requests when all possible request processing threads in Apache Tomcat are in use. Any requests received when the queue is full are refused. The default value is 100. Setting `acceptCount` to a larger number allows the Apache Tomcat queue to accept more HTTP requests if Apache Tomcat is busy. `acceptCount` is modified in the Apache Tomcat `server.xml` file.

The following is a `maxPostSize`, `maxThreads`, `acceptCount` example:

```
<!-- Define a non-SSL HTTP/1.1 Connector on port 8080 -->

<Connector maxPostSize="0" acceptCount="1000" connectionTimeout="20000"
disableUploadTimeout="true" enableLookups="false" maxHttpHeaderSize="8192"
maxSpareThreads="75" maxThreads="130" minSpareThreads="25" port="8080"
redirectPort="8443"/>
```

For more information, see the Apache Tomcat Web site.