SPECjAppServer2004

{scrollbar}

top

This article is a copy of the version for SPECjAppServer2004 v1.05 and Geronimo v1.2. It's now being updated for SPECjAppServer2004 v1.08 and Geronimo v2.0.2, but the updating process is not yet complete.

Running SPECjAppServer2004 benchmark on Geronimo

Still not a success, your help needed!

This article shows how to measure the performance of the Geronimo application server using the industry standard SPECjAppServer2004 benchmark.

Issues still exist that prevent SPECjAppServer2004 from running on Geronimo.

This article is not a success story, but a collection of notes on the progress that has been made in this direction.

Current issue that requires your help is: #Running the benchmark
Disclaimer: This article is created to write down the existing experience and to make it reproducible. It is not targeted to be a comprehensive guide on either product or on merging them together. It's not also a replacement to the products' documentation, but just a step-by-step guide on how to make things work in a simple configuration, as it worked for me. Make sure you at least look through the documentation on both products before you proceed.
SPECjAppServer is a trademark of the **Standard Performance Evaluation Corp.** (SPEC). The official web site for SPECjAppServer2004 is located at http://www.spec.org/jAppServer2004/.

SPECjAppServer2004 version **1.08** introduces a reduced workload called **EAStress2004** that relaxes run and reporting rules, enabling informal results to be shared more easily in open-source research and development projects.

The **EAStress2004** workload in **SPECjAppServer2004 v1.08** can be used as a tool to optimize performance of products under development and to share results in public forums. Unlike **SPECjAppServer2004** results, testing results from the **EAStress2004** workload do not need to be reviewed by **SPEC** prior to being made public.

Results from the **EAStress2004** workload cannot be used for marketing purposes, and comparisons to other **SPECjAppServer2004** results are not permitted.

See full press release on **SPECjAppServer2004/EAStress2004 v1.08** here: http://www.spec.org/jAppServer2004/jAppServer2004v108.html.

All logs, stacks and result files in this article are extracted from **EAStress2004 v1.08** runs.

This article is written for SPECjAppServer2004 v1.08 and Geronimo v2.0.2. For other versions some stages may be different. Older versions of this article for SPECjAppServer2004 v1.05 and older versions of Geronimo can be found here: v1.0, v1.1, v1.2.

The described configuration uses as many Geronimo components as possible, including the built-in Derby database and the built-in Jetty or Tomcat servlet container. In fact, the configuration only uses Java, Geronimo, an external servlet container (e. g. Tomcat) and SPECjAppServer2004. To plug external components (most probably, a database), you have to change your configuration accordingly.

This configuration also assumes that all the components (except, possibly, the SPECjAppServer2004 Driver and the SPECjAppServer2004 Supplier Emulator and its servlet container) are run on the same machine. If you want to run the Distributed workload, your configuration will be different.

This configuration uses the Microsoft Windows XP Professional Service Pack 2 operating system, Cygwin shell, Sun Java SE 5.0 Update 11 and Tomcat v5.0.30 to write this article. If you use another OS, Java or servlet container, some stages may be different.

This article uses forward slashes (/) for command lines, adjust to backslashes (\ ) accordingly if you use Windows command prompt.

This article has the following structure:

- #General information

    - #About Geronimo
    - #About SPECjAppServer2004/EAStress2004
- #Obtaining products
    - #Obtaining Geronimo
    - #Obtaining SPECjAppServer2004
- #Conventions and environment
    - #Hosts
    - #Directories
- #Installing products
    - #Installing Geronimo
    - #Installing SPECjAppServer2004
- #Configuring Geronimo
    - #Adjusting configuration
    - #Starting Geronimo
    - #Accessing the console
    - #Creating the database
    - #Locating the SQL files
    - #Creating the tables
- #Configuring SPECjAppServer2004
    - #Basic configuration
    - #Building the application
    - #Preparing database configuration
    - #Loading the tables
- #Deploying components
    - #Logging in
    - #Deploying database connector
    - #Deploying JMS connector
    - #Deploying the main application
    - #Verifying the deployment
- #Deploying the Supplier Emulator
    - #Using the Geronimo servlet container
    - #Using a stand-alone servlet container
    - #Verifying the deployment
- #Running the benchmark
- #Processing results

General information

About Geronimo

Geronimo is the Apache Software Foundation Java EE 5 certified application server. It is developed under Apache License and can be downloaded freely.

Apache site: http://apache.org

Product site: http://geronimo.apache.org

Latest version, 2.0.2: http://geronimo.apache.org/apache-geronimo-v202-release.html

Release notes: http://cwiki.apache.org/GMOxDOC20/release-notes-202txt.html

Documentation page: http://geronimo.apache.org/documentation.html

The best document available is **"Apache Geronimo: J2EE Development and Deployment"** book by Aaron Mulder: http://chariotsolutions.com/geronimo/

Other important resources are FAQ: http://cwiki.apache.org/GMOxKB and Wiki: http://cwiki.apache.org/geronimo

About SPECjAppServer2004/EAStress2004

SPECjAppServer2004 is a commercial benchmark for measuring the performance of Java EE application servers.

EAStress2004 is a reduced workload that is a part of SPECjAppServer2004 v1.08, it relaxes run and reporting rules, enabling informal results to be shared more easily in open-source research and development projects.

SPEC site: http://www.spec.org

Product site: http://www.spec.org/jAppServer2004/

Press release on v1.08 and EAStress2004: http://www.spec.org/jAppServer2004/jAppServer2004v108.html

FAQ: http://www.spec.org/jAppServer2004/docs/FAQ.html

User's Guide: http://www.spec.org/jAppServer2004/docs/UserGuide.html

Run and Reporting Rules: http://www.spec.org/jAppServer2004/docs/RunRules.html

Back to Top

Obtaining products

Obtaining Geronimo

The latest Geronimo version for now is 2.0.2.

General download page: http://geronimo.apache.org/downloads.html

Two builds of Geronimo exist, with Jetty or Tomcat servlet container enabled by default. You can download either one at http://geronimo.apache.org/downloads.html, they are around 55 MB in size each. This document was written primarily using Jetty version, but Tomcat version works fine also.

Obtaining SPECjAppServer2004

SPECjAppServer2004 costs $2000 ($250 for non-profit/educational purposes), you can order it online. See FAQ for details.

The latest version is 1.08, coming as the `SPECjAppServer2004-Kit-v1.08.jar` file, 12 MB in size.

Back to Top

Conventions and environment

This section contains important notions that mark the important hosts and directories.

Hosts

This article is written in terms of the following machines:

- **`geronimo.host`** – the machine where Geronimo is run, with SPECjAppServer2004 deployed.
- **`emulator.host`** – the machine where the SPECjAppServer2004 Supplier Emulator is deployed.
- **`driver.host`** – the machine where the SPECjAppServer2004 Driver is run. If you use configuration with multiple Drivers, you have to repeat all the operations for this host on all Driver hosts.
- **`master.host`** – the main, Master **`driver.host`** in configurations with multiple Drivers.

The **`emulator.host`** and the **`driver.host`** may be the same machine.

The **`geronimo.host`** and the **`emulator.host`** may be the same machine, moreover, the SPECjAppServer2004 Supplier Emulator may be deployed into a Geronimo built-in servlet container (Jetty or Tomcat).

The **`geronimo.host`** and the **`driver.host`** may be the same machine, but you have to adjust the Geronimo configuration, as both Geronimo and the SPECjAppServer2004 Driver create RMI Registry on the default port (1099) and would conflict on that.

Sharing **`geronimo.host`** with **`emulator.host`** or **`driver.host`** is contradicting with the SPECjAppServer2004 documentation and would impact the performance severely and invalidate the benchmark results. However technically this is possible.

Directories

This section lists important directories that are further addressed in this article. They can be chosen arbitrary, but should not overlap.

- **`<GERONIMO>`** – directory at the **`geronimo.host`** where Geronimo is installed.

- **<SPEC>** – directory at the **geronimo.host** where SPECjAppServer2004 is installed.
- **<KIT>** – directory at the **geronimo.host** containing the files attached to this article.
- **<TOMCAT>** – directory at the **emulator.host** where Tomcat is installed.
- **<DRIVER>** – directory at the **driver.host** that is a copy of the **<SPEC>** directory.
- **<DRIVER_GERONIMO>** – directory at the **driver.host** that is a copy of the **<GERONIMO>** directory.
- **<JAVA_HOME>** – JAVA_HOME location at the **driver.host**.
- **<OUTPUT>** – directory at the **driver.host** where the SPECjAppServer2004 Driver will store its output.
- **<DUMP>** – directory at the **driver.host** where the SPECjAppServer2004 Driver will store its temporal files.

  On Windows some components may work incorrectly if working paths are too long or contain spaces. So it's recommended that you avoid long paths and spaces in them.

Back to Top

Installing products

First, save the files attached to this article to a local directory. This will be your **<KIT>** directory.

Installing Geronimo

You can easily install Geronimo using the `.zip` or `.tar.gz` archive.

Extract the downloaded archive to a local directory. The `geronimo-jetty6-jee5-2.0.2` or `geronimo-tomcat6-jee5-2.0.2` directory is created, that is your **<GERONIMO>** directory.

Installing SPECjAppServer2004

Run:

solid
```
java -jar SPECjAppServer2004-Kit-v1.08.jar
```

Click **Next**, read and accept the license agreement, and type in the directory you want the SPECjAppServer2004 to be installed to. This directory will be your **<SPEC>** directory.

Click **Install**.

Wait until the installation completes, then click **Ready**.

Back to Top

Configuring Geronimo

Adjusting configuration

If your **geronimo.host** and your **driver.host** are the same machine, you have to adjust the port number of the Geronimo RMI Registry (to e. g. 1199), otherwise it would conflict with the SPECjAppServer2004 Driver that uses the default port of 1099. Edit the **<GERONIMO>**`/var/config/config-substitutions.properties` file and change the `NamingPort` variable value.

Starting Geronimo

Go to your **<GERONIMO>** directory.

Start Geronimo by typing:

solid
```
java -Djava.endorsed.dirs=lib/endorsed -javaagent:bin/jpa.jar -Dopenejb.jndiname.failoncollision=true -
Dopenejb.jndiname.format={ejbName} -jar bin/server.jar
```
Please note the two OpenEJB settings, they're necessary to tune Geronimo to the simple format of EJB JNDI names SPECjAppServer2004 uses.

See details at http://cwiki.apache.org/GMOxDEV/client-jndi-names.html and http://cwiki.apache.org/OPENEJB/service-locator.html.

You may also use **<GERONIMO>**`/bin/geronimo.sh` or **<GERONIMO>**`/bin/geronimo.bat` scripts instead, after appropriate adjustments.

Geronimo v2.0.2 needs access to Internet for SPECjAppServer2004 application to be deployed – it tries to fetch XML schemas from http://java.sun.com site.

So if you're behind a firewall, add the appropriate `-Dhttp.proxyHost=` and `-Dhttp.proxyPort=` options to the Geronimo startup line.

This problem is caused by OPENEJB-700 bug which is already fixed and the fix should make it to the next version of Geronimo.

If your installation of Geronimo has no access to Internet at all, or your proxy requires authentication, you may use the following workarounds (thanks to Konstantin Malynkin for describing them). You may download the necessary DTDs to your local drive and specify the local path to them in your deployment descriptors (see **<SPEC>**`/src/deploy/geronimo` directory). For example, you could use `<!DOCTYPE ejb-jar SYSTEM "C:/DTD/ejb-jar_2_0.dtd">` instead of `<!DOCTYPE ejb-jar PUBLIC '-//Sun Microsystems, Inc.//DTD Enterprise JavaBeans 2.0//EN' 'http://java.sun.com/dtd/ejb-jar_2_0.dtd'>`. Another way is to omit verification at all by deleting or commenting out the references to DTDs in the deployment descriptors. Note that in both cases you would have to rebuild and redeploy the SPECjAppServer2004 application for the changes to take effect.

It will take some time to start. After that, you will see:

If you get another result, particularly, if network errors show up, then something has gone wrong.

Sometimes, the startup fails because some local network addresses are inaccessible. This could happen, for example, if you have used a VPN interface that is disconnected now. By default, Geronimo uses the first local address it comes across to access its components, and may try to use a stale address, causing startup errors.

You may try disabling and then re-enabling the unused network interfaces to resolve such issues.

Accessing the console

Open your web browser and connect to the Geronimo Console at **`http://geronimo.host:8080/console/`**

Log in using the user name and password (**system** and **manager** by default).

Now you may investigate the console if you wish.

Creating the database

Go to the **Embedded DB - DB Manager** in the Console Navigation.

Create the benchmark database by typing its name (**SPECDB**) in the **Create DB** field and clicking **Create**.

Locating the SQL files

To create database tables, you can use the default SQL scripts provided in the **<SPEC>**`/schema/sql` directory. However, the directory includes five scripts, and they contain `DROP TABLE` commands that would fail if you try to execute them in the console when tables are not yet created.

Instead, it is recommended that you use the allTablesNoDrop.sql file, if you are creating the tables from scratch, or allTables.sql file if you want to drop and recreate the tables. Both files were created from the **<SPEC>**`/schema/sql` files by simple concatenation, allTablesNoDrop.sql also has `DROP TABLE` commands removed.

Creating the tables

Make sure **SPECDB** is selected in **Use DB** field and then copy-paste the SQL code to **SQL Command/s** frame. Click **Run SQL** button above it.

After a short delay, the frame will clear and the **Result** field below it will say **SQL command/s successful**. If not – check what you did wrong and try again.

If you use multiple SQL scripts, repeat the operations above for each of them.

Back to Top

Configuring SPECjAppServer2004

Basic configuration

`deploy` directory

Go to the **<SPEC>**`/src/deploy` directory and copy the `reference` subdirectory with its contents with the name `geronimo`.

Edit the deployment plans in **<SPEC>**`/src/geronimo`, remove all `message-driven-destination` tags from `mfg.xml`, `orders.xml` and `supplier.xml` – Geronimo v2.0.2 doesn't handle those tags correctly.

Instead of removing those tags, you also can replace them with `<message-destination-type>javax.jms.Queue</message-destination-type>`.

This problem is caused by OPENEJB-701 bug and should disappear after that bug is fixed.

`geronimo.env` file

Go to the **<SPEC>**`/config` directory.

Put the attached geronimo.env template file there. Edit it, make sure you set the values for the following variables:

```
JAS_HOME=<SPEC>
JAVA_HOME=<JAVA_HOME>
J2EE_HOME=<GERONIMO>
JAS_HOST=geronimo.host
EMULATOR_HOST=emulator.host
```

Use forward slashes (/) as directory separators!

You may leave the other variables intact.

`appsserver` file

Edit the **<SPEC>**`/config/appsserver` file – replace the word **default** there with the word **geronimo**.

`run.properties` file

Edit the **<SPEC>**`/config/run.properties` file. Note that it will be used on the **`driver.host`** and make sure the following variables have correct values:

```
Url = http://geronimo.host:8080/SPECjAppServer/app?
outDir = <OUTPUT>
dumpDir = <DUMP>
```

`setenv.bat` file

Edit the **<SPEC>**`/bin/setenv.bat` file, make sure you set the values for the following variables:

```
JAVA_HOME=<JAVA_HOME>
JAS_HOME=<DRIVER>
APPSSERVER=geronimo
```

Building the application

Go to the **<SPEC>** directory.

Clean-up your installation:

solid
```
ant/bin/ant clean
```

Build the application and configure it for Geronimo:

solid
```
ant/bin/ant -Dappserver=geronimo
```

You will get the **BUILD SUCCESSFUL** diagnostic.

Make sure the files `SPECjAppServer.ear` and `emulator.war` are created in the **<SPEC>**`/jars` directory.

Rename `emulator.war` to `Emulator.war`.

Preparing database configuration

In the described configuration, the same database is used for all tables.

Go to the **<SPEC>**`/config` directory. Replace the content of each of the `*db.properties` files you find there with the contents of the attached db.properties template file. Make sure the `pipeDir` variable there points to a valid temporary directory, adjust if necessary.

Loading the tables

Run:

solid
```
ant/bin/ant -Dappserver=geronimo loaddb
```

After some time, you will get the **BUILD SUCCESSFUL** diagnostic.

Back to Top

Deploying components

At this stage you need to deploy the configured components to Geronimo.

Note that if your **`geronimo.host`** and your **`driver.host`** are the same machine, and you changed the port number of the Geronimo RMI Registry, you should specify that port number in all deployer commands, like this:

solid
```
java -jar bin/deployer.jar -port 1199 ...
```

Logging in

To avoid specifying login credentials on any call to deployer, you can login first:

solid
```
java -jar bin/deployer.jar -u system -p manager login
```

Deploying database connector

To deploy a connector to the Derby **SPECDB** database you created earlier, go to the **<GERONIMO>** directory and run:

solid
```
java -jar bin/deployer.jar deploy repository/org/tranql/tranql-connector-derby-embed-xa/1.4/tranql-
connector-derby-embed-xa-1.4.rar <KIT>/sjas-db.xml
```

You will get the **SPECjAppServer2004/DB/1.08/rar** diagnostic.

Deploying JMS connector

To deploy an ActiveMQ JMS connector for SPECjAppServer2004, go to the **<GERONIMO>** directory and run:

solid
```
java -jar bin/deployer.jar deploy repository/org/apache/geronimo/modules/geronimo-activemq-ra/2.0.2
/geronimo-activemq-ra-2.0.2.rar <KIT>/sjas-jms.xml
```

You will get the **SPECjAppServer2004/JMS/1.08/rar** diagnostic.

Deploying the main application

To deploy SPECjAppServer2004 on Geronimo, this configuration uses the deployment plan that was originally found in Geronimo sources at http://svn.apache.org, modified and updated for Geronimo version 2.0.2.

Go to the **<GERONIMO>** directory and run:

solid
```
java -jar bin/deployer.jar deploy <SPEC>/jars/SPECjAppServer.ear <KIT>/sjas-app.xml
```

You will get the **SPECjAppServer2004/Application/1.08/ear** diagnostic.

Verifying the deployment

At this stage you may check that the deployment has been done correctly and that SPECjAppServer2004 is operational.

Manual transactions

Open the deployed SPECjAppServer2004 page: **`http://geronimo.host:8080/SPECjAppServer/`**

In the left-hand menu, click the **Go Trade Autos!** or **Go Build Cars!** link.

Log in with the default credentials (**1**) by clicking **Log in**.

You should see the program interface and be able to perform transactions.

Atomicity tests

Open the deployed SPECjAppServer2004 page: **`http://geronimo.host:8080/SPECjAppServer/`**

In the left-hand menu, click the **Atomicity Tests** link.

You will see the results of three atomicity tests' runs. If all three of them are marked as **PASSED**, your deployment is correct.

Back to Top

Deploying the Supplier Emulator

To deploy the SPECjAppServer2004 Supplier Emulator at the **`emulator.host`**, use one of the following:

The Geronimo built-in servlet container (in case the **`emulator.host`** and the **`geronimo.host`** are the same machine); or a stand-alone servlet container on the **`emulator.host`**.

Note that SPECjAppServer2004 documentation requires that the Supplier Emulator servlet container should have the `keep-alive` option turned off. You can ignore this requirement, but that would impact the performance severely.

Using the Geronimo servlet container

Go to the **<GERONIMO>** directory and run:

solid
```
java -jar bin/deployer.jar deploy <SPEC>/jars/Emulator.war <KIT>/sjas-emulator.xml
```

You will get the **SPECjAppServer2004/Emulator/1.08/war @ /Emulator** diagnostic.

Using a stand-alone servlet container

This configuration assumes that your stand-alone servlet container on the **emulator.host** is Tomcat running on the default port (8080).

Install Tomcat to your **<TOMCAT>** directory on the **emulator.host**.

Do not bother editing **<SPEC>**`/config/tomcat.env` file or running `ant/bin/ant -f tomcat.xml`.

Both files are obsolete, they generate the `Emulator.war` file, which has already been created at #Building the application phase.

Copy the **<SPEC>**`/jars/Emulator.war` file to the **<TOMCAT>**`/webapps` directory and remove the **<TOMCAT>**`/webapps/Emulator` directory if it exists.

Go to the **<TOMCAT>** directory on the **emulator.host** and start Tomcat:

solid
```
bin/catalina run
```

Veryfying the deployment

Go to the page **http://emulator.host:8080/Emulator/**. It should load normally and contain a single directory, `dtd`, with two files inside, `delivery.dtd` and `po.dtd`.

Go to the page **http://emulator.host:8080/Emulator/EmulatorServlet**. You should see a page with text like this:

whitesolid
Emulator Servlet seems to work OK
JAS_HOST : emulator.host
JAS_PORT : 8080
Servlet URL : Supplier/DeliveryServlet

Number of Transactions : 0
Servlet invoked without command specified

Back to Top

Running the benchmark

Copy the **<GERONIMO>** directory to the **driver.host**, the copy will be your **<DRIVER_GERONIMO>** directory (in fact you only need some jars from it).

Copy the **<SPEC>** directory to the **driver.host**, the copy will be your **<DRIVER>** directory.

In the **<DRIVER>**`/config/geronimo.env` file adjust the `JAS_HOME` variable to the **<DRIVER>** directory and `J2EE_HOME` variable to the **<DRIVER_GERONIMO>** directory.

Go to the **<DRIVER>** directory on the **driver.host** and run:

solid
```
bin/setenv.bat
```

This configures the environment to run the Driver.

To start the Driver itself, run:

solid
`bin/driver.bat`

If you wish to run a distributed load with multiple Drivers, then after the Driver is started on the first host (the **`master.host`**), start the Driver on other driver hosts like this:

solid
`bin/driver.bat` **`master.host`**

After starting the Driver, you should see the output like this:

This means the Driver started normally.

Note the times for **Rampup**, **SteadyState**, **Rampdown** and **Finish** to figure out the time needed for the benchmark to complete.

You can interrupt the run at any time with `Ctrl-C`.

Sometimes binding exceptions or other problems may occur at the Driver startup. In such a case, interrupt the test run with `Ctrl-C` and rerun it again. Sometimes it helps.

It's recommended to reload the database tables before each run, particularly if previous run was not finished correctly. Otherwise, errors like this may occur:

During the run, the following diagnostics may appear in the Driver window:

and in the same time, various `TransactionRolledback` and other exceptions of the same kind are being printed in the Geronimo shell.

These diagnostics are probably caused by the fact that TranQL version 1.3 is used in Geronimo v2.0.2 does not provide the necessary transaction isolation level. Hopefully, this problem will be fixed in TranQL version 1.3.1.

After the run has completed successfuly, you will see the output like this:

The number of **JOPS** is a final benchmark metric.

For now these values for Geronimo are terribly low, and don't depend on the hardware being used. Probably this is due to some configuration issues that still exist or due to the TranQL issue mentioned above.

Back to Top

Processing results

When driver is run, a subdirectory with a numerical name is created in the **`<OUTPUT>`** directory. The subdirectory with the largest number corresponds to the latest run. After the Driver run is complete, the `result.props` file is created there, it contains the raw results from the benchmark.

Go to the **<DRIVER>**/reporter directory, copy the file Sample_Submission.txt under arbitraty name (e. g. Your_Submission.txt) and edit the copy, as described in SPECjAppServer2004 User's Guide :: Section 5 – Results, add the raw data from the result.props file.

Run the following command to generate an HTML benchmark report, it would be named Your_Submission.report.html:

solid
```
java -cp reporter.jar reporter Your_Submission.txt
```

Run the following command to generate a text-only benchmark report, it would be named Your_Submission.report.txt:

solid
```
java -cp reporter.jar reporter -a Your_Submission.txt
```

For the details on how yo submit your results, see SPECjAppServer2004 User's Guide :: Section 5.3 – Submitting the Results

Back to Top