

# Assembling a server via Console

[← Assembling a server via command line](#)

[↑ Customizing server assemblies](#)

In this tutorial we will illustrate how to assemble a custom server using an application-centric approach from console. Using this approach our application will define on the contents of server. Custom server will be built around our application and will include only the components required by our application. For the application we have used an existing tutorial from tutorial section. Refer [Stateless Session Bean](#) tutorial for application development and deployment. Later we will assemble a custom server which will only include functional components required by our application.

## Clarification

Application client referred in this tutorial is actually a Web Application Client.

- [Identifying the functional components required by our server](#)
- [Assembling a custom server](#)

## Identifying the functional components required by our server

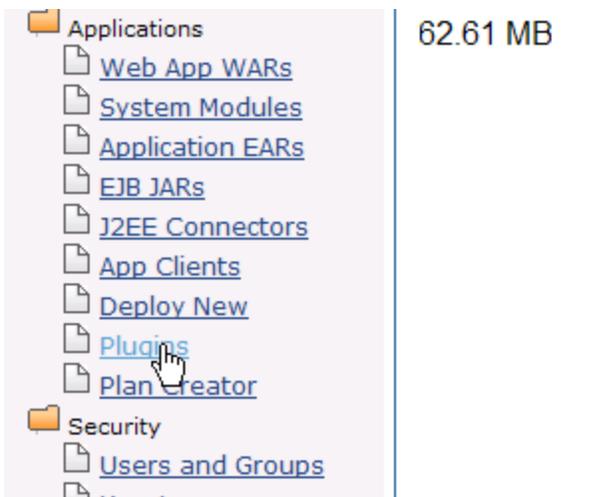
This step is required to identify the various dependencies required by our application. Since we are following an application centric approach we need to find out the various dependencies required by our application.

This is the major work required for our custom server which has been simplified by geronimo administrative console.

1. To assemble a server from existing instance of geronimo we need to include **Geronimo Plugin Group :: Framework** plugin group which is the framework of a functional server and pre-selected by default . Refer [Plugins Group](#) section for more information on **Geronimo Plugin Group :: Framework**.
2. To identify the application plugins to be included, in this tutorial they are:
  - a. Web archive plugin *default/ApplicationClient/1.0/car*
  - b. StatelessSessionEJB plugin *default/StatelessSessionEJB/1.0/car*
  - c. Datasource pool plugin *console/dbpool/jdbc\_users/1.0/car*Information about how to get these plugins, please look into [Converting applications into plugins using the Administration Console](#).
3. To create a new database from console, we need to include the plugin group **Geronimo Plugins, Console :: System Database(Tomcat)**, which includes console interface plugin, embedded Derby database plugin and required resource adapter plugins.  
This completes the identification of all the functional component(dependencies) required by our application centric custom server.

## Assembling a custom server

1. Launch the administrative console and select **Plugins** under **Applications**.



2. Select Assemble a server by choosing Application Centric mode.

**Install Plugins**

### Install Geronimo Plugins

This portlet lets you install Geronimo plugins. This can be used to install new features into a Geronimo server at runtime.

Choose a local or remote repository to inspect for available Geronimo plugins. The repository must have a `geronimo-plugins.xml` file in the root directory listing the available plugins.

You can also download running configurations from another Geronimo server just as if you're browsing and installing third-party plugins. If you want to point to a remote Geronimo administrator username and password in the optional authentication fields.

Repository:

[Update Repository List](#)  
[Add Repository](#)

Optional Authentication: User:  Password:

---

**Create Plugin**

### Create Geronimo Plugin

Choose a configuration in the current Geronimo server to export as a Geronimo plugin. The configuration will be saved as a CAR file to your local filesystem. Note: at present, only one configuration can be exported.

---

**Assemble Server**

### Assemble a server from plugins in this one

Custom Assembly Mode:

- Name the artifacts as suggested: groupId as test, artifactId as TestServer and format as zip or so.

**Assemble Server**

### Name the server to be assembled:

Assembly groupId:

Assembly artifactId:

Assembly version:

Assembly target path:

Assembly format:

- Next step is to select the functional components we identified. Select **Geronimo Plugin Group :: Framework**, WAR, EJB and database pool plugins. You also have to select **Geronimo Plugins, Console :: System Database(Tomcat)** since our application uses derby database and you have to create the database from console. Because application WAR, EJB and database pool plugins are dependent with each other, you are free to choose just one of them and therefore rest of its dependencies will be assembled automatically. You can also choose the Expert mode to select individual plugins instead of plugin groups.
- Final list of plugins/plugin groups we have selected are as follows
  - org.apache.geronimo.pluginGroups/framework/2.2-SNAPSHOT/car
  - org.apache.geronimo.plugins/sysdb-console-tomcat/2.2-SNAPSHOT/car
  - default/StatelessSessionEJB/1.0/car
- Once done select **Assemble**.

## Assemble Server

### Select from plugins in current server:

Expert User (view all system plugins)

Filter by category and name:

### Required plugin in local server:

	Name	Version	Category
<input checked="" type="checkbox"/>	<a href="#">Geronimo Plugin Group :: Framework</a>	2.2-SNAPSHOT	Geronimo Plugin Group

### Application plugins in local server:

	Name	Version	Category
<input checked="" type="checkbox"/>	<a href="#">ApplicationClient</a>	1.0	Unknown
<input type="checkbox"/>	<a href="#">Geronimo Plugins, ActiveMQ v5 :: Console (Tomcat)</a>	2.2-SNAPSHOT	JMS
<input type="checkbox"/>	<a href="#">Geronimo Plugins, ActiveMQ v5 :: Resource Adapter</a>	2.2-SNAPSHOT	JMS
<input type="checkbox"/>	<a href="#">Geronimo Plugins, CA Helper :: Tomcat</a>	2.2-SNAPSHOT	Security
<input type="checkbox"/>	<a href="#">Geronimo Plugins, Console :: Debug Views (Tomcat)</a>	2.2-SNAPSHOT	Administration
<input type="checkbox"/>	<a href="#">Geronimo Plugins, Console :: Monitoring (Tomcat)</a>	2.2-SNAPSHOT	Administration Monitoring
<input type="checkbox"/>	<a href="#">Geronimo Plugins, Console :: Monitoring DataSource</a>	2.2-SNAPSHOT	Administration Monitoring
<input type="checkbox"/>	<a href="#">Geronimo Plugins, Console :: Plan Creator (Tomcat)</a>	2.2-SNAPSHOT	Administration
<input type="checkbox"/>	<a href="#">Geronimo Plugins, Console :: Plugin and Deployment Portlets (Tomcat)</a>	2.2-SNAPSHOT	Administration
<input checked="" type="checkbox"/>	<a href="#">Geronimo Plugins, Console :: System Database (Tomcat)</a>	2.2-SNAPSHOT	Administration
<input type="checkbox"/>	<a href="#">Geronimo Plugins, Console :: Tomcat</a>	2.2-SNAPSHOT	Administration
<input type="checkbox"/>	<a href="#">Geronimo Plugins, Dojo :: Tomcat</a>	2.2-SNAPSHOT	Web Framework
<input type="checkbox"/>	<a href="#">Geronimo Plugins, Dojo Legacy :: Tomcat</a>	2.2-SNAPSHOT	Web Framework
<input type="checkbox"/>	<a href="#">Geronimo Plugins, MEJB :: Config</a>	2.2-SNAPSHOT	EJB
<input type="checkbox"/>	<a href="#">Geronimo Plugins, Monitoring :: Agent</a>	2.2-SNAPSHOT	Monitoring
<input type="checkbox"/>	<a href="#">Geronimo Plugins, Monitoring :: Agent DataSource</a>	2.2-SNAPSHOT	Monitoring
<input type="checkbox"/>	<a href="#">Geronimo Plugins, OpenEJB :: Tomcat</a>	2.2-SNAPSHOT	Administration
<input type="checkbox"/>	<a href="#">Geronimo Plugins, Remote Deploy :: Tomcat</a>	2.2-SNAPSHOT	Deployment
<input type="checkbox"/>	<a href="#">Geronimo Plugins, System Database :: System Database</a>	2.2-SNAPSHOT	Geronimo Core
<input type="checkbox"/>	<a href="#">Geronimo Plugins, UDDI Server :: Tomcat</a>	2.2-SNAPSHOT	Web Services
<input type="checkbox"/>	<a href="#">Geronimo Plugins, Welcome :: Tomcat</a>	2.2-SNAPSHOT	Administration
<input checked="" type="checkbox"/>	<a href="#">StatelessSessionEJB</a>	1.0	Unknown

7. On the next screen each artifact and its associated properties will be displayed. Select **Install**.

### Geronimo Plugins, Console :: System Database (Tomcat)

**Module ID:** org.apache.geronimo.plugins/sydsdb-console-tomcat/2.2-SNAPSHOT/car  
**Category:** Administration  
**Description:** Geronimo Administration Console System Database Portlet Plugin allows users to vi  
**Author:** The Apache Geronimo development community  
**Web Site:** <http://geronimo.apache.org/>  
**License:** The Apache Software License, Version 2.0 (Open Source)

**Geronimo Versions:** 2.2-SNAPSHOT

**JVM Versions:** 1.5

**Dependencies:** org.apache.geronimo.plugins/console-tomcat/2.2-SNAPSHOT/car  
org.apache.geronimo.configs/system-database/2.2-SNAPSHOT/car  
org.apache.geronimo.configs/jsr88-rar-configurer/2.2-SNAPSHOT/car  
org.apache.geronimo.framework/geronimo-gbean-deployer/2.2-SNAPSHOT/car  
org.apache.geronimo.modules/geronimo-converter/2.2-SNAPSHOT/jar  
org.apache.geronimo.framework/plugin/2.2-SNAPSHOT/car  
org.apache.geronimo.modules/geronimo-test-dbean/2.2-SNAPSHOT/jar  
org.tranqil/tranqil-connector-derby-embed-local/1.4/rar  
org.tranqil/tranqil-connector-ra/1.4/rar  
org.tranqil/tranqil-connector-derby-client-local/1.4/rar  
org.tranqil/tranqil-connector-derby-client-xa/1.4/rar  
org.tranqil/tranqil-connector-db2-xa/1.2/rar  
org.tranqil/tranqil-connector-mysql-local/1.2/rar  
org.tranqil/tranqil-connector-mysql-xa/1.2/rar  
org.tranqil/tranqil-connector-oracle-local/1.3/rar  
org.tranqil/tranqil-connector-oracle-xa/1.3/rar  
org.tranqil/tranqil-connector-postgresql-local/1.1/rar  
org.tranqil/tranqil-connector-postgresql-xa/1.1/rar

**Prerequisites:** None

**Obsoletes:** None

**Installable:**

### StatelessSessionEJB

**Module ID:** default/StatelessSessionEJB/1.0/car  
**Category:** Unknown

**Description:**

**Author:**

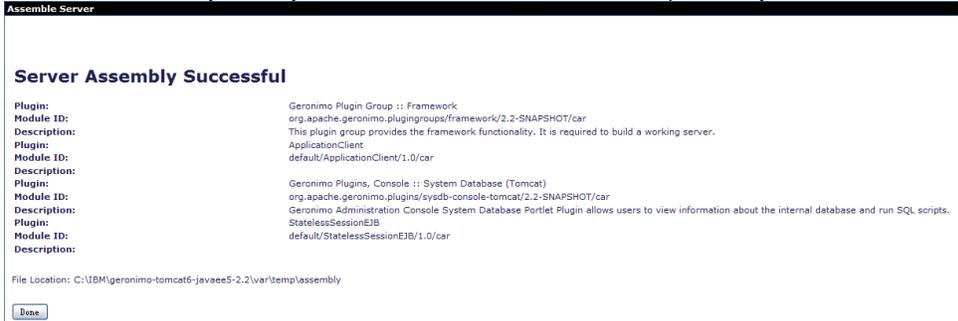
**Web Site:**

**Geronimo Versions:** 2.2-SNAPSHOT

**JVM Versions:** Any

**Dependencies:** console.dbpool/jdbc\_usersd//  
org.apache.geronimo.configs/consanath/car

8. Once done you will receive a confirmation message suggesting the success or failure of server assembly. It will also enlist the location of the custom server assembly which by default is <GERONIMO\_HOME>/var/temp/assembly.



9. The server will be named as `TestServer-1.0-bin.zip`. Go to <GERONIMO\_HOME>/var/temp, you will find the package of newly assembled server with minimizing footprint and the fewest components you need only under a folder named after the groupid of the new server, which is **test** in this case. Unzip the package and start the server.
10. Because our [Stateless Session Bean](#) uses **userdbs**, you need to create database and tables as suggested in the tutorial. This completes our illustration for custom server assembly.