

Locating your application specific configuration files

 [Getting familiar with Geronimo Eclipse Plugin \(GEP\)](#)

 [Getting familiar with the development environment](#)

[Quick Debugging JSPs of your application](#) 

Many applications have configuration files that are intended to be customized by the user and need to be in an easily accessible location. Geronimo is designed so you can create an application specific configuration directory under the `var` directory for this purpose. There are several ways for your application to locate this directory and the files in it. If you deploy your application as a Geronimo plugin you can specify that the original of these files be unpacked from your application plugin when it is installed.

See also the discussion of the system and server locations relevant to [running multiple instances from one installation](#).

Locating the files as files

There are three ways to locate files in a Geronimo installation. They all depend on the *ServerInfo* gbean which is intended to be the universal source of location information for the server.

Direct use of ServerInfo

You can specify a gbean-ref to **ServerInfo** in your geronimo plan:

SNIPPET NEEDED

Look it up in code:

```
ServerInfo serverInfo = (ServerInfo)new InitialContext.lookup("java:comp/env/ServerInfo");
File myconfig = serverInfo.resolveServer("var/myApp/myConfig.xml");
```

Use of geronimo system properties

The **ServerInfo** gbean sets up two system properties that indicate where the geronimo system and server is located.

```
public static final String SERVER_DIR_SYS_PROP = "org.apache.geronimo.server.dir";
public static final String HOME_DIR_SYS_PROP = "org.apache.geronimo.home.dir";
```

Your program can use these to construct the location of your files:

```
String serverDir = System.getProperty(SERVER_DIR_SYS_PROP);
URI serverUri = URI.create(serverDir);
URI configFileURI = serverUri.resolve("var/myApp/myConfig.xml");
File configFile = new File(configFileURI.getPath());
```

Installing app-specific system properties

You can include a *SystemPropertiesGBean* in your geronimo plan and set up application specific system properties that can then be used by your program:

```
<gbean name="MyAppSystemProperties" class="org.apache.geronimo.system.properties.SystemProperties">
  <attribute name="systemPathProperties">
    com.myApp.ConfigFile=var/myApp/myConfig.xml
  </attribute>
  <reference name="ServerInfo">
    <name>ServerInfo</name>
  </reference>
</gbean>
```

```
String configFileLocation = System.getProperty("com.myApp.ConfigFile");
File configFile = new File(configFileLocation);
```

Locating the files in the classpath

It is also possible to extend the classloader of your application to include a directory into which you can put your configuration files by including a *Shared Lib* gbean in the geronimo plan for your application:

```
<gbean name="MyAppSharedLib" class="org.apache.geronimo.system.sharedlib.SharedLib">
  <attribute name="classesDirs">var/myApp</attribute>
  <reference name="ServerInfo">
    <name>ServerInfo</name>
  </reference>
</gbean>
```

```
ClassLoader cl = Thread.currentThread().getContextClassLoader();
URL configFileUrl = cl.getResource("myConfig.xml");
```