Table of Contents

We recommend a specific file structure for JVx applications because every JVx projects should use the same structure. This makes it easier to maintain applications as everything is standardized.

Our recommended structure is:

```
<basedir>
 |- rad              (Configuration.getConfigurationDir())
   |- apps           (Configuration.getApplicationsDir())
     |- <appname>
       |- config.xml (Configuration.getApplicationZone(<appname>))
   |- server         (Configuration.getServerDir())
     |- config.xml   (Configuration.getServerZone())
```

The source code and IDE projects should be saved in <appname> like:

```
- <appname>
 |- src.client
 |- src.server
 |- test
 |- libs
   |- client
   |- server
 |- .classpath
 |- .project
 |- config.xml (Configuration.getApplicationZone(<appname>))
```

The problem with the above structure is that you need a lot of directories and the project itself is in the last level. The advantage is that you can manage multiple applications with one server configuration. However, usually you have at least one application.

We introduced a new search strategy for application and server configuration files. Before JVx 2.1, everything was file based. Every config.xml had to be available as physical file. With 2.1 this is not needed.

The new search strategy is as follows.

Application Zone

```
1: Filesystem: <appsdir>/<appname>/config.xml
2: Filesystem: <appsdir>/../config.xml
   Filesystem: <appsdir>/../../config.xml
   Filesystem: <appsdir>/../../../config.xml
   Filesystem: <appsdir>/../../../../config.xml
3: JNDI:       java:/comp/env/jvx/<appname_lowercase>/config
               (String as resource path), InputStream or IVirtualZone
implementation)
4: Classpath:  Use system property: -
Drad.apps.<appname_lowercase>.config=/path/to/config.xml (must be enabled
via system property)
5: Classpath:  /rad/apps/<appname_lowercase>/config.xml (must be enabled via
system property)
```

```
6: Classpath:  /<appname_lowercase>.xml (must be enabled via system
property)
7: Classpath:  /config.xml (must be enabled via system property)
8: EXCEPTION
```

Server Zone

```
1: Filesystem: <serverdir>/config.xml
2: Filesystem: <serverdir>/../server.xml
   Filesystem: <serverdir>/../../server.xml
   Filesystem: <serverdir>/../../../server.xml
   Filesystem: <serverdir>/../../../../server.xml
3: JNDI:       java:/comp/env/jvx/server/config
               (String as resource path), InputStream or IVirtualZone
implementation)
4: Classpath:  /rad/server/config.xml (must be enabled via system property)
5: Classpath:  /server.xml (must be enabled via system property)
6: EMPTY configuration
```

To enable Classpath search strategy, set the system property: `-DConfiguration.search.classpath=true` otherwise it won't work.

Be careful because ApplicationZone and ServerZone have methods for accessing files and directories:

```
public File getDirectory();
public File getFile()
```

Bot methods will return null if config was loaded via JNDI or classpath.

From:
http://doc.sibvisions.com/ - **Documentation**

Permanent link:
http://doc.sibvisions.com/jvx/common/setup/zones

Last update: **2026/04/21 11:23**