

subdirectories of these directories are not changed and such files should not be changed by users. If an application is deployed and undeployed, full synchronization is not necessary to update the instance with the change.

`config`

This directory contains configuration files for the entire domain.

By default, the DAS resynchronizes files that have been modified since the last resynchronization only if the `domain.xml` file in this directory has been modified.

Note – If you add a file to the `config` directory of an instance, the file is deleted when the instance is resynchronized with the DAS. However, any file that you add to the `config` directory of the DAS is *not* deleted when instances and the DAS are resynchronized. By default, any file that you add to the `config` directory of the DAS is not resynchronized. If you require any additional configuration files to be resynchronized, you must specify the files explicitly. For more information, see [“To Resynchronize Additional Configuration Files” on page 110](#).

`config`

`config/config-name`

This directory contains files that are to be shared by all instances that reference the named configuration *config-name*. A *config-name* directory exists for each named configuration in the configuration of the DAS.

Because the *config-name* directory contains the subdirectories `lib` and `docroot`, this directory might be very large. Therefore, by default, only a change to a file or a top-level subdirectory of *config-name* causes the DAS to resynchronize the *config-name* directory.

`config/domain.xml`

This file contains the DAS configuration for the domain to which the instance belongs.

By default, the DAS resynchronizes this file if it has been modified since the last resynchronization.

Note – A change to the `config/domain.xml` file is required to cause the DAS to resynchronize an instance's files. If the `config/domain.xml` file has not changed since the last resynchronization, none of the instance's files is resynchronized, even if some of these files are out of date in the cache.

`docroot`

This directory is the HTTP document root directory. By default, all instances in a domain use the same document root directory. To enable instances to use a different document root directory, a virtual server must be created in which the `docroot` property is set. For more information, see the `create-virtual-server(1)` help page.

The docroot directory might be very large. Therefore, by default, only a change to a file or a subdirectory in the top level of the docroot directory causes the DAS to resynchronize the docroot directory. The DAS checks files in the top level of the docroot directory to ensure that changes to the `index.html` file are detected.

When the DAS resynchronizes the docroot directory, all modified files and subdirectories at any level are copied to the instance.

If a file below a top-level subdirectory is changed without a change to a file in the top-level subdirectory, full synchronization is required.

generated

This directory contains generated files for Java EE applications and modules, for example, EJB stubs, compiled JSP classes, and security policy files. Do not modify the contents of this directory.

This directory is resynchronized when the `applications` directory is resynchronized. Therefore, only directories for applications that are deployed to the instance are resynchronized.

java-web-start

This directory is *not* resynchronized. It is created and populated as required on each instance.

lib

lib/classes

These directories contain common Java class files or JAR archives and ZIP archives for use by applications that are deployed to the entire domain. Typically, these directories contain common JDBC drivers and other utility libraries that are shared by all applications in the domain.

The contents of these directories are loaded by the common class loader. For more information, see “Using the Common Class Loader” in *GlassFish Server Open Source Edition 3.1 Application Development Guide*. The class loader loads the contents of these directories in the following order:

1. `lib/classes`
2. `lib/*.jar`
3. `lib/*.zip`

The `lib` directory also contains the following subdirectories:

`applibs` This directory contains application-specific Java class files or JAR archives and ZIP archives for use by applications that are deployed to the entire domain.

`ext` This directory contains optional packages in JAR archives and ZIP archives for use by applications that are deployed to the entire domain. These archive files are loaded by using Java extension mechanism. For more information, see [Optional Packages - An Overview](#).