

config/build/deploy process

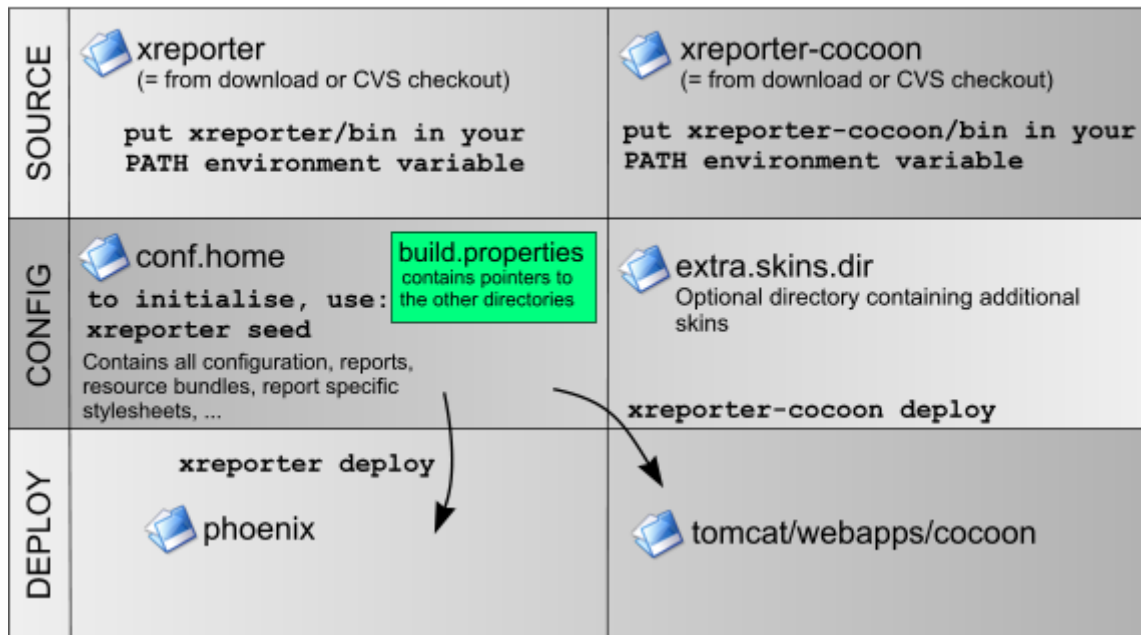
After installation, you may be wondering what all the directories are that were created, and what needs to be done if you change configuration files or add reports. This document should shine some light on this.

NOTE

This document applies to xReporter version 1.2

Overview of the different directories

xReporter deployment structure



There are basically *six different directories* involved:

- The two directories created by extracting the xReporter download package (or checked out from SVN): xreporter and xreporter-cocoon. These directories contain the source files, normally you won't change anything in here except if you are enhancing xReporter itself.
- The "conf.home" directory (the name "conf.home" refers to the property name we use inside the build script, the directory itself can of course have any name you like). You create this directory by creating an empty directory and executing the command "xreporter seed" within it. This directory contains essentially all configuration specific for a certain xReporter installation. You could maintain this directory in CVS. You can have multiple conf.home directories for different xReporter installations.
- The "extra.skins.dir" is an optional directory containing custom skins (again, this is the name of a property, the actual directory will have another name). A "skin" consists of a set of stylesheets, images, ... that will give the xReporter site a customized look. xReporter includes 2 sample skins: a very plain one and a prettier one.
- The phoenix directory contains the Avalon Phoenix application server, it is created by extracting the phoenix download package.

- The tomcat directory contains the servlet container (and webserver) inside which xReporter's publishing layer is deployed. This publishing layer is based on Apache Cocoon, so the default webapplication location is tomcat/webapps/cocoon

Next to the configuration information in the conf.home directory, xReporter also needs an *administrative database*, containing tables with information about users, report access control, and so on. Changes to these administrative tables take immediate effect, thus they do not require a restart of Phoenix or Cocoon.

What is needed at runtime?

Cocoon

After deployment to Tomcat/Cocoon, the Tomcat can be run stand-alone, and doesn't need access to any of the other xReporter directories.

Phoenix

The xReporter application running on Phoenix needs access to the conf.home directory at runtime. It does not require access to any of the other directories.

conf.home details, when to redeploy, when to restart

```

conf.home
|-- build.properties
|-- build.properties.sample
|-- conf
|   |-- authorisationcodes.xml
|   |-- datasources.xml
|   |-- datatypes.xml
|   |-- sequence.txt
|-- lib
|   |-- classes
|   |-- user-extensions.jar
|   |-- xreporter-all.jar
|   |-- xreporter.sar
|-- reports
|   |-- report1.xml
|   |-- report2.xml
|   |-- customlayout
|       |-- fo
|       |-- html
|       |-- messages_en.xml
|-- resourcebundles
|   |-- DataSourceMessages_en.properties
|   |-- ReportMessages_en.properties
|-- sarconf
|   |-- assembly.xml
|   |-- config.xml
|   |-- environment.xml
|-- src
|   |-- java

```

The central configuration file is config.xml. That file contains pointers to the other files and directories, and also contains the connection parameters for the administrative tables (and some other, various configuration parameters). By editing the config.xml file, you can even bypass the default configuration directory structure (though we do not recommend that).

When you change any of the files in the conf directory, you need to restart Phoenix for changes to take effect. If you change any files in the sarconf directory, you need to redeploy and restart Phoenix. If you change any of the files in the src directory, you need to recreate the user-extensions.jar file by executing "xreporter user-extensions.jar" (this will also happen if you execute "xreporter deploy") and restart Phoenix.

Changes to the files in the reports (but not customlayout) and resourcebundles directories do not require a restart of xReporter, xReporter will periodically scan those directories for changes (by default respectively every 5 and 10 seconds, but you can change these intervals in the config.xml).

If you change any of the files in the reports/customlayout directory, you need to redeploy to Cocoon, but you do not need to restart Cocoon.

Deploying to Phoenix

You need to redeploy to Phoenix each time you change something in the conf.home/sarconf directory. If you change any other files, you do NOT need to redeploy.

Redeploying to Phoenix can be done directly from the conf.home directory. First check the values of the properties in the build.properties file, and then execute the following command:

```
xreporter deploy
```

After redeploying, you need to restart Phoenix.

Deploying to Cocoon

You need to redeploy to Cocoon each time you:

- change something to a skin
- change files below the directory conf.home/reports/customlayout

You do NOT need to redeploy to Cocoon when you change any other files such as reports, etc.

Redeploying to Cocoon can be done directly from the conf.home directory. First check the values of the properties in the build.properties file, and then execute the following command:

```
xreporter-cocoon deploy
```

WARNING

Executing this command will delete the contents of the cocoon/mount/xreporter directory. Normally you shouldn't place files there yourself, so this is no problem.

After deploying, you will usually not have to restart Cocoon. You will need to restart Cocoon if you changed any of the XML resource bundles, but not if you changed XSL's or report parameterisation files.

What needs to change when deploying on another system

The Tomcat/Cocoon installation does not contain any absolute path references and can thus be moved freely between systems. If you deploy Tomcat/Cocoon on another system than Phoenix, then you'll have to change the servername in the host element in the cocoon.xconf:

```
<component role="org.outerj.xreporter.client.XReporterClient"  
  class="org.outerj.xreporter.client.XReporterClientImpl">
```

```
<host>myotherserver.com</host>
<port>8888</port>
</component>
```

The Phoenix installation contains absolute pointers to other configuration directories in the config.xml (contained inside the xreporter.sar file). Thus if you move the conf.home and phoenix directories to another system or location, you will need to redeploy to Phoenix. On deployment, the current location of the conf.home directory will be dynamically substituted in the config.xml file.

Invoking the Ant build files directly

The scripts xreporter(.bat) and xreporter-cocoon(.bat) are only launch scripts for Ant build files. What they do is:

- set the classpath
- set the basedir for Ant (should always be the xreporter or xreporter-cocoon directory, respectively)
- find out what the current directory is, and supply that as the value of the "conf.home" property

If you'd like to run these scripts from your IDE you will need to configure your IDE to use the same classpath for Ant as specified in the launch script. Depending on the target you'd like to execute, you will also need to supply a conf.home property pointing to your xReporter configuration directory (not required for a simple compile, but required for creating the sar or deploying).

Additionally, for the xreporter build you need to define a property " phoenix.home" to be able to execute the deploy. For the xreporter-cocoon build you need to define a property "cocoon.home" (also required to compile because we use jars from Cocoon's webapp directly in the compiler classpath). Optionally you can define the property extra.skins.dir, see the document on [skins](#)¹ for its purpose.

1. daisy:95-cd (Skinning)