

# BuildSystem

**Note:** the content of this page is moved to [28.html](#). Updates to the content will be done there. This page will no longer be reviewed for updates and will be removed in due time.

---

## The Cocoon 2.1 Build System

Cocoon's build system and options can be daunting for a first time user, but you do not need to delve deeply into them to get a usable starting point for your own application. If you are just starting out, it's recommended that you build the default build with everything enabled to get a feel for what Cocoon is and can do. After that, come back here and follow the steps below.

### Build properties.

As explained in `INSTALL.txt`, copying `build.properties` and `blocks.properties` to `local.*.properties` is a recommended first step when you're ready to start configuring Cocoon to meet your needs. Editing those files gives you a good amount of control over what is and is not included or enabled in the Cocoon framework you'll assemble your application around.

Most user-oriented options are simple `true/false/omitted` settings but some have values that can be useful to set. For explanations of the options, see [Build Properties](#).

### Preliminary block support has been added.

Read the new `INSTALL.txt` in the root directory (which should get copied here, and to the "official docs"). Here are some more details about how the blocks build works:

Blocks are under

```
src/blocks/
```

The code is in

```
src/blocks/(blockname)/java
```

The configuration `x*` files (which patch `cocoon.roles`, `cocoon.xconf`, `logkit.xconf`, `web.xml`, `sitemap.xmap`, and the `block-samples.xml` file as needed) are in:

```
src/blocks/(blockname)/conf
```

### How is it done?

`module.xml` has been checked in the cocoon root; it's the [Gump](#) descriptor. Each block must have a `<project>` section with dependencies, and other info. See supplied `module.xml` and the [Gump](#) project for details.

Dependencies without the version attribute are intended to be compiled by the build system. Otherwise, there is `version="supplied"`.

The compile target has been changed. It calls a stylesheet in

```
tools/src/build-blocks.xsl
```

on the `module.xml` file and creates an Ant file that can resolve the dependencies between docs, call the main `compile-core` target, and compile each block.

The blocks are built in

```
build/cocoon/blocks/(blockname)
```

Each block will result in a `(blockname)-block.jar` in the build dir.

The confs are copied in the

```
build/cocoon/blocks
```

dir and then all inserted in the relevant descriptors during the `webapp` target.

### How do I disable the compilation of a specific block?

In `blocks.properties`, the new file from which Ant takes properties, you find the possible exclude properties. The build first looks for a `local.blocks.properties` and you should copy the original over to that name, and edit your private copy. Uncomment the excludes for blocks you don't want to compile and run the build.

```
build clean
and then
build
```

To create a .war do:

```
build war
```

*I'm sure that I forgot something ;-P , please ask on the cocoon-dev mailing list.*

---

[CategoryInDaisy](#)