

Frequently Asked Questions

Table of contents

1 Questions.....	2
1.1 1. Getting Started and Building Forrest.....	2
1.2 2. Technical.....	3
1.3 3. Older versions.....	7
1.4 4. General.....	8

Questions

1. Getting Started and Building Forrest

1.1. Where can i read an overview about how to work with Forrest?

See the [Using Forrest](#) guide.

1.2. Which version of Forrest does the website documentation relate to?

The documentation on the Forrest website relates to the current head development version of the source repository. See [further explanation](#).

1.3. What are the system requirements for Forrest?

Forrest includes everything necessary to build and run, except of course for Java. In addition to all the Cocoon JARs, Forrest includes and uses its own version of Ant. The [Cocoon requirements](#) indicate that Java 1.3+ is required.

1.4. The old xml-forrest CVS code repository seems to be stale. What happened?

Forrest switched from a CVS code repository to SVN (subversion) code repository. The old CVS repository is not kept current.

1.5. How can I use SVN to keep up to date with the latest codebase?

Follow these [Building Forrest](#) notes.

The [Using Forrest](#) guide provides further step-by-step assistance in getting started with Forrest for your project.

1.6. When running ./build.sh in cygwin, I get an error: cygpath.exe: * can't create title mutex, Win32 error 6.**

This [appears to be a bug in cygwin](#). Please use the .bat script instead.

1.7. How can I specify the amount of memory to be used by Java?

There are two ways to control this. If you get an OutOfMemoryError when Cocoon is generating pages, see the first paragraph. If you get an OutOfMemoryError when outside of Cocoon (e.g., copying raw files), see the second paragraph.

Frequently Asked Questions

The `maxmemory` property in the `forrest.properties` file controls how much memory Cocoon uses. Like many other properties you can copy them from the default configuration at `src/core/fresh-site/forrest.properties`

Set the `ANT_OPTS` environment variable before you run `forrest`. The exact value you set it to is dependant on your JVM, but something like `ANT_OPTS=-Xmx500M` will probably work.

2. Technical

2.1. How can I generate one pdf-file out of the whole site or selected pages of the site?

Add the following entries to your `site.xml` file:

```
<about tab="home" label="About" href="">
  ...
  <all_site label="Full HTML" href="wholesite.html"/>
  <all_sitePDF label="Full PDF" href="wholesite.pdf"/>
  ...
</about>
```

In this case the menu labeled "About" will have 2 new items: "Full PDF" and "Full HTML". (See also [How to create a PDF document for each tab.](#))

This assumes that you use the [site.xml](#) method for your site structure and navigation, rather than the old `book.xml` method.

2.2. How can I generate html-pages to show a 'clickable' email-address (of the author-element)?

You would override `src/core/context/skins/common/xslt/html/document2html.xsl` and edit the "headers/authors" template.

2.3. How can I generate html-pages to show the revision tag of cvs?

If you have: `<version>$Revision: 1.30 $</version>` The '1.30' will be extracted and displayed at the bottom of the page as "version 1.30". See for example the bottom of the [Using Forrest](#) document.

This technique could also be used for a modification date with `$Date: 2004/01/15 08:52:47 $`

2.4. How do I stop Forrest breaking on links to external files that may not exist, like javadocs?

This can be done by overriding the `cli.xconf` config file, and defining patterns for URLs to exclude.

This means creating a directory `src/documentation/conf` (or wherever `${forrest.conf-dir}` points) and copying `$FORREST_HOME/context/WEB-INF/cli.xconf` to it. Then edit `cli.xconf`, and add any exclude sections you require at the end. The default `cli.xconf` ignores directory links and links containing 'apidocs' or starting with 'api':

```
....
<!-- Includes and excludes can be used to limit which URLs are rendered
-->

<exclude pattern="**/">
<exclude pattern="**apidocs**">
<exclude pattern="api/**">

<uri src="favicon.ico"/>
</cocoon>
```

This is just an example, and you should modify it appropriately for your site.

Note:

Wildcards may be used. These are a powerful feature of Cocoon's [sitemap](#). For example, `foo/*` would match `foo/bar`, but not `foo/bar/baz` — use `foo/**` to match that.

2.5. How do I link to raw files such as `config.txt` and `brochure.pdf`?

Place them in the `src/documentation/content` directory and they will get copied into the output tree where you can link to them. You can also have sub-directories there to reflect your xdocs tree. See the samples documents when you 'forrest seed' a new project for a demonstration of this ability.

For example, if `src/documentation/content/xdocs/tools/downloads.xml` has a `<link href="tool.zip">` then put `tool.zip` in the `src/documentation/content/tools/` directory.

See the explanation and demonstration of "linking" in your local 'forrest seed' site.

2.6. Some of my files are not being processed because they use common filenames.

Certain patterns are claimed by the default sitemaps for special processing. These include: `site`, `changes`, `todo`, `faq`, `images`, `my-images`, `skinconf`, `community`, `howto`

Frequently Asked Questions

Sometimes there are workarounds, e.g. perhaps "my_site". In future versions of Forrest we will attempt to deal with this issue.

2.7. What do the symbols and numbers mean when Forrest lists each document that it has built?

```
* [56/0]      6.281s 23.0Kb index.html
* [0/0]      0.0060s 4.0Kb  images/project-logo.gif
^
* [50/0]      1.582s 18.7Kb todo.html
X [0]        brokenlink.html      BROKEN: reason
* [50/0]      1.222s 20.2Kb dreams.html
* [0/0]      0.535s 11.1Kb dreams.pdf
...
```

Column 1 is the page build status (*=okay X=brokenLink ^=pageSkipped). Column 2 is the number of links that were gathered from that page. Column 3 is the time taken. Column 4 is the page size.

2.8. Images don't display in PDFs. How do I fix this?

Forrest uses [Apache FOP](#) for rendering PDFs. FOP cannot handle all image types natively, and requires third-party jars to be added. FOP natively handles BMP, GIF, JPG, TIFF and EPS (with a few limitations). FOP can also handle SVG (via Batik!) and PNG (see below). For details, see [FOP Graphics formats](#)

To get PNGs working in PDFs with Jimi:

1. Download Jimi from <http://java.sun.com/products/jimi/>
2. Unpack the Jimi distribution and copy JimiProClasses.zip to
\$FORREST/lib/optional/jimi-1.0.jar.

Alternatively you can use JAI (Java Advanced Imaging API at <http://java.sun.com/products/java-media/jai>). For more info, see [FOP Graphics Packages](#)

Note:

Due to Sun's licensing, we cannot redistribute Jimi or JAI with Forrest.

2.9. The tab link in my site incorrectly assumes that 'index.html' is present in the linked-to directory. How do I fix this?

In tabs.xml, use @href instead of @dir, and omit the trailing '/'. Which file to serve is then a concern of the sitemap. For example, if the "User Manual" tab should link to manual/Introduction.html then tabs.xml should contain:

```
<tab label="User Manual" href="manual"/>
```

and add this rule to the sitemap:

```
<map:match pattern="manual">
  <map:redirect-to uri="manual/Introduction.html"/>
</map:match>
```

2.10. When generating PNG images from SVG, I get an error: Can't connect to X11 window server using ':0.0' as the value of the DISPLAY variable.

If you are using JDK 1.4.0 or newer, you can enable *headless* operation by running Forrest with the `forrest.jvmarg` parameter set to `-Djava.awt.headless=true`, like this:

```
forrest -Dforrest.jvmargs=-Djava.awt.headless=true site
```

See also [Cocoon FAQ](#).

2.11. How do i configure my favourite XML editor or parser to find the local Forrest DTDs?

Notes are provided for various tools at [Using Catalog Entity Resolver for local DTDs](#).

2.12. How to use special characters in the labels of the site.xml file?

Use the numeric values for character entities. For example, rather than using `ö` ; use `ö` ;

See the [XHTML Character Entities](#) and see more discussion at [Issue FOR-244](#).

2.13. How to make the site look better and change its skin?

There are [default skins](#) provided, which are configurable and so should meet the needs of most projects. The aim is to provide many capabilities so that extra skins are not needed.

See notes about [configuration](#) of the skins. Some projects may have special needs and can define their [own skin](#).

2.14. How do I enable XSP processing?

First consider whether your needs would be better met by Cocoon itself, rather than Forrest.

That said, there are valid reasons for wanting programmatically generated content, so here is how to enable XSP:

1. Download [jdtcore-2.1.0.jar](#), and copy it to the `$FORREST_HOME/context/WEB-INF/lib`

Frequently Asked Questions

directory (or lib/core/ directory in the source distribution).

2. Override sitemap.xmap in your local project (copy \$FORREST_HOME/context/sitemap.xmap to src/documentation/), and add the following generator definition in the map:generators section:

```
<map:generator name="serverpages"
  pool-grow="2" pool-max="32" pool-min="4"
  src="org.apache.cocoon.generation.ServerPagesGenerator"/>
```

3. Decide how you want to use XSP. For single files, you could just define a *.xml matcher:

```
<map:match pattern="dynamic.xml">
  <map:generate src="content/xdocs/dynamic.xsp" type="serverpages"/>
  ...
  <map:serialize type="xml"/>
</map:match>
```

You may instead wish to override forrest.xmap to define a general mapping for XSPs.

See also the [AddingXSPToForrest](#) Wiki page.

2.15. How do breadcrumbs work? Why don't they work locally?

Breadcrumbs begin with up to three URLs specified in skinconf.xml. Here is what the Forrest site uses:

```
<trail>
  <link1 name="apache" href="http://www.apache.org/" />
  <link2 name="xml.apache" href="http://xml.apache.org/" />
  <link3 name="" href="" />
</trail>
```

If any links are blank, they are not used. After these first links, JavaScript looks at the URL for the current page and makes a link for each directory after the domain. If you are viewing the site locally, there is no domain and so there will be no extra breadcrumbs, only the ones that are specified in skinconf.xml.

3. Older versions

3.1. When invoking Forrest 0.4 from the forrest.antproxy.xml, the build fails because \${project.home} isn't resolved.

This is a bug in 0.4. The following modification to FORREST_HOME/forrest.antproxy.xml should fix it:

```
--- forrest.antproxy.xml      7 Feb 2003 22:21:15 -0000      1.2
+++ forrest.antproxy.xml      22 Feb 2003 09:39:42 -0000
@@ -41,6 +41,7 @@
```

```
</path>
<target name="forrest">
+   <property name="project.home" location="."/>
   <java classname="org.apache.tools.ant.Main" fork="true">
     <classpath refid="forrest-classpath"/>
     <jvmarg value="-Dforrest.home=${forrest.home}"/>
```

3.2. After upgrading to 0.4 my HTML looks significantly different, and table widths are wrong. What happened?

Forrest now uses a HTML 4.0.1 DOCTYPE declaration:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

This causes browsers to render the page in standards-compliance mode, or rather, not in "quirks" mode. For some operations (like determining table column widths), quirks mode results in better looking pages. For the old pre-0.4 behaviour, create a custom sitemap and remove the line:

```
<doctype-system>http://www.w3.org/TR/html4/loose.dtd</doctype-system>
```

3.3. When building my project, I get an validation error: Document root element "site", must match DOCTYPE root "null"..

You are probably trying to build the project with an old version of Forrest (built before 2003-01-08) that is incorrectly trying to validate the `site.xml` file. If so, please update your Forrest installation.

Alternatively, you may be building with an up-to-date Forrest, but have overridden `forrest.validate.xdocs.excludes` in `forrest.properties`. With the introduction of `site.xml`, the above property must have `site.xml` listed to prevent an attempt at DTD-based validation.

3.4. When building my project, I get any of these errors: menu.xmap (The system cannot find the file specified) static.xmap (The system cannot find the file specified) Type 'xml-document' is not defined for 'serialize'

You are using an old version of `sitemap.xmap` (see the document [Upgrading to Forrest 0.5](#)).

4. General

4.1. What is the relationship between `site.xml` and `book.xml`?

Frequently Asked Questions

One `site.xml` file in your project root can replace all the `book.xml` files (one per directory) in your site. Internally, Forrest uses `site.xml` to dynamically generate `book.xml` files. However, Forrest first checks for the existence of a `book.xml` file, so backwards-compatibility is preserved. If a directory has a `book.xml` file, the `book.xml` will be used to generate the menu. This supplement is useful in situations where `site.xml`-generated menus aren't appropriate. See [Menus and Linking](#).

4.2. How do I use DocBook as the xml documentation format?

There are two ways. Forrest can transform the DocBook format into the Forrest "xdocs" format on-the-fly and then render that as normal Forrest documents. Be aware that the stylesheet that does this transformation is deliberately very limited and does not attempt to deal with all DocBook elements.

The other way is to use the full DocBook stylesheets directly. The DocBook DTDs are shipped with Forrest and automatically handled. However, you will need to have the DocBook stylesheets on your system (they are too massive to ship with Forrest) and configure Forrest accordingly. You will need to create a [project sitemap](#) as explained in [Using Forrest](#) and add matches to handle your DocBook documents. Here is an example. Note that you need to change it to suit your situation. The match must be very specific so that only the DocBook documents are matched. The rest of the documents will be handled by Forrest core. Powerful regex capabilities are available.

```
<?xml version="1.0"?>
<map:sitemap xmlns:map="http://apache.org/cocoon/sitemap/1.0">
  <map:pipelines>
    <map:pipeline>
      <map:match pattern="resolver-*.html">
        <map:generate src="{project:content.xdocs}resolver-{1}.xml"/>
        <map:transform
src="file:///usr/share/sgml/docbook/xsl-stylesheets/xhtml/docbook.xsl"/>
        <map:serialize type="xhtml"/>
      </map:match>
    </map:pipeline>
  </map:pipelines>
</map:sitemap>
```

You can also use a mixture of the two methods, some handled automatically by Forrest and some directly using DocBook stylesheets. You can also have a mixture of source files as "document-v*" DTD and DocBook.

Ensure that the document type declaration in your xml instance is well specified. Use a public identifier. The DTD will then be properly resolved by Forrest. If you need to use different DTDs, then see [Using Forrest](#) for configuration guidance.

4.3. How to report which version of Forrest is being used and the properties that are set?

Do `'forrest -projecthelp'` or `'./build.sh'` to find the version number.

To list the properties, add `"forrest.echo=true"` to your `forrest.properties` file and watch the build messages. Doing `'forrest -v'` will provide verbose build messages.

4.4. Where are the log files to find more information about errors?

The logfiles are at `build/webapp/WEB-INF/logs/`

The log level can be raised with the `logkit.xconf` configuration. If you are using Forrest in the interactive webapp mode (which is generally easiest for debugging errors) then see the `build/webapp/WEB-INF/logkit.xconf` file. If you are generating a static site (with command-line `'forrest'`) then copy

`$FORREST_HOME/context/WEB-INF/logkit.xconf` to your project at `src/documentation/content/conf/logkit.xconf` and modify it. See more information and efficiency tips with [Cocoon logging](#).

Doing `'forrest -v'` will provide verbose build messages to the standard output.

4.5. How to help?

Join one of the Forrest project [mailing lists](#) and tell us what you would like to see improved. We regard all feedback as valuable, particularly from newcomers—often, close proximity blinds software developers to faults that are obvious to everyone else. Don't be shy!

4.6. How to contribute a patch?

Please send all contributions via our [issue tracker](#). Here are notes about [making patches](#).

More info about contributing can be found at the [Contributing to Forrest](#) page. It is always a good idea to check the Forrest [issue tracker](#) and [to do list](#) before diving in.