Frequently Asked Questions

Table of contents

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. 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. Java 1.4+ is required.

1.3. 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.

1.4. 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.

1.5. What is the best way to generate "standalone documents" using Forrest?

forrest site -Dproject.start-uri=myfile.pdf

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.

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. Content Creation

2.1. What tools can be used to edit the content?

If you are using the Apache Forrest xml document format or DocBook or other xml document types, then you can use any text editor or even a dedicated xml editor. You must ensure valid xml. See our configuration notes for various editors.

There are content management systems like Apache Lenya.

Remember that Forrest can also use other source formats, such as OpenOffice.org docs or JSPWiki. Use the appropriate editor for those document types and ensure that the document stucture is consistent. Forrest can also use "html" as the source format, in which case you can use text editors or "html editors" such as the one provided with the Mozilla web browser.

2.2. 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.

2.3. How do I insert page breaks into documents?

Page breaks do not make a great deal of sense in HTML documents intended for display on a screen. However, PDF documents are intended for printing and therefore page breaks can be important.

To insert a page break in a PDF document simply add *pageBreakBefore* or *pageBreakAfter* to the class attribute of a section.

If you want these classes to be processed in your HTML documents as well you should add the following to the `extra-css` element in your projects `skinconf.xml`

```
.pageBreakBefore {
  margin-bottom: 0;
  page-break-before: always;
}

.pageBreakAfter {
  margin-bottom: 0;
  page-break-after: always;
}
```

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

You would override `$FORREST_HOME/main/webapp/skins/common/xslt/html/document2html.xsl` and edit the "headers/authors" template.

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. 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.

2.7. 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>
```

2.8. 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 `&ouml;` use `&#246;`

See the XHTML Character Entities and see more discussion at Issue FOR-244.

2.9. Does Forrest handle accents for non-English languages?

Yes, Forrest can process text in any language, so you can include:

- accents: á é í ó ú
- diereses: ä ë ï ö ü
- tildes: ã ñ # õ #

This is because sources for Forrest docs are xml documents, which can include any of these, provided the encoding declared by the xml doc matches the actual encoding used in the file. For example if you declare the default encoding:

```
<?xml version="1.0" encoding="UTF-8"?>
```

but the file content is actually using ISO-8859-1 then you will receive validation errors, especially if you include some non-ASCII characters.

This situation is commonly encountered when you edit the templates created by `forrest seed` with your favorite (probably localized) editor without paying attention to the encoding, or when you create a new file and simply copy the headers from another file.

Although UTF-8 is an encoding well-suited for most languages, it is not usually the default in popular editors or systems. In UNIX-like systems, most popular editors can handle different encodings to write the file in disk. With some editors the encoding of the file is preserved, while with others the default is used regardless of the original encoding. In most cases the encoding used to write files can be controlled by setting the environment variable `LANG` to an appropriate value, for instance:

```
[localhost]$ export LANG=en_US.UTF-8
```

Of course the *appropriate* way to set the encoding depends on the editor/OS, but ultimately relys on the user preferences. So you can use the encoding you prefer, as long as the `encoding` attribute of the xml declaration matches the actual encoding of the file. This means that if you are not willing to abandon ISO-8859-1 you can always use the following declaration instead:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

Another option is to use "character entities" such as `&ouml;` (ö) or the numeric form `&#246;` (ö).

Another related issue is that your webserver needs to send http headers with the matching charset definitions to the html page.

Here are some references which explain further: GT2004 presentation by Torsten Schlabach and Alan Wood's Unicode resources.

3. Technical

3.1. I'm behind a proxy and it's preventing Plugins from being downloaded, what should I do?

You can configure the proxy in the `forrest.properties` file. Set the `proxy.host` and `proxy.port` accordingly (the port will default to port 80).

3.2. 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 $

3.3. 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/main/webapp/WEB-INF/cli.xconf` to it. Declare the location of this file in the forrest.properties configuration, e.g. `project.configfile=${project.home}/src/documentation/conf/cli.xconf`

Then edit cli.xconf, and add any exclude sections that you require. 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.

3.4. 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`

Sometimes there are workarounds, e.g. faq.html or faq-interview.html would fail, but interview-faq.html would be fine. In future versions of Forrest we will attempt to deal with this issue (FOR-217).

3.5. 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
^                           apidocs/index.html
* [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.

3.6. 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.

3.7. 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.

3.8. 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.

3.9. 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/main/webapp/WEB-INF/lib directory (or lib/core/ directory in the source distribution).
2. Add the following generator definition in the map:generators section of your project sitemap

```
  <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.

3.10. 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`.

3.11. How do I make forrest run listen on a different port?

```
forrest run -Dforrest.jvmargs="-Djetty.port=80"
```

Or copy Forrest's main/webapp/jettyconf.xml file to your project's src/documentation directory and set the port number in that file. Then do `forrest run`

4. Older version: 0.6

4.1. 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`

Sometimes there are workarounds, e.g. faq.html or faq-interview.html would fail, but interview-faq.html would be fine. In future versions of Forrest we will attempt to deal with this issue (FOR-217).

5. General

5.1. What is the relationship between site.xml and book.xml?

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.

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

There are two ways. Forrest has a `simplified-docbook` plugin which 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.

5.3. 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.

5.4. Where are the log files to find more infomation 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/main/webapp/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.

5.5. 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!

5.6. 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 before diving in.