

Upgrading to Apache Forrest 0.7

Table of contents

1 Introduction.....	2
2 New Features.....	2
3 Run a clean target after upgrade.....	2
4 New location of \$FORREST_HOME.....	2
5 Java 1.4 (or newer) is required.....	2
6 General upgrade tips.....	2
7 Plugin architecture.....	2
8 Configure plugins.....	3
9 Including raw un-processed content.....	3
10 To be continued.....	3

1. Introduction

This page describes some changes to Apache Forrest that affect people who are upgrading to the 0.7 version. If you have other issues, then please discuss on either the dev or user mailing lists. As more experience is gained, this document will be updated.

(If you are upgrading from a version prior to 0.6 then you will need to see the notes for the previous upgrade.)

2. New Features

The following list shows some of the key new features (for the full list of changes, see the change log).

- Plugin architecture
- Java 1.4

As usual, do a "forrest seed site" in a new directory and compare the forrest.properties and skinconf.xml with that of your project.

3. Run a clean target after upgrade

To avoid any issue with old classes being loaded, run a 'forrest clean' in your project directory, after you upgraded to this version.

4. New location of \$FORREST_HOME

\$FORREST_HOME is now the top-level of the distribution. Also make sure \$PATH gets updated to use the new \$FORREST_HOME/bin

5. Java 1.4 (or newer) is required

Java 1.4 (or newer) is required, starting with this Forrest 0.7 version. For further information, see FAQ.

6. General upgrade tips

Synchronise your project's skinconf.xml and forrest.properties files.

Take advantage of the separation of concerns. In a new workspace, create a fresh 'forrest seed' site, then tweak its forrest.properties and skinconf.xml until it reflects your old site. When it is ready, replace your project's skinconf.xml and forrest.properties files. Any remaining issues would concern other aspects of your configuration, such as site.xml and your actual content.

7. Plugin architecture

See Plugin Infrastructure and Extending Forrest with Plugins and for developing new plugins see How to Build a Plugin. See the list of current plugins and their documentation.

Note that other experimental plugins can be found in the "whiteboard/plugins" directory.

8. Configure plugins

Some functionality has been moved out of the forrest core and into plugins. You will need to declare any plugins that are used by your project, e.g. if you use projectInfo (status, changes, todo) and PDF output, then declare the following in forrest.properties

```
project.required.plugins=org.apache.forrest.plugin.input.projectInfo,org.apache.forrest.plugin.output.pdf
```

9. Including raw un-processed content

The method for including "raw un-processed content" has changed.

In 0.6 version, the raw content was placed in the src/documentation/content/ directory and potential sub-directories. In the generated site, these links would automatically function. Any linked file with .html extension was not processed and not adorned with Forrest skin and navigation menus.

It was also possible to place HTML content in the xdocs directory where it would be copied across if it was linked from the main content.

In 0.7 version, any file that is linked to, needs to be placed in the content/xdocs/ directory structure. Any linked file with .html extension is now processed and is adorned with Forrest skin and navigation menus.

If you wish to emulate the behaviour of 0.6 and earlier then you must add the following to your project sitemap.

```
<map:match pattern="**/*.html">
  <map:select type="exists">
    <map:when test="{project:content}/{0}">
      <map:read src="{project:content}/{0}" mime-type="text/html"/>
      <!--
        Use this instead if you want JTTidy to clean up your HTML
        <map:generate type="html" src="{project:content}/{0}" />
        <map:serialize type="html"/>
      -->
    </map:when>
    <map:when test="{project:content.xdocs}/{0}">
      <map:read src="{project:content.xdocs}/{0}" mime-type="text/html"/>
      <!--
        Use this instead if you want JTTidy to clean up your HTML
        <map:generate type="html" src="{project:content.xdocs}/{0}" />
        <map:serialize type="html"/>
      -->
    </map:when>
  </map:select>
</map:match>
```

If you need to include files that are not linked to, then place them in the src/documentation/content/ directories as with the 0.6 version.

10. To be continued...

...as more issues are discovered/remembered :) Please send feedback to the mailing list.