How to deploy documentation with the Forrestbot "scp" workstage

This How-To describes the building and deployment of a documentation set with the help of the forrestbot "scp" workstage.

Table of contents

1 Intended Audience

Anyone who generates a static documentation set with Forrest will need to deploy the results, whether that be to a remote server or locally.

2 Purpose

This howto will explain one method of using the Forrestbot, by way of the worked example for managing a company or private website, and using "Secure Copy (SCP)" to deploy to a remote webserver. The Forrestbot has a number of deployment methods. After you understand the principles then you will be able to apply that to whatever method is relevant.

The secondary purpose is to explain to ASF committers how to manage project documentation, e.g. for an "Apache Labs" investigation.

This howto also explains the use of the Apache Cocoon "checksums" facility to only process and deploy the changed files.

3 Prerequisites

Refer to the Forrestbot documentation. It is not necessary to have thorough knowledge, but a basic understanding will help.

An account on a server configured to use SSH access. e.g. ASF Guide for new committers.

Review the Forrestbot svn howto documentation. That explains more generally how to use a Forrestbot.

Therefore this "scp" howto can have minimal content.

4 Steps

4.1 Introduction

This Forrestbot uses the "getsrc.local" workstage to retrieve the sources from the local project workspace filesystem, i.e. from the top-level of your Forrest project site.

This Forrestbot uses the "deploy.scp" workstage to copy the generated documents to the remote server. They go into a "stage" directory on the web server which has a .htaccess file to enable selected people to review before putting into production.

4.2 The deploy.scp.settings file

Create a file at the top-level of your project site named `deploy.scp.settings` to provide your ssh credentials. Set its file permissions so that only you can read it.

```xml
<?xml version="1.0"?>
<project>
  <property name="deploy.scp.keyfile" value="/Users/me/.ssh/id_rsa"/>
  <property name="deploy.scp.passphrase" value="#$!*%#..."/>
</project>
```

4.3 The Forrestbot buildfile

The Forrestbot buildfile sets some properties and declares the workstages (i.e. Ant targets) that need to be carried out. Of course this is fully explained in the Forrestbot documentation.

Create a file at the top-level of your project website named, e.g. `publish-stage.xml`

```
<?xml version="1.0"?>
<project name="my-lab-docs" default="main">
  <property name="getsrc.local.root-dir" location="."/>
  <target name="getsrc" depends="getsrc.clean-workdir, getsrc.local"/>
  <property name="deploy.scp.dest"
    value="username@apache.org:public_html/my-lab/stage"/>
  <import file="deploy.settings" optional="true"/>
  <target name="deploy" depends="deploy.scp"/>
  <property environment="env"/>
  <import file="${env.FORREST_HOME}/tools/forrestbot/core/forrestbot.xml"/>
</project>
```

So it gets the sources relative to the current directory. Actually it reads the `forrest.properties` configuration file to find out where other stuff is located.

It deploys the generated files directly to the user's "public_html" webspace into a "stage" sub-directory. Of course this "deploy.scp.dest" location could be a full pathname to an htdocs directory on the remove remote server.

4.4 The Cocoon "checksums" facility

Instruct Forrest's Cocoon to use a "checksums" file to record the checksum of each document that it has processed. This enables subsequent builds to only process the documents that have changed. This means that each file's timestamp will only be touched if it is a changed document, which enables the Forrestbot deploy.scp workstage to only deploy the changed files.

Follow the "checksums" FAQ.

4.5 The "build" workstage

After doing the usual process to edit source documents, review them with 'forrest run', and ensure that things are in order with 'forrest validate'.

Now do the build:

```
forrest -f publish-stage.xml build
```

This does the normal site build to generate the complete set of static documents. Watch for errors.

To review, see the built docs in `build/my-lab-docs` directory.

4.6 The "deploy" workstage

When satisfied, then deploy the built docs:

```
forrest -f publish-satge.xml deploy
```

The files that are being transferred to the remote server will now be listed.

4.7 Moving the documents into production

The reason for deploying the documents into a "stage" directory was so that clients or collaborators could review it before going into production. So ask them to review your people.apache.org/my-lab/stage/ site. When satisfied, then:

```
ssh people.apache.org
cd publish_html/my-lab/stage
cp -Rf * ..
```

That is it.

5 Further Reading

- How to deploy documentation with the Forrestbot "svn" workstage
- Forrestbot - automated building and deploying