

# How to deploy documentation with the Forrestbot "svn" workstage

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

## Table of contents

|   |   |
|---|---|
| 1 Intended Audience.....                      | 2 |
| 2 Purpose.....                                | 2 |
| 3 Prerequisites.....                          | 2 |
| 4 Steps.....                                  | 2 |
| 4.1 Introduction.....                         | 2 |
| 4.2 Follow along on your local system.....    | 2 |
| 4.3 The deploy.svn.settings file.....         | 3 |
| 4.4 The Forrestbot buildfile.....             | 3 |
| 4.5 The "build" workstage.....                | 3 |
| 4.6 The "deploy" workstage.....               | 3 |
| 4.7 Moving the documents into production..... | 4 |
| 5 Frequently Asked Questions.....             | 4 |
| 5.1 1 General issues.....                     | 4 |
| 6 Further Reading.....                        | 4 |

## 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 the Apache Forrest project website. 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 Forrest committers how to manage our project documentation. Other ASF projects will also find it useful.

## 3 Prerequisites

---

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

## 4 Steps

---

### 4.1 Introduction

---

In this document "we/our" refers to the Apache Forrest project committers, who manage the documentation. Even though other developers can help to enhance the documentation by way of contributed patches, only the set of committers can deploy (publish) it.

Our source documents are managed in the project's Subversion (SVN) version control system. Each committer has a local SVN working copy on their office computer. We can prepare source changes and review them with 'forrest run' and 'forrest validate-xdocs'. When satisfied we 'svn commit' those changes. Our Forrestbot uses the "getsrc.local" workstage to retrieve the sources from the local svn checkout.

The generated static documents which form the website are also stored in SVN. So our Forrestbot uses the "deploy.svn" workstage to deploy the generated documents. To publish the website, a regular cron job on the apache.org server automatically does 'svn update' in the website's document space. Elegant. That is what we call a "local forrestbot". Each committer can conduct the workstages on their local computer. This enables a distributed team to manage the documentation.

We have another Forrestbot on our "zone" [demonstration server](#). It runs automatically via cron each hour. Its workstages are not described in this howto, but essentially it gets the sources using the "getsrc.svn" workstage and it deploys the results to the local webserver htdocs directory on that machine using the "deploy.local" workstage. See that [buildfile](#).

### 4.2 Follow along on your local system

---

You have all the sources for the Forrest project in your Forrest distribution in the "\$FORREST\_HOME/site-author" directory. If you are using the release then you have a snapshot of the documentation. Of course if you have SVN trunk then you are up-to-date.

The Forrestbot build files for our site are also available. This means that you can run the "build" workstage. However you will not be able to do the "deploy" workstage unless you are a Forrest committer.

There is a brief explanation of how to publish our site in the document [\\$FORREST\\_HOME/etc/publishing\\_our\\_site.txt](#)

### 4.3 The `deploy.svn.settings` file

As explained in the above document, each committer needs a `deploy.svn.settings` file to provide their credentials.

```
<?xml version="1.0"?>
<project>
  <property name="deploy.svn.user" value="myApacheUsername"/>
  <property name="deploy.svn.password" value="mySvnPassword"/>
</project>
```

### 4.4 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.

Our buildfile is at [\\$FORREST\\_HOME/site-author/publish.xml](#) ...

```
<?xml version="1.0"?>
<project name="forrest-docs" default="main">
  <property name="getsrc.local.root-dir" location="."/>
  <target name="getsrc" depends="getsrc.clean-workdir, getsrc.local"/>
  <import file="../deploy.svn.settings" optional="true"/>
  <property name="deploy.svn.url"
    value="https://svn.apache.org/repos/asf/forrest/site"/>
  <target name="deploy" depends="deploy.svn"/>
  <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 `forrest/site` SVN repository.

### 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:

```
cd $FORREST_HOME/site-author
forrest -f publish.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/forrest-docs` directory.

### 4.6 The "deploy" workstage

When satisfied, then deploy the built docs:

```
forrest -f publish.xml deploy -Ddeploy.svn.commit-message="my commit message"
```

If no commit message is supplied, then forrestbot will use a default.

Now don't get frightened. It might seem like nothing is happening. The first time that the "deploy" workstage is used it will need to do an 'svn checkout' of the site repository. This will take some time. On subsequent deploys, it will need to do an 'svn update'. So be patient.

The files that are being committed will now be listed. That is it. They are now in the "[forrest/site](#)" SVN repository.

## 4.7 Moving the documents into production

---

Next a cron job on the server will automatically publish it. However, if quicker turnaround is required, then do this:

```
ssh people.apache.org
cd /www/forrest.apache.org
umask 002
svn update
```

Next an rsync cronjob will publish it to the production webserver. See other [notes](#) about how the ASF project websites are managed.

## 5 Frequently Asked Questions

---

### 5.1 1 General issues

---

#### 5.1.1 1.1 Why all the svn warnings about "is already under version control"

---

The deploy workstage does 'svn add' for all the generated documents that are new - in these cases the normal "A" messages will be issued. However it just blindly adds all documents that are either new or changed. So the warning messages are issued for the existing, updated documents.

#### 5.1.2 1.2 Why is every PDF document being deployed, even though they have not changed.

---

The PDF plugins FOP library is automatically adding a timestamp to every generated document. We need to find a solution.

## 6 Further Reading

---

- [How to deploy documentation with the Forrestbot "scp" workstage](#)
- [Forrestbot - automated building and deploying](#)