

# Forrestbot

## Table of contents

|                               |   |
|-------------------------------|---|
| 1 Overview.....               | 2 |
| 2 Using Forrestbot.....       | 2 |
| 3 Creating a buildfile.....   | 2 |
| 3.1 Workstages.....           | 3 |
| 3.2 Workstage Properties..... | 3 |
| 4 Forrestbot design.....      | 7 |

## 1. Overview

Forrestbot lets you automate building and deploying websites. The whole process gets the source docs, builds it, then deploys the site where you want it to go. It can also notify you afterwards, and it keeps a log of the build process. To accomplish these tasks Forrestbot uses four "workstages" (getsrc, build, deploy, notify) that are comprised of "implementations" and have various properties that may be set, depending upon the implementations chosen.

### FIXME (AB):

Need a better term than comprised of? Trying give a sense of the building blocks used and how they relate to each other.

## 2. Using Forrestbot

You need to create a customized buildfile directing Forrestbot's work and then simply execute:

```
forrest -f mybuildfile.xml
```

The next section explains how to create your buildfile.

## 3. Creating a buildfile

Within the new buildfile you need to first set properties needed by the workstages you are going to use and then specify what implementations will be used by each workstage.

### FIXME (AB):

This seems backwards since the properties are dependent on the implementations - no?

This sample buildfile can be used as a base from which to customize your own buildfile. The file starts with the project name and default workstage target, then sets the specific properties we will need to get the source, set up notification and indicate the deploy location. It then specifies which implementations we will use and finishes up with importing the forrestbot.xml file.

### FIXME (AB):

Is "default workstage target" the correct term for default="main"?

```
<project name="mysampleproject" default="main">
  <property name="notify.email.host" value="smtp.myhost.com"/>
  <property name="notify.email.to" value="me@domain.com"/>
  <property name="notify.administrator" value="Your Name
&lt;me@domain.com>"/>
  <property name="getsrc.cvs.user" value="anoncvs"/>
  <property name="getsrc.cvs.password" value="anoncvs"/>
  <property name="getsrc.cvs.root" value="/home/cvspublic"/>
  <property name="getsrc.cvs.host" value="cvs.myhost.com"/>
  <property name="getsrc.cvs.module" value="myproject"/>
  <property name="deploy.scp.dest"
value="username@myhost.com:/var/www/mydomain/htdocs"/>

  <!-- here we specify to use two notification implementations -->
  <target name="notify" depends="notify.local, notify.email"/>

  <!-- here we specify to deploy with the scp implementation -->
  <target name="deploy" depends="deploy.scp"/>
```

```

        <!-- the default implementation for getsrc is getsrc.cvs, which is
what we want -->

        <!-- assumes FORREST_HOME has been set as an environment variable
-->
        <property environment="env"/>
        <import
file="${env.FORREST_HOME}/tools/forrestbot/core/forrestbot.xml"/>
</project>

```

### 3.1. Workstages

It is only necessary to include the specific target implementations in the buildfile if we want to override the default implementations. The following tables show the various workstages and which implementations may be used for each and which is the default.

| Workstage              | Implementations  |
|------------------------|--|
| getsrc                 | <ul style="list-style-type: none"> <li>• <a href="#">getsrc.local</a></li> <li>• <a href="#">getsrc.cvs</a> (default)</li> <li>• <a href="#">getsrc.svn</a></li> </ul>                                       |
| build                  | <ul style="list-style-type: none"> <li>• <a href="#">build.forrest</a></li> </ul>  |
| deploy                 | <ul style="list-style-type: none"> <li>• <a href="#">deploy.local</a> (default)</li> <li>• <a href="#">deploy.scp</a></li> <li>• <a href="#">deploy.cvs</a></li> <li>• <a href="#">deploy.svn</a></li> </ul> |
| <a href="#">notify</a> | <ul style="list-style-type: none"> <li>• <a href="#">notify.local</a> (default)</li> <li>• <a href="#">notify.email</a></li> </ul>   |

If you want to do more advanced processing for your project, you can override the 'main' target, which by default is `<target name="main" depends="getsrc, build, deploy, notify" />`, create your own implementation of a workstage, or use any other ant tasks to do additional work.

#### FIXME (AB):

How do you create your own implementation? Can you create a different default target than main - how do you override the default?

### 3.2. Workstage Properties

Each workstage implementation is configurable with properties. The following tables describe each property and whether or not you are required to set it in your buildfile.

Many workstage properties use usernames and passwords. You may want to keep them out of your project's xml buildfile (especially if you store that file in CVS or SVN). A nice way to do this is make a simple buildfile (e.g. my-settings.xml) that just sets those properties (don't include it in CVS/SVN). Then in your project buildfile, have `<import file="my-settings.xml" />`.

#### 3.2.1. Misc Properties

| Property | Description | Default Value | Required? |
|----------|-------------|---------------|-----------|
|----------|-------------|---------------|-----------|

|  |                                       |  |     |
|--|---------------------------------------|--|-----|
| ant.project.name (you specify this by <project name="____"> in your buildfile) | This must be unique for each project. |  | Yes |
|--|---------------------------------------|--|-----|

### 3.2.2. getsrc.clean-workdir

This should be executed before a getsrc implementation is executed. For example, <target name="getsrc" depends="getsrc.clean-workdir, getsrc.svn"/>

### 3.2.3. getsrc.local

| Property              | Description  | Default Value | Required? |
|-----------------------|--|---------------|-----------|
| getsrc.local.root-dir | Absolute path to the project's root directory on the local computer. Use <b>location=</b> instead of <b>value=</b> for this <property> |               | Yes       |

### 3.2.4. getsrc.cvs

| Property            | Description  | Default Value        | Required? |
|---------------------|--|----------------------|-----------|
| getsrc.cvs.user     | CVS username   |                      | Yes       |
| getsrc.cvs.password | CVS password   |                      | Yes       |
| getsrc.cvs.root     | CVS root directory   | /home/cvsroot        | Yes       |
| getsrc.cvs.host     | CVS host   | cvs.apache.org       | Yes       |
| getsrc.cvs.module   | CVS module name (an alias, or full path) to the directory that contains forrest.properties | \${ant.project.name} | Yes       |
| getsrc.cvs.tag      | CVS tag or branch name   |                      | No        |

### 3.2.5. getsrc.svn

| Property            | Description  | Default Value | Required? |
|---------------------|--|---------------|-----------|
| getsrc.svn.url      | Full repository URL for project (this directory must contain forrest.properties) |               | Yes       |
| getsrc.svn.revision | Revision number to fetch   | HEAD          | No        |

### 3.2.6. build.forrest

| Property | Description | Default Value | Required? |
|----------|-------------|---------------|-----------|
|----------|-------------|---------------|-----------|

|                |   |      |    |
|----------------|---|------|----|
| build.work-dir | Directory to temporarily hold working files | work | No |
| build.log-dir  | Directory to hold log files                 | logs | No |

### 3.2.7. deploy.local

| Property         | Description  | Default Value              | Required? |
|------------------|--|----------------------------|-----------|
| deploy.local.dir | Path to deploy site to - the dir that would be the equivalent of build/site dir. Relative paths will be relative to \${bot.home} | sites/\${ant.project.name} | No        |

### 3.2.8. deploy.scp

\${user.home}/.ssh/known\_hosts must properly recognize the host, so you should manually make an ssh connection to the host if you never have before.

| Property            | Description  | Default Value | Required?   |
|---------------------|--|---------------|---|
| deploy.scp.dest     | Full destination reference in the format user@host:/directory/path |               | Yes   |
| deploy.scp.password | Password for user@host   |               | No. You will be prompted for it if it is not set. |

### 3.2.9. deploy.cvs

This is only available on \*nix operating systems.

| Property               | Description  | Default Value                     | Required? |
|------------------------|--|-----------------------------------|-----------|
| deploy.cvs.user        | CVS username to use when committing changes  |                                   | Yes       |
| deploy.cvs.password    | CVS password   |                                   | Yes       |
| deploy.cvs.root        | CVS root   | /home/cvs                         | Yes       |
| deploy.cvs.host        | CVS host   | cvs.apache.org                    | Yes       |
| deploy.cvs.module      | CVS module   | \${ant.project.name}              | Yes       |
| deploy.cvs.commit-mess | Message to use when committing. You probably want to put a machine name or person's name here. | Automatic publish from forrestbot | No        |

### 3.2.10. deploy.svn

| Property | Description | Default Value | Required? |
|----------|-------------|---------------|-----------|
|----------|-------------|---------------|-----------|

|                        |  |                                   |     |
|------------------------|--|-----------------------------------|-----|
| deploy.svn.user        | SVN username to use when committing changes  |                                   | Yes |
| deploy.svn.password    | SVN password   |                                   | Yes |
| deploy.svn.url         | Full repository URL  |                                   | Yes |
| deploy.svn.commit-mess | Message to use when committing. You probably want to put a machine name or person's name here. | Automatic publish from forrestbot | No  |

### 3.2.11. notify

These settings are used by all notify implementations.

| Property                 | Description   | Default Value | Required?                      |
|--------------------------|---|---------------|--------------------------------|
| notify.administrator     | Name and email address of the forrestbot administrator          |               | Yes                            |
| notify.on.failure        | On a build failure, notification will happen if this is true.   | true          | No                             |
| notify.on.success        | On a succesful build, notification will happen if this is true. | true          | No                             |
| notify.log               | Log file  |               | No. Set by other workstage(s). |
| notify.deploy-location   | Deployed location   |               | No. Set by other workstage(s). |
| notify.completion-status | Result of the build   |               | No. Set by other workstage(s). |

### 3.2.12. notify.local

No properties.

### 3.2.13. notify.email

| Property          | Description                                     | Default Value           | Required?   |
|-------------------|---|-------------------------|---|
| notify.email.host | SMTP host through which the email will be sent. | localhost               | Yes   |
| notify.email.to   | Email address to send notification to.          | \${user.name}@localhost | Yes   |
| notify.email.from | From: address in the email                      | Forrestbot              | No, but some mailers may require a valid email address. |

## 4. Forrestbot design

Forrest and forrestbot use ant buildfiles extensively. Ant 1.6's import task is used to import multiple buildfiles into a single build. The following is the flow of control when running forrestbot:

- Your buildfile
  - forrestbot.xml
    - workstage buildfiles
    - forrest.build.xml

The workstage buildfiles set up the properties and files so that the main forrest buildfile (forrest.build.xml) will run. After it is run, other workstage buildfiles can implement reporting, deployment, or other post-build activities.

Your buildfile can specify which workstages you want to use, set properties for them, and do any additional pre- and post-processing.

### **FIXME (AB):**

Which of these (Your buildfile/workstage buildfile) is the "project buildfile" referred to elsewhere on this page, if any? If it is "Your buildfile" can I also create workstage buildfiles? If so, how?