Automated Site Deployment to Windows Server

Overview

Although deploying sites to a Unix based server is easier such an option is not always available. In such a situation it is possible to configure maven for automated site deployment to a Windows box. Unfortiantely, as of this writting only the wagon-ssh and wagon-ssh-external implementations are capable of deploying a directory as required by the site plugin. The actual wagon-ssh implementation goes through the following steps in moving a directory.

- Create a temporary zip file with the directory contents.
- Ensure the remote target directory exists by executing the "mkdir" command via ssh in a non-interactive shell.
- Move the zip file to remote server via scp or sftp (protocol used is dervied from the protocol section of the site url specified in your pom.)
- Unzip the archive on the remote server by executing the "unzip" command via ssh in a non-interactive shell.
- Delete the archive on the remote server by executing the "rm" command via ssh in a non-interactive shell.

To help make the above steps clear it is helpful to show the console output generated when executing the site:deploy goal.

site:deploy console output

```
dos prompt>mvn  site:deploy
[INFO] Scanning for projects...
[INFO] Searching repository for plugin with prefix: 'site'.
[INFO]
--------------------------------------------------------------------------
--
[INFO] Building Maven NDoc Report Plugin
[INFO]    task-segment: [site:deploy]
[INFO]
--------------------------------------------------------------------------
--
[INFO] [site:deploy]
scp://whouatl8z9tn91/cygdrive/c/Apps/Tomcat5.5/webapps/ROOT/mvnsites/maven
-ndoc-plugin/1.0.1-SNAPSHOT - Session: Opened

Executing command: mkdir -p
/cygdrive/c/Apps/Tomcat5.5/webapps/ROOT/mvnsites/maven-ndoc-plugin/1.0.1-S
NAPSHOT/.
Executing command: mkdir -p
/cygdrive/c/Apps/Tomcat5.5/webapps/ROOT/mvnsites/maven-ndoc-plugin/1.0.1-S
NAPSHOT/.
Executing command: scp -t
/cygdrive/c/Apps/Tomcat5.5/webapps/ROOT/mvnsites/maven-ndoc-plugin/1.0.1-S
NAPSHOT/./wagon65147
.zip
Uploading: ./wagon65147.zip to
scp://whouatl8z9tn91/cygdrive/c/Apps/Tomcat5.5/webapps/ROOT/mvnsites/maven
-ndoc-plugin/1.
0.1-SNAPSHOT

###########################
Transfer finished. 110209 bytes copied in 0.081 seconds
Executing command: cd
/cygdrive/c/Apps/Tomcat5.5/webapps/ROOT/mvnsites/maven-ndoc-plugin/1.0.1-S
NAPSHOT/.; unzip -q -o w
agon65147.zip; rm -f wagon65147.zip
```

```
scp://whouatl8z9tn91/cygdrive/c/Apps/Tomcat5.5/webapps/ROOT/mvnsites/maven
-ndoc-plugin/1.0.1-SNAPSHOT - Session: Disconn
ecting
scp://whouatl8z9tn91/cygdrive/c/Apps/Tomcat5.5/webapps/ROOT/mvnsites/maven
-ndoc-plugin/1.0.1-SNAPSHOT - Session: Disconn
ected
[INFO]
------------------------------------------------------------------------
[INFO] BUILD SUCCESSFUL
[INFO]
------------------------------------------------------------------------
[INFO] Total time: 5 seconds
[INFO] Finished at: Thu Sep 07 09:28:07 CDT 2006
[INFO] Final Memory: 3M/6M
[INFO]
------------------------------------------------------------------------
```

```
dos prompt>
```

Non-Interactive Shell Considerations

It is very important to realize that executing a command in a non-interactive ssh shell is different from executing it via an interactive ssh shell. When running a non-interactive command the unix shell is configured with a subset of the dot files used when logged in interactively. For example in the bash shell, only the .bashrc file appears to be read, whereas an interactive bash is typically configured by both .bashrc and .bash_profile.

The wagon-ssh implementation runs the unzip and rm command via a non-interactive shell. Therefore it is important to ensure the commands needed by the wagon-ssh implementation are available in the PATH of the relevant user's non-interactive shell. If your using putty, you can use the plink tool to perform such tests. The example below

Examples of Remote Executions of Various Commands

```
dos prompt>plink remoteuser@remotemachine.somecompany.com 'pwd'
dos prompt>plink remoteuser@remotemachine.somecompany.com 'ls'
dos prompt>plink remoteuser@remotemachine.somecompany.com 'echo ${PATH}'
dos prompt>plink remoteuser@remotemachine.somecompany.com 'unzip --help'
```

Cygwin and OpenSSH

As you can now see, any attempt to deploy to a windows server must make the maven wagon-ssh implementation think the remote server is a unix box. Therefore, just using a simple to install ssh server such as FreeSSHd isn't sufficient, as it doesn't provide a shell which provides mkdir, unzip and rm. Logging into to windows server running FreeSSHd is just like being in a dos shell, and dos shells don't support unix commands. Consequently, the easiest thing to do is to simply get openSSH running via Cygwin, making sure the required commands are in the PATH of the relevant user's non-interactive ssh shell.

A discussion of setting up openSSH via Cygwin is beyond the scope of this document. With a quick search on google for "openSSH cygwin" I quickly found the following how-to document: http://pigtail.net/LRP/printsrv/cygwin-sshd.html. By following this document along with a bit of poking around, the reader will hopefully quickly get openSSH up and running.

Example Snipts from settings.xml and pom.xml

settings.xml snipt

```
<servers>
        <server>
            <id>website</id>
            <username>remoteuser</username>
            <password>remotepass</password>
        </server>
        <!--password and key file based auth both appear to work with the wagon-ssh
implementation
        <server>
            <id>website</id>
            <passphrase>somepassphrase</passphrase>
            <privateKey>C:/sshkeys/id_rsa</privateKey>
        </server>
        -->
</servers>
```

pom.xml snipt

```
<distributionManagement>
        <repository>
            <id>custom-repository</id>
            <url>ftp://somemachine.somecompany.com</url>
        </repository>
        <site>
            <id>website</id>

<url>scp://somemachine.somecompany.com/cygdrive/c/Apps/Tomcat5.5/webapps/ROOT/mvnsites
/mymodule/${project.version}</url>

<!--<url>sftp://somemachine.somecompany.com/cygdrive/c/Apps/Tomcat5.5/webapps/ROOT/mvn
sites/mymodule/${project.version}</url>-->
        </site>
</distributionManagement>
```