

NewAntFeaturesInDetail Ssh

manual pages of [scp](#) and [sshexec](#).

So you want to deploy your distributables to a remote machine via scp?

The first step is to grab the [jsch.jar](#) and drop it into your \$ANT_HOME/lib/optional directory.

The task can be used very easily to copy a selection of files to a specific directory on the secured server. The following example copies the files from the \${dist.dir} to a directory on a host built from a selection of properties:

```
<scp todir="${username}:${password}@${hostname}:${hostpath}">
  <fileset dir="${dist.dir}" />
</scp>
```

A downfall to this task is that it will copy up every file every time, regardless of whether the original has changed or not. This is of particular concern if you're deploying a large website or using a slow connection. But don't fear the all new [<modified/>](#) selector can be used to cut out the unnecessary bandwidth. The task now becomes:

```
<scp todir="${username}:${password}@${hostname}:${hostpath}">
  <fileset dir="${dist.dir}">
    <modified>
      <param name="cache" value="${build.dir}/deploy.cache" />
    </modified>
  </fileset>
</scp>
```

You might also want to consider using the Ant1.5 [<input/>](#) task so that users are prompted for configuration details if the properties haven't already been set. A complete example follows:

```
<target name="get-password" unless="password">
  <input message="Please enter password:" addproperty="password" />
</target>

<target name="deploy" depends="get-password">
  <mkdir dir="${build.dir}" />
  <scp todir="${username}:${password}@${hostname}:${hostpath}">
    <fileset dir="${dist.dir}">
      <modified>
        <param name="cache" value="${build.dir}/deploy.cache" />
      </modified>
    </fileset>
  </scp>
</target>
```

Don't forget to have a look at the `trust` and `knownhosts` attributes in the [scp](#) task's description if you receive an exception like this:

```
com.jcraft.jsch.JSchException: reject HostKey
```

Use `knownhosts` if you have a "known hosts" file containing the key of the host you want to connect to or `trust` if you don't have such a file and want to trust the host anyway.

If you receive an exception like this:

```
com.jcraft.jsch.JSchException: Auth fail
```

it's likely that your remote machine is not allowing password authentication. You have a couple of choices. If you have access to the remote machine's `sshd_config` file, you may have the option of setting password authentication to yes. Alternatively, you can try the patch (I haven't tried it) offered by jcraft. I found it here <http://marc.theaimsgroup.com/?l=ant-dev&m=111959408515300&w=2>.