

Password Encryption

(If you're working with Maven 4, please refer to the Maven 4 edition of this guide ([./guide-encryption-4.html](#)))

- Introduction
- How to create a master password
- How to encrypt server passwords
- How to keep the master password on removable drive
- Tips
 - Escaping curly-brace literals in your password (Since: Maven 2.2.0)
 - Password Security
 - Password Escaping on different platforms
 - Prompting for Password

Introduction

Maven supports server password encryption. The main use case, addressed by this solution is:

- multiple users share the same build machine (server, CI box)
- some users have the privilege to deploy Maven artifacts to repositories, some don't.
 - this applies to any server operations, requiring authorization, not only deployment
- `settings.xml` is shared between users

The implemented solution adds the following capabilities:

- authorized users have an additional `settings-security.xml` file in their `${user.home}/.m2` directory
 - this file either contains encrypted **master password**, used to encrypt other passwords
 - or it can contain a **relocation** - reference to another file, possibly on removable storage
 - this password is created first via CLI for now
- server entries in the `settings.xml` have passwords and/or keystore passphrases encrypted
 - for now - this is done via CLI **after** master password has been created and stored in appropriate location

How to create a master password

Use the following command line:

```
mvn --encrypt-master-password <password>
```

Note: Since Maven 3.2.1 the password argument should no longer be used (see Tips below for more information). Maven will prompt for the password. Earlier versions of Maven will not prompt for a password, so it must be typed on the command-line in plaintext.

This command will produce an encrypted version of the password, something like

```
{jSMOWnoPFgshVpMvz5VrIt5kRbzGpI8u+9EF1iFQyJQ=}
```

Store this password in the `${user.home}/.m2/settings-security.xml`; it should look like

1. `<settingsSecurity>`
2. `<master>{jSMOWnoPFgshVpMvz5VrIt5kRbzGpI8u+9EF1iFQyJQ=}</master>`
3. `</settingsSecurity>`

The encrypted version of the master password is encrypted with a hardcoded key (<https://github.com/apache/maven/blob/fe25a2627c1dafeb44188dad9f45dfd5fe965a98/maven-embedder/src/main/java/org/apache/maven/cli/MavenCli.java#L856>), so please treat it as if the password is stored in the file in plain text.

When this is done, you can start encrypting existing server passwords.

How to encrypt server passwords

You have to use the following command line:

```
mvn --encrypt-password <password>
```

Note: Just like `--encrypt-master-password` the password argument should no longer be used since Maven 3.2.1 (see Tips below for more information.).

This command produces an encrypted version of it, something like

```
{C0QLCE6DU6GtcS5P=}
```

Copy and paste it into the servers section of your `settings.xml` file. This will look like:

```
1. <settings>
2. ...
3.   <servers>
4. ...
5.     <server>
6.       <id>my.server</id>
7.       <username>foo</username>
8.       <password>{C0QLCE6DU6GtcS5P=}</password>
9.     </server>
10. ...
11.   </servers>
12. ...
13. </settings>
```

Please note that password can contain any information outside of the curly brackets, so that the following will still work:

```
1. <settings>
2. ...
3.   <servers>
4. ...
5.     <server>
6.       <id>my.server</id>
7.       <username>foo</username>
8.       <password>Oleg reset this password on 2009-03-11, expires on 2009-04-11 {C0QLCE6DU6GtcS5
      P=}</password>
9.     </server>
10. ...
11.   </servers>
12. ...
13. </settings>
```

Then you can use, say, deploy plugin, to write to this server:

```
mvn deploy:deploy-file -Durl=https://maven.corp.com/repo \
                        -DrepositoryId=my.server \
                        -Dfile=your-artifact-1.0.jar \
```

How to keep the master password on removable drive

Create the master password exactly as described above, and store it on a removable drive, for instance on OSX, my USB drive mounts as `/Volumes/mySecureUsb`, so I store

```
1. <settingsSecurity>
2.   <master>{jSM0wnoPFgsHVpMvz5VrIt5kRbzGpI8u+9EF1iFQyJQ=}</master>
3. </settingsSecurity>
```

in the file `/Volumes/mySecureUsb/secure/settings-security.xml`

And then I create `${user.home}/.m2/settings-security.xml` with the following content:

```
1. <settingsSecurity>
2.   <relocation>/Volumes/mySecureUsb/secure/settings-security.xml</relocation>
3. </settingsSecurity>
```

This assures that encryption only works when the USB drive is mounted by the OS. This addresses a use case where only certain people are authorized to deploy and are issued these devices.

Tips

Escaping curly-brace literals in your password (*Since: Maven 2.2.0*)

At times, you might find that your password (or the encrypted form of it) contains '{' or '}' as a literal value. If you added such a password as-is to your settings.xml file, you would find that Maven does strange things with it. Specifically, Maven treats all the characters preceding the '{' literal, and all the characters after the '}' literal, as comments. Obviously, this is not the behavior you want. What you really need is a way of **escaping** the curly-brace literals in your password.

You can do this with the widely used '\' escape character. If your password looks like this:

```
jSMOWnoPFgsHVPmvz5VrIt5kRbzGpI8u+{EF1iFQyJQ=
```

Then, the value you would add to your settings.xml looks like this:

```
{jSMOWnoPFgsHVPmvz5VrIt5kRbzGpI8u+\{EF1iFQyJQ=}
```

Password Security

Editing `settings.xml` and running the above commands can still leave your password stored locally in plaintext.

You may want to check the following locations:

- Shell history (e.g. by running `history`). You may want to clear your history after encrypting the above passwords
- Editor caches (e.g. `~/.viminfo`)

Also note that the encrypted passwords can be decrypted by someone that has the master password and settings security file. Keep this file secure (or stored separately) if you expect the possibility that the `settings.xml` file may be retrieved.

Password Escaping on different platforms

On some platforms it might be necessary to quote the password if it contains special characters like `%`, `!`, `$`, etc. For example on Windows you have to be careful about things like the following:

The following example will not work on Windows:

```
mvn --encrypt-master-password a!$%^b
```

whereas the following will work on Windows:

```
mvn --encrypt-master-password "a!$%^b"
```

If you are on a linux/unix platform you should use single quotes for the above master password. Otherwise the master password will not work (caused by the dollar sign and the exclamation mark).

Prompting for Password

In Maven before version 3.2.1 you have to give the password on the command line as an argument which means you might need to escape your password. In addition usually the shell stores the full history of commands you have entered, therefore anyone with access to your computer could restore the password from the shell's history.

Starting with Maven 3.2.1, the password is an optional argument. If you omit the password, you will be prompted for it which prevents all the issues mentioned above.

We strongly recommend using Maven 3.2.1 and above to prevent problems with escaping special characters and of course security issues related to bash history or environment issues in relationship with the password.