

## 1.6 Using Expect Scripts

Expect scripts can be used to automate multiple CLI commands in batch mode. Expect is a UNIX scripting and testing utility which can be used with SSH-based applications, like the Oracle VM CLI. Expect scripts can have any file name suffix you like, though they generally have an `.exp` extension. This guide and the sample scripts use either the `.sh` or no extension on Expect script file names. To run an Expect script, use the syntax:

```
expect script
```

Alternatively, make sure you have execute permission set on the Expect script, and run it as would any shell script; without explicitly specifying the **expect** command.

### 1.6.1 Expect Script Prerequisites

To use an Expect script, make sure you have the Expect utility installed. To install Expect using a Yum server on Oracle Linux, enter:

```
# yum install expect
```

To install Expect on other operating systems, see:

<http://expect.sourceforge.net/>

### 1.6.2 Writing Expect Scripts

You can write your own Expect scripts to perform batch jobs with the CLI. This section shows you how to create a very simple Expect script and run it with the CLI, but does not go into details about the programming languages used with Expect. For more information on writing Expect scripts, see:

<http://expect.sourceforge.net/>

A very simple Expect script that lists the Oracle VM Servers is listed in [Example 1.1, "list\\_server.exp Expect script"](#):

#### Example 1.1 list\_server.exp Expect script

```
#!/usr/bin/expect

## Access CLI
set loginUser "admin"
set loginPassword "password"
set mgmtServerAddress manager_host

## Expect Parameters
set timeout 20
set successMsg "Status: Success"
set failureMsg "Status: Failure"

spawn ssh -l $loginUser $mgmtServerAddress -p 10000
```

```
expect_after eof {exit 0}

set timeout 10

##interact with SSH
##expect "yes/no" {send "yes\r"}
expect "password:" {send "$loginPassword\r"}
puts "\n## Starting Generated OVMCLI Script... ##\n"
set timeout 600

expect "OVM> "
send "set OutputMode=Verbose\r"
expect $successMsg {} \
    timeout { puts "\n\nTest Failure: \n\r"; exit}

expect "OVM> "
    send "list Server\r"
    expect $successMsg {} \
        timeout { puts "\n\nScript Failure: \n\r"; exit}
```

Edit the `loginUser`, `loginPassword` and `mgmtServerAddress` variables for your Oracle VM Manager environment.

To execute this Expect script file, use the **expect** command followed by the location of the Expect script, for example, enter:

```
# expect /myscripts/list_server.exp
spawn ssh -l admin localhost -p 10000
admin@localhost's password:
## Starting Generated OVMCLI Script... ##
```

```
OVM> set OutputMode=Verbose
Command: set OutputMode=Verbose
Status: Success
Time: date
OVM> list Server
Command: list Server
Status: Success
Time: date
Data:
    id:00:e0:81:4d:41:01:00:e0:81:4d:40:d6:00:e0:81:4d  name:MyServer1
    id:00:e0:81:4d:40:c6:00:e0:81:4d:40:c7:ff:ff:ff:ff  name:MyServer2
    id:00:e0:81:4d:40:f5:00:e0:81:4d:40:be:00:e0:81:4d  name:MyServer3
OVM>
```

|  |                      |                      |
|--|----------------------|----------------------|
| <a href="#">Prev</a>                             | <a href="#">Up</a>   | <a href="#">Next</a> |
| 1.5 Passing in a CLI Command at the Command Line | <a href="#">Home</a> | 1.7 Sample Scripts   |



Copyright © 2012, 2014, Oracle and/or its affiliates. All rights reserved. [Legal Notices](#)