Managing Remote Systems in Oracle® Solaris 11.1

Part No: E29001–01
October 2012

ORACLE

121203@25097

Contents

Tables

Examples

Preface

Managing Remote Systems in Oracle Solaris 11.1 is part of a multivolume set that covers a significant part of the Oracle Solaris system administration information. This book assumes that you have already installed the Oracle Solaris operating system, and you have set up any networking software that you plan to use.

Note – This Oracle Solaris release supports systems that use the SPARC and x86 families of processor architectures. The supported systems appear in the *Oracle Solaris OS: Hardware Compatibility Lists*. This document cites any implementation differences between the platform types.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info` or visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs` if you are hearing impaired.

Typographic Conventions

The following table describes the typographic conventions that are used in this book.

TABLE P–1 Typographic Conventions

Typeface	Description	Example
`AaBbCc123`	The names of commands, files, and directories, and onscreen computer output	Edit your `.login` file. Use `ls -a` to list all files. `machine_name% you have mail.`
`AaBbCc123`	What you type, contrasted with onscreen computer output	`machine_name%` **`su`** `Password:`
aabbcc123	Placeholder: replace with a real name or value	The command to remove a file is `rm` *filename*.

TABLE P–1 Typographic Conventions *(Continued)*

Typeface	Description	Example
AaBbCc123	Book titles, new terms, and terms to be emphasized	Read Chapter 6 in the *User's Guide*. A *cache* is a copy that is stored locally. Do *not* save the file. **Note:** Some emphasized items appear bold online.

Shell Prompts in Command Examples

The following table shows the default UNIX system prompt and superuser prompt for shells that are included in the Oracle Solaris OS. Note that the default system prompt that is displayed in command examples varies, depending on the Oracle Solaris release.

TABLE P–2 Shell Prompts

Shell	Prompt
Bash shell, Korn shell, and Bourne shell	`$`
Bash shell, Korn shell, and Bourne shell for superuser	`#`
C shell	`machine_name%`
C shell for superuser	`machine_name#`

◆ ◆ ◆ CHAPTER 1

Working With Remote Systems (Overview)

This chapter includes information about working with remote files.

What Is the FTP Server?

The Oracle Solaris release includes ProFTPD. The software implements the server side of the FTP protocol, which is widely used for distribution of bulk data over the Internet. For more information about the ProFPTD project, see `http://www.proftpd.org`.

What Is a Remote System?

For the purpose of this chapter, a *remote system* is a workstation or server that is connected to the local system with any type of physical network and configured for TCP/IP communication.

On systems running an Oracle Solaris release, TCP/IP configuration is established automatically during startup. For more information, see *Configuring and Administering Oracle Solaris 11.1 Networks*.

About FTP Servers in This Release

The old FTP server based on the wu-ftpd distribution has been replaced by the `proftpd` server. The migration of configuration information from the old service to the new is described in `/usr/share/doc/proftpd/proftpd_migration.txt`.

Differences From Standard ProFTPD

The following list discusses the items that are different in the Oracle Solaris 11 implementation of ProFTPD:

- The Oracle Solaris version of ProFTPD runs in stand-alone mode by default.
- This release does not use the `logrotate` command to rotate the service logs.

ProFTPD Components

The following section provides information about the commands, files, and other important components of the ProFTPD service.

ProFTPD Commands

The following table describes the commands and daemons that are associated with the ProFTPD service.

TABLE 1–1 ProFTPD Commands

File Name	Function
`/usr/bin/ftp`	Provides the user interface to the ProFTPD service. See the `ftp`(1) man page for more information.
`/usr/bin/ftpcount`	Shows the current number of connections per server, as well as per virtual host or anonymous configuration. See the `ftpcount`(1) man page for more information.
`/usr/bin/ftpdctl`	Controls the `proftpd` service daemon. See the `ftpdctl`(8) man page for more information.
`/usr/bin/ftptop`	Displays the current status of FTP sessions in a continuously updating format. See the `ftptop`(1) man page for more information.
`/usr/bin/ftpwho`	Shows process information for all active `proftpd` connections and a count of all connected users to each server. See the `ftpwho`(1) man page for more information.

TABLE 1–1 ProFTPD Commands *(Continued)*

File Name	Function
`/usr/sbin/ftprestart`	Restarts FTP connections by using the `ftpshut -R` command. See the `ftpshut`(8) man page for more information.
`/usr/sbin/ftpscrub`	Removes processes that are no longer live from the scoreboard file on demand. See the `ftpscrub`(8) man page and `http://www.proftpd.org/docs/howto/Scoreboard.html` for more information.
`/usr/sbin/ftpshut`	Shuts down FTP connections at a given time. See the `ftpshut`(8) man page for more information.
`/usr/lib/inet/proftpd`	Provides FTP services. See the `proftpd`(8) man page for more information.

ProFTPD Files

The following table lists many of the files associated with the ProFTPD service and their functions.

TABLE 1–2 ProFTPD Files

File Name	Function
`~/.ftpaccess`	Provides an additional control mechanism for each virtual host. The file should be placed in the home directory for the virtual host. See `http://www.proftpd.org/localsite/Userguide/linked/x1021.html` for more information.
`/etc/proftpd.conf`	Includes most of the configuration parameters that need to be defined in order for the ProFTPD service to function.
`/etc/shutmsg`	Includes information used by the `ftpshut` command.
`/etc/ftpd/ftpusers`	Lists the users to be disallowed FTP login privileges. Provided for backward compatibility with the wu-ftpd service.
`/var/log/xferlog`	Lists log information for ProFTPD.
`/var/run/proftpd.scoreboard`	Includes tracking information for each current session, which is used by commands like `ftpcount`, `ftptop`, and `ftpwho`. See `http://www.proftpd.org/docs/howto/Scoreboard.html` for more information.

ProFTPD User

A user and a group named `ftp` are created by the ProFTPD installation process. The ProFTPD server runs under these credentials.

◆ ◆ ◆ CHAPTER 2

Administering the FTP Server (Tasks)

This chapter includes tasks to set up and administer an FTP server.

Administering the FTP Server (Task Map)

The following table describes the procedures that are needed to use the FTP server.

TABLE 2–1 Task Map: Administering the FTP Server

Task	Description	For Instructions
Start the FTP server.	Follow this procedure after changing the `proftpd.conf` file.	"How to Start an FTP Server Using SMF" on page 16
Stop the FTP server.	Follow this procedure before changing the `proftpd.conf` file.	"How to Shut Down the FTP Server Using SMF" on page 16
Shut down the FTP server connections.	Run the `ftpshut` to shut down the FTP connections during file system maintenance or other events that do not require that the service be stopped but access to the files needs to be denied.	"How to Shut Down the FTP Connection" on page 16
Reconfigure the FTP server.	Follow this procedure when changing the `proftpd.conf` file.	"How to Change the ProFTPD Configuration" on page 17

Administering the FTP Server (Tasks)

The following procedures show how to start and stop the FTP server, how to disable the FTP connection and how to make changes to the ProFTPD configuration file.

How to Start an FTP Server Using SMF

1. **Become an administrator.**

   For more information, see "How to Use Your Assigned Administrative Rights" in *Oracle Solaris 11.1 Administration: Security Services*.

2. **Start the FTP server.**

```
# svcadm enable network/ftp
```

How to Shut Down the FTP Server Using SMF

1. **Become an administrator.**

   For more information, see "How to Use Your Assigned Administrative Rights" in *Oracle Solaris 11.1 Administration: Security Services*.

2. **Stop the FTP server.**

```
# svcadm disable network/ftp
```

How to Shut Down the FTP Connection

The `ftpshut`(8) command closes down the FTP server at a particular time. If you want to stop serving FTP only, but not stop the daemon (so it can report the service is not available to clients) then use this procedure. The `ftpshut` command will block connections and stop the current connection, but not shutdown the server daemon itself.

When you run `ftpshut`, a file is generated from command-line options that specify when shutdown occurs, the point at which new connections are refused, and when existing connections are dropped. Users are notified of a server shutdown based on this information. The location of the file that is created by `ftpshut` is `/etc/shutmsg`.

1. **Become an administrator.**

   For more information, see "How to Use Your Assigned Administrative Rights" in *Oracle Solaris 11.1 Administration: Security Services*.

2. **Run the `ftpshut` command.**

```
ftpshut [ -l min] [ -d min] time [warning-message...]
```

`ftpshut`	Command that provides a procedure for notifying users that the FTP server is shutting down.
`-l`	Flag that is used to adjust the time that new connections to the FTP server are denied
`-d`	Flag that is used to adjust the time that existing connections to the FTP server are disconnected
`time`	Shutdown time that is specified by the word `now` for immediate shutdown, or in one of two formats (+ *number* or *HHMM*) for a future shutdown
`[warning-message...]`	Shutdown notification message; see the `ftpshut`(8) man page for more information

3. **Restore access to the files.**

   Use the `ftprestart` command to restart the connections to the FTP server. For further information, see `ftpshut`(8) and `ftprestart`(8).

▼ How to Change the ProFTPD Configuration

Most configuration variations are made by making changes to the `/etc/proftpd.conf` file. Use the following steps when making changes to this file.

1. **Become an administrator.**

   For more information, see "How to Use Your Assigned Administrative Rights" in *Oracle Solaris 11.1 Administration: Security Services*.

2. **Make changes to the configuration file.**

   See the simple examples below for suggestions about what information to add to the configuration file.

3. **Restart the FTP server.**

```
# svcadm restart network/ftp
```

Example 2–1 ProFTPD Configuration File Changes for a Virtual Host

For a virtual host that is using a fixed IP address, use the following directive. You may add multiple IP addresses separated by spaces if needed.

```
<VirtualHost 10.0.0.1>
  ServerName "My virtual FTP server"
</VirtualHost>
```

Example 2–2 ProFTPD Configuration File Changes for Anonymous Access

To provide anonymous ftp access to your site, use these directives:

```
# Deny login access
  <Limit LOGIN>
    DenyAll
  </Limit>

  <Anonymous ~ftp>

# Allow anonymous logins
    <Limit LOGIN>
      AllowAll
    </Limit> ....
  </Anonymous>
```

◆ ◆ ◆ CHAPTER 3

Accessing Remote Systems (Tasks)

This chapter describes all the tasks that are required to log in to remote systems and work with their files. This is a list of the topics in this chapter.

Accessing Remote Systems (Task Map)

This chapter provides tasks that are described in the following table. You can use these tasks to log in and copy files from remote systems.

TABLE 3–1 Task Map: Accessing Remote Systems

Task	Description	For Instructions
Log in to a remote system (`ssh`)	Use the `ssh` command to access a remote system.	"Logging In to a Remote System (`ssh`)" on page 20
Log in to a remote system (`sftp`) to access files.	Use the `sftp` command to send and receive files on a remote system: ■ Open and close an `sftp` connection. ■ Copy files to and from a remote system.	"How to Open and Close an `sftp` Connection to a Remote System" on page 23 "How to Copy Files From a Remote System (`sftp`)" on page 24 "How to Copy Files to a Remote System (`sftp`)" on page 25
Copy remote files with `scp`.	Use the `scp` command to copy files to and from a remote system.	"How to Copy a File Between Two Systems (`scp`)" on page 27
Exit or close a remote system session using `exit` or `logout`.	Use the `exit` or `logout` commands to terminate a session.	"Logging Out of a Remote System (`exit` or `logout`)" on page 22

Logging In to a Remote System (ssh)

The Secure Shell feature of Oracle Solaris provides secure access to a remote host over an unsecured network. In Secure Shell, authentication is provided by the use of passwords, public keys, or both. All network traffic is encrypted. Thus, Secure Shell prevents a would-be intruder from being able to read an intercepted communication. Secure Shell also prevents an adversary from spoofing the system.

Note – Oracle Solaris 11.1 is installed as secure by default. Only the `ssh` command accepts incoming network requests. Therefore, remote login should only be done using `ssh`.

If the system you are logging in to is in a remote domain, be sure to append the domain name to the system name. In the following example, `example.com` is the name of the remote domain, `venus` is the host name, and `jdoe` is the user:

```
ssh –l jdoe venus.example.com
```

Authentication for Remote Logins (ssh)

Authentication (establishing who you are) for `ssh` logins is provided by a combination of system passwords and public host keys. Authentication operations can be performed either by the remote system or by the network environment.

The /etc/ssh_known_hosts and .ssh/known_hosts Files

The `/etc/ssh_known_hosts` and the `.ssh/known_hosts` files contain the list of known host keys on the system or account. By default, the `ssh` command verifies the remote host's key. The `ssh` command next checks the host keys stored in the `/etc/ssh_known_hosts` and `~/.ssh/known_hosts` files. If there is no host key for the remote host in either file, the user is asked whether they trust the new remote host's key. If the user confirms, the remote host's key is then added to the user's `~/.ssh/known_hosts` file before the user is prompted for their password.

In the following example, user `jsmith` is logging in to the remote host `mars.example.com` for the first time.

```
$ ssh -l jsmith mars.example.com
The authenticity of host 'mars.example.com (10.80.226.105)' can't be established.
RSA key fingerprint is 06:55:4d:4e:d2:4a:e6:d9:8a:c4:13:15:18:9a:ef:dd.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'mars.example.com' (RSA) to the list of known hosts.
Password:
Last login: Thu Jun 21 15:44:54 2012 from venus.example
Oracle Corporation      SunOS 5.11      11.1    June 2012
$
```

The RSA key fingerprint for mars.example.com is now in jsmith/.ssh/known_hosts. Subsequent logins by jsmith to mars.example.com will prompt for a system password. For example:

```
$ ssh -l jsmith mars.example.com
Password:
Last login: Thu Jun 21 15:45:47 2012 from mars.example
Oracle Corporation      SunOS 5.11      11.1    June 2012
$
```

Note – By default, you are unable to use the ssh command to log in to a system as root or as a privileged user.

For further information on Secure Shell authentication, refer to "Secure Shell Authentication" in *Oracle Solaris 11.1 Administration: Security Services*.

What Happens After You Log In Using Secure Shell

When you attempt to log in to a remote system with the ssh command, the sshd daemon on the remote system starts the login program. The login program then performs the same tasks as when you log in locally. The program authenticates the user, and if successful, it executes the user's login shell.

If the home directory for the user cannot be found, you are then assigned to the remote system's root (/) directory. For example:

```
Unable to find home directory, logging in with /
```

▼ How to Log In to a Remote System (ssh)

- **Log in to a remote system by using the ssh command.**

  ssh [**-l** *login-name*] *hostname* | *login-name@hostname*

  login-name Logs you in to the remote system with the user name you supply

  hostname The name of the remote system that you are logging into

  If the system's host key is verified, you are asked to provide a password. If the password is typed incorrectly, the system states the number of authentication failures and the last date of successful login.

  If the operation succeeds, the following is displayed: information about your last remote login to that system, the version of the operating system that is running on the remote system, and whether you have mail waiting for you in your home directory.

Example 3–1 Logging In to a Remote System (ssh)

The following example shows the output of a remote login to pluto. The system's host key has not been identified in either the /etc/ssh_known_hosts or the .ssh/known_hosts file, and the user has incorrectly entered in the password at the first attempt:

```
$ ssh -l amy pluto
The authenticity of host 'pluto (10:120:100:12)' can't be established.
RSA key fingerprint is 06:55:4d:4e:d2:4a:e6:d9:8a:c4:13:15:18:9a:ef:dd.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'pluto' (RSA) to the list of known hosts.
Password:
Password:
Warning: 1 failed authentication attempt at Wed Jun 27 12:47 2012 since last successful authentication.

Last login: Wed Jun 27 12:19:04 2012 from venus.us.
Oracle Corporation      SunOS 5.11      11.1    June 2012
$
```

Logging Out of a Remote System (exit or logout)

You can log out of a remote system by using either the exit or the logout command.

```
$ exit
```

```
$ logout
```

The Control-D command sequence also ends a session on a remote system. For example, on the host venus:

```
$ Control-D
Connection to venus closed.
```

Logging In to a Remote System to Copy a File (sftp)

The sftp command is an interactive file transfer program with a user interface similar to ftp. However, sftp uses the SSH File Transfer Protocol to create a secure connection to the server. Not all options available with the ftp command are included in the sftp command, but many of them are.

Essential sftp Commands

The following table lists essential sftp commands.

TABLE 3–2 Essential sftp Commands

Command	Description
`sftp` *remote-system*	Establishes an `sftp` connection to a remote system. For instructions, see "How to Open and Close an sftp Connection to a Remote System" on page 23.
`sftp` *remote-system:file*	Copies the named *file* from *remote-system*.
`bye`	Quits the `sftp` session.
`help`	Lists all `sftp` commands.
`ls`	Lists the contents of the remote working directory.
`lls`	Lists the contents of the local working directory.
`pwd`	Displays the name of the remote working directory.
`cd`	Changes the remote working directory.
`lcd`	Changes the local working directory.
`mkdir`	Creates a directory on the remote system.
`rmdir`	Deletes a directory on the remote system.
`get`	Copies a file from the remote working directory to the local working directory.
`put`	Copies a file from the local working directory to the remote working directory.
`delete`	Deletes a file from the remote working directory.

For more information, see the `sftp(1)` man page.

▼ How to Open and Close an `sftp` Connection to a Remote System

1 Open a connection to a remote system by using the `sftp` command.

```
$ sftp remote-system
```

If the connection succeeds, a confirmation message and prompt are displayed.

2 If prompted, type your password.

```
Password: password
```

If the `sftp` interface accepts your password, it displays a confirmation message and the (`sftp>`) prompt.

You can now use any of the commands that are supplied by the sftp interface, including help. The principal commands are summarized in Table 3–2.

3 **Close the sftp connection.**

```
sftp> bye
```

Example 3–2 Opening an sftp Connection to a Remote System

This sftp session was established to connect to the remote system pluto:

```
$ sftp pluto
Connecting to pluto.
Password: password
sftp>
```

▼ How to Copy Files From a Remote System (sftp)

1 **Establish an sftp connection.**

See "How to Open and Close an sftp Connection to a Remote System" on page 23.

2 **(Optional) Change to a directory on the local system where you want the files copied to.**

```
sftp> lcd target-directory
```

3 **Change to the source directory.**

```
sftp> cd source-directory
```

4 **Ensure that you have read permission for the source files.**

```
sftp> ls -l
```

5 **To copy a file, use the get command.**

Metacharacters may be used with the get command.

```
sftp> get filename
```

6 **Close the sftp connection.**

```
sftp> bye
```

Example 3–3 Copying a File From a Remote System (sftp)

In this example, the user opens an sftp connection to the system pluto, and uses the get command to copy a single file from the /tmp directory.

```
$ sftp pluto
Connecting to pluto...
Password: xxx
```

```
sftp> lcd /tmp
sftp> cd /tmp
sftp> ls
filea
files
ps_data
sftp> get filea
/tmp/filea                                    100%  494     0.5KB/s   00:00
sftp> bye
```

▼ How to Copy Files to a Remote System (sftp)

1. **Change to the source directory on the local system.**

   The directory from which you type the sftp command is the local working directory and thus the source directory for this operation.

2. **Establish an sftp connection.**

   See "How to Open and Close an sftp Connection to a Remote System" on page 23.

3. **You can change to the target directory.**

   ```
   sftp> cd target-directory
   ```

4. **Ensure that you have write permission in the target directory.**

   ```
   sftp> ls -l target-directory
   ```

5. **To copy a single file, use the put command.**

   Metacharacters may be used with the get command.

   ```
   sftp> put filename
   ```

6. **Close the sftp connection.**

   ```
   sftp> bye
   ```

Example 3–4 Copying a File to a Remote System (sftp)

In this example, the user opens an sftp connection to the system pluto, and uses the put command to copy a file from their system to the /tmp directory on system pluto.

```
$ cd /tmp
$ sftp pluto
Password: xxx
sftp> cd /tmp
sftp> put filef
uploading filef to /tmp/filef
filef                                         100%  325     0.3KB/s   00:00
sftp> ls
filea
```

```
filef
files
sftp> bye
```

Remote Copying With the scp Command

The scp command copies files or directories between a local and a remote system or between two remote systems. You can use this command from a remote system (after logging in with the ssh command) or from the local system. The scp command uses ssh for data transfer. Thus, the scp command uses the same authentication and provides the same security as the ssh command.

With scp, you can perform the following remote copy operations:

- Copy a file or directory from your local system to a remote system
- Copy a file or directory from a remote system to your local system
- Copy a file or directory between remote systems from your local system

Security Considerations for Copy Operations

To copy files or directories between systems, you must have permission to log in and copy files.

The scp command, as a component of the ssh command, requires that you have either a user account or host key access to the target system. Consult Chapter 15, "Using Secure Shell," in *Oracle Solaris 11.1 Administration: Security Services* for further information.

Caution – Both the cp and scp commands can overwrite files without warning. Ensure that file names are correct before executing the command.

Specifying the Source and Target for Copy Operations

With the scp command, you can specify the source (the file or directory to be copied) and the target (the location in which to copy the file or directory). You can shorten the path strings by using the tilde character (~) and the shell wildcard characters (*, ?, and so forth).

The tilde character (~) is expanded by all shell programs to be the current user's home directory. The current user is the user under which the shell is executing. If the home directory for the user jack is /export/home/jack, then for the user jack, ~/myfile.txt expands to /export/home/jack/myfile.txt.

This expansion also works for remote paths. If the user `jack` wants to copy a file from his home directory, then these three path descriptions are equivalent:

- `mars:/export/home/jack/myfile.txt`
- `mars:~/myfile.txt`
- `mars:myfile.txt`

This expansion is also useful when referring to another user's remote home directory. In this case, you would include the user's name after the tilde character. For the user `jack`, `mars:~jill/myfile.txt` is equivalent to `mars:/export/home/jill/myfile.txt`, but it is shorter to type.

▼ How to Copy a File Between Two Systems (scp)

1 Ensure that you have permission to copy files on the target system.

The `scp` command requires authentication. Depending upon the method of authentication used, you must have either an account on the target system, or an authorized public key on the target system. You should at least have read permission on the source system and write permission on the target system.

Caution – If you do not have an account on the target system, or if the target system is not configured to allow public keys, you will receive an authentication error. For example:

```
$ scp mars:/var/tmp/testdir/letter.txt .
Permission denied (gssapi-keyex,gssapi-with-mic,publickey,keyboard-interactive)
```

Ensure that you have either a user account or public key access configured on the target system for authentication. See "Secure Shell Authentication" in *Oracle Solaris 11.1 Administration: Security Services*.

2 Determine the location of the source and target.

If you don't know the path of the source or target, you can first log in to the remote system with the `ssh` command, as described in "Logging In to a Remote System (ssh)" on page 20. Then, navigate through the remote system until you find the location. You can then perform the next step without logging out of the remote system.

3 Copy the file or directory.

`$ scp [-r] [[`*user1*`@]`*hostname1*`:]`*file1* `...` `[[`*user2*`@]`*hostname2*`:]`*file2*

`-r`	Use to recursively copy entire directories.
`user1`, `user2`	Login account to use on the remote host.
`hostname1`, `hostname2`	The names of the remote host from or to which the file is to be copied.

file1	The file name or directory name to be copied. Several source file names may be included on one command line.
file2	The destination file name or directory name.

Example 3–5 Using the scp Command to Copy a Remote File to a Local System

In this example, scp is used to copy the file letter.doc from the /home/jones directory of the remote system pluto to the working directory on the local system.

```
$ scp pluto:/home/jones/letter.doc .
The authenticity of host 'pluto (192.168.56.102)' can't be established.
RSA key fingerprint is b4:88:7b:cf:f5:23:d3:ad:0b:14:22:31:74:7b:6c:74.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.56.102' (RSA) to the list of known hosts.
Password:
letter.txt           100% |*****************************|    23       00:00
$
```

In this example, this is the first time the system pluto is accessed, hence, the RSA key fingerprint message.

Here, the " . " symbol at the end of the command line refers to the current working directory on the local system.

Example 3–6 Using the scp Command to Copy a Local File to a Remote System

In this example, scp is used to copy the file notice.doc from the home directory (/home/smith) of the local system earth to the /home/jones directory of the remote system, pluto.

```
$ scp notice.doc pluto:/home/jones
Password:
notice.doc           100% |*****************************|     0       00:00
```

Because no remote file name is provided, the file notice.doc is copied into the /home/jones directory with the same name.

In the following example, the scp operation from the previous example is repeated, but scp is executed from a different working directory on the local system (/tmp). Note the use of the "~" symbol to refer to the current user's home directory:

```
$ scp ~/notice.doc pluto:/home/jones
Password:
notice.doc            100% |*****************************|     0       00:00
```

Index