

OpenSSH for NetWare® Administration Guide

Novell® Open Enterprise Server

2 SP1

December, 2008

www.novell.com



Legal Notices

Novell, Inc. makes no representations or warranties with respect to the contents or use of this documentation, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Novell, Inc. reserves the right to revise this publication and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes.

Further, Novell, Inc. makes no representations or warranties with respect to any software, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Novell, Inc. reserves the right to make changes to any and all parts of Novell software, at any time, without any obligation to notify any person or entity of such changes.

Any products or technical information provided under this Agreement may be subject to U.S. export controls and the trade laws of other countries. You agree to comply with all export control regulations and to obtain any required licenses or classification to export, re-export, or import deliverables. You agree not to export or re-export to entities on the current U.S. export exclusion lists or to any embargoed or terrorist countries as specified in the U.S. export laws. You agree to not use deliverables for prohibited nuclear, missile, or chemical biological weaponry end uses. Please refer to www.novell.com/info/exports/ for more information on exporting Novell software. Novell assumes no responsibility for your failure to obtain any necessary export approvals.

Copyright © 2008 Novell, Inc. All rights reserved. No part of this publication may be reproduced, photocopied, stored on a retrieval system, or transmitted without the express written consent of the publisher.

Novell, Inc. has intellectual property rights relating to technology embodied in the product that is described in this document. In particular, and without limitation, these intellectual property rights may include one or more of the U.S. patents listed at <http://www.novell.com/company/legal/patents/> and one or more additional patents or pending patent applications in the U.S. and in other countries.

Novell, Inc.
404 Wyman Street, Suite 500
Waltham, MA 02451
U.S.A.
www.novell.com

Online Documentation: To access the online documentation for this and other Novell products, and to get updates, see www.novell.com/documentation.

Novell Trademarks

For a list of Novell trademarks, see the [Novell Trademark and Service Mark List \(http://www.novell.com/company/legal/trademarks/tmlist.html\)](http://www.novell.com/company/legal/trademarks/tmlist.html).

Third-Party Materials

All third-party trademarks are the property of their respective owners.

Contents

About This Guide	7
1 Overview of OpenSSH on NetWare	9
1.1 Benefits of OpenSSH	9
1.2 Functions Unique to the NetWare Platform	10
1.3 What's Next	10
2 Setting Up OpenSSH in Your Network	13
2.1 Setting Up SSH on a Server	13
2.1.1 Completing Post-Installation Configuration	13
2.1.2 Public/Private Key Security Risks	20
2.2 Setting Up SSH at Workstations	21
3 Using SSH Commands	23
3.1 Running Commands from a Workstation or Server	23
3.2 Using SSH Command Options.	25
3.3 Running Keyboard Commands at the SSH Server Console Screen	26
A Documentation Updates	29
A.1 December 2008	29
A.2 September 1, 2007.	29
A.3 May 1, 2006	29
A.4 November 1, 2005	29
A.5 July 15, 2005	29
A.6 June 6, 2005	30
A.7 June 1, 2005	30

About This Guide

This guide describes how to set up and use the OpenSSH open source data encryption program that has been integrated with NetWare®. This product provides a secure shell with encryption for use when accessing NetWare servers remotely. This guide is divided into the following sections:

- ♦ Chapter 1, “Overview of OpenSSH on NetWare,” on page 9
- ♦ Chapter 2, “Setting Up OpenSSH in Your Network,” on page 13
- ♦ Chapter 3, “Using SSH Commands,” on page 23

Audience

The majority of this guide is intended for network administrators. A few sections include information for end users.

Feedback

We want to hear your comments and suggestions about this manual and the other documentation included with this product. Please use the User Comments feature at the bottom of each page of the online documentation, or go to [Novell Documentation Web site \(http://www.novell.com/documentation/feedback.html\)](http://www.novell.com/documentation/feedback.html) and enter your comments there.

Documentation Updates

The latest version of this documentation is available at the [OES documentation Web site \(http://www.novell.com/documentation/oes/index.html\)](http://www.novell.com/documentation/oes/index.html).

Additional Documentation

Additional OpenSSH documentation is located on the [Web at www.openssh.com \(http://www.openssh.com\)](http://www.openssh.com).

Documentation Conventions

In this documentation, a greater-than symbol (>) is used to separate actions within a step and items within a cross-reference path.

A trademark symbol (®, ™, etc.) denotes a Novell® trademark. An asterisk (*) denotes a third-party trademark.

When a single path name can be written with a backslash for some platforms or a forward slash for other platforms, the path name is presented with a forward slash to reflect the Linux* convention. Users of platforms that require a backslash, such as NetWare, should use backslashes as required by your software.

Overview of OpenSSH on NetWare

1

OpenSSH is an open source technology that has been integrated with NetWare®. It provides a secure shell that uses encryption provided by Novell® International Cryptographic Infrastructure (NICI) technology rather than SSL to implement 128-bit (and stronger) encryption and contains fewer software import liabilities.

In NetWare 6.5, Novell has integrated [OpenSSH \(http://www.openssh.com\)](http://www.openssh.com) to work on NetWare so that administrators and users can access NetWare servers in their networks using methods that provide secure access and transmission of data.

As Admin or equivalent, you can gain remote access to any server in your network and copy files and directories to and from other servers in your network using SSH utilities. You can also put these commands in script files to automate routine tasks.

Through this shell, end users can securely access and copy files in their home directories or other directories that they have rights to on NetWare servers from remote locations without the use of a browser or proprietary client.

Many users of telnet, rlogin, ftp, and other such programs might not realize that their passwords and data are transmitted across the Internet unencrypted. OpenSSH encrypts all traffic (including passwords) to effectively eliminate eavesdropping, connection hijacking, and other network-level attacks. Additionally, OpenSSH provides a myriad of secure tunneling capabilities.

The OpenSSH suite integrated with NetWare 6.5 includes:

- ♦ The ssh program that replaces rlogin and Telnet
- ♦ scp (replaces rcp)
- ♦ sftp (an alternative to ftp)
- ♦ sshd (server side of the package)
- ♦ Other basic utilities like ssh-keygen or sftp-server

OpenSSH supports SSH protocol versions 1.3, 1.5, and 2.0.

Understanding the following terminology will be helpful as you use this guide:

- ♦ OpenSSH: The open source product
- ♦ SSH: The SSH protocols within the OpenSource product
- ♦ ssh: The client utility

1.1 Benefits of OpenSSH

The following is a brief list of some of the benefits of integrating OpenSSH with NetWare.

- ♦ End users can securely access and copy files in their home directories on NetWare servers from remote locations without the use of a browser or propriety client.
- ♦ Network administrators can gain remote access to any server in their networks and copy files and directories to and from other servers in their networks using ssh utilities. They can also use these commands in script files to automate many routine tasks.

- ♦ Because the SSH client protocols have also been ported to NetWare, network administrators can use the SSH commands from a remote client or from a remote server on the network running NetWare 6.5 to copy files from one server to another server.
- ♦ SSH protocols allow you to connect to the server and automatically send a command, so the server runs that command, then disconnects. This means you can use automated processes.
- ♦ SSH protocols provide security of your data transmissions and communications across the Internet whether you are outside or inside a firewall. You can be confident that hackers cannot access your data.

1.2 Functions Unique to the NetWare Platform

Integrating OpenSSH with NetWare adds the functionality of using SSH on a NetWare server easier. Some commands work differently on NetWare than they do in other SSH implementations.

Added Functionality

- ♦ **OpenSSH Manager:** Any user that belongs to the `sshadmin-Administrators` group is granted access to the OpenSSH Manager to modify the configuration of OpenSSH servers. The OpenSSH Manager can be accessed via Web browser `ssl` connection to port 2200. This tool lets you view SSH connections, change the `sshd_config` file more easily, set log preferences, etc.
- ♦ **SSH Log Daemon:** This agent generates the log files that contain all the logs and errors sent from all `ssh`-type NLM™ programs such as `sshd`, `ssh`, `sftp`, or `scp`.
- ♦ **Authentication:** OpenSSH on NetWare supports two modes of authentication.
 - ♦ Password authentication through LDAP. This authentication gathers all the user's credentials from Novell eDirectory™ 8.7.3.
 - ♦ Public key authentication. This authentication uses the contents of `sshd.bag` to verify a user's key and then eDirectory using the matching user's credentials.

After users have authenticated, the current working directory is their home directory if configured in eDirectory; otherwise, they will be at the root of the server volumes of the server they connected to. Users can navigate like they would with `ftp` to any directory on that server for which they have been assigned rights in eDirectory.

Differences

- ♦ **The localhost commands:** The `ssh localhost` command does not work on a NetWare server; however, the `scp localhost` and `sftp localhost` commands do work.
- ♦ **SSH public/private key administration:** Public keys are stored in a NCI-encoded bag at `sys:/etc/ssh/sshd.bag`. Users' home `.ssh` directory and `AuthorizedKeyFile` configuration settings are not supported.

1.3 What's Next

Now that you know a little about the SSH protocols that have been ported to NetWare and what some of the benefits of using it are, you can continue with the following tasks.

To	See
Set up SSH on your server	Section 2.1, “Setting Up SSH on a Server,” on page 13
Download an SSH-compliant client on a workstation	Section 2.2, “Setting Up SSH at Workstations,” on page 21

Setting Up OpenSSH in Your Network

2

Setting up OpenSSH in your network involves the following tasks:

- ♦ [Section 2.1, “Setting Up SSH on a Server,” on page 13](#)
- ♦ [Section 2.2, “Setting Up SSH at Workstations,” on page 21](#)

2.1 Setting Up SSH on a Server

As a prerequisite, it is recommended that you install the Apache Administration server if it wasn't installed by default. The Apache Administration server is normally installed by default unless you installed a special-purpose server that didn't require it, such as iLogin, DNS/DHCP, Pre-migration NetWare®, Virtual Office, or Novell® Branch Office.

You can install OpenSSH on a server either as an optional component during the NetWare custom installation or after installing NetWare using the following procedure:

- 1 Insert the *NetWare 6.5 Operating System CD* into the CD-ROM drive of the server where you want to install OpenSSH.
- 2 Start the NetWare GUI by entering `startx` at the system console prompt.
- 3 Click *Novell > Install > Add*.
- 4 In the Source Path dialog box, type the path or browse to the CD.
- 5 Select the `postinst.ni` response file, then click OK.
- 6 On the Install Components page, select *Secure Shell* from the products list.
- 7 Click *Next*.
- 8 When prompted, enter the administrator username, password, and context.
- 9 Follow the remaining prompts.
- 10 Click *OK*.

IMPORTANT: After upgrading from a NetWare 5.1 server with eDirectory™ 7.x to a NetWare 6.5 server (which upgrades eDirectory to version 8.7), User objects don't have a uniqueid attribute, which is used by `sshd` for authentication. As a result, `sshd` falls back to the CN attribute, which is no longer public after the upgrade. The admin user must then make the CN attribute public in ConsoleOne® or iManager.

2.1.1 Completing Post-Installation Configuration

After the installation, you need to complete some additional configuration before you or your users can access files on the server.

- 1 Load the `sshd.nlm` file at the server.

- 2 (Optional) Edit the `sys:etc\ssh\sshd_config` file to change any settings from the default.
- 3 (Optional) Add users and public keys into `sshd.bag`.

IMPORTANT: OpenSSH often reports an error trying to configure the product during a remote upgrade. To fix the configuration problems, edit `sys:\etc\ssh\sshd_config` and update the default <Your-Context> tag with the admin user's context. You must also ensure that admin users have the Supervisor trustee right to the NCP™ Server object for each server in the tree that they administer. A local post-install of the OpenSSH product (from the GUI on the server) also corrects the configuration issues.

Understanding the Components

After you set up OpenSSH on your NetWare server, it should contain the components listed in [Table 2-1](#) in the indicated locations.

Table 2-1 *OpenSSH Component Locations*

File	Location	Description
<code>sshd.nlm</code>	<code>sys:/system</code>	OpenSSH version 3.6p1 ported to NetWare 6.5 This is the daemon for the SSH program. It provides secure encrypted communications between two untrusted hosts over an insecure network This daemon listens for the connections from clients
<code>sshd_config</code>	<code>sys:/etc/ssh</code>	System-wide configuration file for the SSH daemon. The daemon reads the configuration file and executes the commands it receives based on the file's settings You can edit this file manually or through the Web administration utility. For more information, see "Editing the Configuration File" on page 15
<code>ssh_host_key</code>	<code>sys:/etc/ssh</code>	Private host key used to authenticate the server for the SSH protocol versions 1.3 and 1.5
<code>ssh_host_rsa_key</code>	<code>sys:/etc/ssh</code>	Private host key used to authenticate the server for the SSH protocol version 2.0 using RSA encryption
<code>ssh_host_dsa_key</code>	<code>sys:/etc/ssh</code>	Private host key used to authenticate the server for the SSH protocol version 2.0 using DSA encryption
<code>sshjni.nlm</code>	<code>sys:/system</code>	Secure Shell JNI Web support
<code>sshlogd.nlm</code>	<code>sys:/system</code>	Secure Shell log daemon that generates the <code>sshd.log</code> file, which contains all errors sent from all ssh-type NLM™ programs such as <code>sshd</code> , <code>ssh</code> , <code>sftp</code> , and <code>scp</code> This NLM is not a standard ssh file. This ssh module only exists on the NetWare platform
<code>ssh-pubuadd</code>	<code>sys:/system</code>	Adds a user plus the user's public key to the local secret store bag

File	Location	Description
ssh-pubudel	sys:/system	Deletes a user from the local secret store bag
ssh-pubulist	sys:/system	Lists users in the local secret store bag

Editing the Configuration File

The `sshd_config` file is located in `sys/etc/ssh/`. You can edit this file manually with any text editor. If your server has been set up with a DNS name, you can make changes to the file using the OpenSSH Admin utility.

We recommend making changes to the configuration using the OpenSSH Manager (OpenSSH Admin) utility because it eliminates syntax errors that you might make editing the file manually. If you manage OpenSSH on multiple servers, we recommend using this utility to import the configuration file to the eDirectory 8.7.3 mode and then also managing the configuration with the utility.

IMPORTANT: The Apache Admin utility must be installed and set up in order to use the OpenSSH Admin utility.

To access this utility from a browser (Netscape* 6.x or later or IE 5.5 or later):

- 1 Enter `https://ip_address_or_server_dns_name:2200`, then click the *SSHD Admin* link under the *OpenSSH Server* heading.
- 2 Type the password information.
- 3 Ensure the information automatically inserted into the following fields is applicable to the user and server that you want to log in to:
 - ♦ User Name
 - ♦ LDAP Provider Domain Name
 - ♦ Port Number 636 (or whatever it has been changed to)
 - ♦ The Use SSL Connection check box (checked)

If this check box is not checked, your password to log in to sshd will be exposed in clear text.
 - ♦ The initial LDAP context

Changing the Options

The following table shows the options that you can change in the `sshd_config` file and the links that you can use for them in the OpenSSH Admin utility. All keyword purposes and options are specified in the `sshd_config` man pages (http://www.openbsd.org/cgi-bin/man.cgi?query=sshd_config&sektion=5&arch=&apropos=0&manpath) unless they are specific to a NetWare implementation.

Table 2-2 *sshd_config Options*

Option	Description	Link in Admin Utility
AllowSSHSessions	<p>Specifies whether or not sshd should allow SSH console session access.</p> <p>The default is Yes.</p>	Security
AuthorizedKeyFile	<p>Path to the file that contains the authorized keys.</p> <p>The default is <code>.ssh\authorized_keys</code>.</p>	Ignored
ChallengeResponseAuthentication	<p>Challenges the user to supply authentication credentials. If the user responds with correct credentials, authentication is allowed. This is currently the only way to authenticate to a NetWare server with OpenSSH.</p> <p>The default is Yes.</p>	Authentication
ClientAliveCountMax	<p>Number of client alive messages that can be sent without sshd requiring any messages back from the client. If this threshold is reached while client alive messages are being sent, sshd disconnects the client, terminating the session.</p> <p>This is very different from KeepAlive. The client alive messages are sent through the encrypted channel and, therefore, are not spoofable. Messages sent by KeepAlive are spoofable.</p> <p>The client alive mechanism is valuable when the client or server depends on knowing when a connection has become inactive.</p> <p>If ClientAliveCountMax is set to 2, unresponsive SSH clients are disconnected after approximately 30 seconds.</p> <p>If ClientAliveInterval is set to 15, and ClientAliveCountMax is left at the default, unresponsive SSH clients are disconnected after approximately 45 seconds.</p> <p>The default is 3.</p>	Connection
ClientAliveInterval	<p>Timeout interval (in seconds) after which, if no data has been received from the client, sshd sends a message through the encrypted channel to request a response from the client.</p> <p>The default is 0 (no messages sent).</p> <p>This option applies to protocol version 2 only.</p>	Connection

Option	Description	Link in Admin Utility
Compression	<p>Enables or disables compression, which reduces traffic on a low-bandwidth connection.</p> <p>The default is Yes (enabled).</p>	Connection
DOSFTPMultiServerNavigation	<p>Enables or disables multi-server navigation.</p> <p>If set to Yes, a server name is added to the path, as in <code>/servername/volume/dir_path</code>.</p> <p>If set to No, the path is <code>/volume/dir_path</code>.</p> <p>The default is Yes (enabled).</p>	File System
eDirNameContext	<p>Search context. Use this to expand or limit access to the tree.</p> <p>To enable users in this context only to authenticate to sshd: <code>o=org</code></p> <p>To allow users in this context and all subcontexts to authenticate to sshd: <code>o=org?scope=subtree</code></p> <p>To search for a user in multiple contexts: context <i>context</i>?scope=subtree</p> <p>This setting is unique to a NetWare implementation.</p>	eDirectory
HostKey	<p>These keys are generated during the OpenSSH installation on NetWare:</p> <p><code>etc\ssh\ssh_host_key</code></p> <p><code>etc\ssh\ssh_host_rsa_key</code></p> <p><code>etc\ssh\ssh_host_dsa_key</code></p>	Host Keys
IgnoreRemoteHomeDir	<p>Specifies whether sshd should ignore the user's home directory unless on the destination server.</p> <p>The default is No.</p>	File System
IgnoreUserKnownHosts	<p>Specifies whether sshd should ignore the user's <code>\$home/.ssh/known_hosts</code> file during RhostsRSAAuthentication or HostbasedAuthentication.</p> <p>This file contains a copy of the key that the host sent the last time a connection was made. If the file is not ignored, the server prompts the user every time that user attempts to connect, asking whether the key should be accepted.</p> <p>The default is No.</p>	Authentication

Option	Description	Link in Admin Utility
KeepAlive	<p>Specifies whether the system should send TCP keepalive messages to the other side. If they are sent, events such as the termination of the connection or the crash of one of the machines are noticed. However, this means that connections terminate if the route is down temporarily. The client detects whether the network goes down or the remote host crashes.</p> <p>The default is Yes.</p>	Connection
KeyRegenerationInterval	<p>Time (in seconds) between regeneration of keys. This prevents decrypting captured sessions by later breaking into the machine and stealing the keys. The key is never stored anywhere. If the value is 0, the key is never regenerated.</p> <p>The default is 3600.</p>	Authentication
ListenAddress	<p>Address for the SSH client to listen on.</p> <p>The default is 0.0.0.0</p>	Listen Address
LoginBannerFile	<p>Path to a file that contains a greeting or specific banner text that displays when the user logs in to the server using an SSH client.</p> <p>Recommended path: <code>sys:\etc\ssh</code></p> <p>The default is None.</p>	Connection
LoginGraceTime	<p>Time interval (in seconds) before the server disconnects if the user has not successfully logged in. If the value is set to 0, there is no time limit.</p> <p>The default is 600.</p>	Connection
LogLevel	<p>Verbosity level that is used when logging messages from sshd.</p> <p>The default is Info.</p>	Log Preferences
LogMaxFileSize	<p>Size (in MB) for the log files.</p> <p>The default is 4.</p> <p>This setting is unique to a NetWare implementation.</p>	Log Preferences
LogMaxRotateFiles	<p>Maximum time (in hours) for logging to occur in one file if the default size is not reached.</p> <p>The default is 7.</p> <p>This setting is unique to a NetWare implementation.</p>	Log Preferences

Option	Description	Link in Admin Utility
LogPath	Path to the log file. The recommended location is <code>sys:\etc\ssh\logs</code> . This setting is unique to a NetWare implementation.	Log Preferences
LogRotationInterval	Maximum time (in hours) for logging to occur in one file if the default size is not reached. The default is 24. This setting is unique to a NetWare implementation.	Log Preferences
PasswordAuthentication	Uses a username and password to verify a user's identity. This is currently the only way to authenticate to a NetWare server with OpenSSH. Even if you do not select Yes to enable Password Authentication, Password Authentication is still used for NetWare servers. The default is Yes.	Authentication
Port	Port for SSH to listen on. The default is Port 22.	Listen Ports
Protocol	Versions of the SSH protocol that are supported.	Miscellaneous
ProxyName <i>username</i> ProxyPassword <i>password</i>	Specifies the proxy user and password for LDAP searches. This is useful when anonymous binds are disabled. The username must be in fully-qualified LDAP format; for example, <code>cn=admin,o=novell</code> .	Authentication
PubKeyAuthentication	Uses cryptographic keys to verify a user's identity. A public key is stored on the server. When a user attempts to authenticate, that user's private key is verified against the public key to authenticate the user. This is currently the only way to authenticate to a NetWare server with OpenSSH. The default is Yes.	Authentication
RSAAuthentication	Allows/disables authentication using identity keys encoded with the Rivest-Shamir-Adleman (RSA) algorithm. This is currently the only way to authenticate to a NetWare server with OpenSSH. The default is Yes. This option applies to protocol version 1 only.	Authentication

Option	Description	Link in Admin Utility
ServerKeyBits	Number of bits in the ephemeral protocol version 1 server key. The larger the number of bits, the more secure the key is. If the server detects a change in this number, there could possibly be a security breach. The default is 768.	Authentication
VerifyReverseMapping	Specifies whether sshd should try to verify the remote hostname and whether the authentication request is coming from the IP address it claims to be coming from. The default is No.	Authentication

2.1.2 Public/Private Key Security Risks

Supporting public/private key authentication in OpenSSH introduces some security issues that you need to be aware of.

One User Can Masquerade as Another

A user could use SSH to send his key with the Fully Distinguished Name (FDN) of another user to gain access to the system as the other user.

For example, say you are user Sally on Linux system Foo and you want to ssh into NetWare system Bar with the intent of gaining admin permissions. On Foo, you generate ssh keys as Sally. On Bar, you add those keys into the local secret store with the Admin user's FDN using the following command:

```
ssh-pubuadd -n cn=admin,o=novell -k ./sally.pub
```

This example assumes that the administrator of Bar has previously added Admin into the secure bag and that the password was set in a previous session.

Now, on Foo, you can enter the following command to gain the eDirectory permissions of the Admin user:

```
sftp cn=admin,o=novell@bar
```

To counter this threat, NetWare administrators must do the following:

- ♦ Verify that the user's FDN matches the name of the user before adding the FDN and key into the secure bag.
- ♦ Secure the console so that commands such as `ssh-pubuadd` can not be run by unauthorized users.

Secure Bag Maintenance

The ssh daemon secure bag, `sshd.bag`, can not be copied to another server. It is good practice to save a copy of the `sshd.bag` file for migration to other servers or in the unlikely case of bag file corruption.

- ♦ To export the secure bag information to a backup text file, enter the following command:

```
ssh-pubulist -b > sshd-bag.bak
```

The resulting text file does not contain passwords. They will have to be re-entered in another interactive session.

- ♦ To import entries from the backup file into a new secure bag, enter the following command:

```
ssh-pubuadd -b ./sshd-bag.bak
```

2.2 Setting Up SSH at Workstations

To access files using SSH commands from a workstation:

- 1 Download and run an SSH-compliant client.

Here is a list of some SSH-compliant clients that you could run:

- ♦ SUSE® Linux openssh-clients package (tested with NetWare 6.5)
- ♦ Red Hat® Linux openssh-clients package (tested with NetWare 6.5)
- ♦ PuTTY (tested with NetWare 6.5)
- ♦ Absolute Telnet
- ♦ MindTerm

You can get these clients from open source software sites on the Internet such as [SourceForge \(http://www.sourceforge.net\)](http://www.sourceforge.net).

NOTE: Terminal should be configured for VT100 emulation when using SSH command line utilities.

- 2 In any of the clients, change the *Window Row* setting from the default to a value greater than 25.

After SSH is set up on the server and at the users' workstations, you can use different SSH commands and utilities to:

- ♦ Perform tasks such as copy files, run scripts, and execute server commands
- ♦ Manage your SSH connections
- ♦ Troubleshoot problems with SSH

For information on using the SSH commands and utilities, see [Chapter 3, "Using SSH Commands," on page 23](#).

Using SSH Commands

3

This section includes instructions for accomplishing the following tasks

- ♦ [Section 3.1, “Running Commands from a Workstation or Server,” on page 23](#)
- ♦ [Section 3.2, “Using SSH Command Options,” on page 25](#)
- ♦ [Section 3.3, “Running Keyboard Commands at the SSH Server Console Screen,” on page 26](#)

3.1 Running Commands from a Workstation or Server

After downloading an SSH-compliant client to your workstation, you can use the commands listed in [Table 3-1](#) to accomplish tasks on the NetWare[®] server. The ssh, scp, and sftp client protocols have been ported to the server so you can execute these commands in server-to-server connections as well.

Table 3-1 SSH command tasks

Type	To
ssh	Connect and log into the specified server (hostname). You must provide your identity to the remote machine. See Section 3.2, “Using SSH Command Options,” on page 25 for a list of SSH command options. For more information, see the ssh information at openssh.com on the Web (http://www.openbsd.org/cgi-bin/man.cgi?query=ssh).
sshd	Control how the daemon logs you in. For options and more information, see the sshd information at openssh.com on the Web (http://www.openbsd.org/cgi-bin/man.cgi?query=sshd).
ssh-add	Not supported on NetWare.
ssh-agent	Not supported on NetWare. NetWare only supports password authentication.
ssh-keygen	Generate, manage, and convert authentication keys for ssh. For more information, see ssh-keygen information at www.openssh.com on the Web (http://www.openbsd.org/cgi-bin/man.cgi?query=ssh-keygen).
ssh-keyscan	Not supported on NetWare.

Type	To
ssh-pubuadd	<p>Add a user plus the user's public key to the local secret store bag.</p> <p>Syntax: <code>ssh-pubuadd [[-n <i>FDN</i>] [-k <i>public_key_filename</i>]] [-b <i>batch_file</i>]</code></p> <p>where:</p> <ul style="list-style-type: none"> ♦ <code>-n</code> specifies the Fully Distinguished Name of the user in LDAP format ♦ <code>-k</code> specifies the public key filename ♦ <code>-b</code> specifies the name of a batch file containing multiple users, with the FDN and public key string for each user on separate lines <p>Usage examples:</p> <pre>ssh-pubuadd -n cn=admin,o=novell -k ./id_rsa.pub ssh-pubuadd -b ./bag.batch</pre>
ssh-pubudel	<p>Delete a user from the local secret store bag.</p> <p>Syntax: <code>ssh-pubudel [-n <i>FDN</i>] [-i <i>id_number</i>]</code></p> <p>where:</p> <ul style="list-style-type: none"> ♦ <code>-n</code> specifies the Fully Distinguished Name of the user in LDAP format ♦ <code>-i</code> specifies the idNum as seen from ssh-pubulist <p>Usage examples:</p> <pre>ssh-pubudel -n cn=admin,o=novell (deletes first entry found with a matching FDN) ssh-pubudel -i 20 (deletes the entry with idNum '20')</pre>
ssh-pubulist	<p>List users in the local secret store bag.</p> <p>Syntax: <code>ssh-pubulist [-l] [-b]</code></p> <p>where:</p> <ul style="list-style-type: none"> ♦ <code>-l</code> specifies long list, which includes idNum, FDN, and the complete public key string ♦ <code>-b</code> specifies batch list, suitable for batch mode input <p>If neither option is specified, the listing defaults to short list which includes idNum, FDN, and the public key comment string.</p> <p>Usage example:</p> <pre>ssh-pubulist -b > ./bag.batch (creates a batch listing for later batch mode input using ssh-pubuadd)</pre>
sftp	<p>Perform secure file transfers with an FTP-like command that works over SSH1 and SSH2 protocol.</p> <p>For command options and more information, see the sftp information at openssh.com on the Web (http://www.openbsd.org/cgi-bin/man.cgi?query=sftp).</p>

Type	To
scp	<p>Copy files between hosts on a network. It uses ssh(1) for data transfer, and uses the same authentication and provides the same security as ssh(1). Scp asks for passwords or passphrases if they are needed for authentication.</p> <p>For command options and more information, see scp information at openssh.com on the Web (http://www.openbsd.org/cgi-bin/man.cgi?query=scp).</p>
sftp-server	<p>Use the SFTP server subsystem (started automatically by sshd). This program speaks to the server side of the SFTP protocol to <code>stdout</code> and expects client requests from <code>stdin</code>.</p> <p>For more information, see ssh information at www.openssh.com on the Web (http://www.openbsd.org/cgi-bin/man.cgi?query=sftp-server).</p>

3.2 Using SSH Command Options

After downloading an SSH-compliant client to your workstation, you can send the options listed in [Table 3-2](#) with the `ssh` command to the NetWare server. The basic command syntax is:

```
ssh option host command
```

Table 3-2 *SSH command options*

Use Option	To
-a	Disable authentication agent forwarding (default)
-A	Enable authentication agent forwarding
-b <i>bind_address</i>	Specify the local IP address to transmit from on machines with multiple address or aliased addresses
-c <i>cipher</i>	Select an encryption algorithm
-C	Enable compression
-D <i>port</i>	Enable dynamic application-level port forwarding.
-e <i>escape_character</i>	Set the escape character; "none" = disable (default: ~)
-f	Fork into background after authentication
-F <i>config_filename</i>	Specify the location of the config file (default: <code>~/etc/ssh/config</code>). Requests <code>ssh</code> to go to the background just before command execution
-g	Allow remote hosts to connect to forwarded ports.
-i <i>filename</i>	Select an identity file for public key authentication (default: <code>~/.ssh/identity</code>)
-l <i>username</i>	Log in using the specified username

Use Option	To
<code>-L listen-port:host:port</code>	Forward local port to remote address This causes ssh to listen for connections on a port and forward them to the other side by connecting to host:port.
<code>-m macs</code>	Specify MAC algorithms for ssh protocol version 2.
<code>-n</code>	Redirect input from . (root)
<code>-N</code>	Do not execute a shell or command
<code>-o option</code>	Process the option as if it is read from a configuration file.
<code>-p port</code>	Connect to the specified port. The server must be on the same port.
<code>-q</code>	Do not display any warning messages
<code>-R listen-port:host:port</code>	Forward a remote port to local address This causes ssh to listen for connections on a port and forward them to the other side by connecting to host:port
<code>-s</code>	Invoke command (mandatory) as SSH2 subsystem
<code>-t</code>	Allocate a tty even if command is given
<code>-T</code>	Do not allocate a tty
<code>-v</code>	Display verbose debugging messages. Using multiple <code>-v</code> increases verbosity.
<code>-V</code>	Display version number only
<code>-z</code>	Autoclose after command execution
<code>-1</code>	Forces <code>ssh</code> to try protocol version 1 only
<code>-2</code>	Forces <code>ssh</code> to try protocol version 2 only
<code>-4</code>	Forces <code>ssh</code> to use IPv4 addresses only
<code>-6</code>	Forces <code>ssh</code> to use <code>ntwk_ipv6_nw</code> addresses only

3.3 Running Keyboard Commands at the SSH Server Console Screen

Table 3-3 lists the keyboard commands that can be executed at the `ssh`, `sftp`, or `scp` server console screen. Each connection generates a new console screen. For example, the console screen generated from a `ssh` connection would appear as `ssh username IP_address`.

Console access is granted only to the Admin user and users with security equal to Admin.

Table 3-3 *Console keyboard commands*

Press	To
Ctrl+A	Escapes the next special character and sends the special keycode to the system console. For example, Ctrl+A [1-9] sends F1 through F9, Ctrl+A[a-f] sends F10 through F15, and Ctrl+A Ctrl+[a-z] sends Ctrl-A through Ctrl-Z
Ctrl+B	Begin (Home)
Ctrl+D	Move the cursor down (Down-arrow)
Ctrl+L	Move the cursor to the left (Left-arrow)
Ctrl+U	Move the cursor to the up on the screen (Up-arrow)
Ctrl+R	Move the cursor to the right (Right-arrow)
Ctrl+F	Switch to a different server console screen. The server GUI screen is not supported
Ctrl+P	Page up
Ctrl+N	Page down
Ctrl+G	Delete
Ctrl+O	Insert
Ctrl+X	Exit
Ctrl+T	Reboot server
Ctrl+E	End
Ctrl+Z	Select screen
Ctrl+H	Backspace
Ctrl+S	Setting screen
Ctrl+Q	Display SSH keyboard help screen
Ctrl+K	Access the kernel debugger screen

Documentation Updates

A

These dates indicate when the OpenSSH documentation has been updated, and what changes have been made.

A.1 December 2008

- ♦ Guide updated to the latest file template.
- ♦ Edited for couple of changes.

A.2 September 1, 2007

Documented a new command option **to autoclose the ssh screen.**

A.3 May 1, 2006

Added information about new support for public key authentication.

A.4 November 1, 2005

The following updates were made:

- ♦ Page design reformatted to comply with revised Novell® documentation standards.

A.5 July 15, 2005

The following issues were added:

- ♦ OpenSSH often reports an error trying to configure the product during a remote upgrade. To fix the configuration problems, edit `sys:\etc\ssh\sshd_config` and update the default `<Your-Context>` tag with the admin user's context. You must also ensure that admin users have the Supervisor trustee right to the NCPTM Server object for each server in the tree that they administer. A local (from the GUI on the server) post-install of the OpenSSH product will also correct the configuration issues.
- ♦ After upgrading from a NetWare® 5.1 server with eDirectory™ 7.x to a NetWare 6.5 server (which upgrades eDirectory to version 8.7), User objects don't have a uniqueid attribute, which is used by sshd for authentication. As a result, sshd falls back to the CN attribute, which is no longer public after the upgrade. The admin user must then make the CN attribute public in ConsoleOne® or iManager.

A.6 June 6, 2005

The following updates were made:

- ♦ The text “this is currently the only way to authenticate to a NetWare server with OpenSSH” was added to the descriptions for ChallengeResponseAuthentication, PubKeyAuthentication, and RSAAuthentication.
- ♦ The second sentence of the last paragraph of the description for RSAAuthentiacion, “Version 2 uses the Digital Signature Algorithm (DSA),” was removed because SSH 2 can use either RSA or DSA.

A.7 June 1, 2005

The following updates were made:

- ♦ References to eDirectory versions were changed to “eDirectory 8.7.3.”
- ♦ Links to NetWare 6.5 documentation were changed to point to the OES documentation site.