

Fujitsu Services

OpenSSH Support Guide

Ref: DE/SPG/003
RS/MAN/?
??

Version: 1.0

COMMERCIAL IN-CONFIDENCE

Date: 30/06/2003

Document Title: OpenSSH Support Guide**Document Type:** Support Guide**Release:** BI3 S30R**Abstract:** This document describes the support and use of OpenSSH, giving information for both users and administrators of the system.**Document Status:** APPROVED**Originator & Dept:** Tony Dolton, Development/Cryptography**Contributors:****Internal Distribution:** Mik Peach**External Distribution:****Approval Authorities:**

Name	Position	Signature	Date
Mark Taylor	Development Manager		
Mik Peach	Operations and Support Services Manager (CS)		

0.0 Document Control

0.1 Document History

Version No.	Date	Reason for Issue	Associated CP/PinICL No.
0.1	29/05/03	Initial issue	CP3283, PC0086150, PC0089341, PC0089347, PC0089649, PC0089936, PC0090223, PC0090234, PC0090245
1.0	30/06/03	Updated for comments received. In particular, updated and expanded sections 4 and 5. Further information added for users in sections 8.6 and 8.7.	PC0090224, PC0090226

0.2 Review Details

Review Comments by :	
Review Comments to :	

Mandatory Review Authority	Name
Customer Service	Mik Peach * (by proxy)
Developer	Mike Garrett *
Team Leader	Will Dawson *
Designer	Simon Fawkes *
ITU	Alan D'Alvarez
Optional Review / Issued for Information	
Mike Stewart	
Nigel Taylor	
Chris Bates	

(*) = Reviewers that returned comments this review cycle

0.3 Associated Documents

Reference	Version	Date	Title	Source
DE/HLD/002			OpenSSH Secure Access and Logging HLD	PVCS
DE/LLD/003			OpenSSH Secure Access and Logging LLD	PVCS
SY/SOD/009			Secure Support System Outline Design	PVCS

Unless a specific version is referred to above, reference should be made to the current approved versions of the documents.

0.4 Abbreviations/Definitions

Abbreviation	Definition
Cygwin	A Linux-like environment for Windows, which uses a DLL to implement a POSIX layer on top of the Windows API.
DLL	Dynamic Link Library
ISD / OSD	Infrastructure Services Division / Operational Services Division. The current / former name for the unit responsible for day to day operation of the Live Horizon system.
Linux	A free Unix-like open source operating system.
OpenSSH	The version of SSH produced by the OpenBSD project, published as "open source".
POSIX	Portable Operating System Interface. A set of standard operating system interfaces based on Unix.
SAS	Secure Access Support. The SAS Server is the platform from which OpenSSH sessions are initiated.
SSH	The Secure Shell, a particular software-based approach to network security. Also used to describe the protocol used on such a system.
Unix	An operating system first developed at Bell Labs in 1969.
Windows	An operating system for PCs produced by Microsoft.

Fujitsu Services

OpenSSH Support Guide

Ref: DE/SPG/003
RS/MAN/?
??

Version: 1.0

COMMERCIAL IN-CONFIDENCE

Date: 30/06/2003

0.5 Changes in this Version

Version	Changes
1.0	Updated for comments received. In particular, updated and expanded sections 4 and 5. Further information added for users in sections 8.6 and 8.7.

0.6 Changes Expected

Changes
Changes as a result of comments received during review.

0.7 Table of Contents

1	INTRODUCTION.....	6
2	SCOPE.....	6
3	SYSTEM OVERVIEW.....	7
4	USER SETUP.....	8
4.1	BASIC PROCEDURE.....	8
4.2	SUPPLEMENTARY PROCEDURE.....	8
5	LOGGING SERVER.....	10
5.1	OVERVIEW.....	10
5.2	SECURE SERVERS.....	10
5.3	AUDIT FILES.....	11
6	USING OPENSSSH.....	12
6.1	CONNECTING TO THE SAS SERVER.....	12
6.2	CONNECTING TO THE TARGET PLATFORM.....	12
6.3	CONNECTION FAILURES.....	14
6.4	AFTER CONNECTION.....	14
7	TROUBLESHOOTING.....	15
7.1	PERMISSIONS PROBLEMS.....	15
7.2	PATH PROBLEMS IN “BASH” AND “SH”.....	15
7.3	DELETE KEY IN “BASH”.....	16
7.4	USER SHOWN AS “ADMINISTRATOR”.....	16
7.5	FAILURES AFTER UPDATING PASSWORD AND GROUP FILES.....	16
8	USEFUL TIPS FOR OPENSSSH USERS.....	17
8.1	ACCESSING OTHER FILESTORE AND DRIVES.....	17
8.2	POSIX/WINDOWS PERMISSIONS.....	17
8.3	UNPREDICTABLE PATH BEHAVIOUR.....	17
8.4	DELETE KEY IN “BASH”.....	17
8.5	THE “KILL” COMMAND.....	17
8.6	COMMAND HISTORY AND TYPEAHEAD.....	18
8.7	CHANGING WINDOW SIZE.....	18
	APPENDIX A – CYGWIN COMMANDS FOR NORMAL USERS.....	19
	APPENDIX B – CYGWIN COMMANDS FOR ADMINISTRATORS.....	19

1 Introduction

This document is the OpenSSH Support guide. It is intended for those staff needing to understand the configuration, use and support of OpenSSH within the Post Office Project.

It is necessary, for security and auditing purposes, to provide a method that allows datacentre and counter systems to be managed interactively but for all these management actions to be captured. When these actions have been captured (or logged) it must be possible to audit the actions. This, in turn, means the logs must be in an easily understandable format.

OpenSSH is used to provide access to these systems. This provides a 'command-line' interface to remote machines. This consists of a 'service' running on the target machine and a 'client' that allows access to the service from another machine.

The OpenSSH client has been modified so that it saves the data that flows between the client and the server to another system. This is done in such a way that no interaction is possible with the target machine without the interactions being logged.

When connecting to data centre platforms users log in using their own names and passwords. When connecting to counters the users log in using a 'special' user, and the OpenSSH client will be configured such that user authentication is achieved by the Public Key method. In this case a 'Pass Phrase' will be supplied by the user to effect the OpenSSH client server connection.

An NT service captures the data sent by OpenSSH clients into files that can be used later for auditing. This is referred to as the 'Logging Server'.

2 Scope

This document describes the configuration, support and use of the OpenSSH client, server and logging server.

It gives an introduction to and overview of OpenSSH (sections 1 to 3).

It specifies setup activities necessary for users that are to use OpenSSH (section 4).

It describes the Logging Server (section 5).

It describes how to connect to servers using OpenSSH (section 6).

It describes particular problems that may be experienced by administrators and users, with actions for rectification (sections 7 and 8).

It lists the Cygwin tools that can be used over an OpenSSH connection (Appendices A and B).

3 System Overview

OpenSSH is used to provide secure access to all remotely managed systems. Each system to be managed includes the OpenSSH server within the platform build. An amended OpenSSH client is installed on a number of support terminal servers which are located within the data centres. Access to the terminal servers is via a terminal server client installed on the operational and third line support users' workstations.

The following diagram shows this architecture:

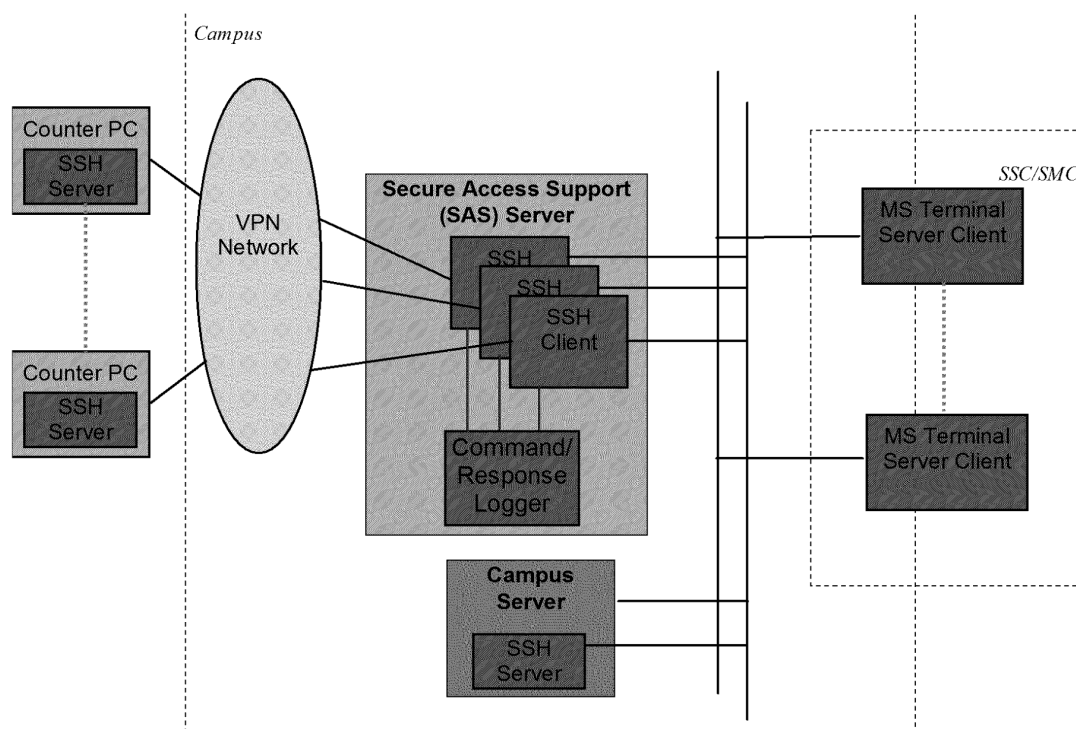


Figure 1: Overall OpenSSH Architecture

The OpenSSH Client is the executable `ssh.exe`. As can be seen, it is located only on the SAS Servers. Multiple instances can operate at the same time (even invoked by the same user, and connected to the same target platform).

The OpenSSH Server is a service called `CYGBWIN sshd` (short name `sshd`, executable `sshd.exe`), located on all target platforms (including the SAS Servers). It can be started and stopped by the usual methods, although it should normally be running at all times, to allow support staff access to the target platform. It spawns a new thread to service each client connection received. It is protected by a Tivoli sentry which will restart it on failure.

The Logging Server (shown as "Command/Response Logger" above) is a Windows service called `SSH_Logging_Server` (executable `SSHlogsvr.exe`), located on the SAS Servers. It can be started and stopped by the usual methods, although it should normally be

running at all times. It spawns a new thread to service each client connection received. It is protected by a Tivoli sentry which will restart it on failure.

4 User Setup

4.1 Basic Procedure

When using OpenSSH to connect to central servers, support users must be set up as Cygwin users on the target platforms. The support users are in different domains, depending on the domain of the target platforms. The relevant domains are as follows:

Domain of Target Platform	Support User Domain
PWYKMS	PWYKMS
PWYHQ, SIGF, CORPPWY, CONFMAN	PWYHQ
HUTHTIP, PDRTIP	(local users only)
(any other)	PWYDCS

Hence whenever such a user is added to the relevant domain, the necessary setup must occur on all potential target platforms.

The following procedure should be followed when support users are added or deleted:

1. Create or delete the user in the relevant domain as appropriate.
2. Rename the administrative username on the KMA Server to “Administrator” (to overcome a known problem in Tivoli).
3. Run the Tivoli task “Cygwin_Task” to populate the password and group files on the relevant central servers.
4. Reset the administrative username on the KMA Server to the original value (see step 2).

Note that certain platforms will be unable to access the relevant domain information, causing the Tivoli task in step 3 to fail. See the Supplementary Procedure below for how to set these platforms up.

4.2 Supplementary Procedure

The above procedure will not succeed on certain platforms which cannot access the relevant domain information. These include:

- FTMS Remote Gateways, which cannot access the relevant domain controller.
- Other platforms which may experience a NetBIOS problem which prevents contact with the domain controller.

The following manual procedure should be used to overcome the above problems, and is again to be used when support users are added or deleted:

1. Login to the domain controller for the domain for the users that manage the platform.

2. Carry out the following actions within a cmd shell:

```
cd \support\tools\generic\cygwin
cygwin
cd /cygdrive/c/support/config
sh pway-ssh-domain-mkpasswd <domain name>
```

3. Copy the resultant passwd.<domain name> and group.<domain name> files from the c:\support\config directory on the domain controller to the c:\support\tools\generic\cygwin\etc directory on the target server.

4. Login to the target server.

5. Carry out the following actions within a cmd shell:

```
cd \support\tools\generic\cygwin
cygwin
cd /cygdrive/c/support/config
sh pway-ssh-local-mkpasswd
cp tmp/passwd.local /etc/passwd
cp tmp/group.local /etc/group
cd /etc
cat passwd.<domain name> >> passwd
cat group.<domain name> >> group
```

5 Logging Server

5.1 Overview

The Logging Server records all OpenSSH sessions for audit purposes. Each command issued by the OpenSSH client, and all the output returned by the OpenSSH server on the target platform is written to a protected file on the SAS Server.

The Logging Server is a Windows service called `SSH_Logging_Server`. It can be started and stopped by the usual methods, although it should normally be running at all times.

If an OpenSSH client is unable to contact the Logging Server, it will not allow a connection, and will display the following output:

```
Checking host <hostname>
Unable to connect to SSH Logging Server
IP address was <IP address>
Port was <port number>
fj_connect FAILED
This version of ssh will only work if it can connect
to a Logging server (for auditing purposes).
Please contact your system administrator for instructions.
```

In this case, it is likely that the Logging Server is not running and must be restarted.

5.2 Secure Servers

Certain platforms are considered particularly sensitive and information from them should not be placed in the audit log. These platforms are specified in the protected file at `D:\SSHLogging\config\SecureServers.txt` on SAS Servers. The platforms are specified by IP address or hostname, the format matching that specified on the `ssh` command line (see section 6.2). All possible methods of referencing a platform should be included (e.g. hostnames, aliases, IP addresses) to ensure that information from the platform is not logged in error. When connecting to such a platform, the following text should be displayed during login:

```
*****
* You are connecting to a system which has private *
* data. This means that none of the responses from *
* the target system will be logged.                *
*****
```

The administrator should check that this message appears when connecting to the relevant servers. If this is not the case, they should check for the platform's presence (in the specified format) in the `SecureServers.txt` file.

5.3 Audit Files

Once a connection has been made, all commands submitted and the resulting output (except secure platforms) are written to a log file in D:\SSHLogging\Live on the SAS Server. Once the session is completed, the log file is moved to D:\SSHLogging\Completed, from where it is archived (by a separate process). The log file names are of the following form:

<destination platform>-<domain>-<username>-<source platform>.txt.<date><time>

Where:

<destination platform>	is the target platform's host name, or its IP address in dotted quad format with the dots replaced by "@"
<domain>	is the domain name of the user initiating the session
<username>	is the username of the user initiating the session
<source platform>	is the name of the platform at which the ssh connection was initiated
<date>	is in ccyyymmdd format
<time>	is in hhmmss format

The contents of the audit file is in XML format. This begins with information to identify the session, followed by the contents of the session, and finally records the ending of the session. The following XML tags record the contents of the session:

<KP>	A key pressed at the client.
<KL>	A line of key input at the client (a series of key presses followed by a carriage return).
<RS>	Response from the server. May be the results of a command, or echoing a keypress or its results (e.g. when editing the command line). Not logged when the server is secure (see section 5.2 above).
<TM>	A timestamp.

6 Using OpenSSH

6.1 Connecting to the SAS Server

To use OpenSSH, users must first connect to the SAS Server using MS Terminal Server.

Once on the SAS Server, a Cygwin session, using a 'bash' shell, can be started by running the batch file at `c:\support\tools\generic\cygwin\cygwin.bat`.

6.2 Connecting to the Target Platform

Once connected to the SAS Server, the user performs one of the following:

- **Connecting to a Counter PC**

To connect to a counter PC, use the following command:

```
ssh -l csash <counter IP address>
```

As shown, all connections to counter PCs use the single `csash` user. The counter is specified by `<counter IP address>`, a dotted quad format IP address.

Connections to counter PCs use RSA public key authentication. During the login, the user must supply the pass phrase to allow OpenSSH to access the RSA private key and authenticate the login.

- **Connecting to a Central Server**

To connect to a central server, use the following command:

```
ssh <server name or IP address>
```

Connections to central servers use password authentication. During the login, the user must resupply their Windows password to complete authentication.

6.2.1 Connecting to a Platform for the First Time

(Some of the behaviour described in this section is controlled by the `StrictHostKeyChecking` setting in the `ssh_config` configuration file on the SAS Server, the current setting being "no".)

Each target platform possesses a unique host key to identify itself to clients. The first time a user connects to any platform, the following message is displayed during the login sequence:

```
Warning: Permanently added '[<name>,<IP address>' (RSA1) to the list of  
known hosts.
```

As indicated, the platform and the public part of its host key is then added to the user's known hosts file (at `.ssh/known_hosts` under their home directory). As a result, subsequent logins will not display this message.

After the user has been validated, the following message may appear:

```
Could not chdir to home directory /cygdrive/c/sshadmin/users/<user name>:  
No such file or directory
```

The indicated directory will immediately be created as part of the login process, and will be available for this, and all future sessions. The error message can thus be safely ignored.

6.2.2 Platform Changes

(Some of the behaviour described in this section is controlled by the `StrictHostKeyChecking` setting in the `ssh_config` configuration file on the SAS Server, the current setting being “no”).

If the host key of the target platform doesn't match that recorded in the known hosts file, the following warning message is displayed during login:

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@  
@      WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!      @  
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@  
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!  
Someone could be eavesdropping on you right now (man-in-the-middle attack)!  
It is also possible that the RSA1 host key has just been changed.  
The fingerprint for the RSA1 key sent by the remote host is  
<key value>.  
Please contact your system administrator.  
Add correct host key in <user's known hosts file> to get rid of this  
message.  
Offending key in <user's known hosts file>:<line number>  
Password authentication is disabled to avoid man-in-the-middle attacks.
```

It is possible that the target platform's host key has legitimately been changed. However, if this is not known to be the case, the system administrator should be informed. To prevent the message from occurring, the correct host key should be placed in the known hosts file. The easiest way of achieving this is to delete the platform's entry from the user's known hosts file, so that it is treated as a new platform on the next login (see section 6.2.1 above). (Alternatively, the correct host key could be obtained from another user, or from the target platform itself.)

If both the host key and IP address of a central server has changed, this could indicate a DNS spoofing attack. A warning message similar to the following is displayed during login:

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@  
@      WARNING: POSSIBLE DNS SPOOFING DETECTED!      @  
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@  
The RSA1 host key for <server name> has changed,  
and the key for the according IP address <IP address>  
is unknown. This could either mean that  
DNS SPOOFING is happening or the IP address for the host  
and its host key have changed at the same time.
```

As indicated, it is possible that the target platform's host key and IP address have legitimately been changed. However, if this is not known to be the case, the system administrator should be informed.

6.3 Connection Failures

The connection may fail for a number of reasons.

The client may fail to gain a connection to the Logging Server (see section 5).

If the OpenSSH server is not running on the target platform, the following message is displayed:

```
ssh: connect to address <IP address> port <port number>: Connection refused
```

If the connection has been made, but all attempts to authenticate the user have failed, then the message "Permission denied" is displayed.

The user should then check that all the information supplied (whichever of server name, IP address, user name, password or passphrase) is correct.

Further detail on the failure may be obtained by appending the `-v` command line option to the `ssh` command, although development staff may be required to interpret the results. Up to three `-v` options may be supplied on the same command line, indicating successively more detailed levels of tracing, which is written to the standard error output.

6.4 After Connection

Once connected to the target platform, the user can use the Cygwin commands documented in Appendix A, as well as any other facilities available on the platform. All commands submitted to the session, and the result of those commands, are recorded by the Logging Server.

Note the tips in section 8 regarding the use of OpenSSH.

To terminate the OpenSSH session, simply exit the session by typing `exit` or `Control-D`. Logging stops and control returns to the Terminal Server session on the SAS Server.

7 Troubleshooting

This section describes potential problems that may be encountered when configuring, supporting or using OpenSSH, giving possible solutions. They are supplied for reference by those supporting OpenSSH, but many will be relevant to users of OpenSSH also (in which case they are also contained in section 8).

7.1 Permissions Problems

When attempting to diagnose problems with OpenSSH (even those not apparently related to permissions – for example, see section 7.2), it should be noted that the permissions displayed by OpenSSH don't necessarily reflect the full set of permissions applied by Windows. This is because the rich set of permissions supported by Windows, with access specified individually for multiple users and groups, cannot generally be mapped to the simple user/group/other model offered by POSIX. Hence OpenSSH will generally only display an approximation of the permissions in POSIX form, but will usually apply the full set of Windows permissions. The permissions displayed and applied are also affected by the setting of the `CYGWIN` environment variable (`ntsec` or `nontsec`).

As a result, you should not rely on the permissions information displayed by Cygwin commands such as `ls`; instead use Windows facilities (e.g. `cacls`).

7.2 Path Problems in “bash” and “sh”

In certain circumstances, the “bash” shell will not execute the first executable version of a command in the `PATH`. This is because of the difficulties of converting complex Windows permissions to a POSIX equivalent (see section 7.1). This can result in OpenSSH believing that a utility is not executable for the current user, when it actually is. The “bash” shell may then find a version later in the `PATH` which it finds to be executable according to POSIX, which it will then execute. This is a particular problem with commands that are also provided by Windows, namely `find`, `sort` and `hostname`, but there can be clashes with other command sets as well. (If the command is not found elsewhere in the `PATH`, the first command with a matching name is used; this version usually turns out to be executable after all, in which case the system appears to work as expected.)

The “bash” built-in command `type` can be used to determine when this problem is occurring. If found to be a problem, then “bash” can be forced to use the correct version by either supplying the full pathname, or cutting down the `PATH` environment variable. The user may also consider using the “sh” shell (but see the following).

There is a similar but unrelated problem in the “sh” shell. The shell itself is more reliable than “bash” in that it always tries to execute the versions in the order they are found in the path, and thus always executes the first version that is executable according to its Windows permissions. However, the `type` command in “sh” uses a simplistic (and lenient) algorithm to determine execute permission, and can indicate that a version earlier in the path will be run, whereas it is actually not executable for the current user. So do not rely on the results of

type when using “sh”. (The `which` command generally matches the behaviour of “sh” more closely.)

7.3 Delete Key in “bash”

As described in section 8.4, the “delete” key can be made to operate as expected within the “bash” shell, by placing the following line in a file named `.inputrc` within the relevant user’s home directory.

```
"\e[3~": delete-char
```

To make this change apply to all users on a server, the line should be placed in a well-known file (usually `/etc/inputrc`), and that file referred to by the system environment variable `INPUTRC`. This variable will be read by “bash” on login and cause it to read the indicated file (and ignore the user’s `.inputrc` file).

7.4 User Shown as “Administrator”

If a user has not been set up as a Cygwin user on the SAS Server as described in section 4, before attempting to invoke Cygwin, they will be shown as an administrative user, e.g. with user name “Administrator” displayed by `id` and in the “bash” prompt.

This user name is only displayed by Cygwin for convenience, in the absence of any meaningful user name. The user will not have access to any facilities other than those normally available.

This situation can only be rectified by setting up the user correctly as shown in section 4.

7.5 Failures After Updating Password and Group Files

After new users or groups have been configured for Cygwin by adding to the `/etc/passwd` or `/etc/group` files on the relevant platforms (see 4), it will usually be necessary to restart all Cygwin programs, including the server process, `sshd.exe`, before these users/groups can be used. This is because these files are cached by the Cygwin software and are generally only read at startup by the Cygwin dll.

Failure to restart processes will result in user/group related failures. A specific example is where the new user tries to connect to the affected server using `ssh`. If the target machine’s group file has been updated, but `sshd` hasn’t been restarted (which is unlikely if the procedure in section 4 is followed), then the login will fail with the following message:

```
setgid: Invalid argument
```

8 Useful Tips for OpenSSH Users

8.1 Accessing Other Filestore and Drives

The visible filestore under Cygwin's root directory (/) reflects filestore mounted for Cygwin (normally C:\Support\Tools\generic\cygwin). However, it is still possible to access other parts of filestore (subject to normal access controls), including other drives, using built-in "cygdrive" mount points. For example, /cygdrive/d/ equates to the Windows path D:\. Cygwin will also accept the "D:" form in some circumstances.

8.2 POSIX/Windows Permissions

Note that the POSIX permissions displayed by OpenSSH don't necessarily reflect the full set of permissions applied by Windows, due to the greater complexity of the latter's security model. They are also affected by the setting of the CYGWIN environment variable (ntsec or nontsec).

As a result, you should not rely on the permissions information displayed by Cygwin commands such as `ls`; instead use Windows facilities (e.g. `cacls`).

8.3 Unpredictable PATH Behaviour

Be aware that there may be several versions of a named command on your path (as well as commands built in to your shell). For instance, `find`, `sort` and `hostname` are all tools supplied within both Cygwin and Windows. Note that the "bash" shell does not always choose the first executable version in your path (see Troubleshooting, section 7.2). If you aren't sure that the shell is picking up the correct version of a command (you can use the `type` shell builtin to confirm this), either specify the full path or restrict the `PATH` to the relevant location(s). You could also try using the "sh" shell.

8.4 Delete Key in "bash"

To enable the "delete" key to operate as expected within the "bash" shell, place the following line within a file named ".inputrc" within your user's home directory:

```
"\e[3~": delete-char
```

"Bash" will read this file on login and henceforth interpret the delete key as expected (deleting the character under the cursor).

8.5 The "kill" Command

Note that several versions of the `kill` command may be available on the Cygwin system, each with different characteristics.

The "bash" shell has a built-in `kill` command. It is only able to terminate Cygwin processes.

The Cygwin `kill` command will normally only terminate Cygwin processes, although the `-f` flag can be used to kill Windows (i.e. non-Cygwin) processes (as displayed by “`ps -W`”). To run this version instead of the built-in version in “`bash`”, the full pathname `/usr/bin/kill` should be specified.

Other versions of “`kill`” may also exist, for example that in the NT Resource Kit. To use this version from Cygwin, the full pathname should be specified.

Note that although care should be taken when referring to processes with regard to Cygwin and Windows process ids, it should not normally be possible to kill the wrong process, although you may fail to kill a Windows process as described above.

The “`bash`” built-in `alias` command could also be used to permanently specify the version to be used (e.g. specify `kill` to be the required version, or set up separate aliases for each version).

(In future, system-wide aliases may be set up to avoid the need for users to specify full paths or set up aliases.)

8.6 Command History and Typeahead

Command history facilities are normally available in the “`bash`” shell to allow previous commands to be rerun, after editing if necessary. Pressing the “up arrow” key will recall the previous command.

Care should be taken when trying to invoke such facilities when previous commands are still running. When the command prompt does eventually appear, the recalled command may not be displayed, and it will not be obvious that command history has been invoked; if “Enter” is pressed will the command be displayed, and immediately run.

A simple workaround in this instance is to press the “End” key once the command prompt has reappeared. If a command has been recalled, it will then be displayed correctly, edited appropriately.

8.7 Changing Window Size

Certain Cygwin commands (e.g. `less` and `ls`) try to tailor their output to the size of the window from which they are invoked (although not always with complete success – e.g. `less` and window width).

However, this facility does not operate correctly when using OpenSSH; commands in an OpenSSH session generally behave as if the output window were the same size as when the session began, which can cause confusing output after the window is resized.

It is thus recommended that the window size not be altered after an OpenSSH session has been started. If it is found to be necessary to change the window size, then a new session should be started, after setting the window size appropriately.

An alternative workaround can be used to enable the window size to be changed; it relies on the fact that changes while the session is suspended are propagated correctly. First type `~`, `Control-Z` to suspend the server session, noting the number of the stopped job. Then

Fujitsu Services

OpenSSH Support Guide

Ref: DE/SPG/003
RS/MAN/?
??

Version: 1.0

COMMERCIAL IN-CONFIDENCE

Date: 30/06/2003

change the window size as required, and resume the stopped job using “fg <*job number*>”. Subsequent commands should then use the new window size.

Appendix A – Cygwin Commands for Normal Users

The following Cygwin commands are supplied for use by normal users:

awk	basename	bash	cat	chgrp	chmod
chown	chroot	cmp	cp	cut	cygpath
date	dd	df	diff	dirname	du
echo	egrep	env	expr	false	fgrep
find	fold	gawk	grep	groups	gunzip
gzip	head	hostname	id	kill	less
ln	login	ls	md5sum	mkdir	mount
mv	nice	nl	nohup	od	paste
printf	ps	pwd	regtool	rm	rmdir
sed	sh	sleep	sort	tail	tar
tee	test	touch	tput	true	tset
umount	uname	wc	which	who	

Appendix B – Cygwin Commands for Administrators

The following Cygwin commands are supplied for administrator use only:

cygrunsrv	mkgroup	mkpasswd
ssh-add	ssh-agent	ssh-keygen
ssh-keyscan	ssh-keysign	ssh-rand-helper