Command Line SSH

☞ **Secure Shell Access**

Command line access via a Unix terminal/shell is the most reliable, powerful method of accessing the Digital Music Studios. With it users can login and use the functionality and processing power of the CDMS computers remotely.

To login, open a terminal shell (on Mac, Applications --> Terminal; on Windows use Cygwin; on Linux or other Unix just open a standard xterm). Then at the command prompt type:

ssh *REMOTE_MACHINE_NAME*

...where REMOTE_MACHINE_NAME is the name of the machine you wish to access, for example digital.music.cornell.edu. If you use a different username on your local system than the one on the system you are logging into, you will also need to specify your remote username :

ssh *USERNAME@REMOTE_MACHINE_NAME*

CEMC systems are named as follows:

jack.music.cornell.edu (B27 Linux system)

devel.music.cornell.edu (B27 Mac system)

b25b.music.cornell.edu ("Film" studio, B25B)

b25c.music.cornell.edu (Undergraduate Studio 1, B25C)

b25d.music.cornell.edu (Undergraduate Studio 2, B25D)

On CEMC systems, convenient command shortcuts are also available using a standard syntax:

sshjack (same as ssh jack.music.cornell.edu)

sshdev (same as ssh devel.music.cornell.edu)

sshdig (same as ssh digital.music.cornell.edu)

sshb25b (same as ssh b25d.music.cornell.edu)

sshb25c (same as ssh b25c.music.cornell.edu)

sshb25d (same as ssh b25d.music.cornell.edu)

When logging in from the outside you will be prompted for your password. Enter it and you will be logged in immediately. Logins from within the studios are governed by secure authorized keys meaning your account will automatically be verified when a remote shell is requested and you will not have to type your password. This will become important later on as we seek to run remote processes inline with local programs.

You may also wish to create password keys between systems, enabling passwordless movement

among studio systems. To set this up, type “cornellkeygen” and follow the directions entering no passphrase.

☞ Secure Shell Copying

To copy files to and from local or remote computers, use the scp (secure copy) command. The syntax is identical to ordinary copying with cp.

scp *FILE_TO_COPY [MORE_FILES_TO_COPY] DESTINATION*

Unlike cp, the file to copy and/or their destination may be a remote computer. The user must specify the remote machine name and directory as a prefix. So for example, to copy a local file named myfile to my remote home directory ("/home/kevinernste") on digital.music.cornell.edu, I would type:

scp myfile digital.music.cornell.edu:/home/kevinernste/

Note the colon ":" between the machine name and the directory in the destination argument. This syntax should be followed carefully.

As with cp (and other Unix commands), all unix "wildcards" will work, so:

scp * digital.music.cornell.edu:/home/kevinernste/

...will copy all files in the current directory to the destination directory on b27. To copy directories and subdirectories, just add "-r" (for “recursive), so:

scp -r * digital.music.cornell.edu:/home/kevinernste/

☞ GUI SSH tools

If you wish, there are also several graphical SSH/SCP/SFTP clients available. Most limited to simple file copying, but some include terminal shell functions as well. To peruse the list of clients available for your operating system of choice, please visit openssh.com and follow the links in the left-hand panel.