# UNIX Training

## An overview of PAREXEL's SAS UNIX Platform

PAREXEL.
Right where you need us

# Objectives

- How to request access to PAREXEL's SAS UNIX platform
- How to logon to PAREXEL's SAS UNIX platform
- How to use PAREXEL's SAS UNIX platform
- Allow users to be more efficient when using UNIX
- Understand the file system and navigation in the UNIX environment
- Understand how to edit files
- Understand system usage
- Understand how to resolve issues
- Awareness of UNIX software

Pre-requisites and logging in

# Part One: Before we begin

# Before we begin: What is UNIX?

UNIX is a computer operating system (OS).

HP-UX differs from the Windows OS by being text based and not graphics based.

# Before we begin: Prerequisites

To use UNIX at PAREXEL, you must meet the following requirements

- Complete this training, LMS
- Have a UNIX account
- Have Citrix desktop set-up
- Have WinSCP client installed

- All the above can be obtained by using the Service Desk, CAPS and LMS
- Navigate to the PAREXEL Connect Site http://ep.pxl.int/portal/server.pt
- In the “My Hot Links” box, select either “CAPS” or “Service Desk” http://service.pxl.int/

# Before we begin: Prerequisites with Optional SAS Enterprise Guide

To use SAS Enterprise Guide at PAREXEL, you must meet the following requirements

- Complete this training, LMS
- **Complete SAS Enterprise Guide Workshop, LMS
- Have a UNIX account
- Have Citrix desktop set-up, **with SAS Enterprise Guide
- Have WinSCP client installed

- All the above can be obtained by using the Service Desk, CAPS and LMS
- Navigate to the PAREXEL Connect Site http://ep.pxl.int/portal/server.pt
- In the “My Hot Links” box, select either “CAPS” or “Service Desk” http://service.pxl.int/

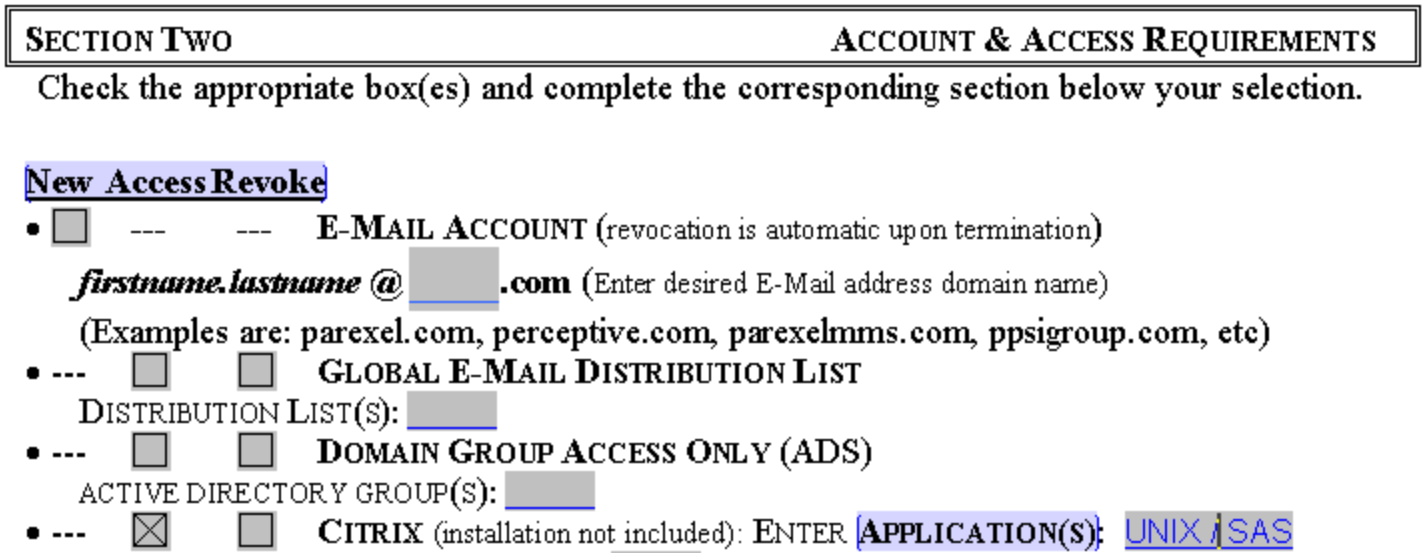
# Before we begin: Citrix set-up

Users should have Citrix accounts requested by their manager when they start at PAREXEL.

However, if required, a global access request form may be completed with this request.

Next to the **CITRIX** line, click the access box and in the text box type in **UNIX / SAS, or UNIX / SAS / EG** ** to include Enterprise Guide

The form should then be signed by your line manager or an appropriate person as detailed on the top of the form



SECTION TWO — ACCOUNT & ACCESS REQUIREMENTS

Check the appropriate box(es) and complete the corresponding section below your selection.

New Access Revoke

- ☐ --- --- E-MAIL ACCOUNT (revocation is automatic upon termination)

  *firstname.lastname* @ _____.com (Enter desired E-Mail address domain name)

  (Examples are: parexel.com, perceptive.com, parexelmms.com, ppsigroup.com, etc)
- --- ☐ ☐ GLOBAL E-MAIL DISTRIBUTION LIST

  DISTRIBUTION LIST(S): _____
- --- ☐ ☐ DOMAIN GROUP ACCESS ONLY (ADS)

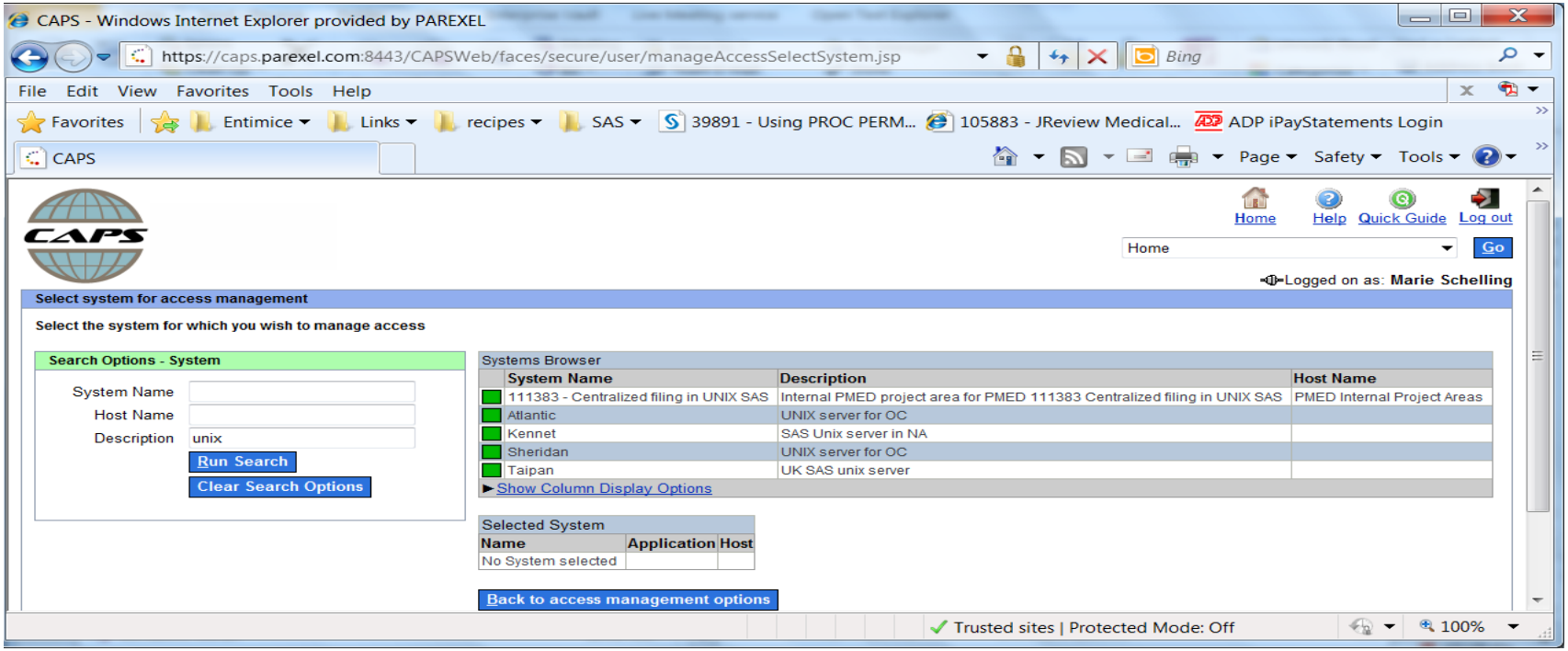
  ACTIVE DIRECTORY GROUP(S): _____
- --- ☒ ☐ CITRIX (installation not included): ENTER APPLICATION(S): UNIX / SAS

# Before we begin: Requesting a UNIX account

To request a UNIX account – you need to submit a CAPS request

A prerequisite of this is the users should be trained in the CAPS system

CAPS User Training CT-CAPS, (self-paced)



# Before we begin: WinSCP client set-up

WinSCP is a free SFTP (Secure Shell File Transfer Protocol) client for windows. It's main function is to allow safe copying of files between a local and remote computer (e.g. the transfer of files from your local machine to a UNIX directory)

sFTP is used rather than FTP as it is more secure and uses encrypted methodology

WinSCP can be requested by submitting a Service Desk to have it installed on your computer

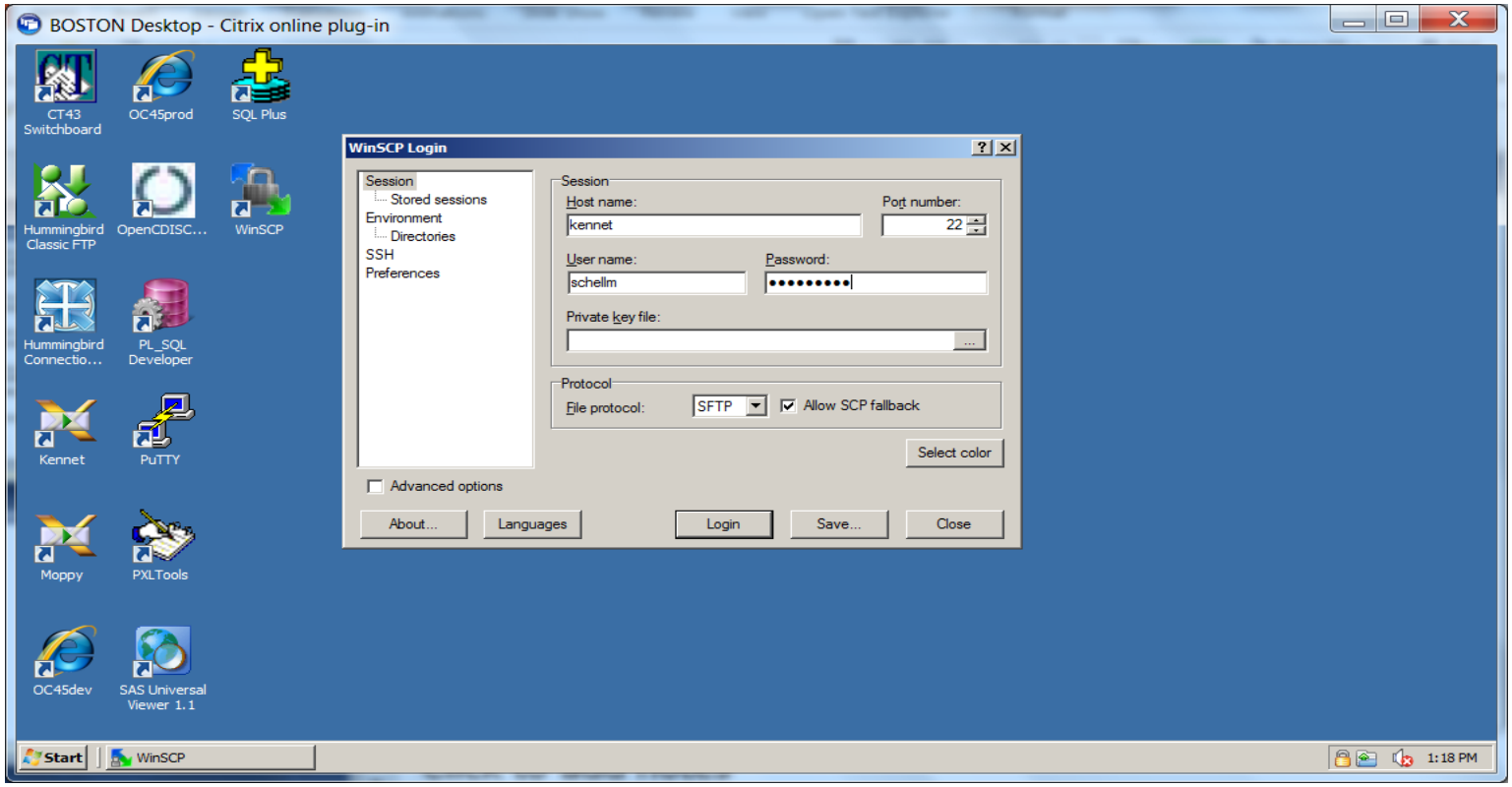
Atlernately WinSCP can also be accessed on Boston Desktop

The minimum required version of WinSCP is 4.1.8

# Before we begin: WinSCP Login

Enter Host name: kennet

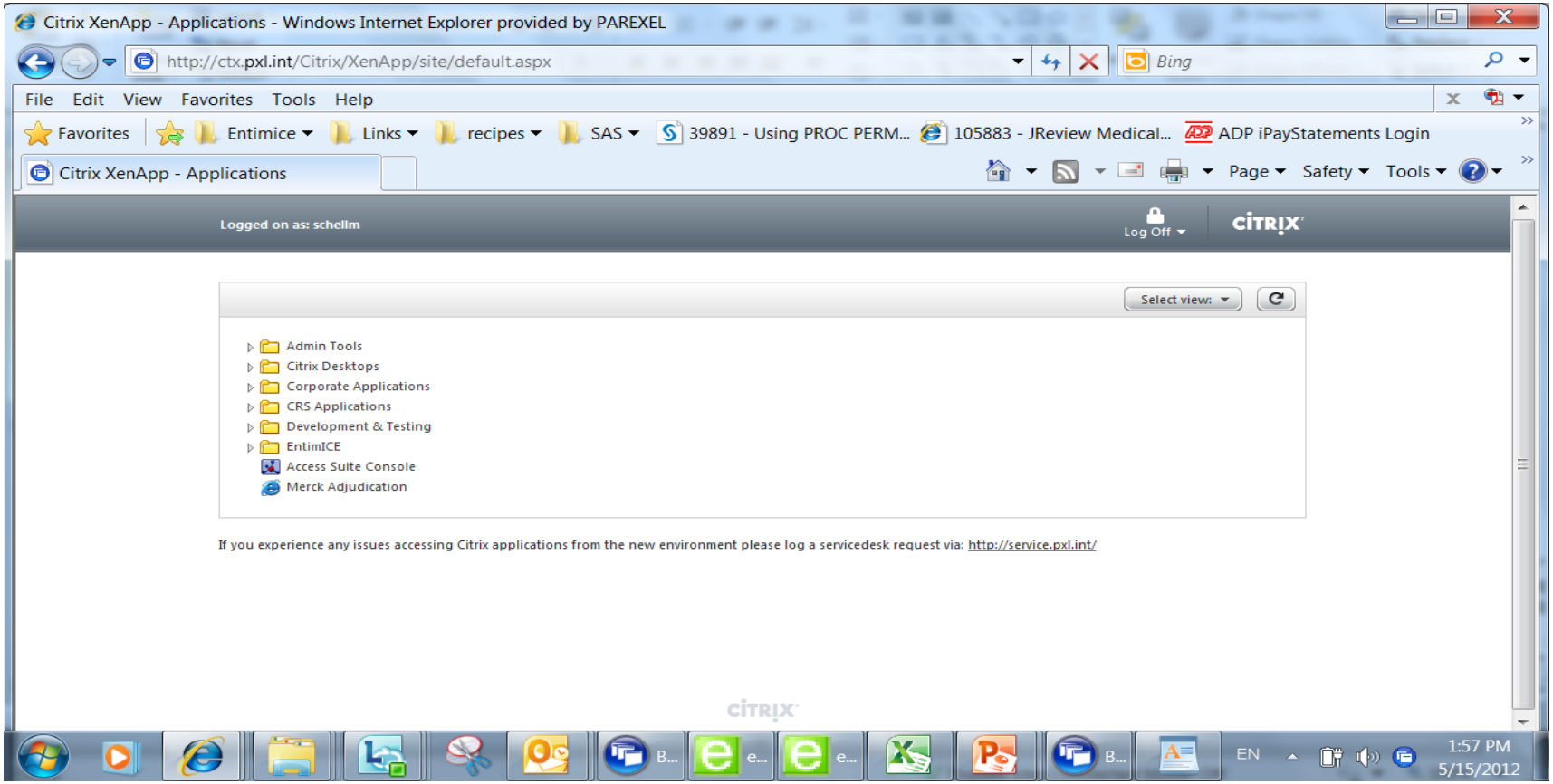
Enter User name and password, Click Login



# Before we begin: Logging on via Citrix

To access UNIX, the user needs to login via Citrix and use the desktop folders to access the remote desktop where UNIX is installed via ie link

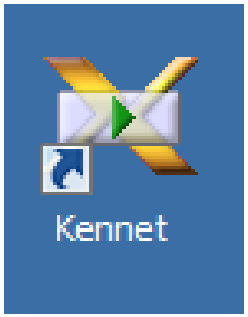
http://ctx.pxl.int/Citrix/XenApp/site/default.aspx



# Before we begin: Logging on via Citrix

Citrix Desktops, BOSTON Desktop contains the Unix tools for all US SAS environments. **moppy** is Development environment and **kennet** is Production environment

Once the BOSTON desktop is available on screen, the user should double click on the **kennet** icon to log in to the UNIX server.



Project access, navigation, basic commands

# Part Two: Now you're logged on

# Now you're logged on: Citrix, Exceed and UNIX

- Citrix is used to remotely connect to another PC, the remote desktop contains a link to the UNIX instance
- Exceed is a PC program which supplies window management services for unix applications. As Windows is incompatible with UNIX systems, therefore Exceed needs to be used to allow access to UNIX X Window applications from the comfort of the Windows desktop.
- Exceed will begin upon clicking on the UNIX shortcut.
- Whereas windows is mainly graphically driven, UNIX is a command line interface.

# Now you're logged on: Naming conventions

UNIX is case sensitive so will differentiate between the files test.sas and TEST.SAS as different files.

File names can be up to 256 characters long

Avoid using special characters in file names such as

| ; , ! @ # $ ( ) < > / \ " ' ` ~ { } [ ] = + & ^ <space> <tab>

Instead of spaces in file names and directory names, use characters such as _ - . (underscore, subtraction or period)

# Now you're logged on: UNIX manual

The **man** command in UNIX provides an on-line system manual on providing information which can be used to

To display details about using the system manual, use **man man**

To display details about the listing (**ls**) command, use **man ls**

Pressing enter moves through the on-line manual line by line

Pressing space moves through the on-line manual page by page

To search for a keyword **passwd** in the manual, use **man –k passwd**

To return a specific section (say section 1 on the passwd command), use **man 1 passwd**

# Now you're logged on: Changing your password

UNIX account passwords expire every **90** days

So it is important that you are able to change your password. When the password expires, the system will prompt you for a new password at logon.

If you wish to change your password at any time manually, then the steps below should be followed

At the UNIX prompt, type **passwd**

The system will ask for your old password, type this and press return

If successful, the system will then ask for your new password, again type this and press return

The system will then ask for your new password again to confirm the change, type your new password and press return

The system will display "Passwd successfully changed"

Your UNIX passwords are not linked to your smartcard so must be entered each time the user logs in.

# Now you're logged on: Password rules

A password must follow the following rules to be allowed:

- Must be at least 8 characters in length
- Must be alphanumeric characters and numbers
- Must differ from the users name

e.g. aardvar6 would be a **valid** password
aardvark would be an **invalid** password
aard5 would be an **invalid** password

A new password must differ from the old one by at least three characters

Can't use a previous password from the last 10 passwords used in the history, e.g. your password from 2 times before can not be reused

# Now you're logged on: Listing project access

For each project area on UNIX, access is restricted to those users with authorised access

You can list which projects you have access to by typing **studies**

By typing **studies** {username} you can check access for other users

For example typing **studies cantrea** will list all the projects that the user cantrea has access to

**ls /ep** can be used to list all early phase projects on the system

# Now you're logged on: The project area

To navigate to a project area, type cd /ep/blinded/e999999_sponsorname

Or cd /ep/unblinded/e999999_sponsorname and press return

Within each project directory, there are 4 top level folders

- dm
- stats
- transfer
- pkpd

The dm folder is accessible by all project staff in the dm group

The stats folder is accessible by all project staff in the stats group

The transfer folder is accessible by all project staff and can be used to transfer files between the dm and stats groups

The pkpd folder is accessible by all project staff in the pkpd group

# Now you're logged on: Navigation

To navigate to an existing folder (export), type **cd** (change directory) followed by the folder name,

e.g. **cd export**

The prompt will change from /ep/unblinded/e999999_sponsorname/dm to /ep/unblinded/e999999_sponsorname /dm/export which demonstrates that the user has navigated to the export folder

By typing **cd ..** you can navigate up one level in the directory you are in, by typing this when in the direct /ep/unblinded/e999999_sponsorname/dm/export, it will change the folder to /ep/unblinded/e999999_sponsorname/dm

# Now you're logged on: Navigation

Additionally typing **cd** the user will be taken to their home area, e.g. /home/users/cantrea

To navigate up multiple levels at once, use **cd ../../transfer** this would navigate up 2 directory levels, then change into the transfer directory

# Now you're logged on: Navigation in unblinded area

To create a new project on UNIX, a request must be submitted using a global new project request form which should be attached to a service desk ticket

The /ep (blinded), /unblinded and /randomisation areas are created up front when a new UNIX project is set-up however access is only granted through CAPS.

The /unblinded area is used for storing programs and data related to activities that refer to data that should not be viewed by blinded programmers

The /randomisation area is used for the generation of randomization codes

# Now you're logged on: Listing files

The **ls** command lists the files and directories in a directory

By adding additional options to this command, the output can be tailored

**ls –l** or **ll** will display a long list of the files and directories

**ls –al** will display a long list of all files including hidden files

**ls – trl** will display a long list in reverse date order, newest at the bottom

**ls –Rl** will display a recursive list of the files and directories

*Note: As UNIX is case sensitive, it is important to use R instead of r*

**ls –sl** will display a long list with 512 byte blocks listed in the left hand column

**ls –Rasl** will display a long recursive list of the files and directories with 512 byte blocks listed in the left hand column

*Note: As UNIX is case sensitive, it is important to use R instead of r*

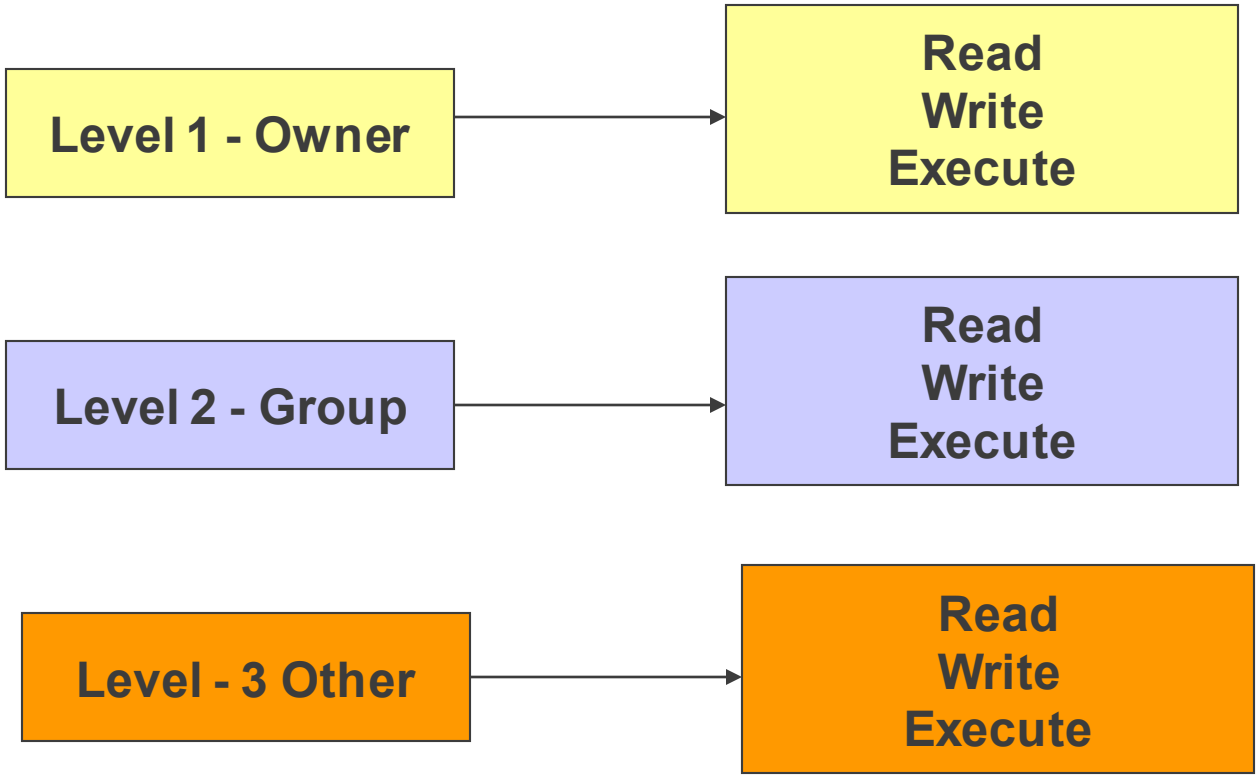
# Now you're logged on: Listing files

When using **ls -l** to list the files, the following columns will be displayed

- File (-) or directory (d)
- User permission
- Group permission
- Others permission
- File owner
- Group owner
- File size
- Modification date and time
- File name

```
drwxrwx---   2 cantrea    dm              96 Dec 16  2005 scripts
-rw-rw----   1 cantrea    dm               4 Feb  8 14:36 test.123
-rw-rw----   1 cantrea    dm              67 Feb  2  2009 test.csv
-rw-rw----   1 cantrea    dm              11 Nov  1  2007 test.dat
-rwxrwx--x   1 cantrea    dm              22 Feb  8 14:38 test.sas
-rw-rw----   1 cantrea    dm             193 Oct 14  2009 test.sql
-rw-rw----   1 cantrea    dm             382 Jun 22  2009 test.zip
-rwxrwxr-x   1 cantrea    dm               0 Sep  2  2008 test2.sas
-rw-rw----   1 cantrea    dm            8606 Dec 10  2007 tmslice_compare.sas
-rw-rw----   1 cantrea    dm          167608 Sep  4  2009 track1.csv
-rw-r--r--   1 cantrea    dm         1360896 Jan 24  2008 tracking.sas7bdat
```

# Now you're logged on: File permissions 1



# Now you're logged on: File permissions 2

The **chown** command can be used to change the owner of a file, to change the owner of a file (test4.txt) from smithj to doej, use **chown doej test4.txt**

Only the owner can change ownership or permissions of a file

The **chmod** command changes the permissions of a file according to the parameters specified.

Only the owner of a file or with appropriate privileges can change its mode.

There are three levels of file permissions (owner, group and others).

**The "others" permission should generally have no access**

Within each level, the permissions can be set as read access, write access and execute access (the typical permission granted is 770 – owner and group / read, write & execute)

```
400  (= u=r)  Read by owner
200  (= u=w)  Write by owner
100  (= u=x)  Execute (search in directory) by owner
040  (= g=r)  Read by group
020  (= g=w)  Write by group
010  (= g=x)  Execute/search by group
004  (= o=r)  Read by others
002  (= o=w)  Write by others
001  (= o=x)  Execute/search by others
```

# Now you're logged on: File permissions 3

To deny write permissions to others on a file (export.sas), use **chmod 775 export.sas**

To make a file (listing.sas) executable by all, use **chmod 777 listing.sas**

To assign read, write,execute access to the owner and read and execute access to group (run.sas), use **chmod 770 run.sas**

At PAREXEL, the general rule should not be granting project directories / files to the others group

```
400  (= u=r)  Read by owner
200  (= u=w)  Write by owner
100  (= u=x)  Execute (search in directory) by owner
040  (= g=r)  Read by group
020  (= g=w)  Write by group
010  (= g=x)  Execute/search by group
004  (= o=r)  Read by others
002  (= o=w)  Write by others
001  (= o=x)  Execute/search by others
```

# Now you're logged on: File permissions 4

The **chgrp** command can be used to change the group that has access to a file or directory. To change the group that can access a file (test5.txt) from dm to stats, use **chgrp stats test5.txt**

To recursively change the privileges, owner or group values, can specify the –R suffix

To provide 770 access to all files in a directory (export), use the command **chmod –R 770 export/**

```
400  (= u=r)  Read by owner
200  (= u=w)  Write by owner
100  (= u=x)  Execute (search in directory) by owner
040  (= g=r)  Read by group
020  (= g=w)  Write by group
010  (= g=x)  Execute/search by group
004  (= o=r)  Read by others
002  (= o=w)  Write by others
001  (= o=x)  Execute/search by others
```

# Now you're logged on: Searching for files

The **find** command searches through a directory structure searching for files that match the required criteria.

To find a file (test.sas) in the current directory and all directories below the current directory use **find . –name test.sas**

To find any file in the current directory and all directories below the current directory that contains the string oak use **find . –name "*oak*"**

To find all sas programs in a project, use

**find /ep/unblinded/{project name}/dm –name "*.sas"**

# Now you're logged on: Copying and moving files

The **cp** command lets you copy a file from one location to another

To copy a file (file1) to another file (file2) use **cp file1 file2**

To copy a file (test.sas) to a directory (/sasprogs) use **cp test.sas ./sasprogs**

To copy a file /ep/unblinded/e999999_sponsorname/dm/export.sas to your current directory, use

**cp /ep/unblinded/e999999_sponsorname/dm/export.sas .** *(this dot refers to the current directory)*

*Check the file permissions and groups after you copy, you might need to run **chgrp** andor **chmod**, to have the groups and permissions requried*

The **mv** command lets you move a file from one location to another

The command is similar to the cp command but deletes the initial file

To rename a file (file1) to another file (file2) use **mv file1 file2**

To move a file (file1) to a directory (/temp) use **mv file1 ./temp**

To move all files in the directory to the /tmp directory use **mv * ../tmp**

# Now you're logged on: Removing and creating files

The **rm** command lets you remove a file

To remove a file (export_test.sas) use **rm export_test.sas**

The **rmdir** command lets you remove a directory

To remove an *empty* directory (temp) use **rmdir temp**

**rm –r –i <directory name>** will recursively (in turn) remove all files within a directory and the directory itself and will prompt the user to confirm each deletion

When using the rm and rmdir commands, it is suggested you always use the –i option

The **mkdir** command lets you create a directory

To create a directory (temp1) use **mkdir temp1**

# Now you're logged on: Searching within files 1

The **grep** (Global Regular ExPression) command lets you search a file for a specific text string

To search a file (validate.sas) for the text ("program") use **grep program validate.sas**

The system will display any occurrences of the text searched for, so if the file contains the text "program" and "programming", both results will be returned

Note: **grep –i** does a case insensitive search, so text, TEXT and tEXt will all be found for the word text

# Now you're logged on: Searching within files 2

The head command lets you view the header of a file (by default 10 lines)

To view the first 20 lines of a file (test.txt) use **head –n 20 test.txt**

The tail command lets you view the tail of a file (by default 10 lines)

To view the last 15 lines of a file (test.txt) use **tail –n 15 test.txt**

To view a log file (export.log) that is currently being created (e.g. additional lines are being written to the file as a program is running), use **tail –f export.log**

This will refresh the display as additional information is written to the log – can be useful for seeing at what point a program has reached in its execution

The pipe symbol | can be used to pass output from one command as input to the next. For example to get a list of the 10 most recent created files, pass the output of the files in date order to the tail –n 10 command, to do this, use

**ls –trl | tail –n 10** This command passes the list if files in date order to the tail command which displays the last 10 files based on the input list provided

# Now you're logged on: Using xedit

There are a number of text editors available in UNIX, xedit is one of the easiest and straight forward text editors to use so is suggested to be used at PAREXEL.

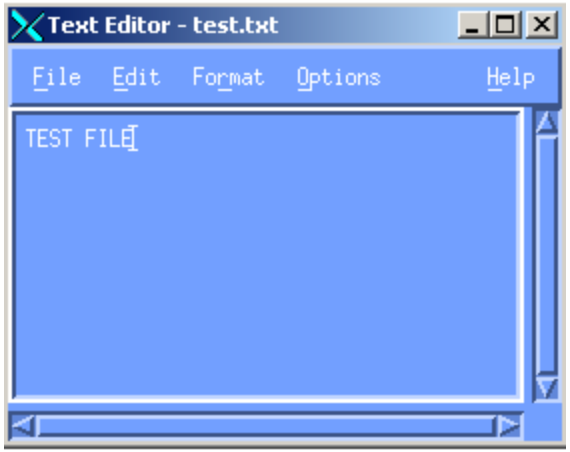
To use xedit, type **xedit** at the UNIX command prompt

To open a file (test.txt) using xedit, type **xedit test.txt** at the command prompt

xedit is essentially the UNIX equivalent of a windows text editor similar to notepad

It should be used for viewing log files and text files.

It is recommended **not to** update SAS code using xedit but to use sas (SAS 9.1), sas91 (SAS 9.1) or sas92 (SAS 9.2) program editor



# Now you're logged on: SAS in UNIX

To start SAS in UNIX, use the command explicity, to start SAS 9.2, type **sas92** at the unix prompt, or type **sas91** to start SAS version 9.1

A version of SAS should consistently be used across the project and should be discussed with the DOL.

To start SAS in background mode, so that you can continue using the UNIX session as well as SAS, use the command **sas92 &**

To run a SAS program (export.sas) in batch mode (the program is run without visually opening SAS), use the command **sas92 export.sas** or **sas92 export**

To run a SAS program (export.sas) in interactive mode, SAS should be started and the program submitted from within.

If you run any programs in batch mode, you can see which programs are running, by using the **jobs** command

For example, if you run export.sas in batch mode (sas export.sas &), you can see this, by typing **jobs**. The UNIX session will show **Running sas92 export.sas&**

# Now you're logged on: Verifying SAS version

- With both SAS 9.1.3 and 9.2 versions available it becomes important to verify which SAS version is used by a programmer. To do this, simply use automatic SYSVER variable. Adding the following code to setup.sas of a SAS 9.x project will guarantee the correct version of SAS is used.

```
* Set the project SAS version (possible values 9.1/9.2);
 %let _projectSasVersion = 9.2;
* Check SAS 9.X compatibility;
data _null_;
    if "&sysVer." ne "&_projectSasVersion." then do;
        message = "ERROR:[PXL] SAS version &_projectSasVersion. is required for
        the project.";
        put message;
        rc = system("echo " || message);
        abort abEnd;
    end;
run;
```

# Now you're logged on: Transferring files 1

Multiple systems used at PAREXEL are:

- **winSCP recommended system**
- **Samba**
- **sftp – (transfers between unix systems, e.g. kennet)**

Samba is software that is run on UNIX and allows UNIX and Windows to communicate as if UNIX, hence allowing you to view files in Windows which are being hosted on UNIX.

You can map a Samba drive to your UNIX home directory as \\kennet\<UNIX account>, and a project drive as \\kennet\<e999999_sponsorname>

- If you use Samba to transfer files, be aware that Samba carries the win \r\n through and do not fix them as ftp/scp programs do in ascii mode.
- If 2 files with names that only differ in case, only one is accessible via samba. the content of the second one is hidden

To convert a file (testwin.txt) to a UNIX formatted file (testUNIX.txt), the command **dos2ux testwin.txt > testUNIX.txt** should be used.

# Now you're logged on: Transferring files 2

WinSCP is a free SFTP (Secure Shell File Transfer Protocol) and SCP (Secure Copy Protocol) client for windows. It's main function is to allow safe copying of files between a local and remote computer (e.g. the transfer of files from your local machine to a UNIX directory)

To use WinSCP, you can start this from **Start – All Programs – WinSCP**

You can set up an instance, by clicking the **New** button

You then need to specify the instance (e.g. kennet), username (e.g. cantrea) and password

You can change transfer options, by using **Options – Preferences –** for example ensuring that WinSCP transfers SAS files in ascii format

http://winscp.net/eng/docs/guides

# Now you're logged on: Some other commands 1

By using the up and down arrows at the prompt you can recall previous / recent commands executed at the prompt.

The tab key can be used to autocomplete text you are typing, for example, if you want to edit a file areallyreallylongfilename.txt using xedit, the command

**xedit areally** followed by the user pressing the tab key would result in

**xedit areallyreallylongfilename.txt** being displayed which will save the user time in typing filenames in.

Note: the autocomplete option will only autocomplete as much as possible for uniqueness of file names.

e.g. if you have files test1.txt and test2.txt, then typing **te** followed by the tab key will result in **test** being displayed, as the next character could either be a 1 or a 2 so can not be autocompleted.

If you press tab twice, UNIX will list all the filenames that can be autocompleted

# Now you're logged on: Some other commands 2

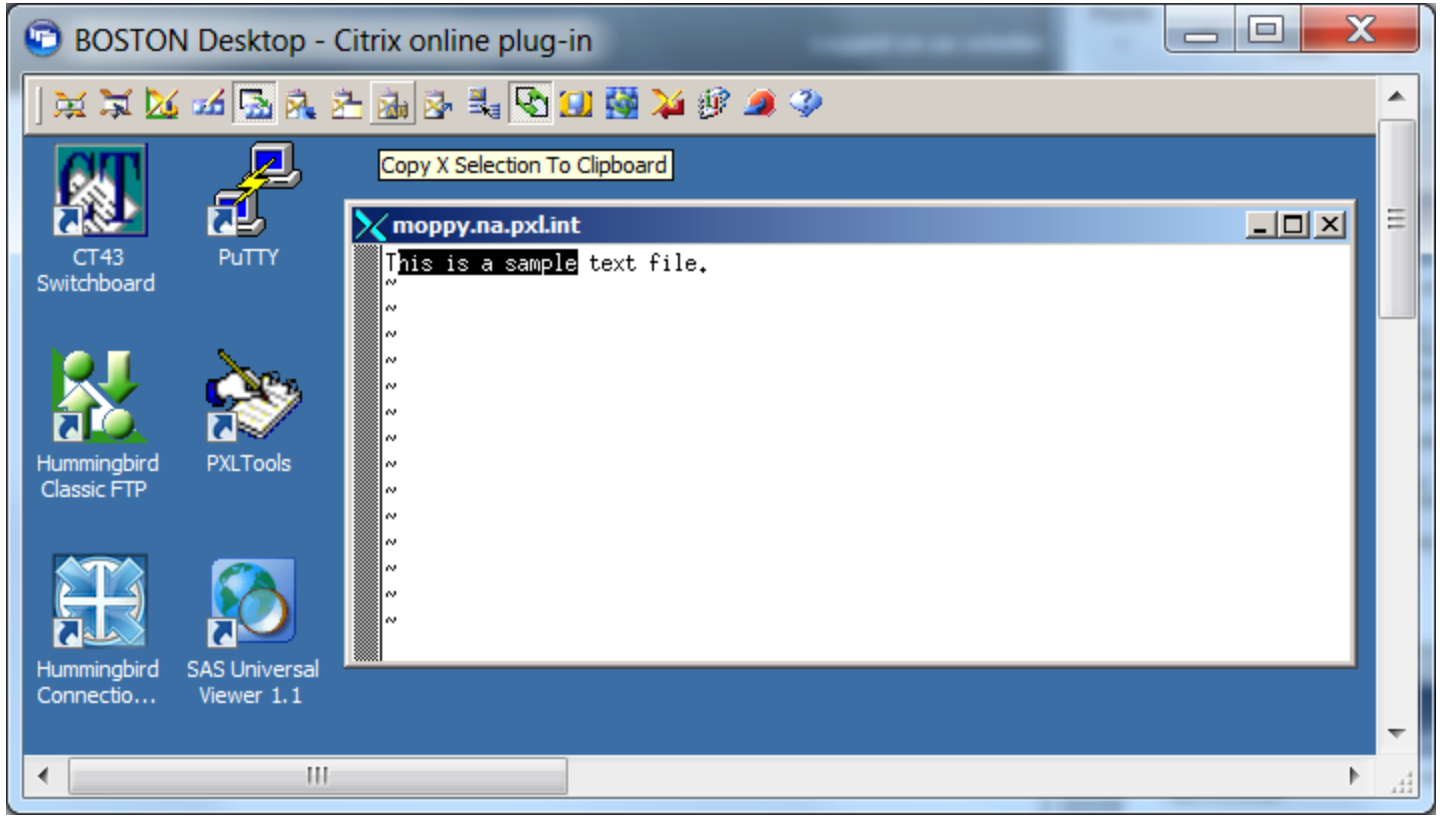
The **clear** command will clear the UNIX window with the prompt appearing at the top of the screen.

The middle mouse button can be pressed to paste text held within the paste buffer into UNIX – so could copy a word text file straight to UNIX

By hovering the mouse pointer over the scroll bar, pressing the middle mouse button and moving the mouse up and down, you can scroll up and down in the UNIX window.

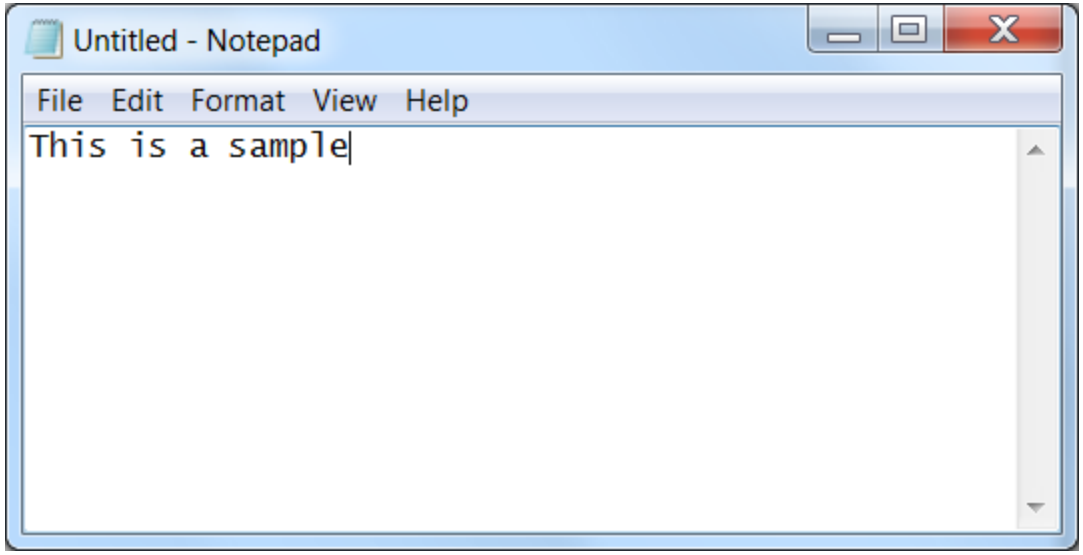
# Now you're logged on: Some other commands 2

To copy text, highlight the text to copy, click Copy X Selection from Exceed Toolbar, to place the text in the Clipboard



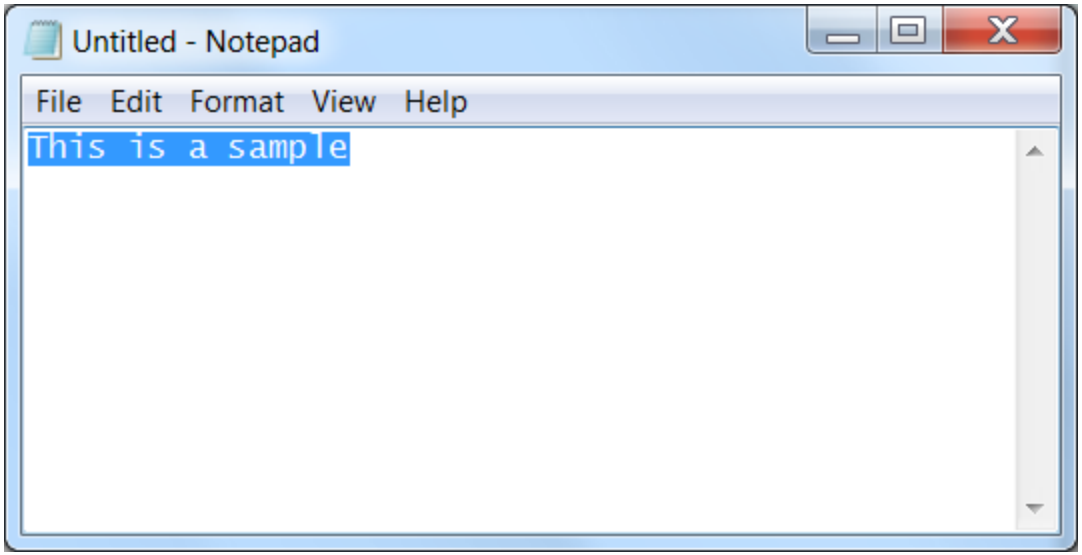
# Now you're logged on: Some other commands 2

Click paste, to past the contents of the Clipboard



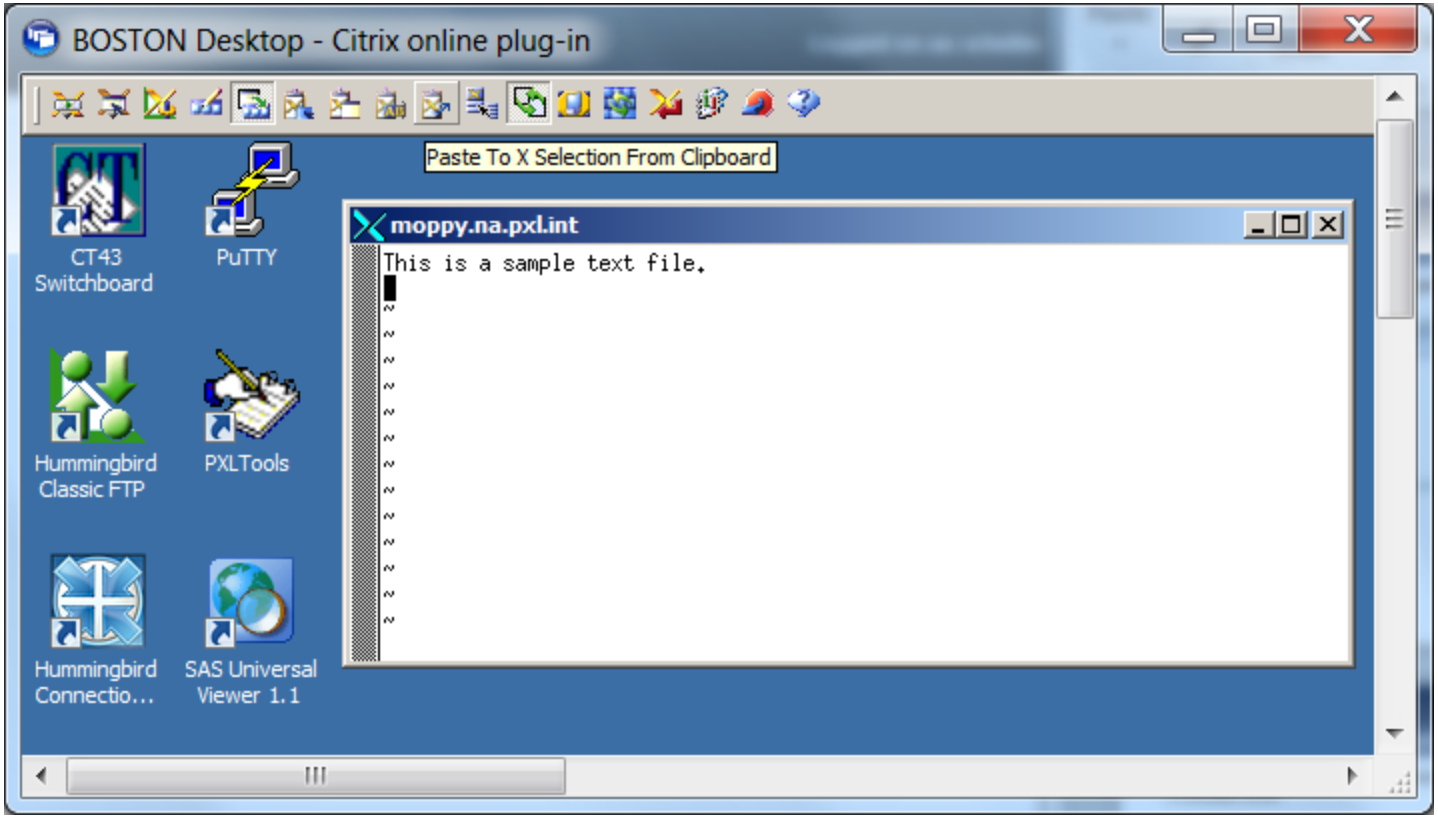
# Now you're logged on: Some other commands 2

Copy text from windows, by highlight the text to copy, click Copy to place the text in the Clipboard



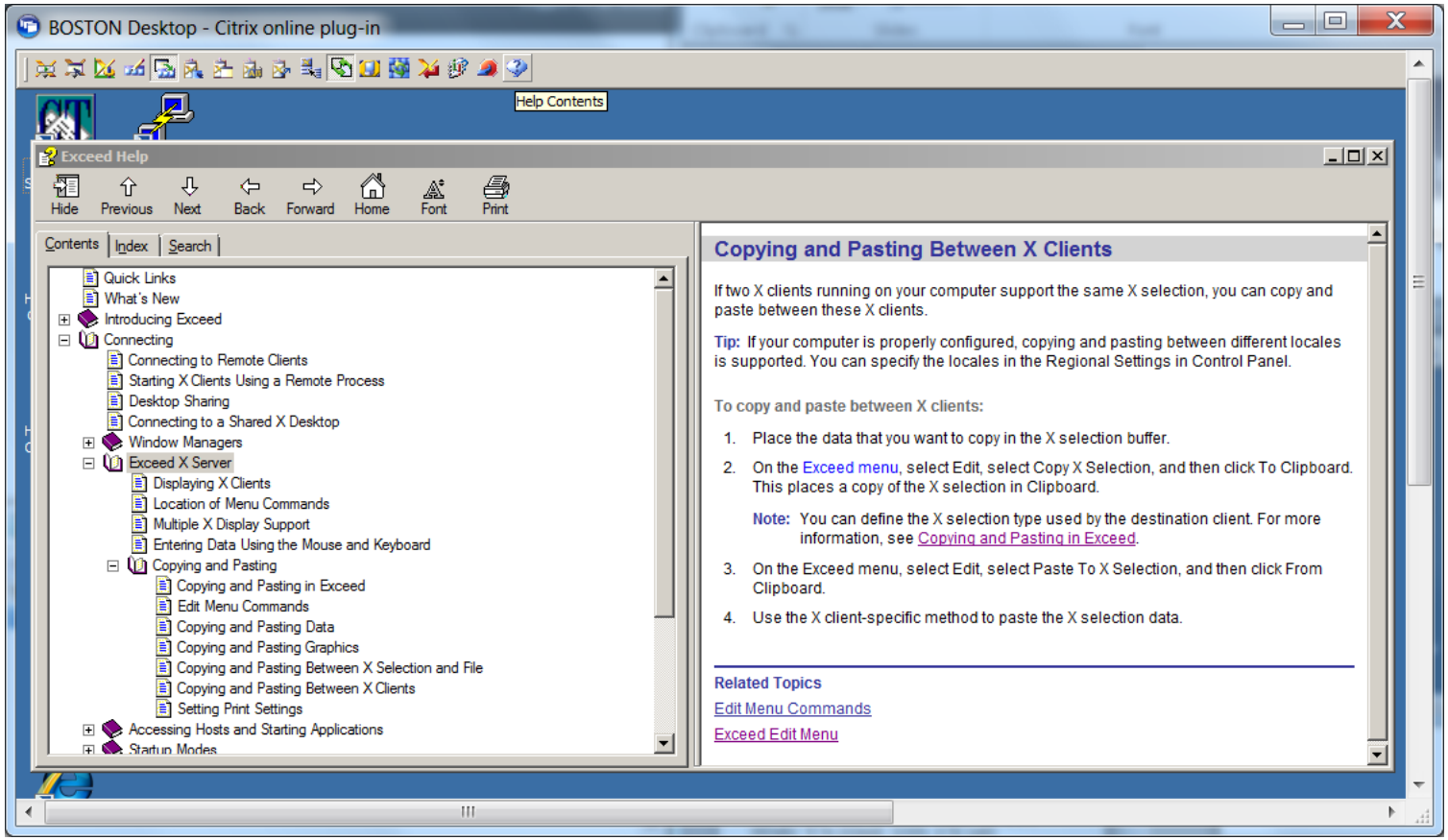
# Now you're logged on: Some other commands 2

Place cursor where to past text, Click Middle Mouse button to paste



# Now you're logged on: Some other commands 2

Exceed Copying and Pasting help can be viewed by Clicking on Exceed toolbar Help Contents



Further information

# Part Three: Additional Information

# Additional Information: Process Control 1

The **ps** command can be used to display the list of current processes being run by the user

- When ps is used without any options, it displays four items of information for the processes currently on the system
- ps itself is a process and it is terminated as soon as its output is displayed.

- The four items are labelled PID, TTY, TIME and CMD.
- TIME is the amount of CPU time in minutes and seconds that the process has been running, CMD is the name of the command that launched the process, TTY is the name of the console or terminal that the user logged into, which can also be found by using the tty command. PID is the process ID on the system.

# Additional Information: Process Control 2

```
/home/users/cantrea/unix_exercises . ps
   PID TTY          TIME COMMAND
 26588 ttyp1    0:00 ps
 17290 ttyp1    0:00 motifxsassm
 17274 ttyp1    0:01 sas
 17277 ttyp1    0:00 elssrv
 17191 ttyp1    0:00 dtpad
 13622 ttyp1    0:00 -bash
```

The command **psaux** may be used to display more complete and complex information on the processes currently running on the system

```
/home/users/cantrea/unix_exercises . psaux less | tail -n 10
lp         1548  1499  0.00    9     36 S  Dec 19     00:00 cat /var/spool/lp/request/printer20/dA4278taipan
lp         1427  1409  0.00   62    272 S  Dec 19     00:00 /usr/sbin/lpsched
cantrea   14825 14814  0.00   54    536 S 14:25:23    00:00 sort -nrk 4
cantrea   14823 14822  0.00   80    344 R 14:25:23    00:00 ps -e -o ruser,pid,ppid,pcpu,sz,vsz,state,stime,time,args
cantrea   14822 14814  0.00  383   1692 S 14:25:23    00:00 /usr/local/bin/bash /usr/local/bin/psaux less
cantrea   14815 10543  0.00   22    104 S 14:25:22    00:00 tail -n 10
cantrea   14814 10543  0.00  383   1692 S 14:25:22    00:00 /usr/local/bin/bash /usr/local/bin/psaux less
cantrea   11515 11512  0.00   45    208 S 14:12:59    00:00 /opt/SAS/SAS_9.1/utilities/bin/elssrv  6 3 3 2
bacons    14817 21056  0.00   13     56 S 14:25:22    00:00 sleep 1
RUSER       PID  PPID  %CPU   SZ    VSZ S    STIME     TIME COMMAND
```

# Additional Information: Process Control 3

Processes can either be run in the foreground (default) or the background.

To run a process (sas) in background mode, type **sas &** at the command prompt.

If a process is not run in background mode, then the UNIX prompt can not be accessed until the process is closed or completed.

In some instances, processes may become out of control and can not be accessed (such as a SAS session).

The **kill** command is used terminate a process, to terminate a process (17274), use **kill 17274**

Use **kill -9 17274** to force the process to be killed - should be used if the command above does not result in the process being stopped

You can only kill your own processes (e.g. processes that your account is running)

# Additional Information: Disk usage

As a user of UNIX, you need to be aware of your impact on the system both from a disk usage and disk space aspect.

The **du** command displays the number of 512-byte blocks used by all files and directories

The command **du –xsk * | sort –n** will display the total file space (in 1024-byte blocks) used by each file listed in increasing disk usage order (if a directory contains multiple files, only the total directory size will be listed)

The **bdf** command displays disk space information, including disk space used, disk space remaining for all file systems present.

To see the space used on the disk you are currently on, use the **bdf .** command.

# Additional Information: zip / unzip

The **zip** command creates a compressed file containing specified files, this can reduce the amount of disk space used by project files.

To create a zip file (test.zip) containing files a.txt and b.txt use

**zip test.zip a.txt b.txt**

By also removing nonessential files and compressing large files, disk space can be used sensibly.

The **unzip** command uncompresses a zip file.

To unzip a zip file (test.zip) containing files a.txt and b.txt use

**unzip test.zip a.txt b.txt**

The zip functionally can be used to allow easier transfer of files to another file system, e.g. it is easier to move a 1MB file to Windows rather than an unzipped 500MB file.

Similarly SAS datasets should ideally be stored in SAS transfer files for a similar file compression.

# Additional Information: Logging off

To exit the UNIX session cleanly, type **exit** at the command prompt.

Note: Ctrl+D can be used on the UNIX screen – should ensure that you have closed all applications before doing this or you may lose work

Then use the Start button – Log off option to log off from the Citrix desktop session.

It is good practice to do this to ensure that no stray UNIX sessions or processes are left running (which leads to system slowdown).

# Additional Training

- SAS Enterprise Guide \ UNIX SAS Windows Workshop, LMS
- Subversion Version Control, LMS

# Thank you