



Supporting Your Voyager Server

Managing your Voyager System Series

Copyright Statement

All of the information and material inclusive of text, images, logos, product names is either the property of, or used with permission by Ex Libris Ltd. The information may not be distributed, modified, displayed, reproduced – in whole or in part – without the prior written permission of Ex Libris Ltd.

TRADEMARKS

Ex Libris, the Ex Libris logo, Aleph, SFX, SFXIT, MetaLib, DigiTool, Verde, Primo, Voyager, MetaSearch, MetaIndex and other Ex Libris products and services referenced herein are trademarks of Ex Libris, and may be registered in certain jurisdictions. All other product names, company names, marks and logos referenced may be trademarks of their respective owners.

DISCLAIMER

The information contained in this document is compiled from various sources and provided on an "AS IS" basis for general information purposes only without any representations, conditions or warranties whether express or implied, including any implied warranties of satisfactory quality, completeness, accuracy or fitness for a particular purpose.

Ex Libris, its subsidiaries and related corporations ("Ex Libris Group") disclaim any and all liability for all use of this information, including losses, damages, claims or expenses any person may incur as a result of the use of this information, even if advised of the possibility of such loss or damage.

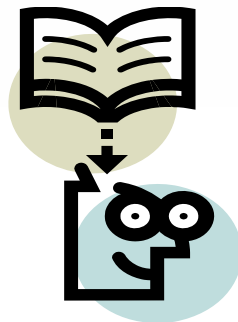
© Ex Libris Ltd., 2011

Agenda

- **Stuff you should know**
- Talking to your server
- Listening to your server
- Log files
- Be proactive
- Upgrading
- vi
- Recap and resources

Stuff You Should Know

- Basic O.S. and shell commands ([handout](#))
- vi Unix editor; vi = “visual” ([handout](#))
- Starting and stopping “processes”
- PuTTY (secure “Telnet”)
- WinSCP (secure “FTP” for uploading/downloading files and more!)
- WinMerge (file comparison tool)
- Your configuration, software versions, etc.



Terminology

- File – bits on disk or tape
- Program – executable, binary
- Script – ASCII file (editable) talks to shell
- Shell – command interpreter (UI); talks to kernel
- Kernel – core or key components of the O.S.; talks to hardware and includes process management

Terminology

- Database – a system that organizes, stores, and retrieves large amounts of data.
- Oracle – RDBMS
- VGER instance – Oracle application
- Tablespace – a **logical** unit comprised of one or more datafiles
- Database schema – database objects associated with a unique “username” (e.g., xxxdb)

Terminology

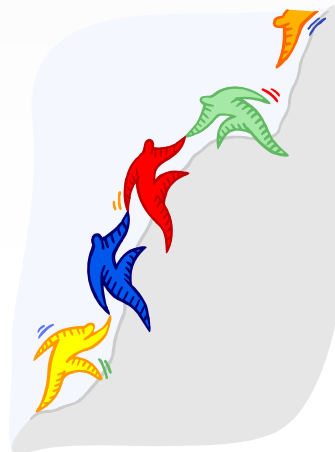
- / – “root” directory of the server
- /m1 – base directory for Voyager
- /m1/voyager – all Voyager files on the Unix server are under this directory
- /m1/voyager/xxxdb – database directory containing all database-specific files
- Use WinSCP to look around the file structure

Detecting Versions

- **Solaris/Linux** version plus patch level of the OS. Will also tell you what type of server you are running:
 - `uname -arv`
- To find out **Oracle**, you simply can run sqlplus:
 - `/export/home/voyager => sqlplus`
- For **Voyager** version check `voyager.env` file
- **Apache**: `cd /m1/shared/apache2/bin`
 - `httpd -v`

Our Major Players

- Apache – Web Server
- Tomcat – Java virtual machine
- Voyager – ExLibris ILS software
- Oracle - rdbms (database software)
- Operating System - Solaris, Linux, Windows...
- Clients - Cataloging, Acquisitions, Circulation...



When You Login...

- A typical “terminal session” is when you login to gain access, execute commands to do work, then logout to terminate your connection.
- User → Shell → Kernel → Hardware/O.S.
- Command Syntax:
 - Command [-options] [arguments]
 - For example: `ls -l -a` (or `ls -la`)
 - For example: `tail -20 log.voyager`

Three Important Things

- UNIX commands can't be abbreviated
- UNIX commands are case sensitive
- There are two metacharacters:
 - **?** (match one character)
 - ***** (match varied number of characters)

- Example: `find .??* -prune`

64.94.37.24 - PuTTY

```
$ find .??* -prune  
.TTauthority  
.Xauthority  
.bash_history  
.bash_logout  
.bash_profile  
.bashrc  
.dt  
.dtprofile  
.emacs  
.history  
.java  
.kde  
.lesshst  
.profile  
.profile-11-5-2011  
.profile.pre.7.0.1  
.sh_history  
.ssh  
.sunw  
.viminfo  
.zshrc  
$ █
```

Voyager versus Root User

- Login via console or remotely.
- Always be aware of **who you are!**
 `whoami` command
- Use the `pwd` command to find out **where you are.**
- Most work is done as voyager user.
- Be very, very careful when you're root user.
- voyager and root users each
 have their own *crontabs...*

su versus sudo

- The su (short for substitute user) command makes it possible to change a login session's owner
 - `su - oracle`
- The sudo (superuser do) command allows a permitted user to execute a *command* as root, or another user, as specified in the *sudoers* file (configuration is site dependent!)
 - `sudo su -`

dc02vg0179na.hosted.exlibrisgroup.com - PuTTY

```
/home/voyager => whoami
voyager
/home/voyager => pwd
/home/voyager
/home/voyager => su - oracle
Password:
Welcome to dc02vg0179na.hosted.exlibrisgroup.com
[VGER] oracle@dc02vg0179na.hosted.exlibrisgroup.com : oracle/ => whoami
oracle
[VGER] oracle@dc02vg0179na.hosted.exlibrisgroup.com : oracle/ => pwd
/home/oracle
[VGER] oracle@dc02vg0179na.hosted.exlibrisgroup.com : oracle/ => exit
/home/voyager => whoami
voyager
/home/voyager => pwd
/home/voyager
/home/voyager => sudo -s
[sudo] password for voyager:
/home/voyager => whoami
root
/home/voyager => id
uid=0(root) gid=0(root) groups=0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10(wheel),104(pkcs11)
/home/voyager => exit
/home/voyager => whoami
voyager
/home/voyager => pwd
/home/voyager
/home/voyager => █
```

Agenda

- Stuff you should know
- **Talking to your server**
- Listening to your server
- Log files
- Be proactive
- Upgrading
- vi
- Recap and resources

Talking To Your Server

- When you log in, you are said to be at the command line or command prompt. This is where you enter UNIX commands:

```
$date Fri Mar 1 22:59:28 EDT 2011
```

```
$uptime 10:59PM up 259 days, 9:44, 5 users, load average:  
3.81, 14.27, 13.71
```

```
$hostname gonzo
```

- You can combine commands in a file called a **script** that allows you to run them one after another.

Scripting Essentials

- Scripting allows you to automate tasks.
- If you do it more than once, use a script.
- If you script it, document your script.
- If you don't have a doc system, start one.
- If you change something, document it!

Writing Scripts

- Scripts are cool!
- A script can be a simple UNIX command-line script
- Or it can be a shell script
- There are a number of different scripting shells



Example of Korn Shell Script

```
#!/bin/ksh
```

```
date
```

```
uptime
```

```
hostname
```

- To run: `./example.ksh`
- Note the file must be **executable!**

File Permissions

- When you first create a shell script, it will usually not be executable. Use the symbolic `chmod` command with the `+x` option to add the execute permissions
 - `chmod a+x example.ksh`
- Use the `ls -l` command to list a file's permissions
- 3 permissions levels: read, write, execute

chmod Command

- Used to set/modify a file's permissions.
- Read the man pages!
- You can change the entire permission pattern of a file in a single go using one number:
 - `chmod 755 example.ksh` (*755: I don't mind if other people read or execute this file, but only I should be able to modify it*)
- **Symbolic** notation versus **Octal** notation.

Critical Task Scripts

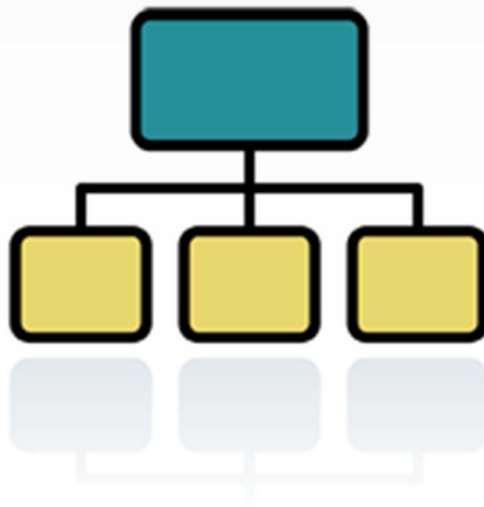
- Master Script (or scripts):
 - Shuts down services
 - Runs backup
 - Rotates logs
 - Brings services back up
 - Runs Circ and Acq batch jobs
 - Loads patrons

Other “Less Critical” Scripts

- Log CPU, memory, disk and pagefile stats with mpstat, vmstat, iostat and sar respectively, to warn you if anything hits 90% of capacity.
- Look for errors or warnings in critical Oracle and System log files.
- Report on /data and /mfhd.data file sizes (regen warning).

Nice-To-Have Scripts

- Deletion of rotated log/report archive directories
- Making sure the OPAC is up
- Making sure Email is working
- Any other tasks you do frequently...



Backup Script

- Backups are mandatory (in your contract!)
- Full easier to recover from than incremental.
- “Cold” is best where you **shut down everything**.
- Use tape rotation – multiple tapes protect you if one goes bad.
- Rotate tapes off-site.
- Listen to Ex Libris and follow their advice.
- Monitor your backup logs **daily**.

cron

- cron is a UNIX utility that allows tasks (such as running a script!) to be automatically run in the background at regular intervals by the cron daemon. These tasks are often termed as cron jobs.
- crontab (CRON TABLE) is a file which contains the schedule of cron entries to be run and at specified times.



crontab Commands

- `crontab -e` Edit your crontab file, or create one if it doesn't already exist.
- `crontab -l` Display your crontab file.
- `crontab -r` Remove your crontab file.
- **Note** : If you inadvertently enter the crontab command with no argument(s), do not attempt to get out with Control-d. This **removes all entries** in your crontab file. Instead, exit with Control-c.

crontab

- A crontab file has five fields for specifying day, date and time followed by the command to be run at that interval.

```
* * * * *      command to be executed
- - - - -
| | | | |
| | | | +----- day of week (0 - 6) (Sunday=0)
| | | +----- month (1 - 12)
| | +----- day of month (1 - 31)
| +----- hour (0 - 23)
+----- min (0 - 59)
```

cron Output

- By default cron jobs send an email to the user account executing the cronjob.
- Use: `>/dev/null 2>&1` to stop job email
- To collect the cron execution log in a file:

```
30 18 * * * rm /home/someuser/tmp/* >  
/home/someuser/cronlogs/clean_tmp_dir.log
```
- To append the log *and* stop the email:

```
30 18 * * * rm /home/someuser/tmp/* >>  
/home/someuser/cronlogs/clean_tmp_dir.log>/dev/null 2>&1
```
- ExL: DO NOT CRON REGENS!

cron Example: OPAC Test

```
#  
# Check the status of the OPAC. Every day at 5:07am,  
# email Tim and Laura the results  
#  
07 5 * * * /usr/local/utils/TestCatalog.pl | mailx -s  
'Voyager OPAC status' tim@coalliance.org lguy@mines.edu  
Panderse@mines.edu  
#
```

```
#!/m1/shared/bin/perl
#
# TestCatalog.pl 2010/03/02 Ken Herold
# Hamilton College
# See if OPAC is available
#

use LWP::UserAgent;

$url = "http://voytest.coalliance.org";
$ua = LWP::UserAgent->new;
$response = $ua->get($url);
$status = $response->status_line;

if ($status =~ m/200/) {
print "OPAC is OK\n";
}

if ($status =~ m/503/) {
print "OPAC is DOWN\n";
}

exit(0);
```


Weekly Restart crontab Example

```
#  
# Reboot the server 8:00am every Sunday  
# Changed to init 6 as per ExL 11/2009  
0 8 * * 0 /usr/sbin/init 6  
#
```

Agenda

- Stuff you should know
- Talking to your server
- **Listening to your server**
- Log files
- Be proactive
- Upgrading
- vi
- Recap and resources

Processes

- UNIX is a true multi-tasking system.
- The way it can manage many tasks at once is through processes.
- Each system and end-user task is contained within one of these processes.
- You can use the `ps` command to look at UNIX processes. Other commands allow you to control them.

Process Status Command

- The `ps` command can be used to list all running processes
- The `ps` command has a legion of options
- The first process is called "init." It is created by the kernel during the boot sequence
- Important system processes are owned by root.
- Some processes run from startup to shutdown (like init), others come and go rapidly

Process Management

- `ps` command
 - `ps -ef` – Gives you list of all processes running including their unique process identification number (pipe it through the **less** command!)
- `kill` command
 - `kill 485` – Kill PID 485 gracefully
 - `kill -9 485` – *Strongly* kill PID 485

Useful Process Commands

- `ps -ef | grep -i voyager`
- `ps -ef | grep -i ora_`
- `ps -ef | grep -i httpd`
- `ps -ef | grep -i java`
- `ps -ef | grep -i listener`
- `ps -ef | grep -i tomcat`

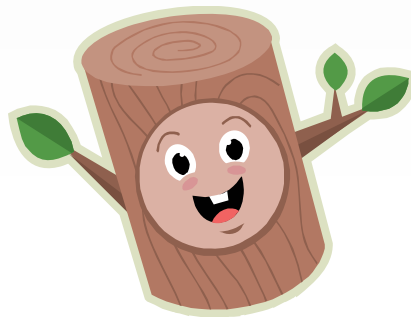


Agenda

- Stuff you should know
- Talking to your server
- Listening to your server
- **Log files**
- Be proactive
- Upgrading
- vi
- Recap and resources

Log files

- In general the Voyager and other logs are more useful for **diagnosis** than prevention.
- Default output often voluminous and includes spurious errors and warnings, and may simply be not meaningful.
- There are O.S. logs, Oracle logs, Apache logs, Voyager logs, Tomcat logs....
- If you're having issues make copies of logs!



Voyager Log Files

- Voyager Server Logs
 - /m1/voyager/xxxdb/log/log.voyager
 - /m1/voyager/xxxdb/log/z3950svr_access.log
- Voyager Deleted Records Logs
 - /m1/voyager/xxxdb/rpt/delete.*
- Logs for Voyager Jobs
 - /m1/voyager/xxxdb/rpt/*job.log (circjob, etc.)

Tomcat Log Files

- Tomcat WebVoyage logs
 - /m1/voyager/xxxdb/tomcat/vwebv/logs/catalina.out
 - /m1/voyager/xxxdb/tomcat/vxws/logs/catalina.out
 - /m1/voyager/xxxdb/tomcat/logs/*

Oracle Log Files

- Instance – level logging:
 - Solaris/Linux/AIX:
\$ORA_LOG/alert_VGER.log
- Oracle networking logs:
 - \$ORA_NET/../../log/sqlnet.log
 - \$ORA_NET/../../log/listener.log
- Note the use of aliases here (e.g., \$ORA_LOG)

O.S. Logs

- /var/adm/messages (Solaris)
- /var/log/messages (Linux)

Apache Logs

- /m1/shared/apache2/logs
 - access_log
 - error_log

- /m1/shared/apache2/xxxdb/logs
 - access_log
 - error_log

Using tail in *Real Time*

- tail -f log.voyager
- press Enter key twice
- replicate your issue
- review log.voyager in “real time”
- look for WARNING, ERROR, UNABLE, FAILURE

Logwatch

- Will catch things like failed ssh attempts, su attempts and what commands were run with su and sudo, give you a snapshot of filesystem free space, etc.

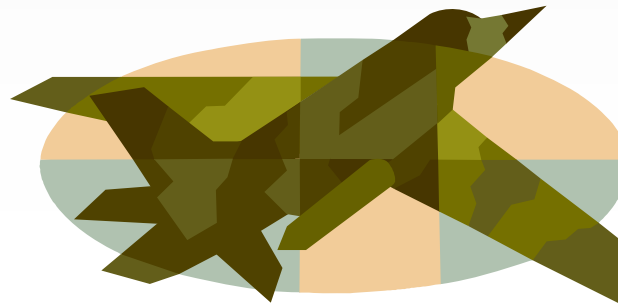
<http://sourceforge.net/projects/logwatch/files/>

Agenda

- Stuff you should know
- Talking to your server
- Listening to your server
- Log files
- **Be proactive**
- Upgrading
- vi
- Recap and resources

Be Proactive

- How is a Systems Administrator like a Fighter Pilot?
- Decision-making occurs in a recurring cycle of Observe-Orient-Decide-Act.
- Avoid *reacting* (that is the way a fighter pilot will lose a dogfight)
- Instead proactively identify and fix



Disk Usage

- Keep an eye on your available disk space
- The `df` (disk free) command can be used to look at available disk space
 - `df -k`
 - `df -h` (h=human readable)
- The `du` (disk usage) command can be used to see disk usage for specific files and directories
 - `du -h /m1/voyager/xxdb/data/* | sort -n | tail`
- Disk errors show up in the O.S. log

```
voytest.coalliance.org - PuTTY
/home/voyager => df -h /m1
Filesystem      Size  Used Avail Use% Mounted on
/dev/hdv1       316G   52G  248G  18% /
/home/voyager =>
/home/voyager => du -h /m1/voyager/csmdb/data/*
247M    /m1/voyager/csmdb/data/bib_text.1.tbt
3.6M    /m1/voyager/csmdb/data/bib_text.1.tdr
4.0K    /m1/voyager/csmdb/data/csmdb.1.bif
480M    /m1/voyager/csmdb/data/csmdb.1.dc
4.0K    /m1/voyager/csmdb/data/csmdb.1.kil
4.0K    /m1/voyager/csmdb/data/dynamic.bif
2.1M    /m1/voyager/csmdb/data/dynamic.dc
4.0K    /m1/voyager/csmdb/data/dynamic.que
/home/voyager =>
```

Memory

- For an Oracle server memory is critical.
- Check `/var/log/messages` (a system log file) for `swap` errors.
- Run the `free -m` command for how much memory you are currently using
- The `ps -aux` for percentage of memory used by processes
- Run `vmstat` to see if you're swapping too much
- A word about Zombies

```
voytest.coalliance.org - PuTTY
/home/voyager => free -m
      total        used         free       shared    buffers     cached
Mem:      8192        3109         5082           0           0         1295
-/+ buffers/cache:      1813         6378
Swap:      8192           0         8191
/home/voyager => █
```

“Tuning”

- Oracle has “tuning.”
- It’s all about resource allocation.
- Tuned per hardware configuration (RAM, disks, # bytes/records) & your contract.
- Badly tuned shared memory, pagefiles, swap space, tablespace, bandwidth, etc. can all cause issues...
- Your O.S. is tuned, too!

R U Firewalled?

- Incoming and outgoing packets MUST be allowed through your firewall(s), routers(s), & server(s)
- What's your timeout? Do you know it?
- Are you suddenly having problems? Did your I.T. folks make a change?



Security is Critical

- Keep current with patches
- Monitor your logs using [Nagios](#) or etc.
- Enforce strong passwords and change them often
- Use firewalls, SSH (PuTTY), SFTP, WinSCP
- Have good backups with tested restores
- Physical security: no passwords on post-its!

Stay Current!

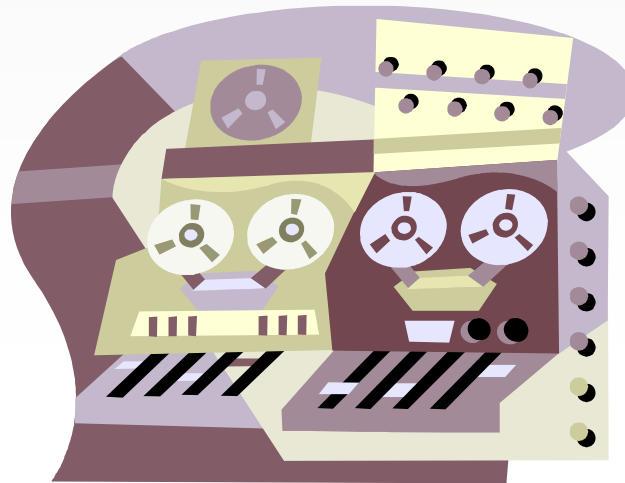
- Patch your O.S. (ExL will not do this for you)
 - MUST stop Apache, Voyager, & Oracle first
 - Check vendor's web site.
- Keep your Voyager version current.
- Keep 3rd party software (Tomcat, Apache, etc.) current.
- Make sure your hardware is under warranty.
- Consider an IT infrastructure support firm.

Agenda

- Stuff you should know
- Talking to your server
- Listening to your server
- Log files
- Be proactive
- **Upgrading**
- vi
- Recap and resources

Reasons to Upgrade

- Bug Fixes
- Enhancements
- Is your version supported?
- Keeping Tomcat current
- Not necessarily a case of “If it ain’t broke don’t fix it.”



So You Think You Want to Upgrade?

- Check server hardware requirements
- Check server O.S./software requirements
- Check server file system and disk space
- Check server backups and restores
- Check PC hardware requirements
- Check PC O.S./software requirements

Different Ways to Upgrade

- Schedule with the installation team
- Voyager Installation Kit ("VIK")
- Patch

Have Ex Libris Do It

- Open an incident; specify preferred dates
- ExL will take care of all 3rd party products (Oracle, Apache, Tomcat, etc.) and all modules
- You must do customizations, O.S. upgrades and patches, site-specific add-ons (like newBooks), client installations
- Can take several days (extra for Oracle, Analyzer)
- It's free!

Have Ex Libris Do It

- Once you have a date, work out your timeline.
 - **Before:** Preview Server, customizations, server cleanup, connectivity for technician, decide about options available (if applicable), worksheet
 - **During:** Clients, patience
 - **After:** Tomcat, add-ons, testing, worksheet
- Importance of good communication and reading all the documentation

Voyager Installation Kit

- Do It Yourself. Includes Apache/Tomcat/Perl
- With Voyager 8 it will upgrade **Oracle** too.
- Are your extension modules all supported?
- Double-check your backups/restores.
- Decide when. Carve out enough time.
- Get your documentation together.
- Know License Key & passwords (e-Service)
- Test run VIK installation through step 4.1.

Should You Run The VIK?

- ExL will still do upgrades for you *for free*
- Takes time – just how much depends...
- Familiarity with your server and Voyager
- Extension modules all supported?
- Oracle/Voyager/O.S. versions okay?
- Easy but not stress-free (“Intestinal Fortitude”)
- Read the Manual!
- See my 2010 ELUNA presentation on the VIK

Patch

- Do It Yourself.
- Or... Ex Libris will do it for you.
- Applies to Service Packs (e.g., 7.2.1 to 7.2.3)
- Download simple script.
- Read/understand the documentation.
- You need to test Voyager after you are done.
- Don't forget clients! And Tomcat!
- **NEW!** Voyager 8.1 UTIL Menu

Customer Center

- No matter which method you choose.
- Will help with planning.
- Provides access to all documentation.
- Go to the [Documentation Center](#), choose Voyager and look for the “Upgrade Release Planning” section.
- Not all versions have identical documents.
- <http://customercenter.exlibrisgroup.com>

Agenda

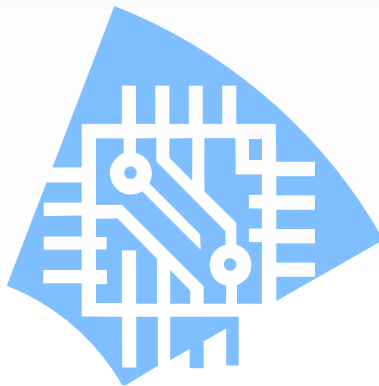
- Stuff you should know
- Talking to your server
- Listening to your server
- Log files
- Be proactive
- Upgrading
- **vi**
- Recap and resources

What is vi?

- A screen oriented editor
- Included with most UNIX system distributions
- Command driven
- Categories of commands include:
 - General administration
 - Cursor movement
 - Insert text
 - Delete text
 - Paste text
 - Modify text

Modes in vi

- There are three basic modes of vi:
 - Command mode (default; invoke using `Esc`)
 - Insert mode (invoke using `i` or a command)
 - Line mode (invoke using `:` command)



Starting and ending a vi session

- vi filename
- If the file doesn't exist, it will be created
- Otherwise it will open existing file
-
- :q! quit without saving
- :wq write (save) and quit
- :w write
- :w! *newfile* write file to '*newfile*' (BEWARE!)

Cursor Control

- **h** – one space to the left (also try left arrow)
- **j** – one line down (also try down arrow)
- **k** – one line up (also try up arrow)
- **l** – one space to the right (also try right arrow)
- **\$** – end of current line
- **^** – beginning of current line
- **G** – end of file

Input Mode

- **i** – will insert character at present cursor position
- **I** – inserts at beginning of the line
- **a** – appends at present position
- **A** – appends at end of the line
- **o** – inserts a blank line below
- **O** – inserts a blank line above

Deleting Text

- **x** – deletes current character
- **D** – deletes from cursor to end of line
- **dw** – deletes current word
- **dd** – deletes current line
- **dG** – deletes all lines to end of file including current line
- **u** – undo last change

Copying/Moving/Changing Text

- **yy** – copy the line
- **yw** – copy the word
- **dw** – cut the word
- **dd** – cut the line
- **p** – paste yanked lines below
- **P** – paste yanked lines above

Searching for Text

- Do in Line Mode (invoke using : command)
- **/ERROR** will locate first occurrence of pattern
- **n** – locates next occurrence
- **N** – locates previous occurrence

Agenda

- Stuff you should know
- Talking to your server
- Listening to your server
- Log files
- Be proactive
- Upgrading
- vi
- **Recap and resources**

Recap and Resources

- PuTTY - Free terminal emulator application which can act as a client for Telnet and SSH
- WinSCP - Free FTP and SFTP client for Windows
- WinMerge – Differencing and merging tool
- UltraEdit - Commercial Text Editor/File Compare
- ExL Customer Center, EL Commons
- The Internet, man pages, books and more books
- Voy-L, Voy-Admin lists
- ExL Support