p4 grep

Engineering Tech Talk 2010.1

Presenter: Sven Erik Knop

Perforce has grep!

- grep: **g**lobal/**r**egular **e**xpression/**p**rint
- **Search** through **text files** on the **server**
- Search:
  - Fixed or regular expressions
  - Case sensitive or insensitive
  - Match or inverse match
- Text files:
  - Default is to only consider text files
  - -t overrides this for binary files

Command syntax

- `p4 grep`
  `[ -a -i -v -n`
  `-A after -B before -C context`
  `-l -L -t -s -F -G ]`
  `-e pattern file[revRange]...`

- file[revRange] identical to 'p4 files/p4 print'
  - -a searches all revisions within the range

-e pattern

- Why '-e'?
  - Follows the Perforce arguments usage
  - Alternative syntax for command line grep

- Examples:

```
p4 grep –e "Copyright" //depot/main/p4/....h
p4 grep –e foo –a //depot/....txt@1000,2000
```

Case sensitive search

- -i
  - Case-insensitive search (slower)
  - Default is case-sensitive search

```
p4 grep –i –e copyright //depot/main/p4/....h
```

  - Matches COPYRIGHT, Copyright etc.

Match and no-match

- **-v**
  - Inverse search
  - Matches all lines not containing pattern

```
p4 grep –i –v –e completed //.../todo.txt
```

  - Matches every line in every todo.txt file that does not contain the word “completed” (in any case spellings)

Standard output and –n option

- Regular output is

```
filename#rev:matchedLine
```

```
> p4 grep –e “bar” //.../foo
//depot/test/foo#99:This line contains bar
//depot/o/foo#1:A bartender walks into a pub
```

- -n shows the line number

```
filename#rev:line:matchedLine
```

```
> p4 grep -n –e “bar” //.../foo
//depot/test/foo#99:125:This line contains bar
//depot/o/foo#1:1:A bartender walks into a pub
```

Tagged mode is supported

```
> p4 -ztag grep -e “bar” //depot/test/foo
... depotFile //depot/test/foo
... rev 99
... matchedLine This line contains bar

> p4 -ztag grep -n -e “bar” //depot/o/foo
... depotFile //depot/o/foo
... rev 1
... line 1
... matchedLine A bartender walks into a pub
```

Showing context

- Three options
  - -A <n> : show <n> lines **a**fter match
  - -B <n> : show <n> lines **b**efore match
  - -C <n> : show <n> lines around match
- -A and –B take precedence
- Contexts separated by '--' in the output
  - Output designed to be equivalent to 'grep'
- Matching line indicated by colon ':'
  - Context lines indicated by dash '-'

Showing context (example)

```
> p4 grep -C 1 -e 3 //depot/...
//depot/bar#1-2
//depot/bar#1:3
//depot/bar#1-4
--
//depot/foo#1-2
//depot/foo#1:3
//depot/foo#1-4
--
//depot/foo#1-12
//depot/foo#1:13
//depot/foo#1-14
```

Showing context (tagged mode)

```
> p4 -ztag grep -C 1 -e 3 //depot/bar
... depotFile //depot/bar
... rev 1
... type before
... matchedLine 2

... depotFile //depot/bar
... rev 1
... type match
... matchedLine 3

... depotFile //depot/bar
... rev 1
... type after
... matchedLine 4
```

-l and -L: Showing matching files

- Option -l list files with matches
  - Matching ceases after first match
- Option -L lists files that have no match
  - Again, matching ceases after first match

```
> p4 grep -L -i -e copyright //depot/src/....h
//depot/src/internal/simple.h#4
//depot/src/internal/complex.h#99
```

-t and -s

- **-t** interprets file as text
  - Overrides the rev filter that hides non-text files
  - Lines longer than 4096 characters cause error
    - (in any case, not just with the -t option)
  - p4 grep will print out warning
- **-s** (silent) prevents the printing of the warning
  - (but still stops the matching for this file)

-F for fixed string search (fgrep)

- Useful if the search string contains Meta characters such as ‘*’
- Does not use regular expressions
- -i for case-insensitive search still works

-G for regex search (default)

- Mixture between grep and egrep
- V8 (Henry Spencer) regular expressions
  - (adapted from Jam)
- Regular expressions special characters:
  - ‘.’, ‘?’, ‘+’, ‘*’, ‘[‘, ‘]’ , ‘(‘, ‘)’ , ‘|’, ‘^’, ‘$’, ‘\’

Regular expressions

- Zero or more branches, separated by '|'
  - Match if at least one branch matches
- Branch is zero or more concatenated pieces
- Piece is an atom possibly followed by '*', '+' or '?'
  - '*' matches zero or more atoms
  - '+' matches one or more atoms
  - '?' matches zero or one atoms

Atom

- Regular expression in parentheses
- Range ‘[ ]’
- ‘.’ - matching a single character
- ‘^’ - beginning of the line
- ‘$’ - end of the line
- ‘\’ followed by a character
- Any other character

Range

- Sequence of characters enclosed in '[]'
- Matches every character of the sequence
  - Unless first character is a '^' - then it matches every character not in the sequence
- If two characters are separated by a '-', this is a shorthand for all ASCII characters between them.
  - For example, [0-9] matches any decimal digit

Regular expression examples

- [1-9][0-9][0-9]-[0-9]+
  - Matches "513-0", "999-12345"
- foo|bar
  - Matches "foo" and "bar"
- ([cC]opyright)|([cC]opyleft)
  - Matches "Copyright" and "copyleft"
- [[(].*[])]
  - Matches "[something]" and "(else)"

Tunables

- dm.grep.maxfiles 10K
  - Maximum number of revisions that can be searched (including ranges)
- dm.grep.maxlinelength 4096
  - Maximum line length that can be searched

- Grep can be disabled by setting dm.grep.maxfiles to 0

Protection table

- **p4 grep** requires ‘read’ permission for all files to be searched

Audit log

- If P4AUDIT is defined, 'grep' will be audited using the tag 'grep':

```
2010/03/08 14:49:56 sknop@alita 127.0.0.1 grep //depot/a2#1
2010/03/08 14:49:56 sknop@alita 127.0.0.1 grep //depot/ab#1
2010/03/08 14:49:56 sknop@alita 127.0.0.1 grep //depot/b2#1
2010/03/08 14:49:56 sknop@alita 127.0.0.1 grep //depot/ba#1
2010/03/08 14:49:56 sknop@alita 127.0.0.1 grep //depot/bar.txt#3
```

Conclusion

- `p4 grep`
  `[ -a -i -v -n`
  `-A after -B before -C context`
  `-l -L -t -s -F -G ]`
  `-e pattern file[revRange]...`

Search through file content on the server

Questions?