

Introduction to Perforce for Users

Introduction

- Introductions
- Class Schedule
- GUI vs. CLI
- About the Exercises

Course Contents (Day 1)

- Overview
- Help!
- File Operations
- File Reporting and Revision Specifiers
- Changelist Management
- Handling File Conflicts

Course Contents (Day 2)

- Client Workspace Management
- Branching and Integration
- Streams
- Labels
- Defect Tracking

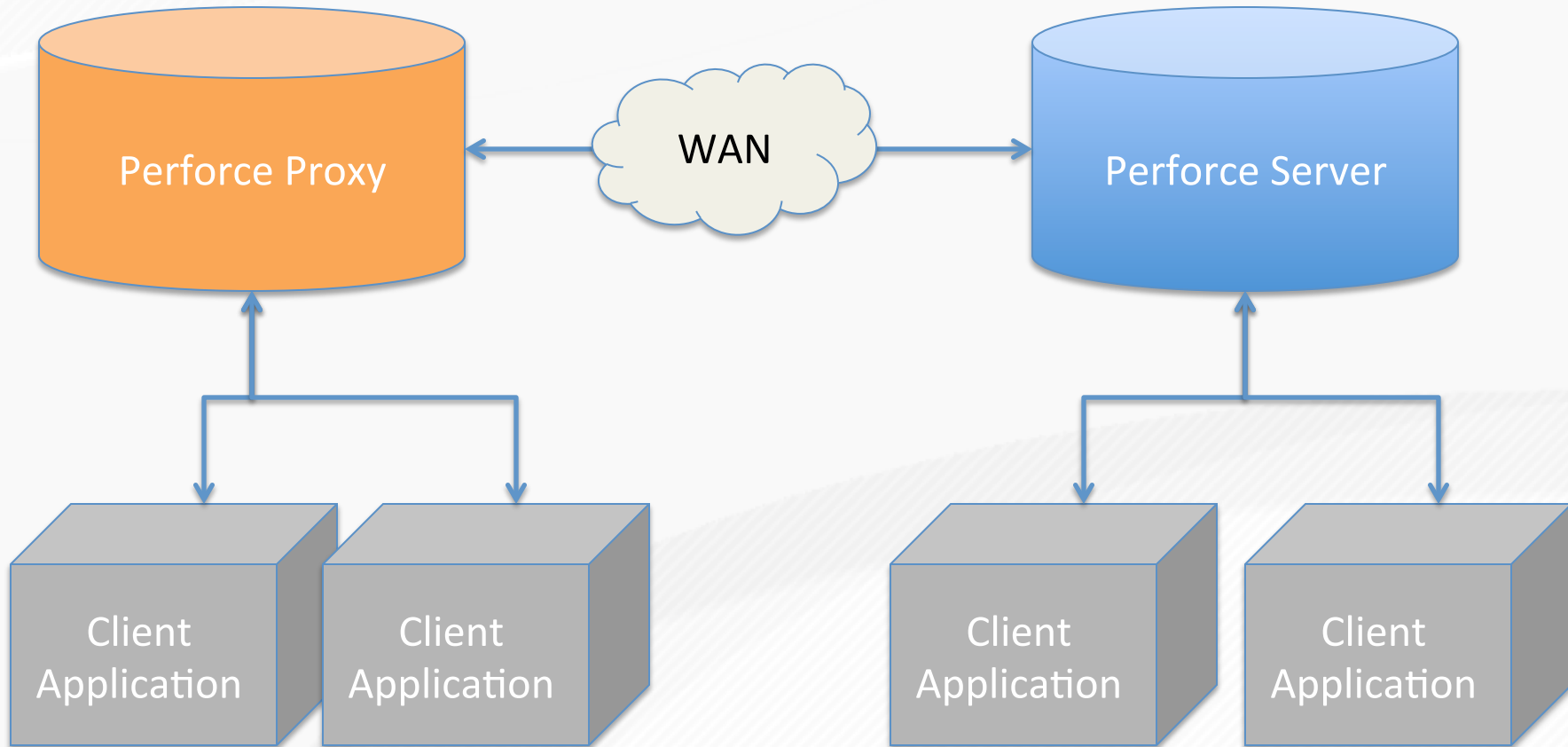
Introduction to Perforce for Users

Overview

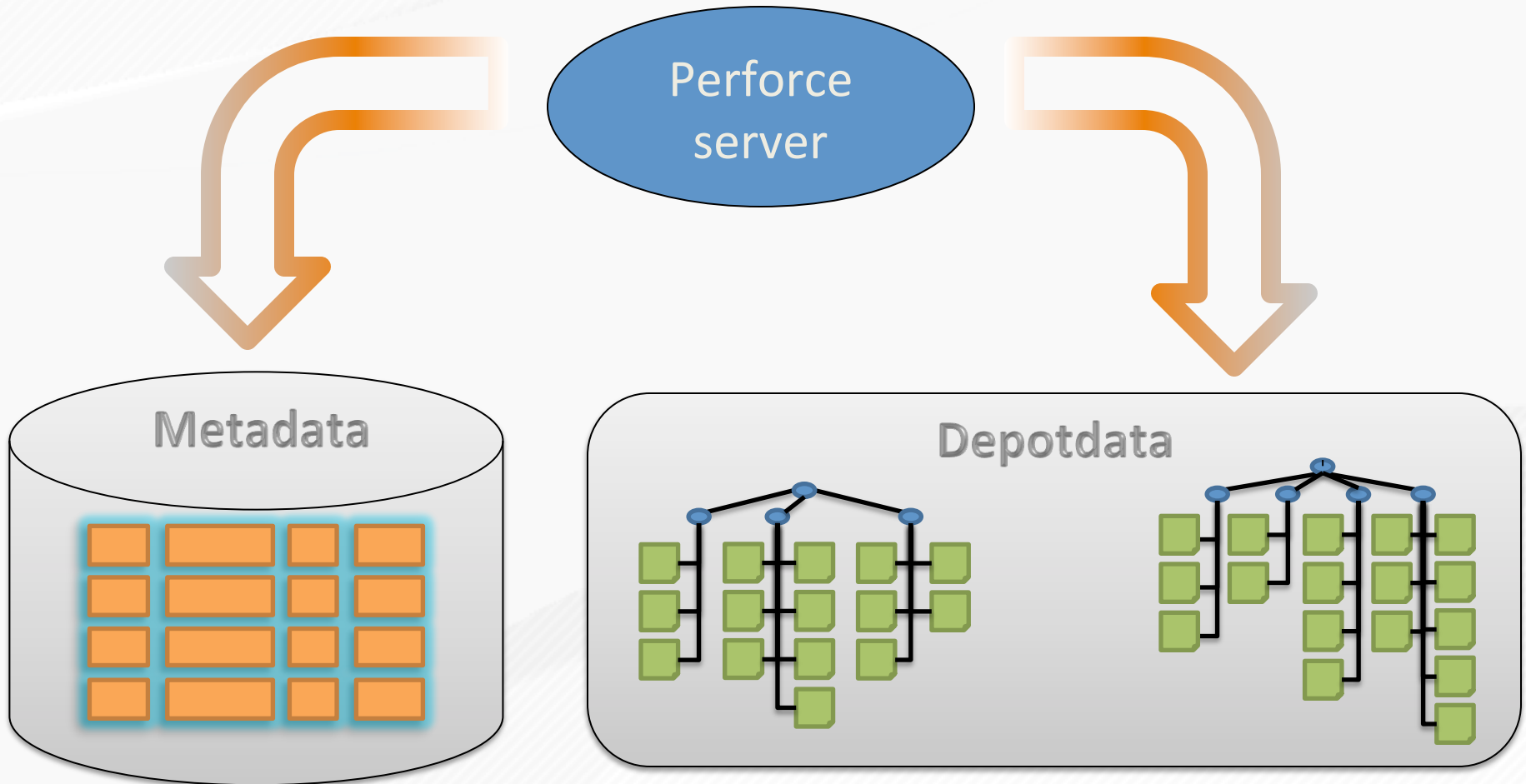
Overview

- Client/Server Model
- Server
- Client Programs
- Changelists
- Client...

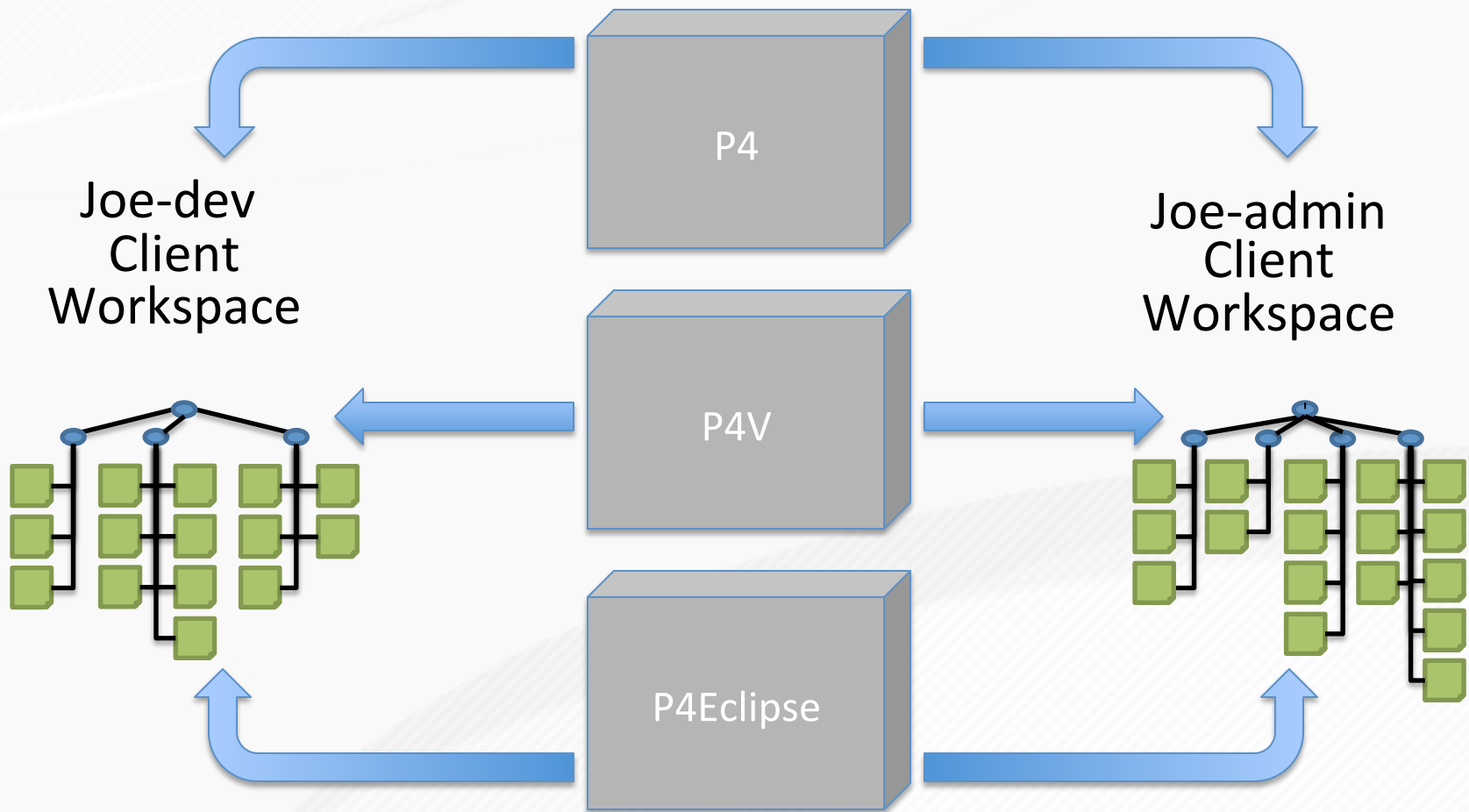
Perforce Client-Server model



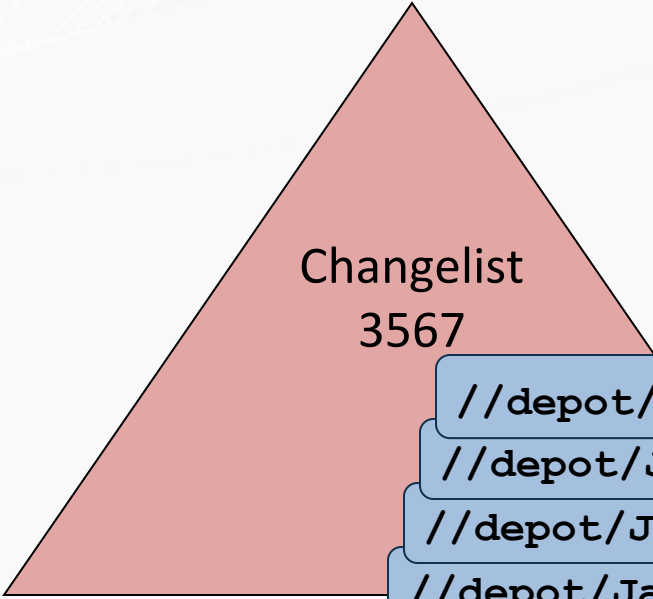
Perforce server



Perforce Client applications



Changelists



Changelist
3567

`//depot/Jam/MAIN/src/ape.c#23 edit`

`//depot/Jam/MAIN/src/zebra.c#9 edit`

`//depot/Jam/MAIN/src/lion.c#3 delete`

`//depot/Jam/MAIN/src/eland.c#14 edit`

`//depot/Jam/MAIN/src/bear.c#1 add`

Client ...

- Client machine
- Client programs
- Client workspaces

Showing connection information

p4 info

User name: bruno

Client name: bruno_ws

Client host: bruno-win.perforce.com

Client root: c:\work

Current directory: c:\work\Jam\MAIN\src

Client address: 103.14.25.211:1719

Server address: london:1666

Server root: /usr/perforce

Server date: 2012/03/26 11:21:36 +0100 BST

Server uptime: 382:14:58

Server version: P4D/DARWIN90U/2011.1/... (2012/01/25)

Server license: Perforce 200 users...

Case Handling: insensitive

Commands in this Chapter

- `p4 info`

Introduction to Perforce for Users

Help!

Help!

- Command Summary Help
- Help for a Specific Command
- Online Perforce Documentation
- E-mail Perforce
- Perforce Forums

Command summary help

p4 help

p4 is Perforce's client tool for the command line.
Try:

p4 help simple	list most common commands
p4 help commands	list all standard commands
p4 help command	help on a specific command
...etc...	

Perforce commands

p4 help commands

Perforce client commands:

add	Open a new file to add it to the depot
admin	Perform administrative operations on the server
annotate	Print file lines along with their revisions
archive	Archive obsolete revisions to archive depots
branch	Create or edit a branch specification
branches	Display list of branches
...etc...	

Help for a specific command

p4 help delete

`delete -- Open an existing file to delete it from the depot`

`p4 delete [-c changelist#] [-n -v] file ...`

Opens a depot file for deletion.

If the file is synced in the client workspace, it is removed. If a pending changelist number is specified using with the `-c` flag, the file is opened for delete in that changelist. Otherwise, it is opened in the default pending changelist.

Perforce Documentation

- [Introducing Perforce](#)
- [P4 User's Guide](#)
- [Perforce Command Reference](#)
- [Perforce System Administrator's Guide](#)
- [Release Notes](#)
- [Knowledge Base](#)
- [White Papers](#)

Help via E-mail

- E-mail Perforce at:
 - support@perforce.com
 - sales@perforce.com
 - consulting@perforce.com
- Perforce Forums
 - Subscribe at <http://forums.perforce.com/>
 - Exchange ideas with other Perforce users

Commands in this Chapter

- `p4 help`

Introduction to Perforce for Users

Basic File Operations

Basic Operations

- Populating a Client Workspace
- Editing, Adding and Deleting Files
- Moving Files
- Reverting Files
- Assessing Current Status
- Submitting Changes

Populating a client workspace

p4 sync

```
//depot/Jam/MAIN/src/README#26 - added as c:\work
  \Jam\MAIN\src\README
//depot/Jam/MAIN/src/regexp.c#2 - added as c:\work
  \Jam\MAIN\src\regexp.c
//depot/Jam/MAIN/src/regexp.h#1 - added as c:\work
  \Jam\MAIN\src\regexp.h
//depot/Jam/MAIN/src/RELNOTES#77 - added as c:\work
  \Jam\MAIN\src\RELNOTES
//depot/Jam/MAIN/src/rules.c#5 - added as c:\work
  \Jam\MAIN\src\rules.c
...etc...
```

Adding new files

```
notepad alias.c
```

```
notepad check.c
```

```
p4 add alias.c check.c
```

```
//depot/Jam/MAIN/src/alias.c#1 - opened for add
```

```
//depot/Jam/MAIN/src/check.c#1 - opened for add
```

Editing existing files

```
cd work\Jam\MAIN\src
```

```
p4 edit rules.c rules.h
```

```
//depot/Jam/MAIN/src/rules.c#5 - opened for edit
```

```
//depot/Jam/MAIN/src/rules.h#2 - opened for edit
```

```
notepad rules.c
```

```
notepad rules.h
```

Deleting files

```
p4 delete exec.c //depot/Jam/MAIN/dir.c  
//depot/Jam/MAIN/src/exec.c#2 - opened for delete  
//depot/Jam/MAIN/dir.c#3 - opened for delete
```

Moving files

```
p4 edit filemac.c
```

```
//depot/Jam/MAIN/src/filemac.c#3 - opened for edit
```

```
p4 move filemac.c fileosx.c
```

```
//depot/Jam/MAIN/src/fileosx.c#1 - moved from  
//depot/Jam/MAIN/src/filemac.c#3
```

Undoing your changes

```
p4 revert alias.c fileosx.c  
rules.h //depot/Jam/MAIN/dir.c  
//depot/Jam/MAIN/images/logo.pdf
```

```
//depot/Jam/MAIN/dir.c#3 - was delete, reverted  
//depot/Jam/MAIN/images/logo.pdf#none - was add, abandoned  
//depot/Jam/MAIN/src/alias.c#none - was add, abandoned  
//depot/Jam/MAIN/src/rules.h#2 - was edit, reverted  
//depot/Jam/MAIN/src/filemac.c#3 - was move/delete, reverted  
//depot/Jam/MAIN/src/fileosx.c#none - was move/add, deleted
```


What files were opened?

p4 opened

```
//depot/Jam/MAIN/src/check.c#1 -  
    add default change (text)  
  
//depot/Jam/MAIN/src/exec.c#2 -  
    delete default change (text)
```

p4 opened -u bob

```
//depot/Jam/MAIN/src/make.c#5 -  
    edit default change (text) by bob@bojam  
  
//depot/Jam/MAIN/src/rcp.c#7 -  
    edit default change (text) by bob@bojam  
  
//depot/Jam/MAIN/src/trans.h#9 -  
    edit default change (text) by bob@jelly
```

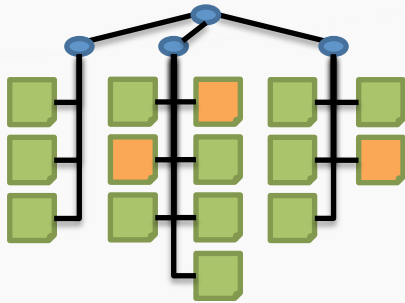

What changes have you made?

p4 diff rules.c

```
==== //depot/Jam/MAIN/src/rules.c#5 -  
c:\work\Jam\MAIN\src\rules.c ====  
76c76  
<         TARGET target, t = &target;  
---  
>         TARGET target, *t = &target;
```

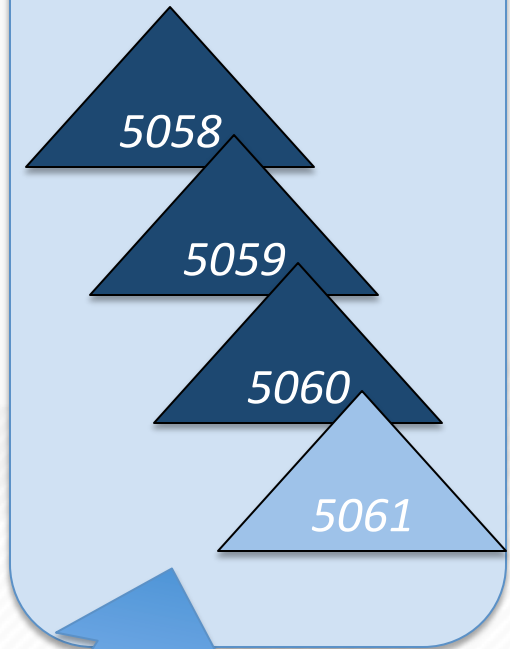
Submitting a changelist

Client Workspace



rules.c#5
check.c#1
exec.c#2

Depot



Submitting a changelist

p4 submit

Change: new

Description:

 Added new check function

Files:

```
//depot/Jam/MAIN/src/check.c    # add
//depot/Jam/MAIN/src/exec.c      # delete
//depot/Jam/MAIN/src/filemac.c  # move/delete
//depot/Jam/MAIN/src/fileosx.c  # move/add
//depot/Jam/MAIN/src/parse.c    # edit
//depot/Jam/MAIN/src/rules.c    # edit
```

Previewing workspace update

p4 sync -n

```
//depot/Jam/MAIN/src/README#27 - updating
```

```
c:\work\Jam\MAIN\src\README
```

```
//depot/Jam/MAIN/src/regexp.c#3 - updating
```

```
c:\work\Jam\MAIN\src\regexp.c
```

```
//depot/Jam/MAIN/src/regexp.h#2 - updating
```

```
c:\work\Jam\MAIN\src\regexp.h
```

```
//depot/Jam/MAIN/src/RELNOTES#79 - updating
```

```
c:\work\Jam\MAIN\src\RELNOTES
```

```
//depot/Jam/MAIN/src/varexp.c#8 - added as
```

```
c:\work\Jam\MAIN\src\varexp.c
```

...etc...

New Commands in this Chapter

- `p4 sync`
- `p4 edit`
- `p4 add`
- `p4 delete`
- `p4 move`
- `p4 revert`
- `p4 opened`
- `p4 diff`
- `p4 submit`

Introduction to Perforce for Users

Advanced File Operations

Advanced Operations

- Supported Wildcards
- Referencing files
- File types
- Reconcile Offline Work

Supported Wildcards

- Two types

```
p4 sync ...
```

```
p4 sync //depot/Jam/MAIN/...
```

```
p4 sync //depot/Jam/R1.0/....txt
```

```
p4 sync //depot/Jam/MAIN/doc/J*
```

```
p4 sync //depot/Jam/MAIN/src/*
```

- Local vs. depot expansion

```
p4 sync *.c
```

```
p4 sync "* .c"
```

Referencing files

- Depot Syntax

```
p4 sync //depot/Jam/MAIN/src/util.c
```

- Local syntax

```
p4 sync util.c
```

```
p4 sync c:\work\Jam\MAIN\src\util.c
```

```
p4 sync ..\src\util.c
```

File types

- Base file types

`text`

`binary`

`symlink`

`unicode`

`utf16`

- Workspace modifiers

`+x` - executable

`+w` - always writable

`+l` - exclusive open

`+k` - RCS keyword

`+m` - sync vs. submit modtime

- Server storage attributes

`+S` - latest revision

`+D` - deltas

`+C` - compressed

`+F` - full file

Setting file type

- Use the `'-t'` flag:

```
p4 add -t binary *.pdf
```

```
p4 edit -t text genfiles
```

- For already opened files, use `p4 reopen`

```
p4 reopen -t binary+lm doit
```

```
p4 reopen -t +w ....txt
```

List files altered offline

p4 diff -se

- Contents changed since last sync

p4 diff -sd

- Files removed since last sync

Reconcile offline work

```
cd c:\sean-jam\Jam\MAIN\src
```

```
dir /s/b/a-d      | p4 -x - add
```

```
p4 diff -sd ... | p4 -x - delete
```

```
p4 diff -se ... | p4 -x - edit -t auto
```

Reconcile offline work - EASIER

```
cd c:\sean-jam\Jam\MAIN\src
```

```
p4 status
```

```
p4 reconcile ...
```

- export P4IGNORE=.p4ignore

New Commands in this Chapter

- `p4 add -t`
- `p4 edit -t`
- `p4 reopen -t`
- `p4 reconcile`
- `p4 status`

Introduction to Perforce for Users

File Reporting

File Reporting

- Depot File Information
- Directory information
- Changelist Information
- What's in your workspace
- Search for patterns in files
- Show file contents with revisions

Depot File Information

- List files in a depot

```
p4 files //depot/Jam/MAIN/...  
//depot/Jam/MAIN/utils.c#3 - edit change 5121  
    (text)  
...etc...
```

- List revision history of a file

```
p4 filelog utils.c  
//depot/Jam/MAIN/utils.c  
... #3 change 5121 edit ... 'add reuse feature'  
... #2 change 5119 edit ... 'fix startup bug'  
... #1 change 4967 add ... 'add Gizmo src files'
```

Directory Information

- List all directories under a given path

```
p4 dirs //depot/Jam/*
```

```
//depot/Jam/MAIN
```

```
//depot/Jam/REL2.1
```

```
//depot/Jam/REL2.2
```

- Artifact of file path
- Directories are not versioned

Changelist information

- List all numbered changelists

p4 changes

Change 5120 on 2011/06/15 by bruno@ws ...

Change 5119 on 2011/06/15 by bruno@dyn_ws ...

...etc...

- Restrict to a path or file

p4 changes //pb/...

- Restrict to a user

p4 changes -u bruno

What's in your workspace?

p4 have

```
//depot/Jam/MAIN/src/Build.com#7 -  
    c:\work\Jam\MAIN\src\Build.com  
//depot/Jam/MAIN/src/Build.mpw#1 -  
    c:\work\Jam\MAIN\src\Build.mpw  
//depot/Jam/MAIN/src/command.c#8 -  
    c:\work\Jam\MAIN\src\command.c  
...etc...
```

Search for patterns in files

```
p4 grep -e "rules.h" //depot/Jam/...  
//depot/Jam/MAIN/src/command.c#8:  
    # include "rules.h"  
//depot/Jam/MAIN/src/compile.c#25:  
    # include "rules.h"  
//depot/Jam/MAIN/src/compile.c#25:  
    * directly to the rule (as defined in rules.h).  
//depot/Jam/MAIN/src/headers.c#5:  
    # include "rules.h"  
...etc...
```


Show file contents with revisions

p4 annotate jam.c

```
//depot/Jam/MAIN/src/jam.c#35 - edit change 627 (text)
1:      /*
1:      *
16:     *      Copyright 1993, 1996 Chris.
...etc...
```

p4 annotate -a -c jam.c

```
//depot/Jam/MAIN/src/jam.c#35 - edit change 627 (text)
1-627:   /*
1-627:   *
1-1:     *      Copyright 1993 Chris.
30-248:  *      Copyright 1993, 1995 Chris.
...etc...
```

- P4V's Time-Lapse View gives a graphical view

New Commands in this Chapter

- `p4 files`
- `p4 filelog`
- `p4 changes`
- `p4 have`
- `p4 grep -e`
- `p4 annotate`

Introduction to Perforce for Users

Revision Specifiers

Revision Specifiers

- Revision specifiers
- Revision ranges
- Comparing different revisions
- Backing out a changelist

What are Revision Specifiers?

- Identify a file or group of files

Changelist	@5123	@5179	@5234	@5259	@5310
file1	#1	#2		#3	#4
file2		#1	#2		#3
file3	#1			#2	

Revision Specifiers

Type

Revision number

Changelist number

Workspace name

Label name

Date

Latest revision

Have revision

Non-existent revision

Syntax

#7

@30

@sams_ws

@rlos_label

@2012/03/01

#head

#have

#none or #0

Why use Revision Specifiers?

- Restrict the scope of a command; for example:

- Sync to a particular revision

```
p4 sync //depot/Jam/MAIN/rules.h#12
```

- List files updated up to a known changelist

```
p4 files //depot/Jam/...@5234
```

- List changes submitted up to a known date

```
p4 changes //depot/Jam/MAIN/...@2012/04/21
```

Revision Specifier Examples

```
p4 sync utils.c#2
p4 sync utils.c@431
p4 diff utils.c@431
p4 sync @431
p4 changes ...@431
p4 changes //depot/Jamgraph/DEV/gizmo/*.c@431
p4 sync @sams_ws
p4 sync @janslabel
p4 sync util.c#head
p4 files //depot/Acme-api/...#have
p4 sync //depot/notes/....txt
p4 sync #none
p4 sync util.c@2001/05/29
p4 files //depot/Jam/...@2001/05/29:15:37:00
```


Revision ranges

- Specify two revisions within the command:

```
p4 changes abc.c@20,@32
```

```
p4 files //depot/Jam/MAIN/...@431,@431
```

- Only files/revisions within range apply
- Some commands accept revision ranges

```
p4 changes
```

```
p4 integ
```

```
p4 sync
```

```
p4 jobs
```

```
p4 files
```

```
p4 print
```

```
p4 fixes
```

Comparing different revisions

p4 diff2 hash.c#1 hash.c#2

```
==== //depot/Jam/MAIN/src/hash.c#1 -  
      //depot/Jam/MAIN/src/hash.c#2 ====
```

```
2c2,4
```

```
<  * Copyright 1993 Christopher Seiwald.
```

```
---
```

```
>  * Copyright 1993, 1995 Christopher Seiwald.
```

```
>  *
```

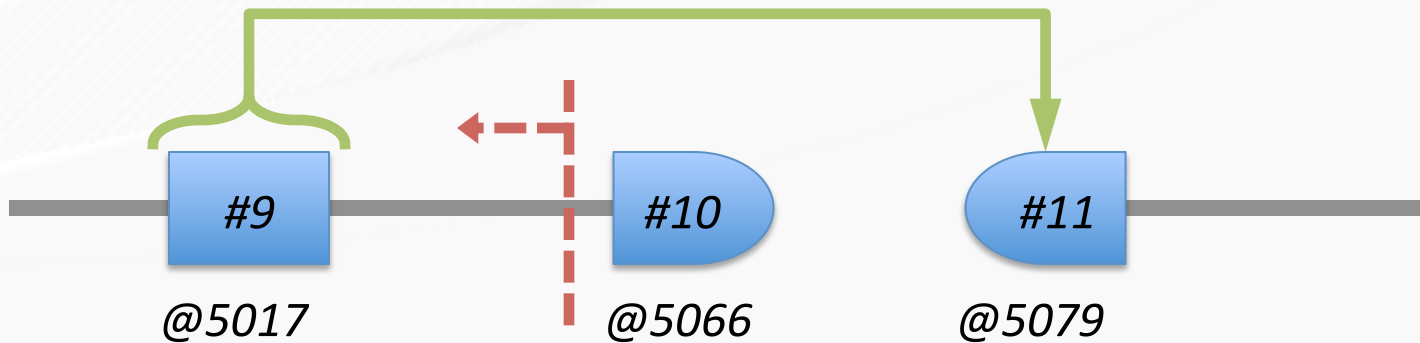
```
>  * This file is part of Jam - see jam.c for  
    Copyright information.
```

What Files Have Changed?

```
p4 diff2 -q //depot/...@231  
//depot/...@250
```

```
==== //depot/Jam/MAIN/src/jam.c#13 (text) -  
    //depot/Jam/MAIN/src/jam.c#15 (text) ==== content  
==== //depot/Jam/MAIN/src/jam.h#17 (text) -  
    //depot/Jam/MAIN/src/jam.h#18 (text) ==== content  
==== <none> - //depot/Jam/MAIN/src/Jambase.ps#1 ====  
==== <none> - //depot/Jam/MAIN/src/Jamfile.ps#1 =====
```

Recovering a Deleted File



- Sync the revision to be re-added:

p4 sync parse.h@5065

or

p4 sync parse.h#9

- Open the file for add

p4 add parse.h

- Submit

New Commands in this Chapter

- `p4 diff2`

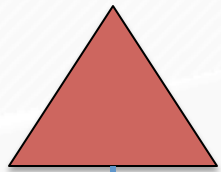
Introduction to Perforce for Users

Changelist Management

Changelist Management

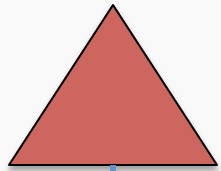
- Managing pending changelists
- Shelving opened files
- Showing changelist detail

Multiple Pending Changelists



Default

- //depot/Jam/MAIN/src/rules.c#1 <add>
- //depot/Jam/MAIN/src/utils.c#5 <edit>
- //depot/Jam/MAIN/src/utils.h#9 <edit>



5069

- //depot/Jam/MAIN/src/compile.c#23 <edit>
- //depot/Jam/MAIN/src/compile.h#17 <delete>
- ...

Creating a New Changelist

p4 change

```
Change: new
Client: bob-jam
User:    bob
Status: pending
Description:
    Fix off-by-one error.
Files:
    //depot/Jam/MAIN/src/compile.c #edit

Change 5069 created with 1 open file(s).
```

Editing a Pending Changelist

p4 change 5069

Change: 5069

Client: bob-jam

User: bob

Status: pending

Description:

Fix off-by-one error. Use UTC time.

Files:

//depot/Jam/MAIN/src/compile.c #edit

Using Numbered Changelists

- Open files into specific changelist

```
p4 edit -c 5061 //cl/Jam/MAIN/src/make.c
```

```
p4 add -c 5062 hash.h hash.c
```

```
p4 delete -c 5062 //depot/Jam/MAIN/src/fileent.c
```

- Move open files between changelists

```
p4 reopen -c 5061 *.sh
```

```
p4 reopen -c default //...
```

Submitting a Numbered Changelist

```
p4 submit -c 5069
```

```
Submitting change 5069.
```

```
Locking 3 files ...
```

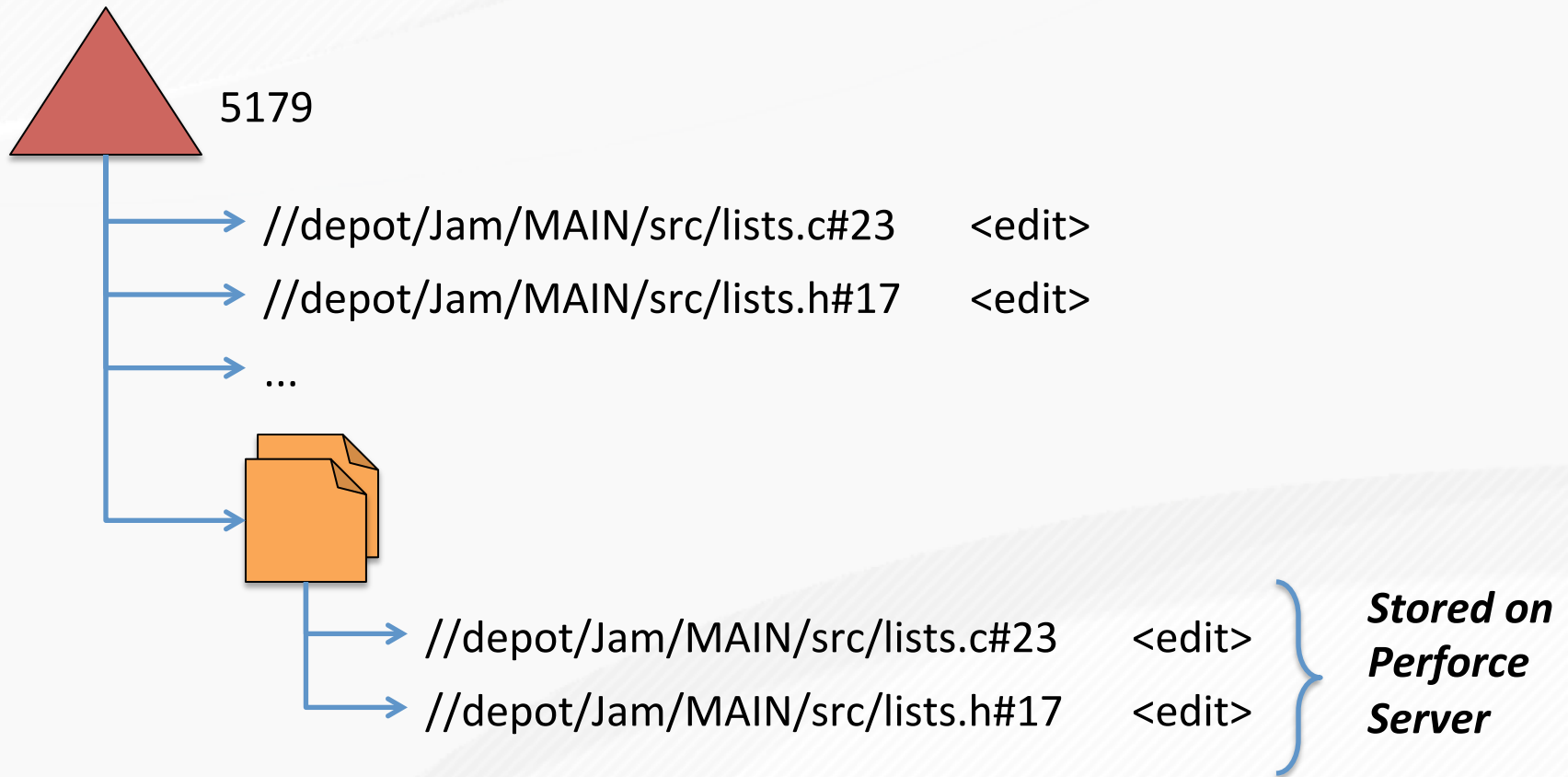
```
add //depot/Jam/MAIN/src/newstr.c#1
```

```
delete //depot/Jam/MAIN/src/execmac.c#3
```

```
edit //depot/Jam/MAIN/src/rules.c#7
```

```
Change 5069 renamed 5072 and submitted.
```

Shelved Changes



Shelving Opened Files

p4 shelve

Change: new

Client: bob-jam

User: bob

Status: new

Description:

Shelving current edits

Files:

//depot/Jam/MAIN/src/lists.c # edit

//depot/Jam/MAIN/src/lists.h # edit

Accessing Shelved Files

```
p4 unshelve -s 5179
```

```
//depot/Jam/MAIN/src/lists.c#6 - unshelved,  
opened for edit
```

```
//depot/Jam/MAIN/src/lists.h#4 - unshelved,  
opened for edit
```

```
p4 diff lists.c@=5179
```

```
==== //depot/Jam/MAIN/src/lists.c#45 -  
c:\earl-os2\Jam\MAIN\src\lists.c ====
```

```
37,39d36
```

```
< # ifdef NTFS
```

```
< # include "daylight.h"
```

```
< # endif
```


Deleting Shelved Files

p4 changes -s shelved

```
Change 5179 on 2011/11/06 by bob@bob-jam  
*pending* 'Shelving current edits. '
```

p4 shelve -d -c 5179

```
Shelve 5179 deleted.
```

Showing Changelist Detail

```
p4 describe -s 5185
```

```
Change 5185 by earl@earl-dev on 2010/07/02 10:15:38
```

```
Windows NT changes.
```

```
Affected files ...
```

```
... //depot/Jam/MAIN/src/Jambase#52 edit
```

```
... //depot/Jam/MAIN/src/Jamfile#18 edit
```

```
... //depot/Jam/MAIN/src/RELNOTES#15 edit
```

New Commands in this Chapter

- `p4 change`
- `p4 shelve`
- `p4 unshelve`
- `p4 describe`

Introduction to Perforce for Users

Handling File Conflicts

Handling File Conflicts

- When can conflicts occur?
- Automatically resolve
- Interactively resolve
- Reporting Commands
- Backing out a Changelist

When can conflicts occur?

- File conflicts
 - Submitting a change to an out-of-date file
 - Re-syncing an opened file that is out-of-date
 - Moving to an existing file
 - Unshelving onto an opened file
- *Integration conflicts*

When can conflicts occur?

- Submitting changes to an out-of-date file

p4 submit

Change 5085 created with 4 open file(s).

Submitting change 5085.

```
//depot/Jam/MAIN/src/rules.c -  
    must resolve before submitting
```

```
//depot/Jam/MAIN/src/rules.c -  
    must resolve #6,#7
```

Out of date files must be resolved or
reverted.

```
Submit failed -- fix problems above then use 'p4  
submit -c 5085'
```


When can conflicts occur?

- Re-syncing opened files

```
p4 sync -q ...
```

```
... //depot/Jam/MAIN/src/rules.c - must resolve  
#2,#5 before submitting
```

```
... //depot/Jam/MAIN/src/rules.h - must resolve  
#2,#14 before submitting
```

When can conflicts occur?

- Moving to an existing file

```
p4 move -f //depot/Jam/REL2.1/...  
//depot/Jam/MAIN/...
```

```
//depot/Jam/MAIN/src/regexp.c#2 - moved from  
//depot/Jam/REL2.1/src/regexp.c#1  
//depot/Jam/MAIN/src/regexp.h#1 - moved from  
//depot/Jam/REL2.1/src/regexp.h#1
```

When can conflicts occur?

- Unshelving onto opened file

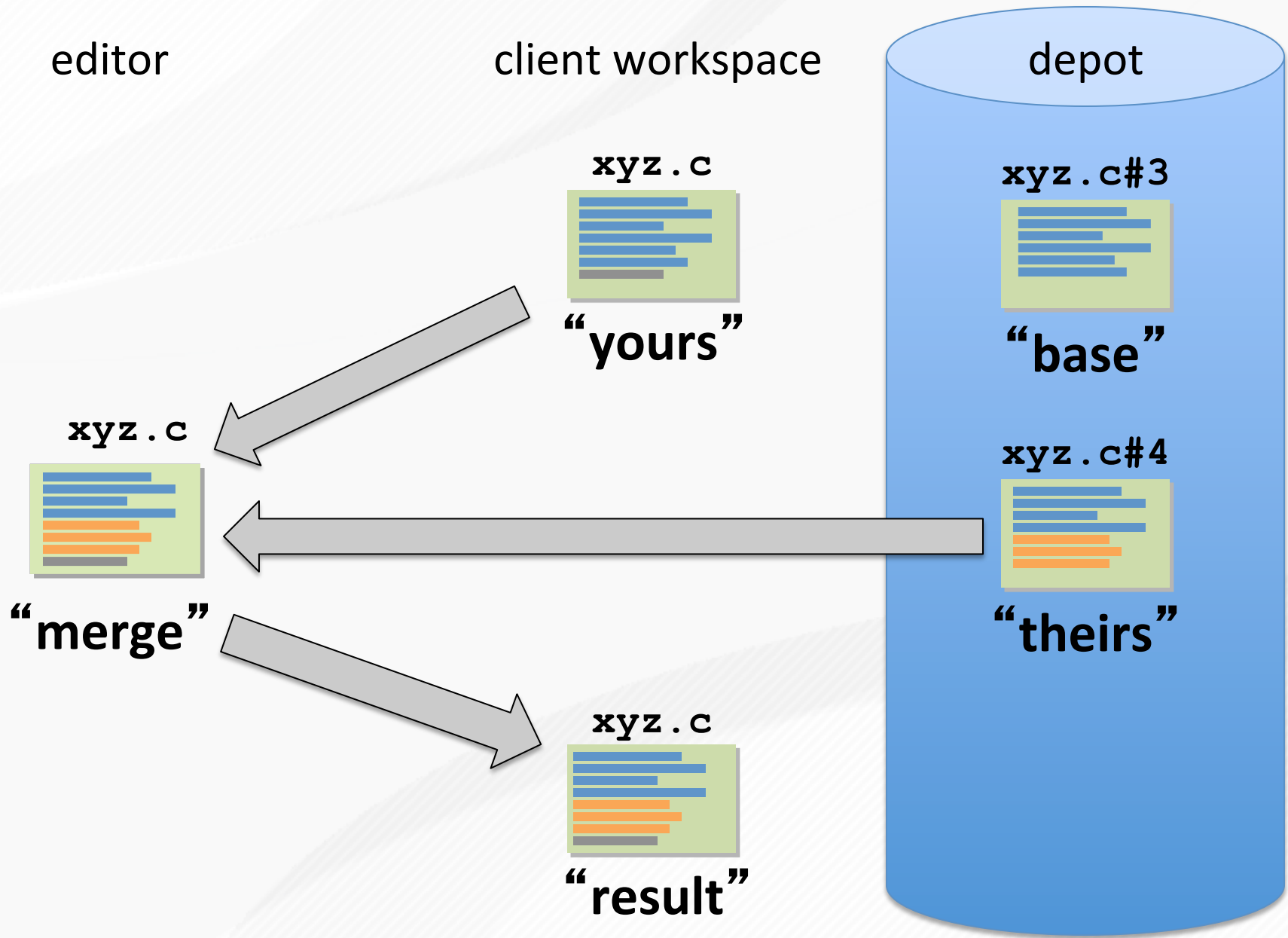
```
p4 unshelve -s 12023
```

```
//depot/Jam/MAIN/src/rules.c#5 - unshelved,  
opened for edit
```

```
... //depot/Jam/MAIN/src/rules.c - must  
resolve //depot/Jam/MAIN/src/rules.c@=12023  
before submitting
```

What happens?

- Depot file remains unchanged
- A resolve is scheduled



Automatic resolve options

- Ignore (accept your workspace file)
`p4 resolve -ay`
- Copy (accept their depot file)
`p4 resolve -at`
- Safe (accept yours or theirs, if only one differs)
`p4 resolve -as`
- Merge (accept yours, theirs or merge, if no conflicts)
`p4 resolve -am`

Interactive resolve

p4 resolve

```
c:\work\Jam\MAIN\src\rules.c
```

```
- merging //depot/Jam/MAIN/src/rules.c#6,#7
```

```
Diff chunks:
```

```
  0 yours + 1 theirs + 0 both + 1 conflicting
```

```
Accept(a) Edit(e) Diff(d) Merge (m)
```

```
Skip(s) Help(?) e:
```

- Iterates over each file scheduled for resolve.

Diff chunk types

yours

- Unique diff chunk in client file only

theirs

- Unique diff chunk in depot file only

both

- Unique diff chunk common to depot and client file

conflicting

- Unique diff chunk conflicting between depot, client and base.

Interactive resolve actions

- Accept yours (**ay**)
 - Ignore 'theirs', leave workspace file as-is
- Accept theirs (**at**)
 - Copy 'theirs' to workspace
- Accept merged (**am**)
 - Replace workspace file with 'merge' file
- Accept edited (**ae**)
 - Replace workspace file with edited 'merge' file

Before resolving, you can...

- **Edit(e)**
 - merged file
 - your file
- **Skip(s)**
- **Help(?)**
- **Diff(d)**
 - merged vs. yours
 - yours vs. base (dy)
 - theirs vs. base (dt)
 - merged vs. base (dm)

Editing the “merge” file

theirs

```
# include "jam.h"
# include "option.h"
# include "patchlev.h"
# include "make.h"
```

yours

```
# include "jam.h"
# include "option.h"
# include "patchlevel.h"
# include "make.h"
```

merged

```
# include "jam.h"
# include "option.h"
>>>> ORIGINAL //depot/Jam/MAIN/xyz.c#3
==== THEIRS //depot/Jam/MAIN/xyz.c#4
# include "patchlev.h"
==== YOURS //bruno_ws/Jam/MAIN/xyz.c
# include "patchlevel.h"
<<<<
# include "make.h"
```

result (after editing)

```
# include "jam.h"
# include "option.h"
# ifdef FATFS
# include "patchlev.h"
# else
# include "patchlevel.h"
# endif
# include "make.h"
```

Textual vs. binary resolves

- Three-way resolves
 - Operate on text files
 - Use Yours, Theirs, and Base
 - Files can be merged
- Two-way resolves
 - Operate (by default) on binary files
 - No binary diffs stored, so files can't be merged
 - Resolve choices: accept yours or theirs

Resolve reporting commands

- List pending resolves

```
p4 resolve -n
```

```
c:\work\Jam\MAIN\src\rules.c -  
    merging //depot/Jam/MAIN/src/rules.c#6,#7
```

- List resolved files

```
p4 resolved
```

```
c:\work\Jam\MAIN\src\hash.c -  
    merged from //depot/Jam/MAIN/src/hash.c#4,#6
```

- Re-do a resolve

```
p4 resolve -f
```

Locking a file

p4 lock rules.c

//depot/Jam/MAIN/src/rules.c - locking

- Why?
 - Prevent others submitting
 - Avoid multiple resolves
- How is a file unlocked?
 - Revert
 - Submit
 - **p4 unlock <file>**

Backing out a changelist

```
p4 sync @6024
```

```
p4 edit index.c command.h
```

```
p4 sync
```

```
p4 resolve -ay
```

```
p4 submit
```

```
Change 6026 created with 2 open file(s).
```

```
Locking 2 files ...
```

```
Submitting change 6026.
```

```
edit //depot/Jam/MAIN/src/index.c#11
```

```
edit //depot/Jam/MAIN/src/command.h#6
```

```
Change 6026 submitted.
```

New Commands in this Chapter

- `p4 resolve`
- `p4 resolved`
- `p4 lock`

Introduction to Perforce for Users

Workspace Management

Workspace Management

- What is a workspace?
- Why have multiple workspaces?
- What does a workspace need?
- Creating a Client Workspace
- Client Workspace View Mappings

What is a workspace?

- View into a Perforce Server
- Local files in the workspace are managed by Perforce
- Uniquely identified by its name (**P4CLIENT**)

Why have multiple workspaces?

- Project/Branch oriented
- Better performance
 - For both you and your colleagues!

What does a workspace need?

- Name
 - Uniquely identifies the workspace
- Root
 - The workspace's root directory on your client machine
- View
 - Maps areas of a Perforce Server to client workspace

Creating a client workspace

p4 client

```
Client:      bob-dev
Update:       2009/11/15 12:39:47
Access:       2010/06/03 16:33:25
Owner:        bob
Host:         perseus
Description:
              Bob's main development workspace.
Root:       c:\p4work
Options:      noallwrite noclobber nocompress unlocked
              nomodtime normdir
SubmitOptions: submitunchanged
LineEnd:      local
View:
              //depot/Jam/MAIN/... //bob-dev/Jam/MAIN/...
```

Client workspace view

Client: bob-jam

Root: c:\p4work

View:

//depot/Jam/MAIN/src/...	//bob-jam/Jam/MAIN/src/...
-//depot/Jam/MAIN/src/tests/...	//bob-jam/Jam/MAIN/src/tests/...
//depot/Jam/MAIN/src/*.h	//bob-jam/Jam/MAIN/src/hfiles/*.h
//depot/Jam/MAIN/cfg/*	//bob-jam/Jam/MAIN/config/*
//depot/Acme-api/....cpp	//bob-jam/Acme-api/....CC
//depot/Jam/REL1.0/src/...	//bob-jam/Jam/REL1.0/src/...
+//depot/Jam/PATCH1.0/src/...	//bob-jam/Jam/REL1.0/src/...

Client workspace options

- Options
 - `[no]allwrite` `[no]compress`
 - `[no]clobber` `[un]locked`
 - `[no]modtime` `[no]rmdir`
- SubmitOptions
 - `submitunchanged[+reopen]`
 - `revertunchanged[+reopen]`
 - `leaveunchanged[+reopen]`
- LineEnd
 - `local` `unix` `mac` `win` `share`

Creating workspaces

- Creating a new workspaces

```
p4 client bob-jam-project_a
```

- Copy a workspace's view and options

```
p4 client -t bob-jam-dev mary-jam-dev
```

- List all workspaces for a user

```
p4 clients -u bob
```

Testing workspace view

- Show depot and client workspace paths

p4 where command.c

//depot/Jam/MAIN/src/command.c

//mel-dev/Jam/MAIN/src/command.c

d:\mel-dev\Jam\MAIN\src\command.c

- File doesn't have to exist for **p4 where** to work

New Commands in this Chapter

- `p4 client`
- `p4 clients`
- `p4 where`

Introduction to Perforce for Users

Intermission!

Introduction to Perforce for Users

Workspace Management
(continued)

Workspace Management

- Configuring Perforce Settings
- Setting up your client machine
- Checking your client configuration

Environment Variables

- Important for client applications:
 - P4USER** – Perforce User name
 - P4CLIENT** – Perforce workspace name
 - P4PORT** – Perforce Server hostname and port
 - P4CONFIG** – Client config filename

Setting in the command

- Command syntax

p4 *(usage flags)* **command** [**flags**] **args**

- **-p** <port> **-u** <user> **-c** <workspace>

p4 **-p** mars:1666 **-u** amir **-c** amir-api **client**

p4 **-c** rlo-jam-main **sync** **-n**

Environment/Registry setup

- Unix (probably in .profile/.bashrc/...)
`export P4PORT=neptune:1666`
`export P4CLIENT=lisas-jam-dev`
`export P4USER=lisa`
- Windows/Mac
`p4 set P4PORT=neptune:1666`
`p4 set P4CLIENT=lisas-jam-dev`
`p4 set P4USER=lisa`

Config file setup

- Set **P4CONFIG** to a file name
p4 set P4CONFIG=config.txt
- Create config files in client workspace root
- List environment variables in config file
P4PORT=lemon:1909
P4CLIENT=graphics_proj

Order of precedence

- 1) Command-line flags
- 2) Variables defined in **P4CONFIG** file
- 3) Environment variable settings
- 4) Registry variable settings
- 5) Default environment variable values

Recommended client setup

- General settings in environment/registry
- Workspace specific settings in **P4CONFIG** files

Checking your settings

- Use **p4 set** to list your configuration:

```
p4 set
```

```
P4CLIENT=bruno_ws
```

```
P4CONFIG=.p4 (set) (config 'noconfig')
```

```
P4PORT=1666
```

```
P4USER=bruno
```

Setting a password

- Use **p4 passwd** to set password on server
- A ‘strong’ password...
 - ...is at least eight characters
 - ...includes two of:
 - Upper case characters
 - Lower case characters
 - Non-alphabetic characters

Logging in to Perforce

- Start a session

```
p4 login
```

```
Enter password:
```

```
User marta logged in.
```

- End a session

```
p4 logout
```

```
User marta logged out.
```

Other environment variables

- P4CHARSET, P4DIFF, P4MERGE
- P4EDITOR, P4TICKETS
- P4IGNORE
- TMP, TEMP

New Commands in this Chapter

- `p4 set`
- `p4 passwd`
- `p4 login`
- `p4 logout`

Introduction to Perforce for Users

Branching and Integration

Branching and integration

- Terminology
- Why branch?
- Integration steps – creating
- Integration steps – propagating

Terminology

- Branch
 - (noun) A set of related files, created as a copy of its parent
 - (verb) To create a branch
- Integrate
 - To create a new branch
 - To propagate changes between existing branches

Why branch?

- Conflicting policies
- Task oriented development
- Discussed in more detail in:

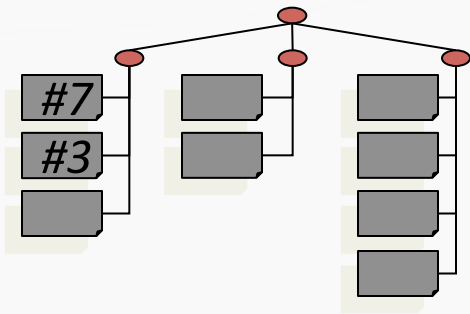
High-Level SCM Best Practices

http://www.perforce.com/customers/white_papers/highlevel_scm_best_practices

Creating a branch

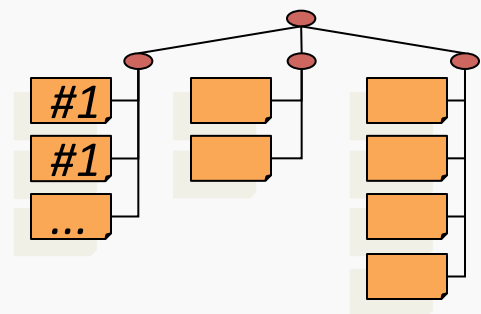
source

//depot/Jam/MAIN/...



target

//depot/Jam/R1.0/...



MAIN

R1.0

How to integrate

- Using **p4 integ**
- The source and target can be set in
 - A branch specification
 - stores source/target pairs
 - Create new branch
 - Re-use to propagate changes
 - A file specification
 - Source/target set on the command line
 - Once use
 - Useful for simple views.

Integration steps - creating

- Create a branch spec (if you're using one)
- Add target branch to workspace view
- Open files for integration
- Submit

Integration steps - creating

- Create a branch spec

p4 branch JamMAIN-to-R1.0

Branch: JamMAIN-to-R1.0

Owner: bob

Description:

Jam Release 1.0 branch for Server & API

Options: unlocked

View:

//depot/Jam/MAIN/svr/... //depot/Jam/R1.0/svr/...

//depot/Jam/MAIN/api/... //depot/Jam/R1.0/api/...



source paths



target paths

Integration steps - creating

- Add target branch to workspace view

p4 client

Client: ben-jam

Description:

Ben's Jam client workspace.

Root: c:\demo

View:

//depot/Jam/MAIN/... //ben-jam/Jam/MAIN/...

//depot/Jam/R1.0/... //ben-jam/Jam/R1.0/...

Integration steps - creating

- Open files for integration

- Branch spec

```
p4 integ -b JamMAIN-to-R1.0
```

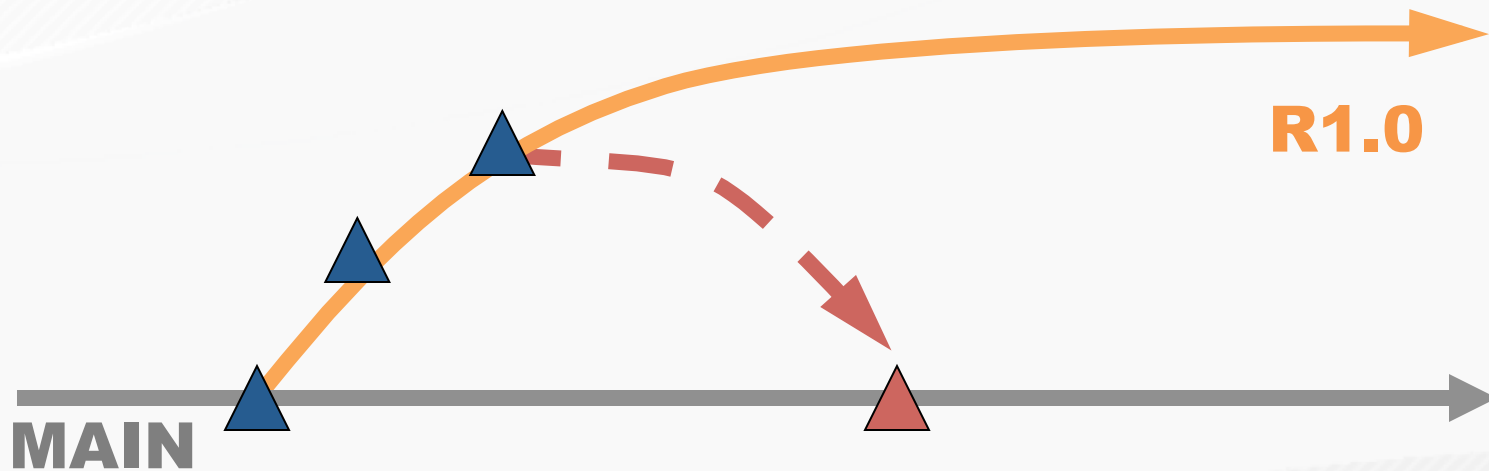
- File spec

```
p4 integ //depot/Jam/MAIN/...  
//depot/Jam/R1.0/...
```

- Submit

```
p4 submit
```

Propagating changes across branches



Integration steps – propagating

- Check client workspace view
- Open files for integration
- Resolve
- Submit

Integration steps – propagating

- Open files for integration

- Branch spec

```
p4 integ -r -b JamMAIN-to-R1.0
```

- File spec

```
p4 integ //depot/Jam/R1.0/... //depot/Jam/MAIN/...
```

- Perforce tracks integration history
- Only changed files are propagated

Integration steps – propagating

- Resolve

`p4 resolve`

- Submit

`p4 submit`

Branched revision history

p4 filelog -i

//depot/Jam/REL2.1/src/hash.c

```
//depot/Jam/REL2.1/src/hash.c
... #2 change 869 edit ...
... #1 change 749 branch ...
... ... branch from //depot/Jam/MAIN/src/hash.c#1,#3
//depot/Jam/MAIN/src/hash.c
... #3 change 137 edit ...
... ... branch into //depot/Jam/REL2.1/src/hash.c#1
... #2 change 30 edit ...
... ... branch into //depot/Jam/REL2.0/src/hash.c#1
... #1 change 1 add ...
```

- P4V's Revision Graph gives a graphical view

New Commands in this Chapter

- `p4 branch`
- `p4 integ`
- `p4 filelog -i`

Introduction to Perforce for Users

Branching and Integration
(continued)

Branching and Integration

- Different resolve types
- Integration reporting
- The Flow of Change

Different resolve types

- Content
- Moved files
- Filetype changes
- Branched files
- Deleted files

For all resolves...

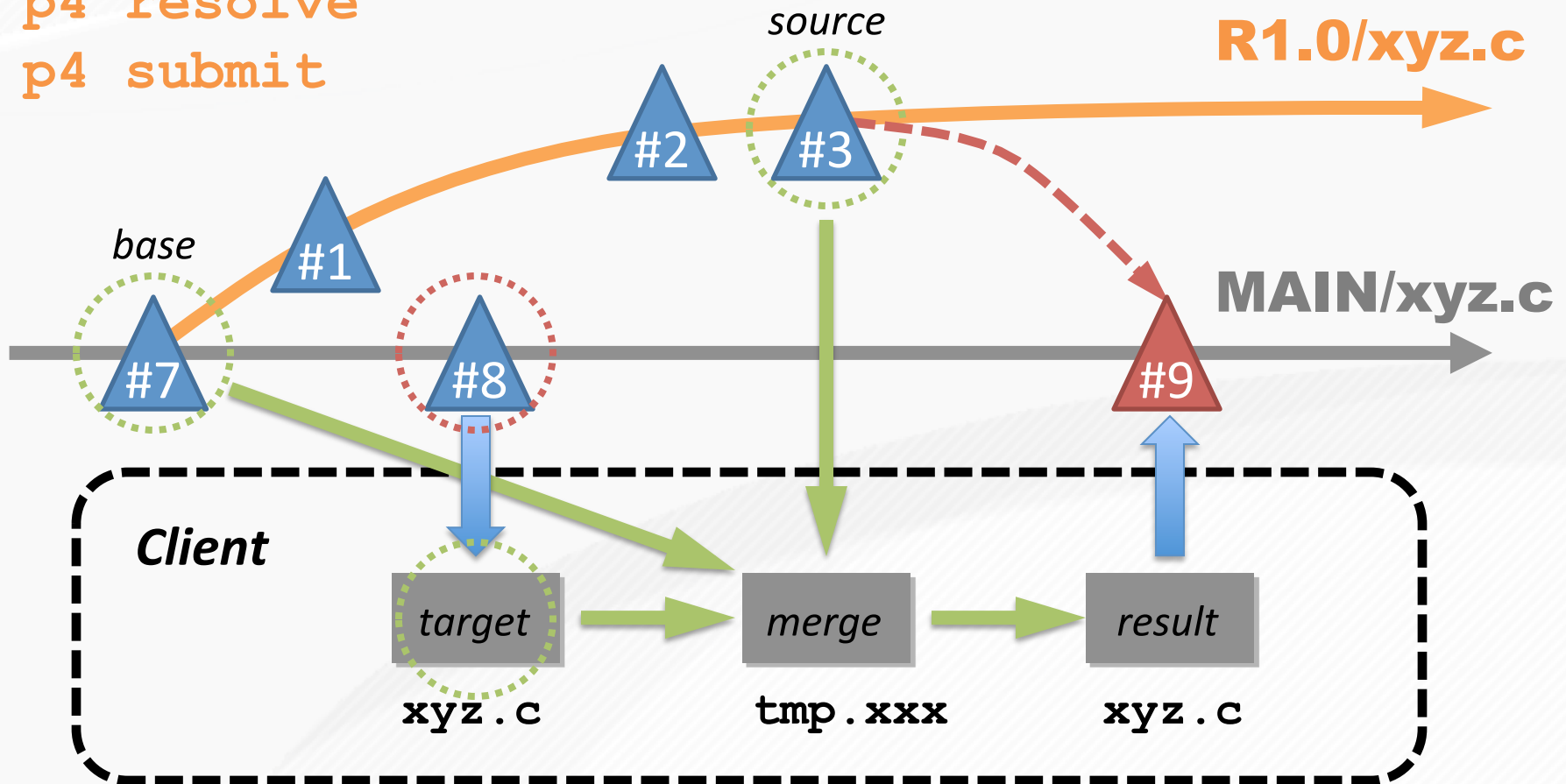
- *Theirs* = Source
- *Yours* = Target
- *Base* = Closest common ancestor

Resolves during integration

p4 integ -r -b JamMAIN-to-R1.0

p4 resolve

p4 submit



Moved files

- Source or target renamed using **p4 move**
 - Copy source file name (**at**)
 - Ignore source file name (**ay**)

p4 resolve

```
c:\work\Jam\DevA\src\fileapple.c -  
    resolving move to //Jam/DevA/src/fileosx.c
```

```
Filename resolve:
```

```
at: //Jam/DevA/src/fileosx.c
```

```
ay: //Jam/DevA/src/fileapple.c
```

```
Accept(a) Skip(s) Help(?) at:
```

Filetype changes

- Filetype changed in source, target or both
 - Copy source file type (**at**)
 - Ignore source file type (**ay**)
 - Merge file types (**am**)

p4 resolve

```
c:\work\Jam\DevA\src\search.c -
```

```
    resolving filetype from //Jam/MAIN/src/search.c#2
```

```
Filetype resolve:
```

```
at: (binary+m)
```

```
ay: (binary+l)
```

```
am: (binary+lm)
```

```
Accept(a) Skip(s) Help(?) am:
```

Branched Files

- Integrate branches new files, by default
- Schedule branch resolve with **p4 integ -Rb**
 - Branch file (**at**)
 - Ignore file (**ay**)

p4 resolve

```
c:\work\Jam\DevA\src\newfile.txt - resolving branch  
from //Jam/MAIN/src/newfile.txt#1
```

Branch resolve:

at: branch

ay: ignore

Accept(a) Skip(s) Help(?) at:

Deleted Files

- Replaces the need for **-d** and **-D[sti]** flags
- Schedule delete resolve with **p4 integ -Rd**
 - Delete file (**at**)
 - Ignore delete (**ay**)

p4 resolve

```
c:\work\Jam\DevA\src\search.c - resolving delete  
from //Jam/MAIN/src/search.c#3
```

```
Delete resolve:
```

```
at: delete
```

```
ay: ignore
```

```
Accept(a) Skip(s) Help(?) at:
```

What needs to be integrated?

p4 interchanges -r -b JamREL2.1toMAIN

Change 183 on 2001/02/06 by earl@earl-dev-guava

'Straightened out platform suppo'

Change 184 on 2001/02/20 by earl@earl-os2-buckeye

'Altered hash function fo'

Change 185 on 2001/02/20 by earl@earl-os2-buckeye

'Rework Jambase to confor'

Change 186 on 2001/02/23 by earl@tosh

'Have jam's makeDirName return .'

Change 187 on 2001/03/05 by earl@earl-dev-guava

'New LOCATE_SOURCE for jam. '

What has been integrated?

p4 integrated //depot/Jam/REL2.1/...
//depot/Jam/REL2.1/src/Build.com#1 - branch from
//depot/Jam/MAIN/src/Build.com#1,#5
//depot/Jam/REL2.1/src/Build.com#1 - branch into
//jam/rel2.1/src/Build.com#1
//depot/Jam/REL2.1/src/command.c#1 - branch from
//depot/Jam/MAIN/src/command.c#1,#5
//depot/Jam/REL2.1/src/command.c#1 - branch into
//jam/rel2.1/src/command.c#1
...etc...

What changes were integrated?

p4 changes -i //depot/Jam/R1.0/...

Change 5047 on 2008/12/29 by bob@bob-jam 'Create release branch'

Change 5028 on 2008/12/23 by baron@baron-bios 'Upgrade to latest metrowerks on'

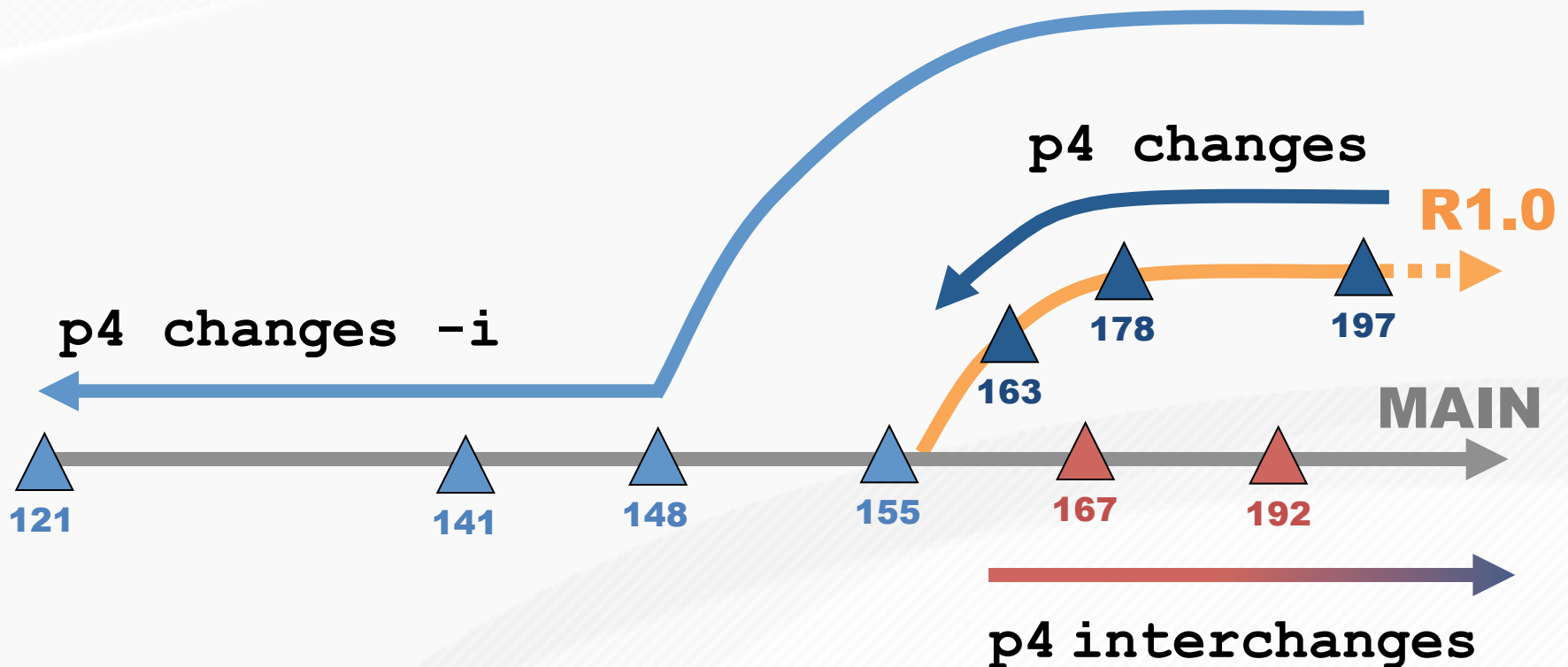
Change 5026 on 2008/12/22 by jean@jean-spice 'Added all the old press release'

Change 5025 on 2008/12/22 by jean@jean-spice 'Press releases now just use standard'

Change 5014 on 2008/12/19 by berger@berger-spice 'Put in fix for jam's NT handle '

...etc...

What changes are reported?

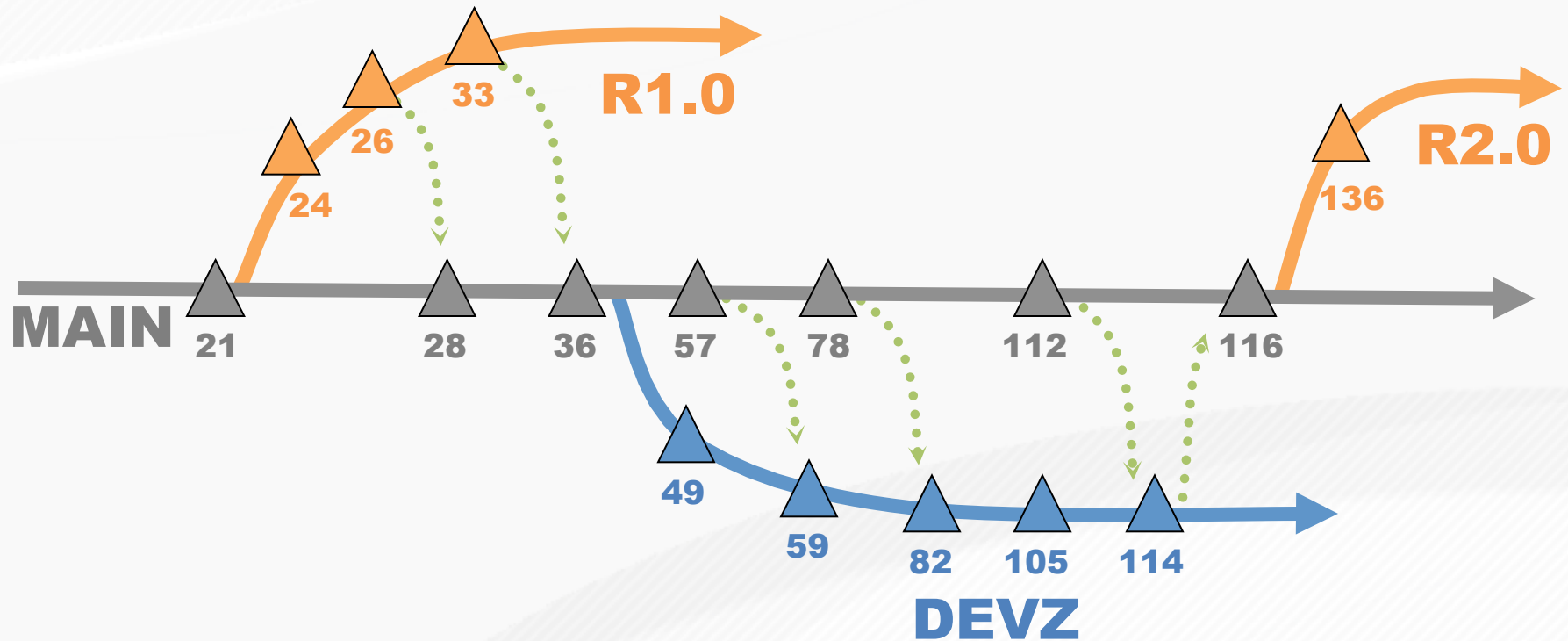


What differs between branches?

p4 diff2 -q -b MAIN-to-R1.0

```
//depot/Jam/MAIN/src/command.c#9      (text) -  
    //depot/Jam/R1.0/src/command.c#1    (text) content  
//depot/Jam/MAIN/src/regexp.c#2        (text) -  
    //depot/Jam/R1.0/src/regexp.c#2     (text) content  
//depot/Jam/MAIN/src/RELNOTES#77       (text) -  
    //depot/Jam/R1.0/src/RELNOTES#2     (text) content  
//depot/Jam/MAIN/src/scan.c#18 -       <none>  
//depot/Jam/MAIN/src/scan.h#12 -       <none>  
<none> - //depot/Jam/R1.0/src/truescan.h#1  
...etc...
```

The Flow of Change



Merge down, Copy up

p4 merge <source> <target>

- Merge from source to target
- Simplified form of **p4 integ**
- Files still need to be resolved

p4 copy <source> <target>

- Copy from source to target
- Ignores previous integrations
- No resolve scheduled

New Commands in this Chapter

- `p4 interchanges`
- `p4 copy`
- `p4 integrated`
- `p4 merge`

Introduction to Perforce for Users

Streams

•201
1.1
1.13.1

Streams

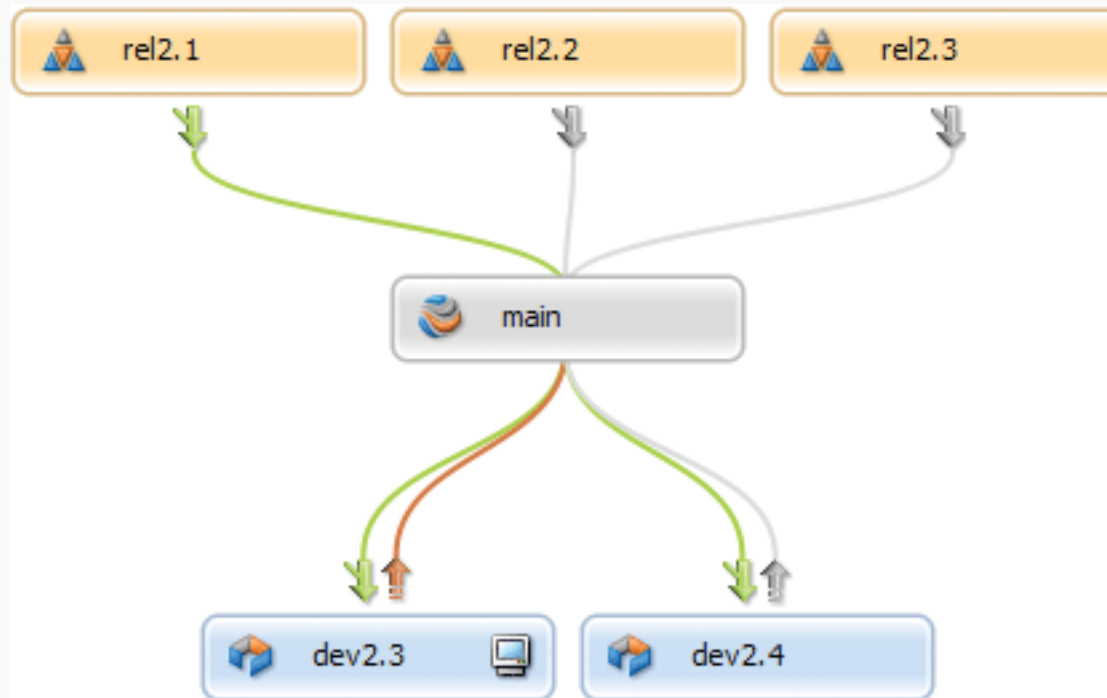
- Introduction
- Creating a stream
- Populating a stream
- Working in a stream
- Streams views

Introduction

- Intelligent branching
- Policies govern workflow
- Control the flow of change
- 1:1 relationship
- All are decedents of the mainline
- Defined relationship to its parent...
...not to its children

Stream relations

- Controlled flow of change



Stream Types

‘release’

Highly stable

(merge ‘down’ to parent, copy ‘up’ from parent)

‘mainline’

Stable per your policy

(merge ‘down’ to dev, copy ‘up’ from dev)

*(merge ‘down’ from release, **copy ‘up’ to release**)*

‘development’

Unstable

(merge ‘down’ from parent, copy ‘up’ to parent)



Creating a mainline Stream

- Requires a streams depot (e.g. `//projX`)
- Create a 'mainline' stream spec

```
p4 stream -t mainline //projX/main
```

Creating a Streams Workspace

- Creating a Streams workspace
 - Run **p4 client** as normal
 - Add stream field and name:
Stream: //projX/main
 - The '**View:**' field is automatically generated
 - user updates are ignored/not saved
- Switching to another stream (updates local files)
p4 client -s -S //projA/dev

Populating the mainline

- Add files from local file system

```
dir /s/b/a-d      | p4 -x - add  
p4 reconcile
```

- Or branch from another depot (local or stream depot)

```
p4 copy -v //src/... //projX/main/...
```

Editing a Stream Spec

p4 stream //projX/main

```
Stream:      //projX/main
Update:      2011/12/07 17:51:56
Access:      2012/01/04 14:41:06
Owner:       bruno
Name:        main
Parent:      none
Type:        mainline
Options:     allsubmit/ownersubmit
             [un]locked  [no]toparent
             [no]fromparent
```


Creating a Child Stream

- Creating a 'development' stream

```
p4 stream -t development -P //projX/main  
//projX/dev
```

- Creating a 'release' stream

```
p4 stream -t release -P //projX/main  
//projX/rel
```

Populating a child stream

- Switch client workspace to target:

```
p4 client -s -S //projX/dev bruno.projX.ws
```

- Populate child stream:

```
p4 merge -S //projX/dev -r
```

‘-S’ maps child to parent

‘-r’ reverse direction

- Submit populated revisions:

```
p4 submit -d "Branching from mainline"
```

Populating streams

- Populate a child from its parent

```
p4 populate -S //new/dev -r
```

‘-s’ maps child to parent

‘-r’ reverse direction

- Populate from an unrelated parent

```
p4 populate -S //other/main -P //new/main
```

‘-P’ target stream

Propagating Changes

- Propagate changes using:

```
p4 merge -S <stream>
```

```
p4 copy -S <stream>
```

```
p4 integ -S <stream>
```

- To merge down from 'main' to 'dev'

```
p4 merge -S //projX/dev -r
```

- To copy up from 'dev' to 'main'

```
p4 copy -S //projX/dev
```

Listing Streams

- List all streams

```
p4 streams
```

- Restrict list to a specific depot

```
p4 streams //projX/...
```

- Use '-F' to filter results

```
p4 streams -F "type=release" //projX/...
```

New Commands in this Chapter

- `p4 stream`
- `p4 streams`

Introduction to Perforce for Users

Labels

About Labels

- Labels vs. Changelist Numbers
- Creating a Label
- Tagging Files with a Label
- Static vs. Automatic Labels
- Locking a Label

Labels vs. Changelist numbers

- User determined file set
- Can be changed
- Meaningful name

Tagging files with a label (static)

```
p4 tag -l rel1 //depot/Jam/R1.0/...  
//depot/Jam/R1.0/svr/jamfile#1 - added  
//depot/Jam/R1.0/svr/libsvc.c#3 - added  
//depot/Jam/R1.0/svr/libsvc.h#5 - added  
//depot/Jam/R1.0/svr/rmtsvc.c#8 - added  
//depot/Jam/R1.0/svr/rmtsvc.h#2 - added  
//depot/Jam/R1.0/svr/userhelp.c#4 - added  
...etc...
```

Updating a label

- Use **p4 tag**

```
p4 tag -l rel1 //depot/Jam/MAIN/...@123
```

- Adding/Updating references to files

```
P4 tag -l rel1 //depot/Jam/MAIN/...
```

- Delete references to files

```
p4 tag -d -l rel1 //depot/Jam/MAIN/svr/...
```

Automatic Labels

p4 label Main-build-Apr23_10

Label: Main-build-Apr23_10

Owner: bob

Description:

Build label for development work.

Options: unlocked noautoreload

Revision: @10357

View:

//depot/Jam/MAIN/svr/...

//depot/Jam/MAIN/api/...

Unloaded Labels

```
p4 label Main-build-Apr23_10
```

```
Label:    Main-build-Apr23_10
```

```
Owner:    bob
```

```
Description:
```

```
    Build label for development work.
```

```
Options:  unlocked autoreload
```

```
View:
```

```
    //depot/Jam/MAIN/svr/...
```

```
    //depot/Jam/MAIN/api/...
```

```
p4 tag -l Main-build-Apr23_10 @10357
```

Using labels

- Syncing to it

```
p4 sync @rel1
```

```
p4 sync //depot/Jam/...@rel1
```

```
p4 sync @rel1,@rel1
```

- Listing files tagged by it

```
p4 files @rel1
```

```
p4 files //depot/Jam/...@rel1
```

Locking a Label

```
p4 label jam1.0
```

```
Label:    jam1.0
```

```
Owner:    bob
```

```
Description:
```

```
    Label for Release 1.0 - production
```

```
Options:  locked noautoreload
```

```
Revision: @10357
```

```
View:
```

```
    //depot/Jam/REL1.0/...
```


Listing Labels

```
p4 labels -e "jam-*
```

```
Label jam-1.1.0 2008/01/09 'Created by earl. '  
Label jam-2.0.5 2009/01/01 'Created by earl. '  
Label jam-2.1.0 2009/05/04 'Created by earl. '  
...etc...
```

```
p4 labels //depot/Jam/MAIN/...
```

```
Label jam-1.1.0 2008/01/09 'Created by earl. '  
Label jam-2.0.5 2009/01/01 'Created by earl. '  
Label jam-2.1.0 2009/05/04 'Created by earl. '  
...etc... (only static and not unloaded labels)
```

New Commands in this Chapter

- `p4 tag`
- `p4 label`
- `p4 labels`

Introduction to Perforce for Users

Job Tracking

Job Tracking

- Perforce Jobs
- Creating a Job
- Searching Jobs
- Linking Jobs to Changelists

Perforce jobs

- Perforce's defect tracking system
- Textual descriptions of unit of work
- Can be customized
- Can integrate with external defect tracker

Creating a job

p4 job

Job: new

User: bob

Status: open

Date: 2009/08/01

Description:

Add buttons to Gizmo space window.

Editing an existing job

p4 job job100123

Job: job100123

User: bob

Status: open

Date: 2009/08/01

Description:

Add buttons to Gizmo space window.
Buttons should be mauve or taupe.

Listing jobs

p4 jobs

```
job000001 on 1999/01/19 by aban *closed*  
  'Add clean uninstall functionality '  
job000002 on 1999/02/28 by aban *closed*  
  'Need release notes for jam project '  
job000003 on 1999/02/28 by leslie *closed*  
  'Make serialized release notes '  
job000004 on 1999/07/19 by aban *closed*  
  'Jam port to FreeBSD. '  
job000005 on 1999/10/23 by aban *closed*  
  'Jam port to SCO. '  
...etc...
```

Listing fixed jobs against a branch

```
p4 jobs //depot/Jam/MAIN/src/...
```

```
job001001 on 2008/01/23 by tsmith *closed*  
  'execunix.c blows up on fputs()'
```

```
job001162 on 2008/03/21 by tsmith *closed*  
  'MAXLINE on NT can't account for'
```

```
job004053 on 2009/01/22 by hjones *closed*  
  'Jam port to QNX'
```

Searching jobs

```
p4 jobs -e "filter email"
```

```
p4 jobs -e "status|email"
```

```
p4 jobs -e "status=open user=edk"
```

```
p4 jobs -e "date>=2008/07/14"
```

```
p4 jobs -e "description=filter*"
```

```
p4 jobs -e "doesn't return zero"
```

Setting a job view

p4 user

```
User:      bob
Email:     bob@caniche.com
Update:    2008/08/09 13:45:59
Access:    2010/06/07 16:45:05
FullName:  Bob Everly
JobView:   user=bob status=open
```

Linking jobs to changelists

- Link changelist and job

```
p4 fix -c changenum jobname
```

- Submit sets job status to **fixStatus**

- Override using **p4 submit -s status**

Job fix reporting

- What changelists fixed a job?

```
p4 fixes -j jobname
```

- What jobs were fixed by a changelist?

```
p4 fixes -c changenum
```

- Same output for both

```
p4 fixes -j job001234 or p4 fixes -c 12345
```

```
job001234 fixed by change 12345 on 2010/01/22 by  
jrice@rover
```

New Commands in this Chapter

- `p4 job`
- `p4 jobs`
- `p4 user`
- `p4 fix`
- `p4 fixes`

Finally...

Visit Perforce

<http://www.perforce.com>

and our Public Depot

<http://public.perforce.com/public/index.html>

for current listings of available software

Perforce Highlights

FAST

Easy to learn

Superior technical support

Cross-platform compatibility

Atomic change transactions

Powerful integration algorithm

The End

All Perforce manuals and technical notes are available
at www.perforce.com.

Report problems and get technical help from
support@perforce.com.

Share tips and ideas with other users on
<http://forums.perforce.com>