

Introduction to Perforce for Users

Introduction

- Introductions
- Class Schedule
- GUI vs. CLI
- About the Exercises
- The meaning of *live* (P4V Demonstrations)

Course Contents (1 Day)

- Overview
- Help!
- Basic File Operations
- File Reporting and Revision Specifiers
- Changelist Management
- Workspace Management
- Handling File Conflicts
- Branching and Integration
- Streams
- Labels (optional)
- Job Tracking (optional)

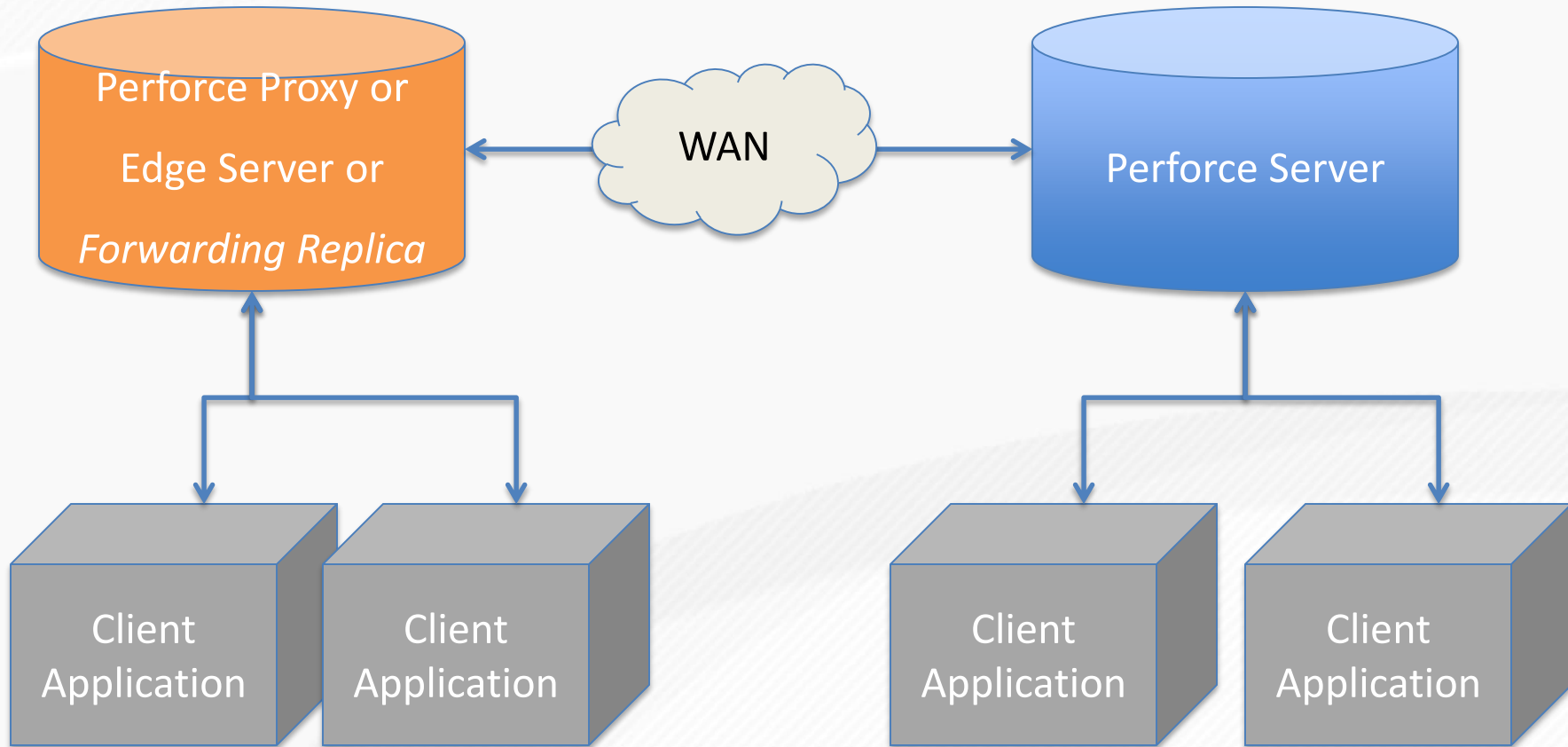
Introduction to Perforce for Users

Overview

Overview

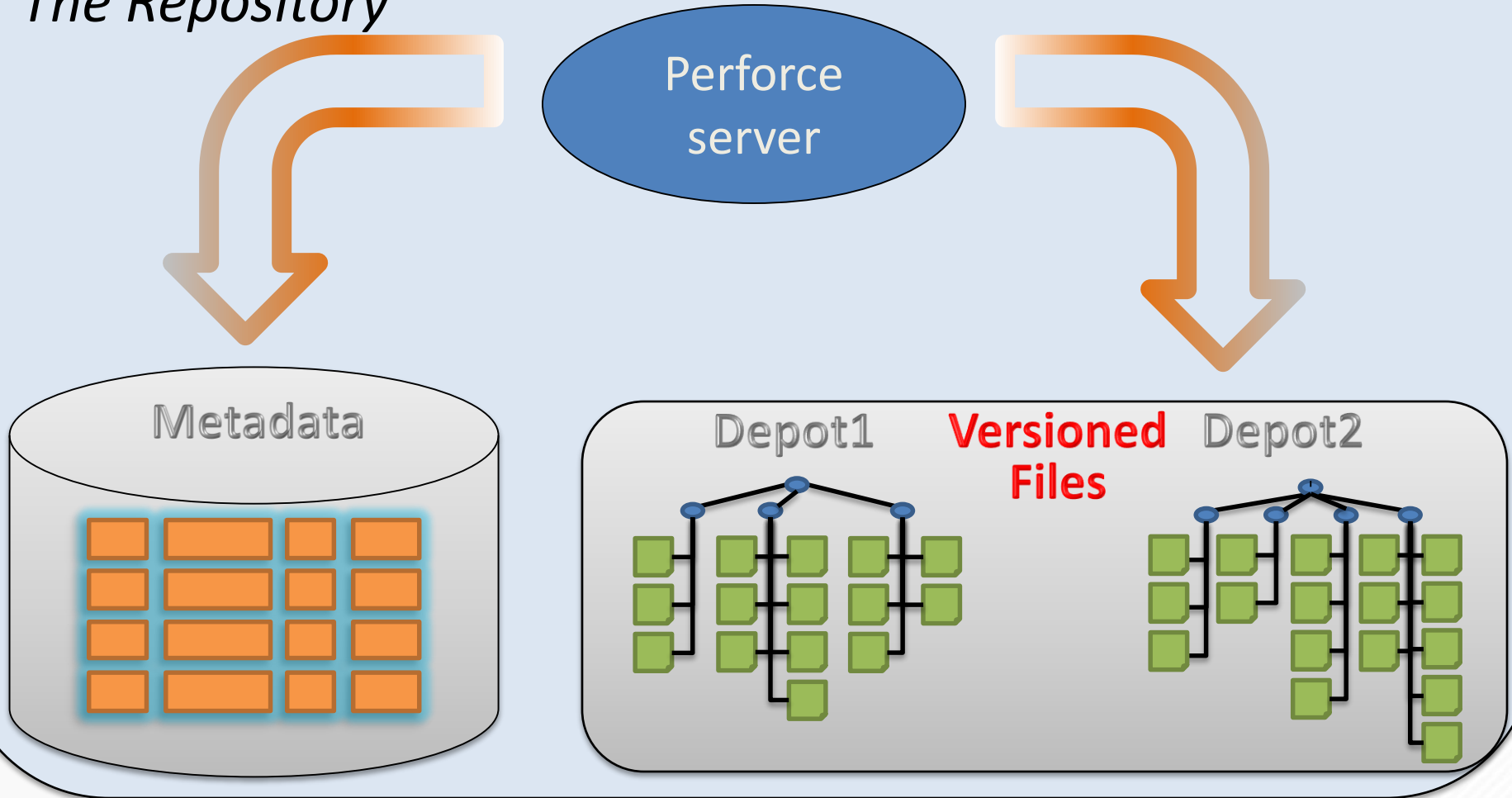
- Client/Server Model
- Server
- Client Programs
- Changelists
- Terminology: Client...
- Connecting to Perforce

Perforce Client-Server model

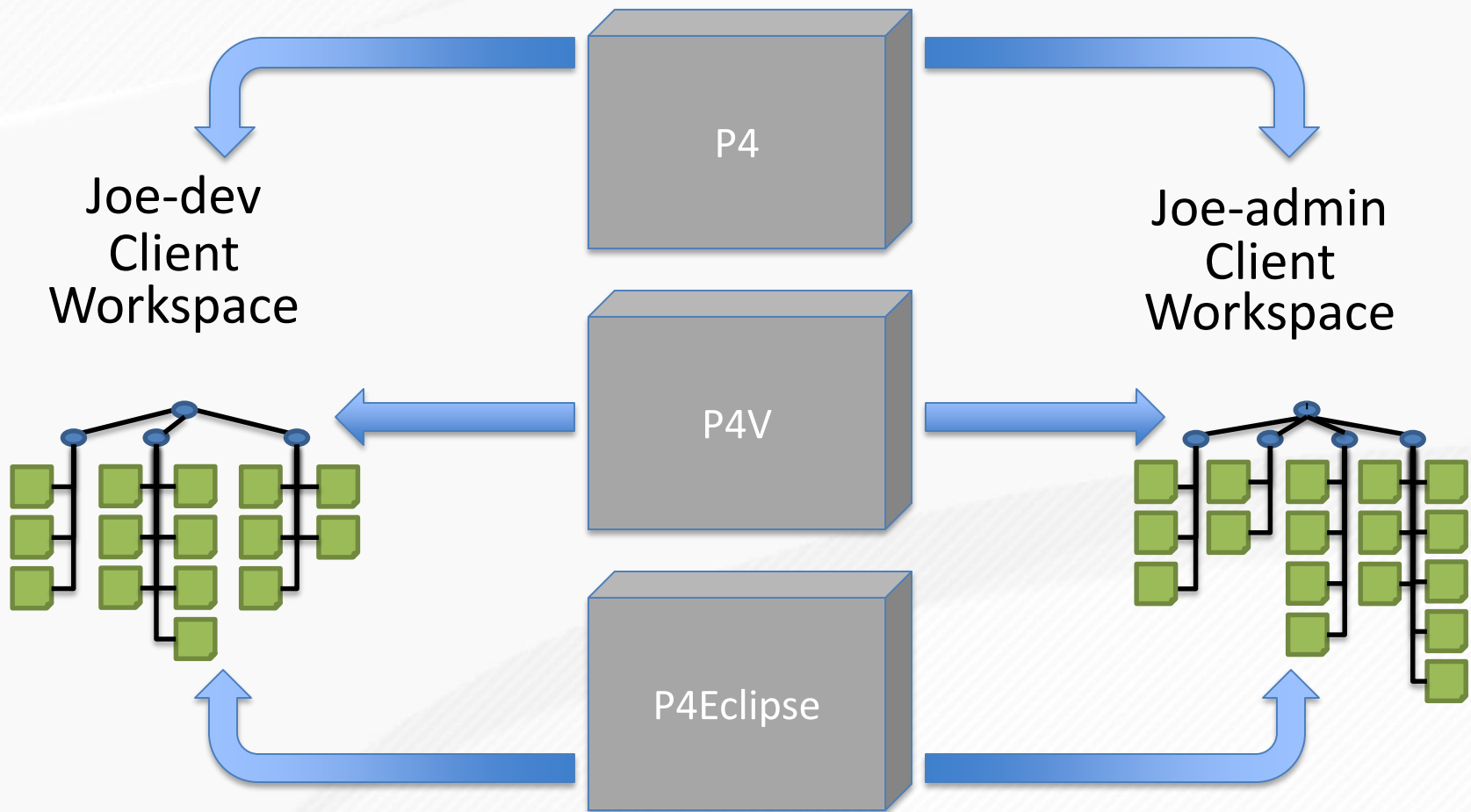


Perforce server

The Repository



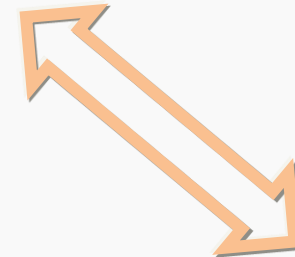
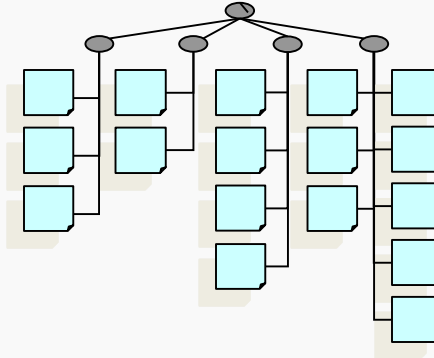
Perforce Client applications



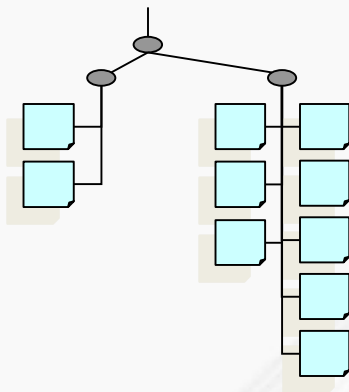
Server & Workspaces

Some-server:1701

//ABCdepot

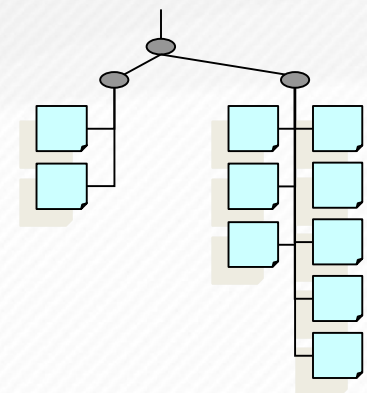


***“Joe-dev”
Workspace***

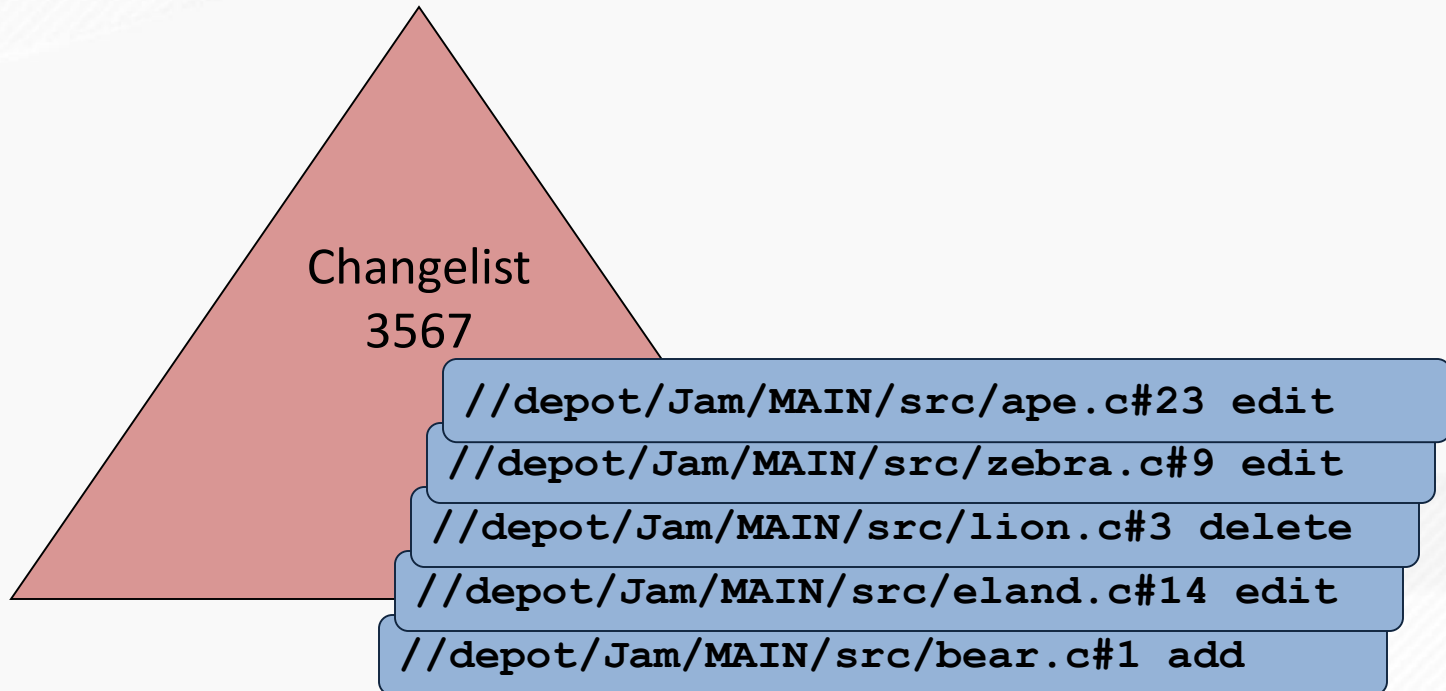


***Joe-dev and Fred-dev
might contain the same
depot files but on
different machines***

***“Fred-dev”
Workspace***



Changelists



Terminology - Client ...

- Client machine
- Client programs
- Client workspaces

Connecting to the server ***Live***

- Basic connection - specify
 - <Server-machine>:<port-number>
 - Username
 - Workspace (optional)
- You may be prompted for password

Commands in this Chapter

- Connecting to Perforce

Introduction to Perforce for Users

Help!

Help!

- P4V Help Files
- Online Perforce Documentation
- E-mail Perforce
- Perforce Forums / Community
- P4Blog / Newsletter

Getting help in P4V

live

- Task oriented context sensitive help
- Getting started with P4V manual
- Showing system information

Documentation on Perforce's web site

Support > Documentation

- Introducing Perforce
- Getting Started with P4V
- P4/P4V Cheat Sheet
- Perforce Command Reference
- Perforce System Administrator's Guide
- Release Notes
- Knowledge Base
- White Papers

Help via E-mail and other sources

- E-mail Perforce at:
 - support@perforce.com
 - sales@perforce.com
 - consulting@perforce.com
- Perforce Community
 - www.perforce.com/community
 - Forums / Blog / Ideas
 - Exchange ideas with other Perforce users

Customize P4V

live

- General settings
- Editor choices
- Diff applications
- Merge applications
- File content font selection

New Actions in this Chapter

- Help →
 - P4V Help
 - Getting Started with P4V
 - System Info

Introduction to Perforce for Users

Basic File Operations



Basic Operations

- Populating a Client Workspace (Get Latest Revision)
- Editing, Adding and Deleting Files
- Renaming/Moving Files
- Reverting Files
- Assessing Current Status
- Submitting Changes

Get latest revision of files

live

Get latest revision

- Select  Depot or  Workspace tree
- Select desired folder or files
- Selected files are transferred to your workspace from the server
- Files are read-only by default

Editing workspace files

live



Check out . . .

- or
- Check out and open...
- Make local files writeable
- Optionally, open files in editor
- Other users can see which files you have checked out

Adding new files

live

Mark for Add...

- Save new files in your workspace tree
- Select the Workspace tree
- Refresh P4V and navigate to your files
- Optional:
- Change Filetype...
 - Add file type modifiers
 - Change base file type

Deleting files

live

Mark for Delete...

- Select files for deletion on the server
- Marked files are removed from workspace

Renaming/moving

live

- **Rename/Move . . .**
- Rename a file or move to another server folder
- New file name is marked for move/add
- Old file name is marked for move/delete

Reverting workspace files

live

- **Revert**
- Discard action on marked files
- Edits overwritten
- Deleted files copied to workspace
- Files marked for add unmarked

Assessing work in progress

live

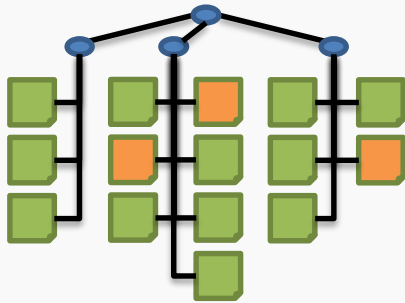


Pending changelist tab

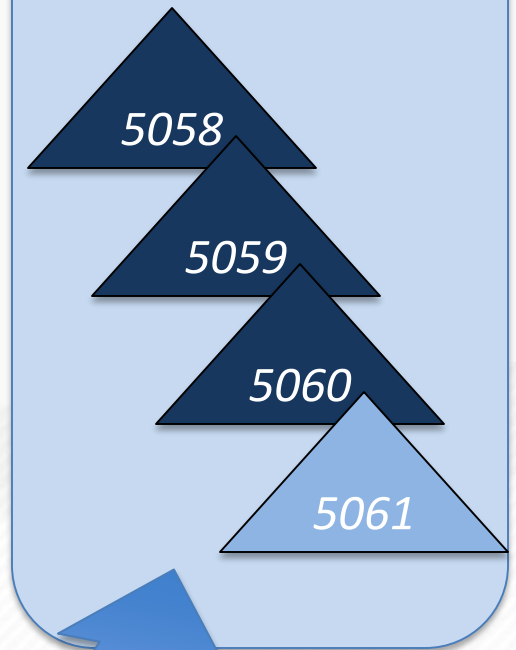
- Which files are you working on?
- **Diff Against Have Revision**
- What changes have you made?

Submitting a changelist

Client Workspace



Depot



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

Submitting a changelist

live

Submit...

- Opens the submit dialog
- Enter a description for the default changelist
- Submit
 - Completes the submit action
 - Files are transferred to the server

New Actions in this Chapter

- **Context-click →**
 - **Get Latest Revision**
 - **Check Out**
 - **Mark for Add**
 - **Mark for Delete**
 - **Rename/Move**
 - **Revert**
 - **Diff Against Have Revision**
 - **Submit**
- **Marked files show in pending changelist**

Introduction to Perforce for Users

File Information / Reporting
Underlying principles and P4V
implementation

File Reporting

- Finding files
- Wildcards for filenames
- Revision specifiers
- Showing file and folder history
- Comparing and diffing files and folders

Searching for files

live



Find File...

- List files matching search pattern

Show In

- Depot Tree
- Workspace Tree
- File browser
- Pending Changelist

Supported Wildcards

- P4V vs CLI
- 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/*
```

Revision Specifiers – CLI/API syntax

Type	Syntax
Revision number	#7
Changelist number	@30
Workspace name	@sams_ws
Label name	@rlos_label
Date	@2012/03/01
<i>Latest revision</i>	<i>#head</i>
<i>Have revision</i>	<i>#have</i>
<i>Non-existent revision</i>	<i>#none or #0</i>

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	

- A changelist specifier can refer to:
 - The set of files contained in the changelist
 - A point in time in the life of your repository (like a label)

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
```

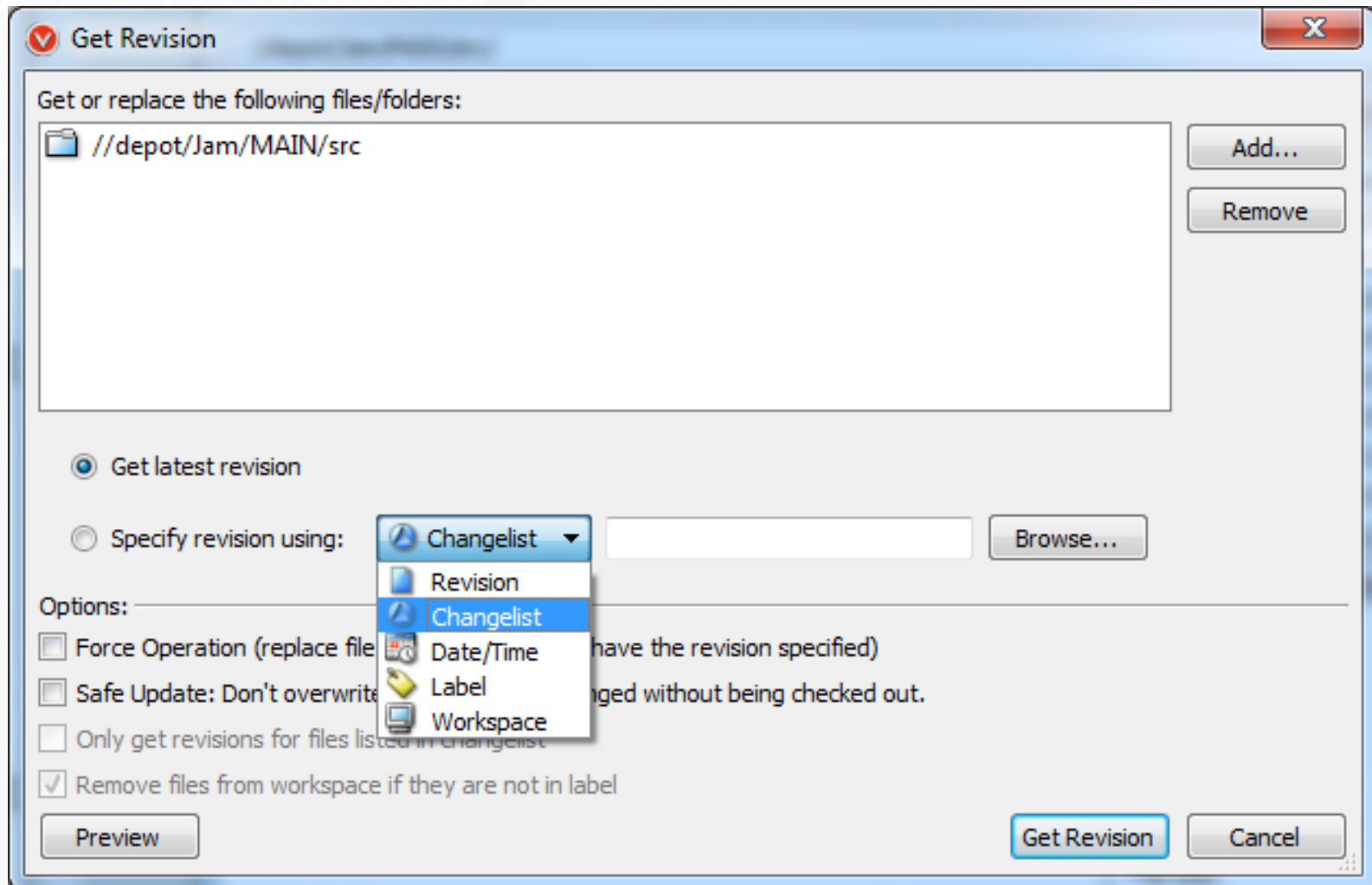
- List files updated up to a known changelist

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

- List changes submitted up to a known date

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

Revision Specifiers - P4V



Revision Specifiers - CLI

```
p4 sync utils.c#2
p4 sync utils.c@431
p4 sync @431
p4 sync ...@431
p4 changes //depot/Jamgraph/DEV/gizmo/*.c@431
p4 sync @janslabel
p4 files //depot/Acme-api/...#have
p4 sync //depot/notes/....txt
p4 sync #none
p4 sync util.c@2014/05/29
p4 files //depot/Jam/...@2014/05/29:15:37:00
```

Revision ranges – CLI only

- Limiting commands to a revision range:

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

```
p4 changes xyz.c@rel1,@rel2
```

```
p4 files //depot/api/...@431,@431
```

```
p4 files //depot/api/...@=431
```

- Some commands accept revision ranges:

```
p4 changes
```

```
p4 integ
```

```
p4 sync
```

```
p4 jobs
```

```
p4 files
```

```
p4 print
```

```
p4 fixes
```

Displaying file properties

live

Files in Folder

- Display file information

Details

Checked Out By

Preview

Displaying file history

live



File History

- List summary of each submitted revision
- Use drag-and-drop for showing file diffs

Details

Integrations

Labels

Preview

Displaying revision graph

live

Revision Graph

- View a file's history graphically
- View file differences
- Display file information

Details

Integrations

Labels

Preview

Displaying Time-lapse View

live

Time-lapse View

- Recently added lines color-coded
- Lifetimes of adjacent chunks of text
- Which user made the change?
- Find text in a file
- View a single revision, incremental diffs or a range of revisions

Removing workspace files

live

Remove from workspace

- Delete local files under Perforce control
- Leave marked files untouched

Displaying folder history

live

Folder History

- Drag-and-drop changelists to view **Folder Diff**

Using the Diff dialog

live



Diff Against...

- Select files or folders in the server
- Use a revision specification

File types

- Base file types:

`text`
`binary`

`symlink`
`apple`

`unicode`
`resource`

`utf16`

- Workspace storage attributes

`+x` - *executable*

`+w` - *always writable*

`+k` - *RCS keyword*

`+ko` - *RCS keyword (old style)*

`+l` - *exclusive open*

`+m` - *sync vs. submit modtime*

- Server storage attributes

`+S` - *latest revision*

`+C` - *compressed*

`+D` - *deltas*

`+F` - *full file*

- See **Command Reference Guide** for definitive explanation

New Actions in this Chapter

- Edit → Find File
- View → Files in Folder
- Context Click →
 - *Get Revision*
 - *Revision Graph*
 - *Remove from Workspace*
 - *Diff Against*
 - *Change Filetype*
 - *File History*
 - *Time-lapse View*
 - *Folder History*
- Tree Filter pull-down menus

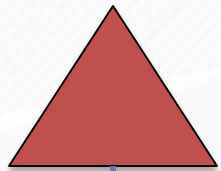
Introduction to Perforce for Users

Changelist Management

Changelist Management

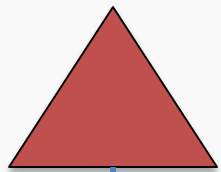
- Managing pending changelists
- Shelving opened files
- Showing changelist detail

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>
- ...

Working with multiple changelists *live*

Pending Changelists

- Organize your work
- Rearrange files before submitting
- Add to the description


Shelving opened files

live

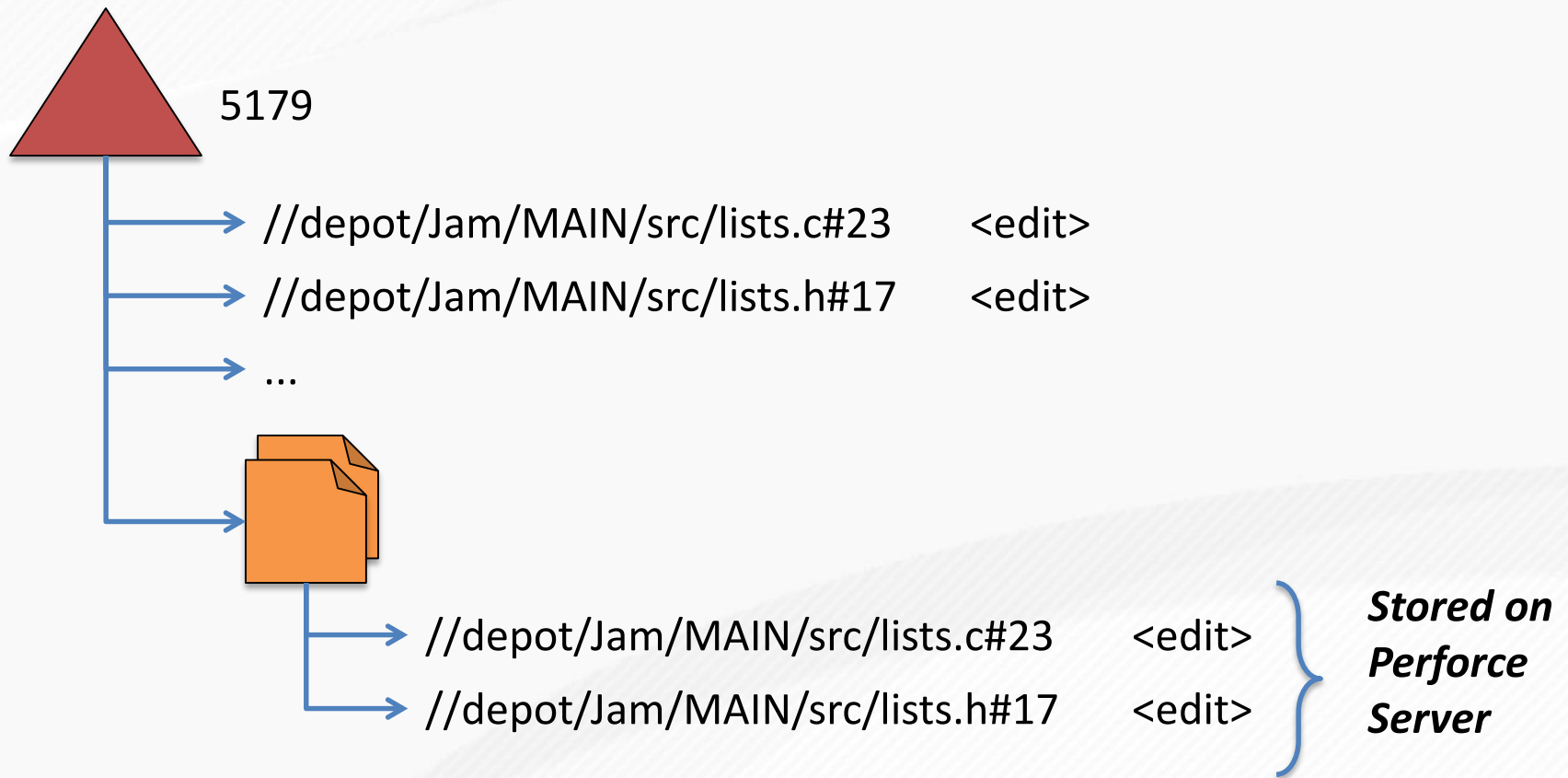
Shelve Files...

- Transfer edited files to a “shelved” changelist on the server
- Revert checked out files
- Clear previously shelved files

Working with shelved files *live*

-  Shelved files are available to other users
 - Unshelve and open for edit
 - View file differences
 - Transfer ownership
- Shelved changelists can be updated
- Owners can delete shelved changelists

Shelved Changes



Filtering changelists

 Pending or

 Submitted Changelists

- Filter by:
 - Folder or file path
 - User name
 - Workspace name

New Actions in this Chapter

- File → New → Pending Changelist
- Drag and drop files among changelists
- Context click →
 - Edit Pending Changelist *<nnn>*
 - Submit
 - Shelve Files
- View → Submitted changelists
- Context click →
 - View Submitted Changelist *<nnn>* →
 - Show File Diffs
- Filter Submitted Changelists

Introduction to Perforce for Users

Workspace Management

Setting up a Workspace

- Creating a Workspace
- Setting Workspace View Mappings
- Referencing Files
- Configuring Workspaces and Connections

What is a workspace?

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

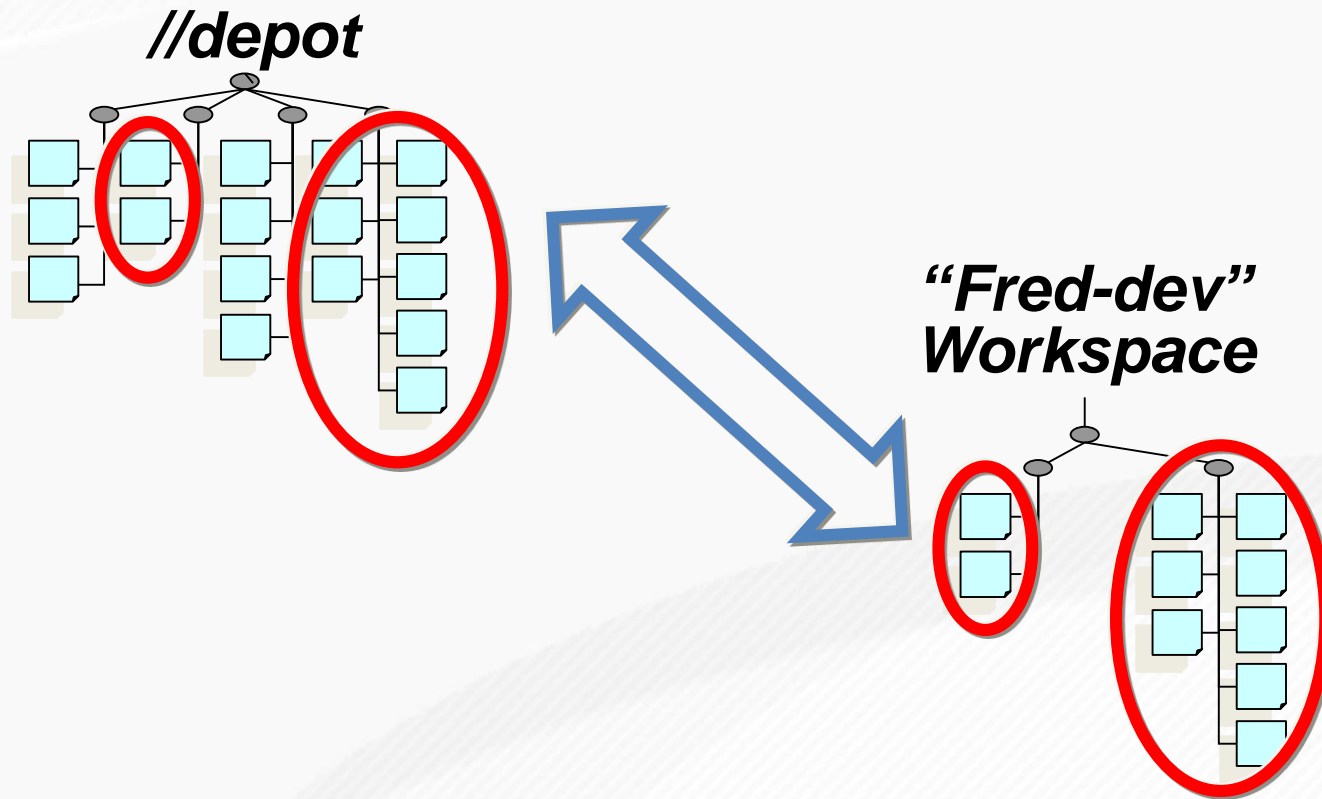
Why have multiple workspaces?

- 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

Workspace Views map between server and local file system



Creating a workspace specification

Workspace: bruno_ws (localhost:1666, bruno)

Basic Advanced

Workspace name:

Workspace root: Browse...

Stream:

Stream at change:

▼ Workspace Mappings:

//depot/...	//bruno_ws/...
-------------	----------------

☐ Automatically get all revisions

OK Cancel Apply

Workspace: bruno_ws (localhost:1666, bruno)

Basic Advanced

Workspace name:

Workspace root: Browse...

Stream: Browse...

Stream at change:

▼ Workspace Mappings:

☒ ☐ ☐ ☐ ☐

Depot Tree	Client Expression
▶ <input checked="" type="checkbox"/> depot	//bruno_ws/...
▶ <input type="checkbox"/> gwt	
▶ <input type="checkbox"/> gwt-streams	
▶ <input type="checkbox"/> HR	

☐ Automatically get all revisions

OK Cancel Apply Help

Creating a workspace specification *live*



Create a new workspace

- When launching P4V

Connection → New Workspace...

- In the **Depot** or **Workspace** tree
- In the **Workspaces** pane

Client workspace view

Client: bob-jam

Root: c:\p4work\bob-jam

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/A/B/config/*	//bob-jam/Jam/MAIN/cfg/*
//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/...

Workspace view tab options *live*

- Include or exclude
 - Files or folders
 - An entire tree
 - Files under one folder
 - Files by extensions
 - Files or folders mapped to different names

Workspace customization

live

- Specify a root folder
- Add alternate root folders
- Set workspace options
- Specify line ending translation
- Choose submit option

Viewing workspaces

live

- Filter workspaces by:
 - Owner
 - Names containing a string
 - Available for use on your computer

Copying workspaces

live

- Clone a workspace's
 - Options
 - View
 - Submit Options
 - Line Endings

New Actions in this Chapter

- Open Connection → New → Workspace
- Details → Form/View tabs
- Connection →
 - Edit Current Workspace
 - New Workspace
- View → Workspaces →
 - Name contains: Filter
 - Context click → Create Workspace from *<workspacename>*

Introduction to Perforce for Users

Handling File Conflicts

Handling File Conflicts

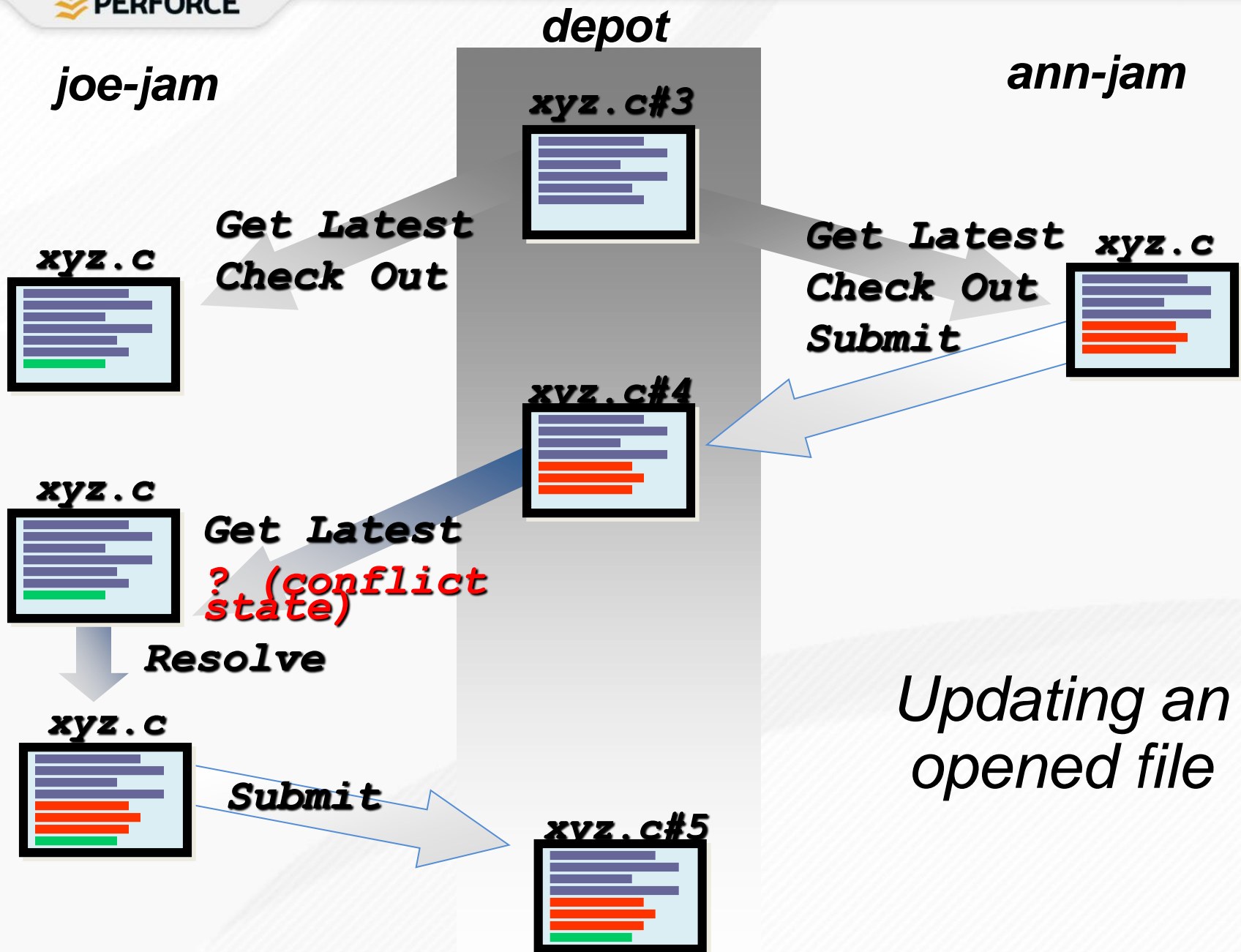
- File Conflicts
- Resolving Conflicts
- Conflicting Diffs
- Reporting Actions

Classic conflict scenario

- Two users edit the same file
- First user submits
- Second user gets latest revision
- Resolve is 'scheduled'

Other conflict scenarios

- 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*



Updating out-of-date workspace files

live

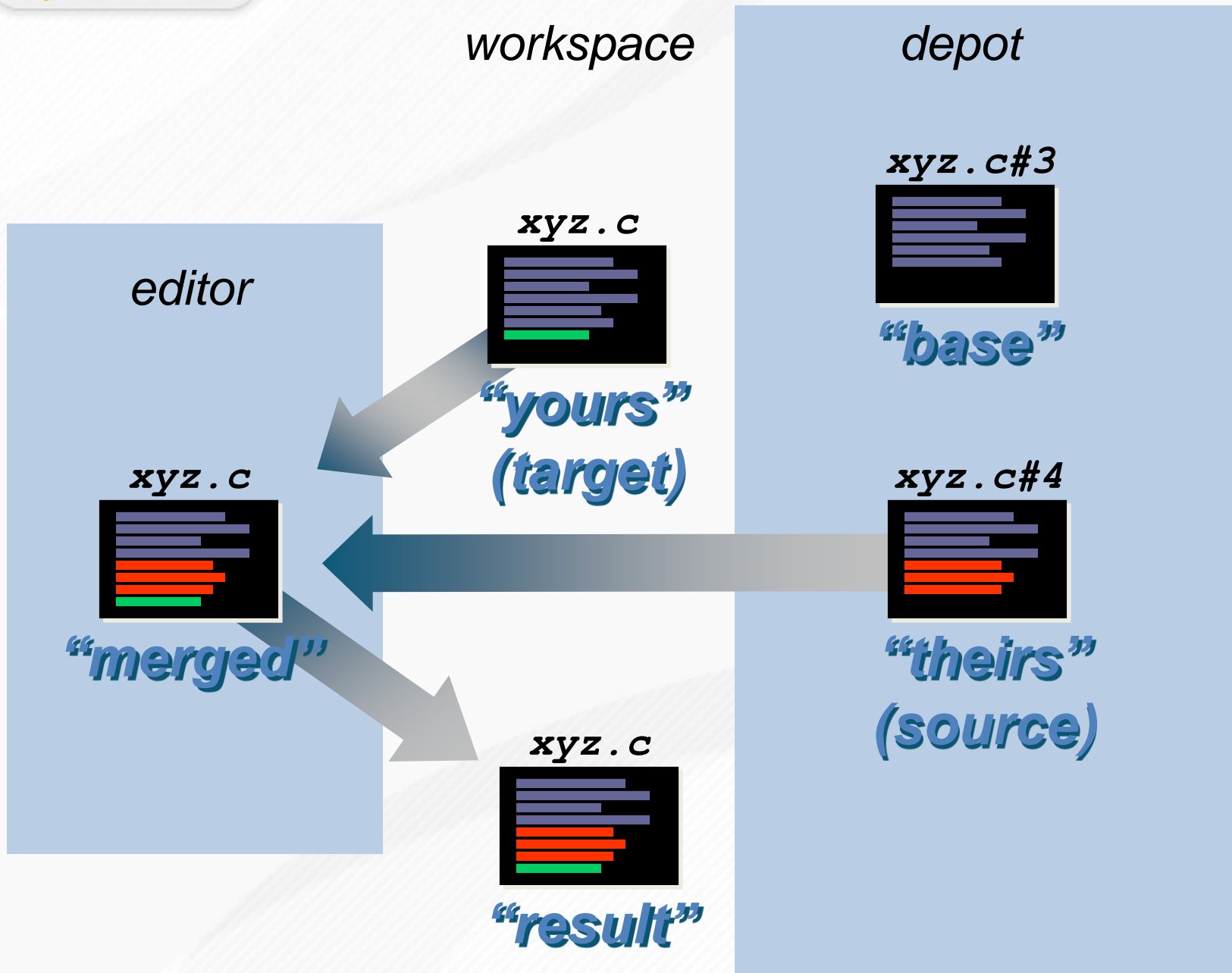
 File checked out not at the head revision

- Get latest revision to “schedule a resolve”
- File checked out will not be changed

 File ready to resolve

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

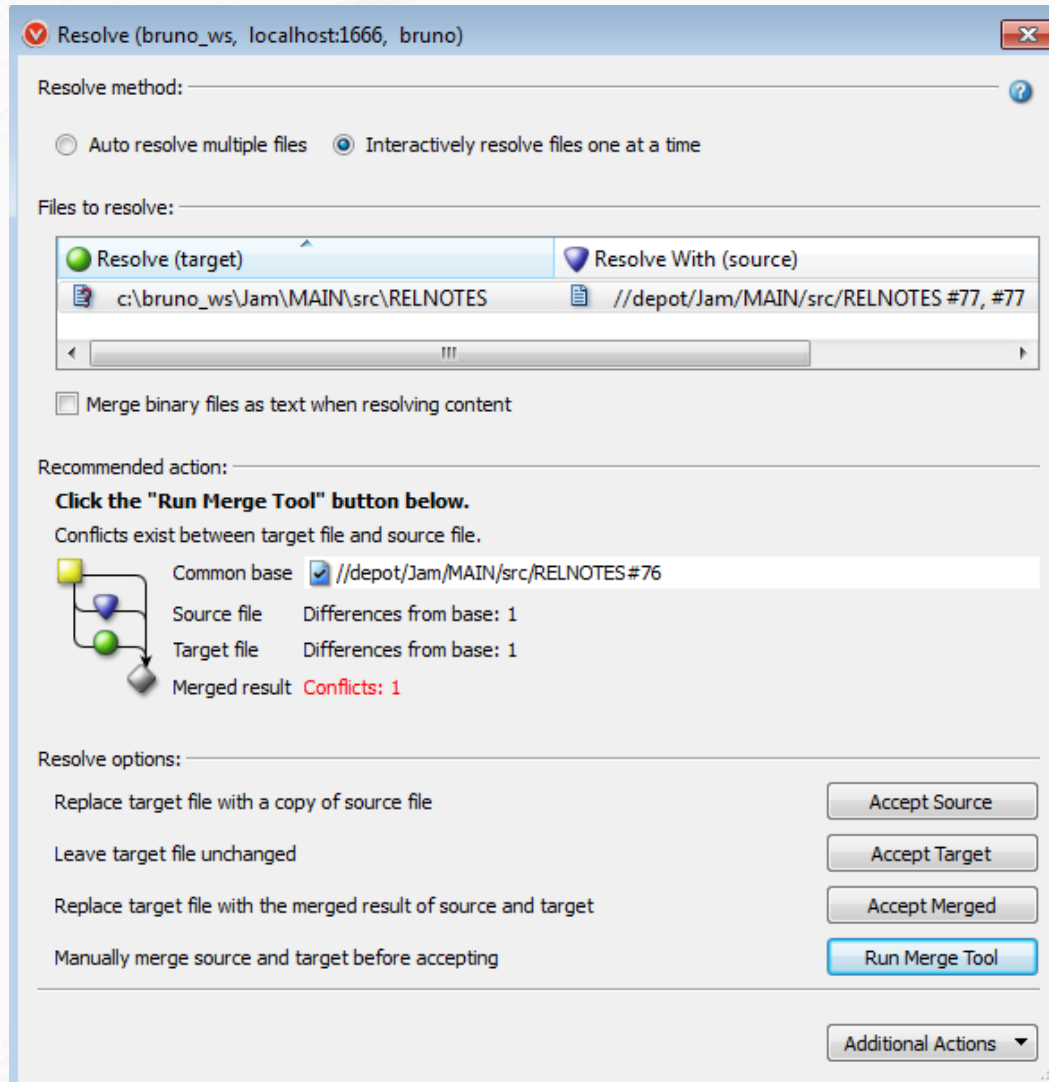


Resolving files interactively

Resolve . . .

- Accept Source (Theirs – depot file)
- Accept Target (Yours – workspace file)
- Accept Merged
- Run Merge Tool
- Data mining / Additional Actions

Resolving options



Diff chunk types

yours

- Unique diff chunk in client file only

theirs

- Unique diff chunk in depot file only

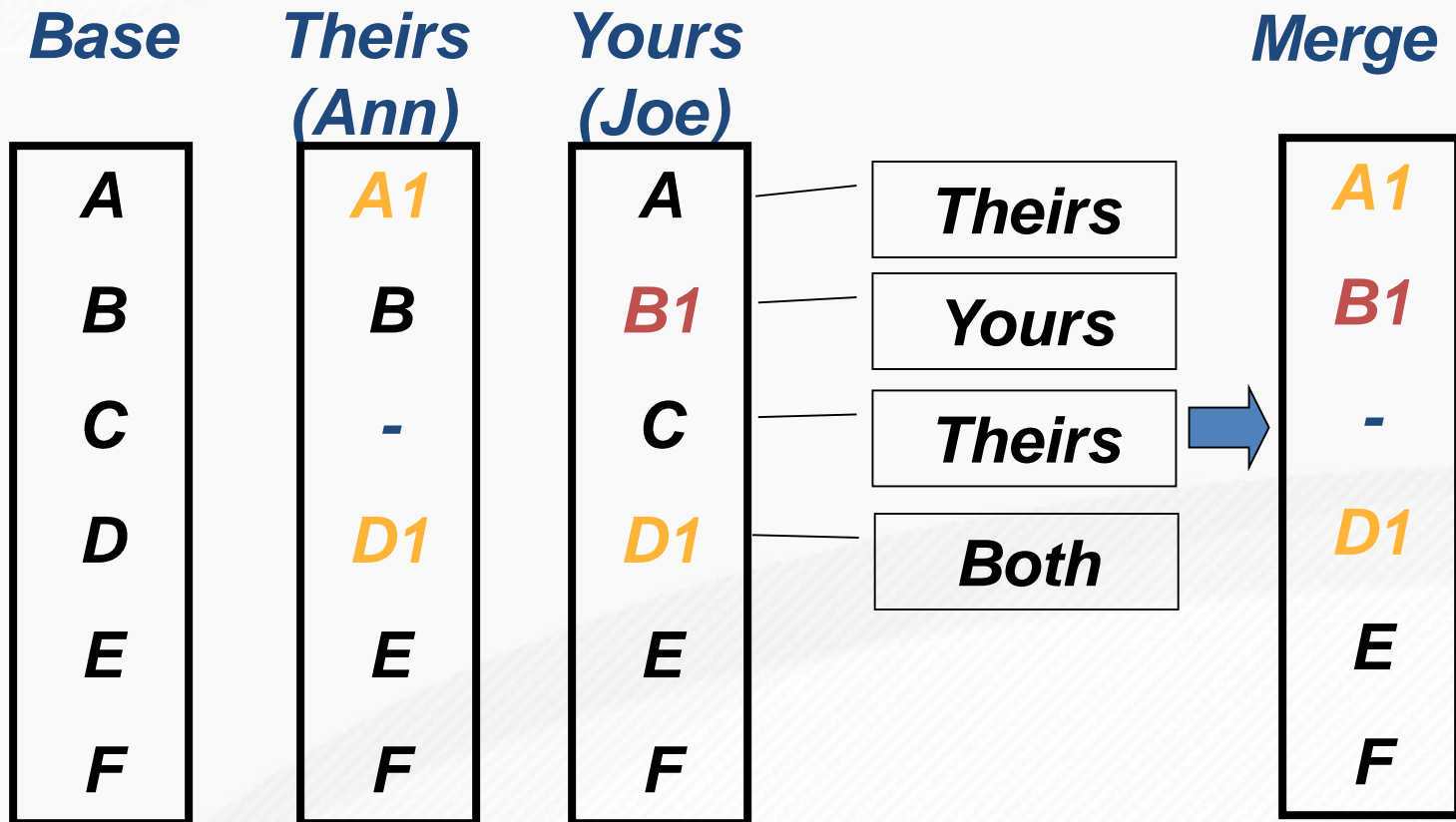
both

- Unique diff chunk common to depot and client file

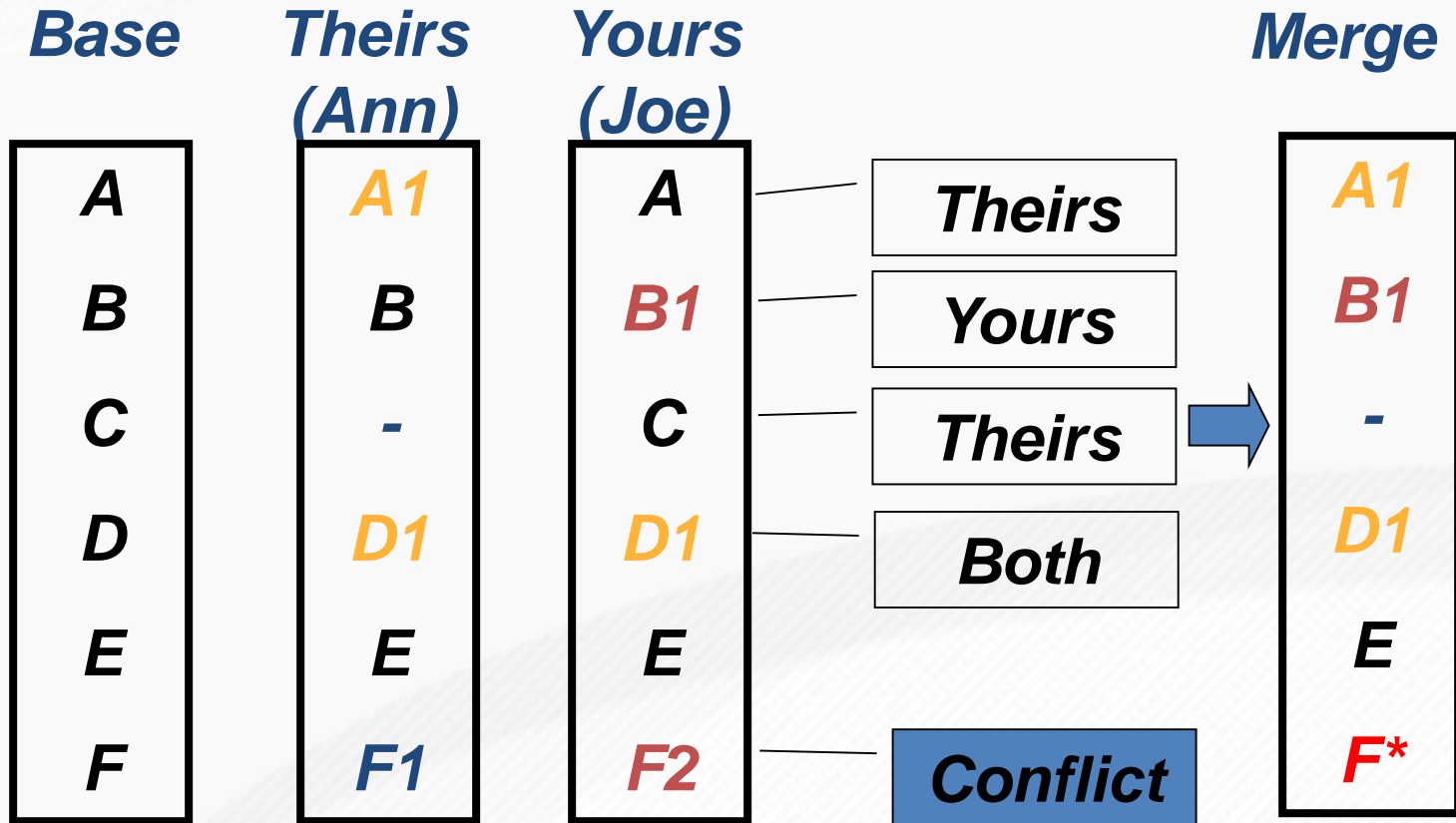
conflicting

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

Diff Chunks – Easy 3-way Merge



Diff Chunks – 3-way Merge conflicts



Merging files using P4Merge

headers.c - Perforce P4Merge

File Edit View Search Help

1 diffs (Ignore line ending differences) | Tab spacing: 4 | File format (Encoding: System Line endings: Windows)

Base: headers.c#5
Theirs: headers.c#6
Yours: headers.c

Differences from base: 0
Differences from base: 0

Merge: Merge file used for resolve Conflicts: 1

//depot/Jam/MAIN/src/headers.c#6	//depot/Jam/MAIN/src/headers.c#5	c:\bruno_ws\Jam\MAIN\src\headers.c
72 p[0].string = hrrule->	72 p[0].string = hrrule->	72 p[0].string = hrrule->
73 p[0].left = &p[1];	73 p[0].left = &p[1];	73 p[0].left = &p[1];
74 p[1].l1ist = list_new(74 p[1].l1ist = list_new(74 p[1].l1ist = list_new(
75 p[1].left = &p[2];	75 p[1].left = &p[2];	75 p[1].left = &p[2];
76 p[2].l1ist = headers1(76 p[2].l1ist = headers1(76 p[2].l1ist = headers1(
77 p[2].left = 0;	77 p[2].left = 0;	77 p[2].left = 0;
78	78	78
79 if(p[2].l1ist)	79 if(p[2].l1ist)	79 if(p[2].l1ist)
80 {	80 {	80 {

Merge file used for resolve

```

p[0].left = &p[1];
p[1].l1ist = list_new( L0, t->name );
p[1].left = &p[2];
p[2].l1ist = headers1( headlist, t->boundname, rec, re );
p[2].l1ist = headers1( headlist, fname, rec, re );
p[2].l1ist = headers1( headlist, fname, rec, re, ro );
p[2].left = 0;
if( p[2].l1ist )

```

Make this file be the "correct" result.

Locking files

live



- Other users cannot submit until you submit or unlock the file
- Exclusive lock filetype prevents other users even checking out

Recovering a deleted file

live

Show deleted depot files

Rollback...

Submit...



Reconciling offline work

live

- Mark files for:
 - add
 - edit
 - delete

New Actions in this Chapter

- Context click →
 - Get Latest Revision
 - Resolve →
 - Open File
 - Diff
 - Run Merge Tool
 - Lock
- View → Submitted Changelists →

Introduction to Perforce for Users

Classic Branching and Integration
Branching Files and Propagating
Changes
(and Branch Mappings)

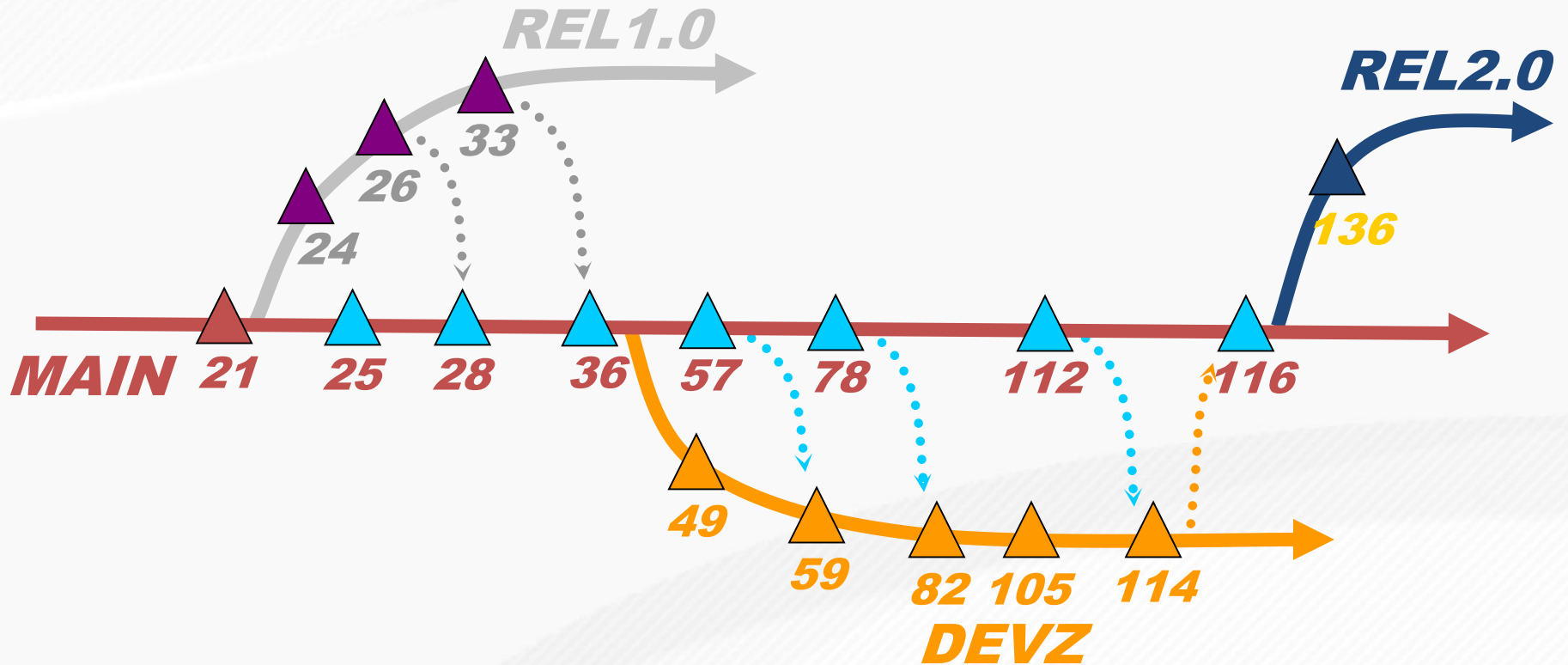
Branching and integration

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

Terminology

- Branch
 - (noun) A set of related files (a.k.a. codeline), created as a copy of its parent
 - (noun) A branch mapping (in P4V)
 - (verb) To create a new codeline (branch)
- Integrate
 - To create a new branch/codeline ([Branch Files...](#))
 - To propagate changes between existing branches/codelines ([Merge/Integrate...](#) or [Copy...](#))

Some branching patterns



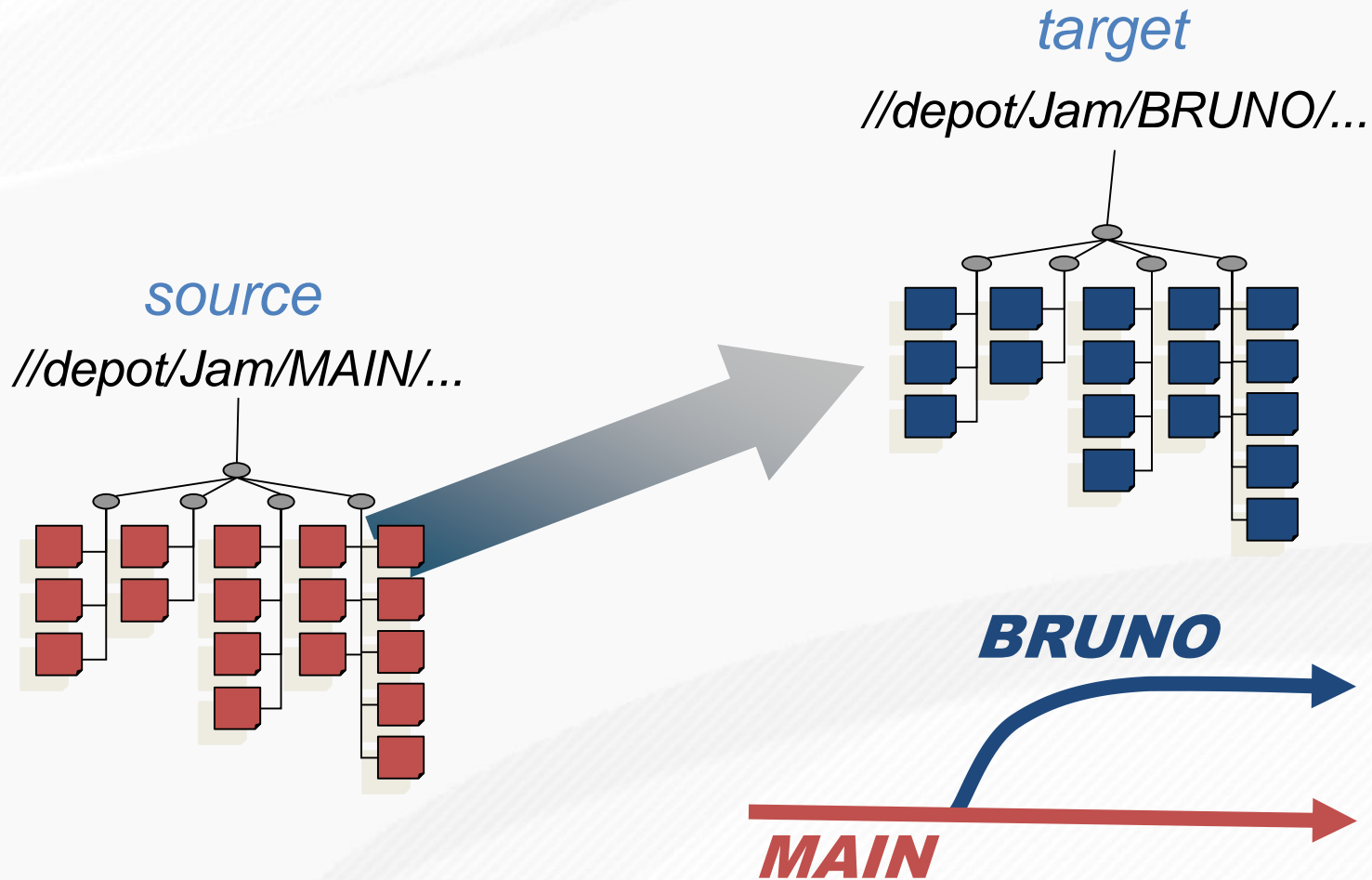
Why branch?

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

High-Level SCM Best Practices

<http://www.perforce.com/sites/default/files/pdf/high-level-perforce-best-practices.pdf>

Creating a branch

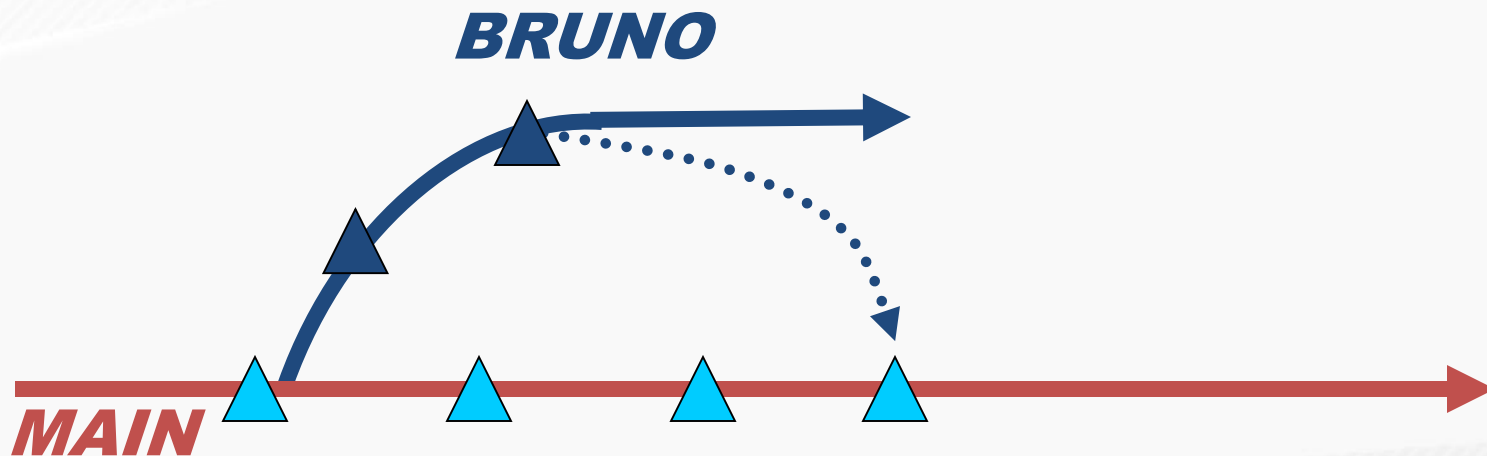


Creating a new branch

live

- **Branch Files...** specifying source and target
- **Submit...**

Propagating changes across branches



Propagating changes

live

- The target branch must be mapped in the current workspace view – we are changing those files!
- **Merge/Integrate...**
- **Resolve...**
- **Submit...**

Resolving for merge/integrate

- *Theirs* = Source
- ***Yours* = Target (the one being changed by merge)**
- *Base* = Closest common ancestor

workspace

depot

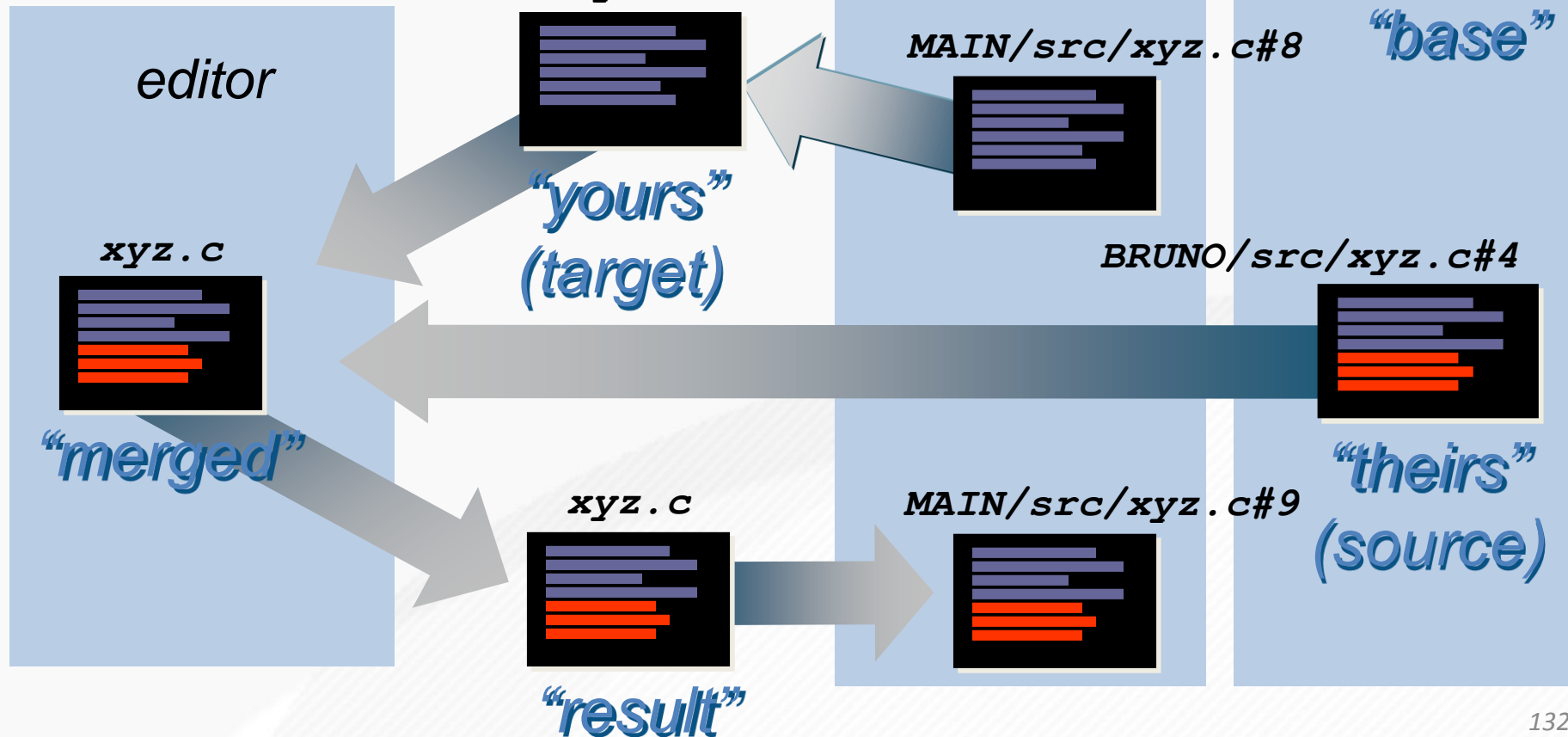
Connection → Edit Current Workspace...

Merge/Integrate... → Specify file paths...

//depot/Jam/BRUNO/... //depot/Jam/MAIN/... BRUNO/src/xyz.c#3

Resolve...

Submit...

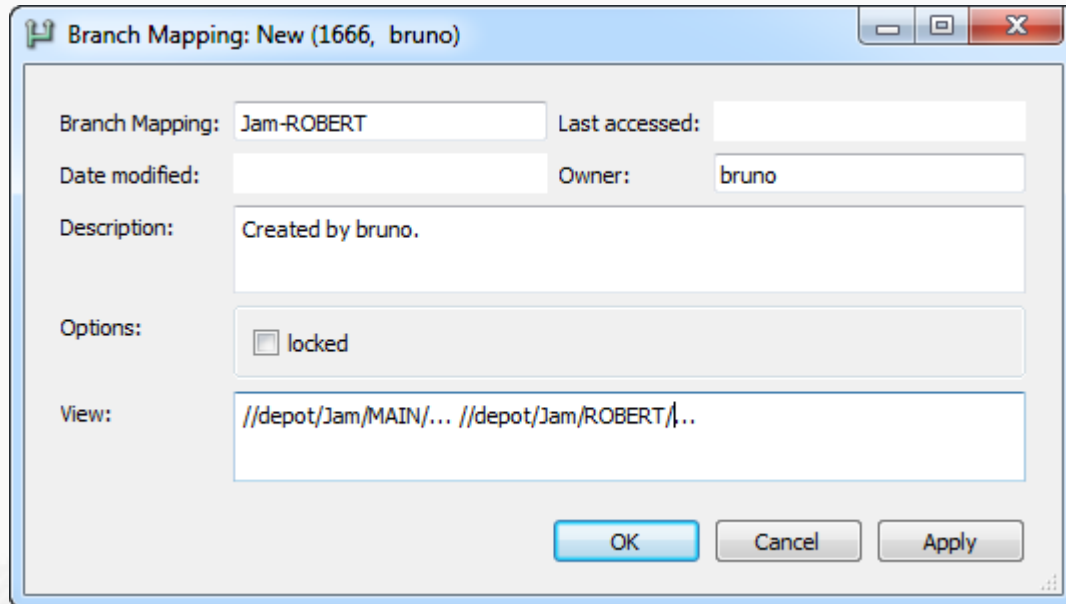


Integrating using branch mappings

- Branch mappings store source/target pairs
- Integrate using branch mappings
 - Create new branch or Save
 - Re-use to propagate changes

Creating branch mappings

New Branch Mapping...



Branch Mapping: Last accessed:

Date modified: Owner:

Description:

Options: ☐ locked

View:

OK Cancel Apply

Integrating using branch specifications*live*

- **Merge/Integrate...**
- Choose branch spec
- **Resolve...** when propagating changes
- **Submit...**

Introduction to Perforce for Users

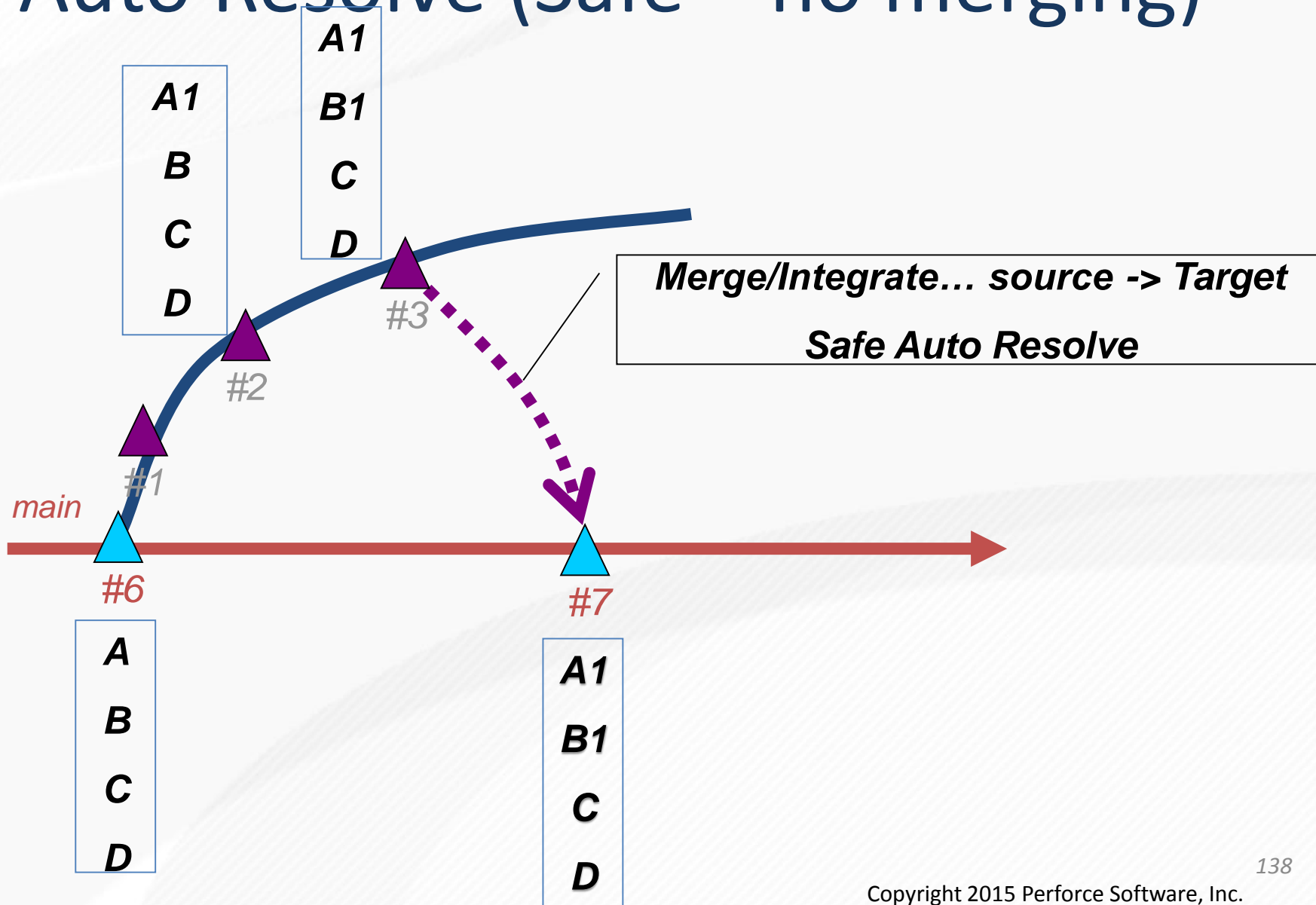
Automatic Resolving

Automatic Resolve Options

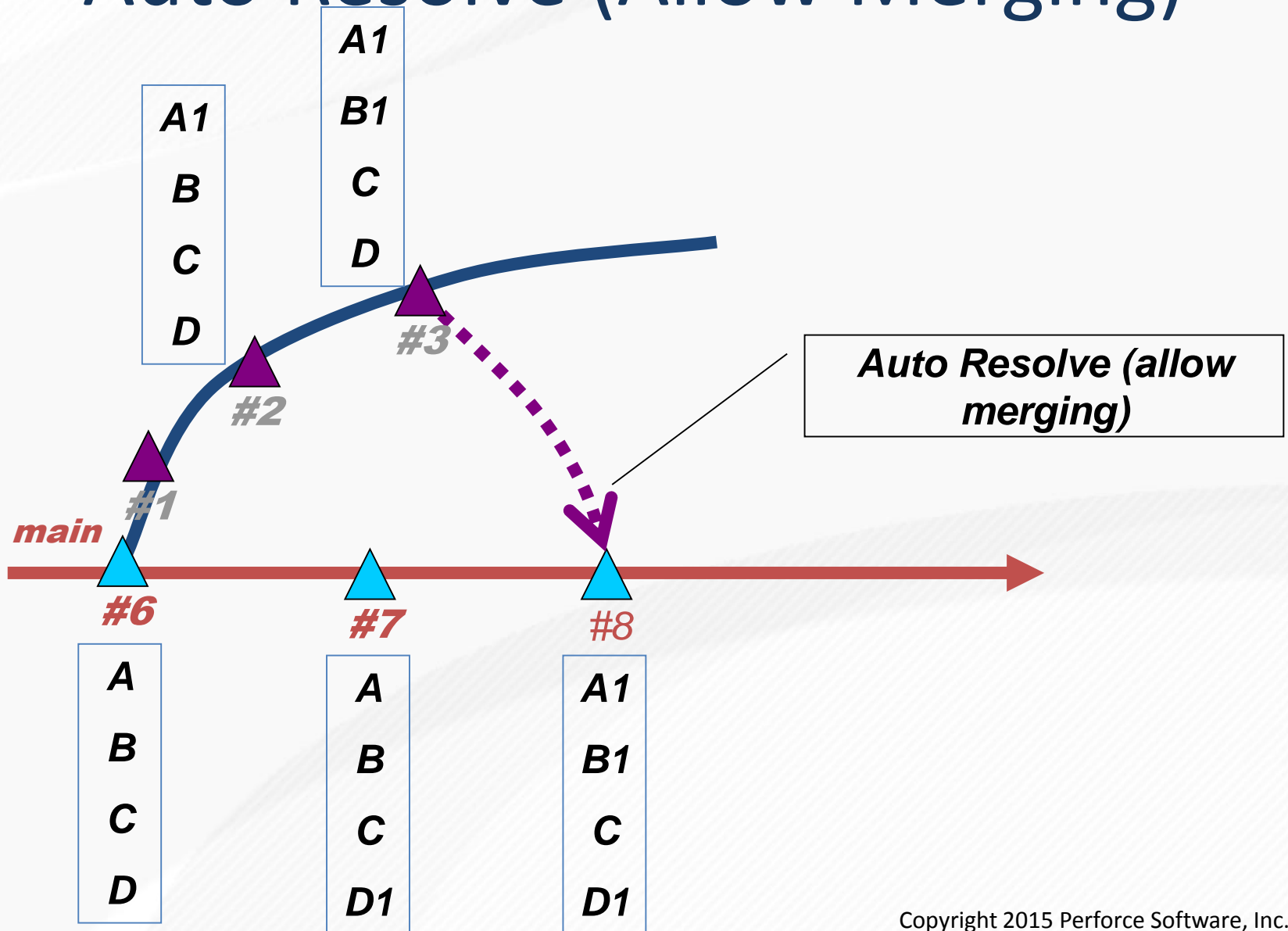
Resolve . . .

- Safe Automatic (no merging)
- Automatic (merging allowed)
- Interactive (conflicts present)

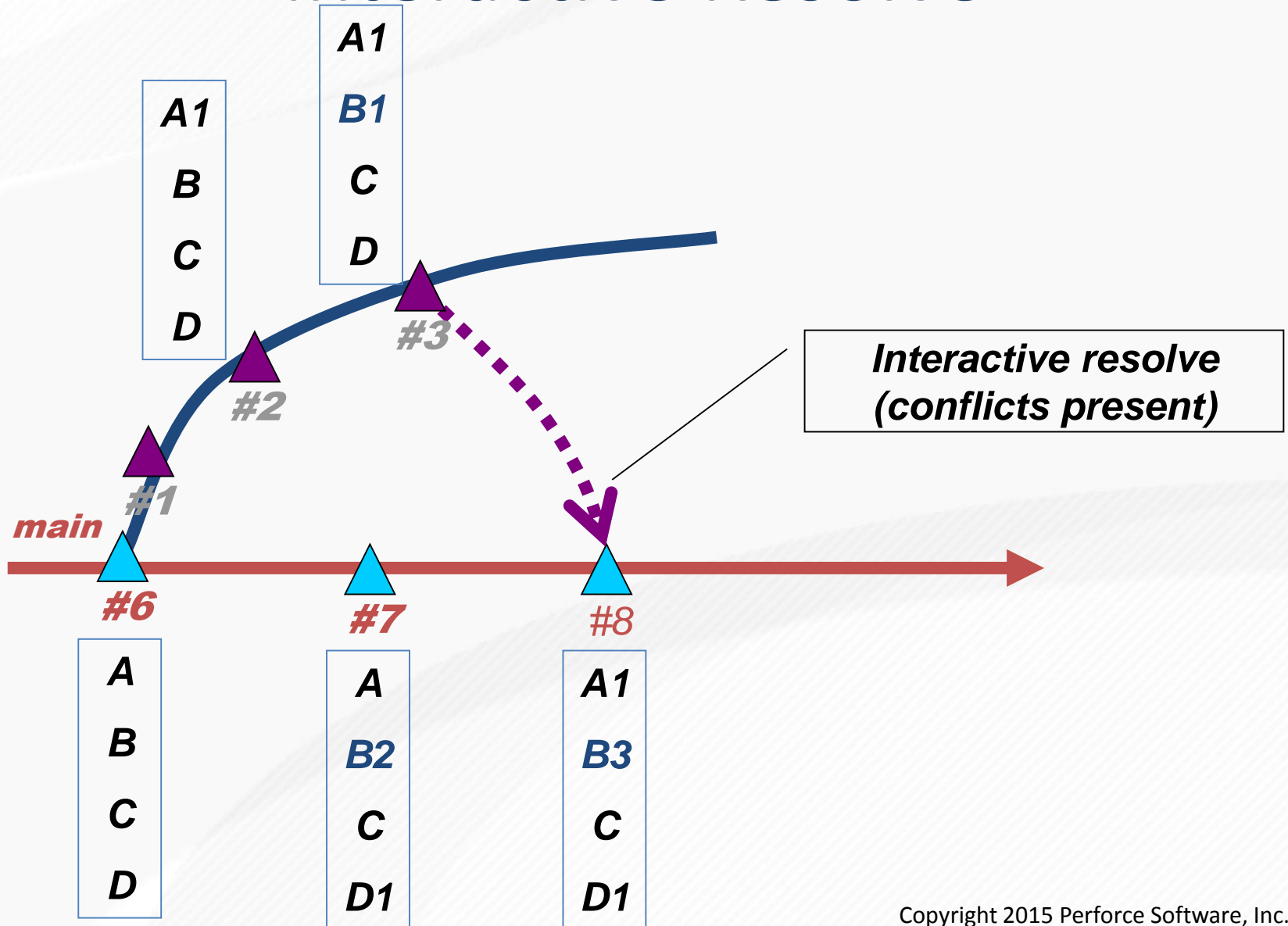
Auto Resolve (Safe – no merging)



Auto Resolve (Allow Merging)



Interactive Resolve



Automatic Resolve Options

Recommended every time

- Safe Automatic (50 - 80%)
- Automatic (35 - 15%)
- Interactive (15 - 0%)

Selective integrations

live

- Use any revision specification to limit the range of integration
- Integrate up to or only specific revisions
- Select **Advanced Options...** and store selections

New Actions in this Chapter, Part 2

- Context click → Branch Files → Specify target
- Context click → Merge/Integrate → Specify target
- Context click → Merge/Integrate → Save Branch Mapping
- Context click → Merge/Integrate → Select branch mapping
- Resolve...
- Context click →
 - Integrate... →
 - Specify source and target files →
 - Limit the range of the integration:
 - Advanced Options

Introduction to Perforce for Users

Using Streams

Streams

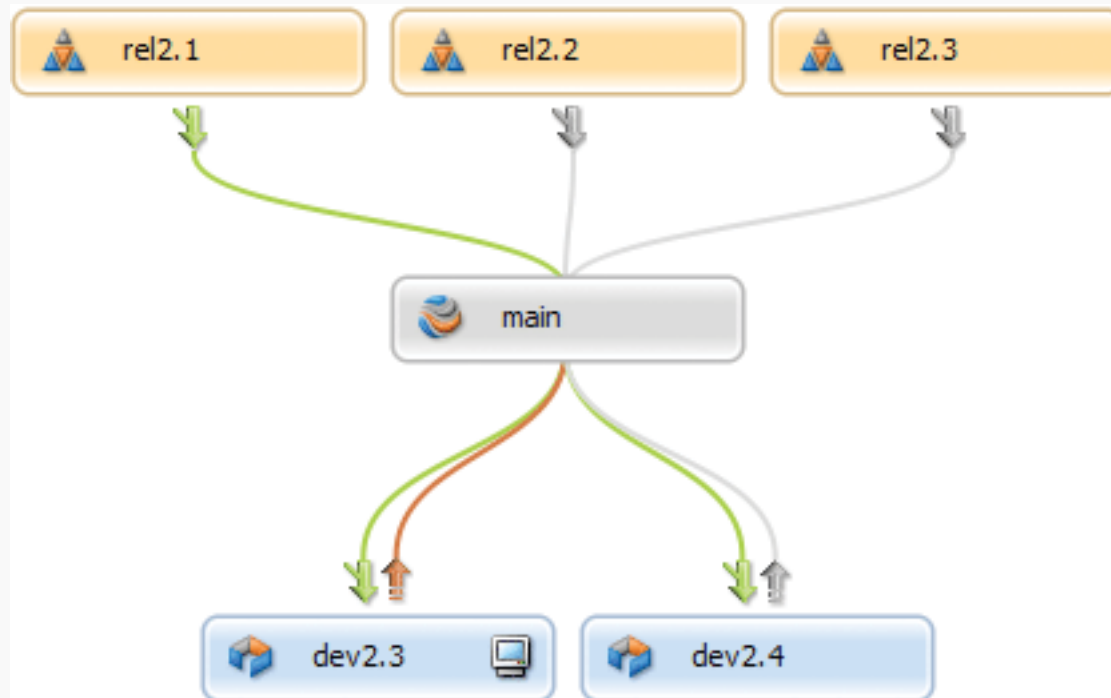
- Introduction
- Creating a stream
- Populating a stream
- Working in a stream
- Streams views

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, typically no copy ‘up’)

‘mainline’

Stable per your policy

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

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

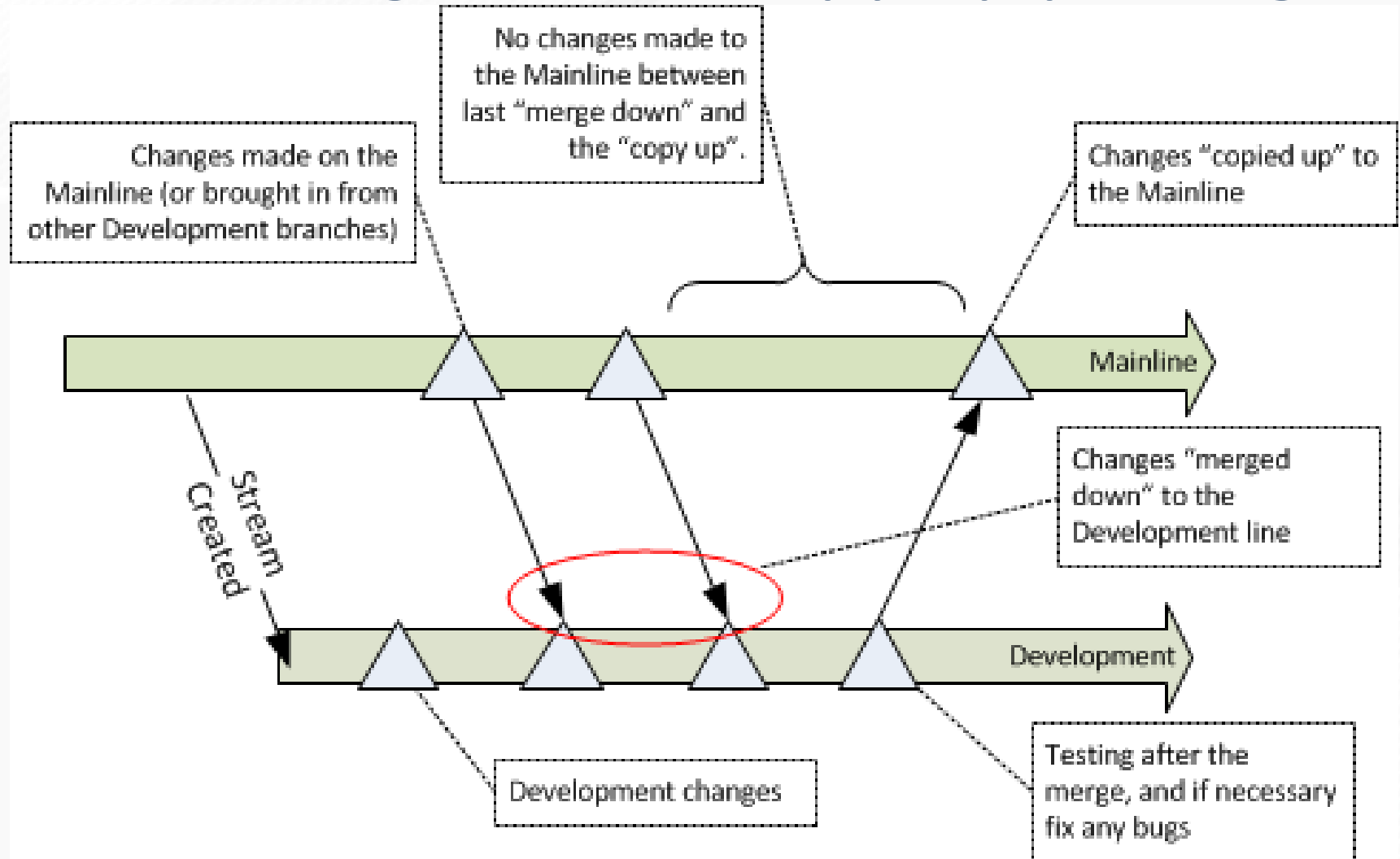
‘development’

Unstable

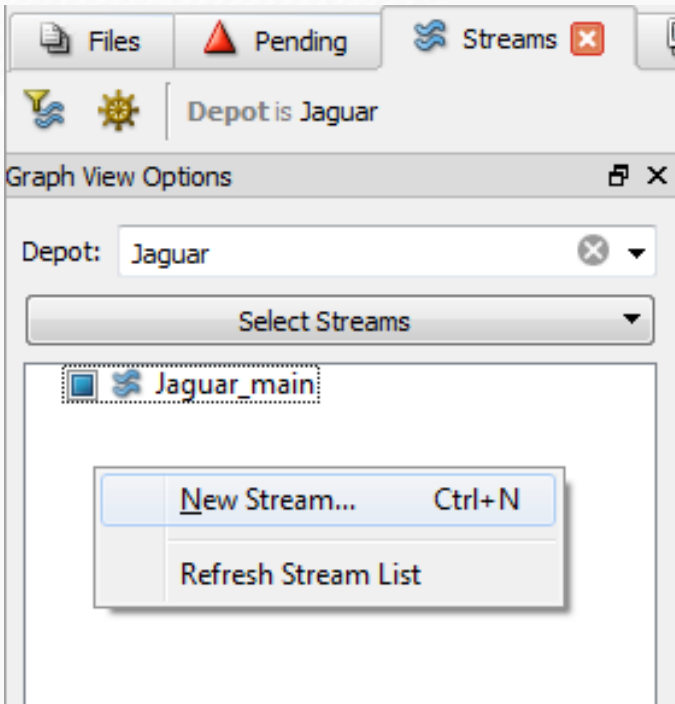
(merge ‘down’ from parent, copy ‘up’ to parent)



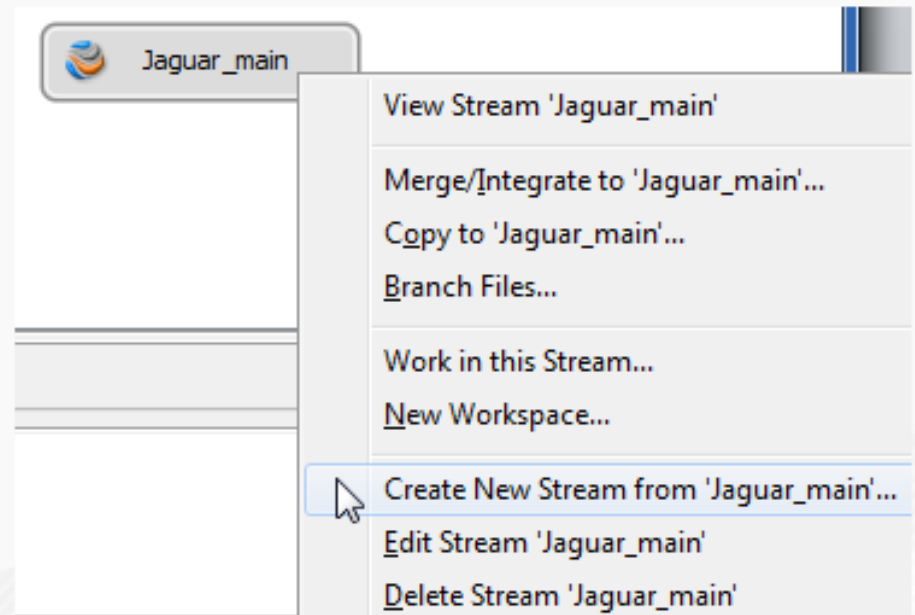
The Merge-down, Copy-up paradigm



Creating a stream



Right click
options to
create a new
stream



Stream workspaces

Workspace: New (1666, bruno)

Basic Advanced

Workspace name:


Workspace root: Browse...

Stream: X Browse...

Stream at change:

▼ Workspace Mappings:

To edit these mappings, you must edit the stream's path configu ✓✕✎

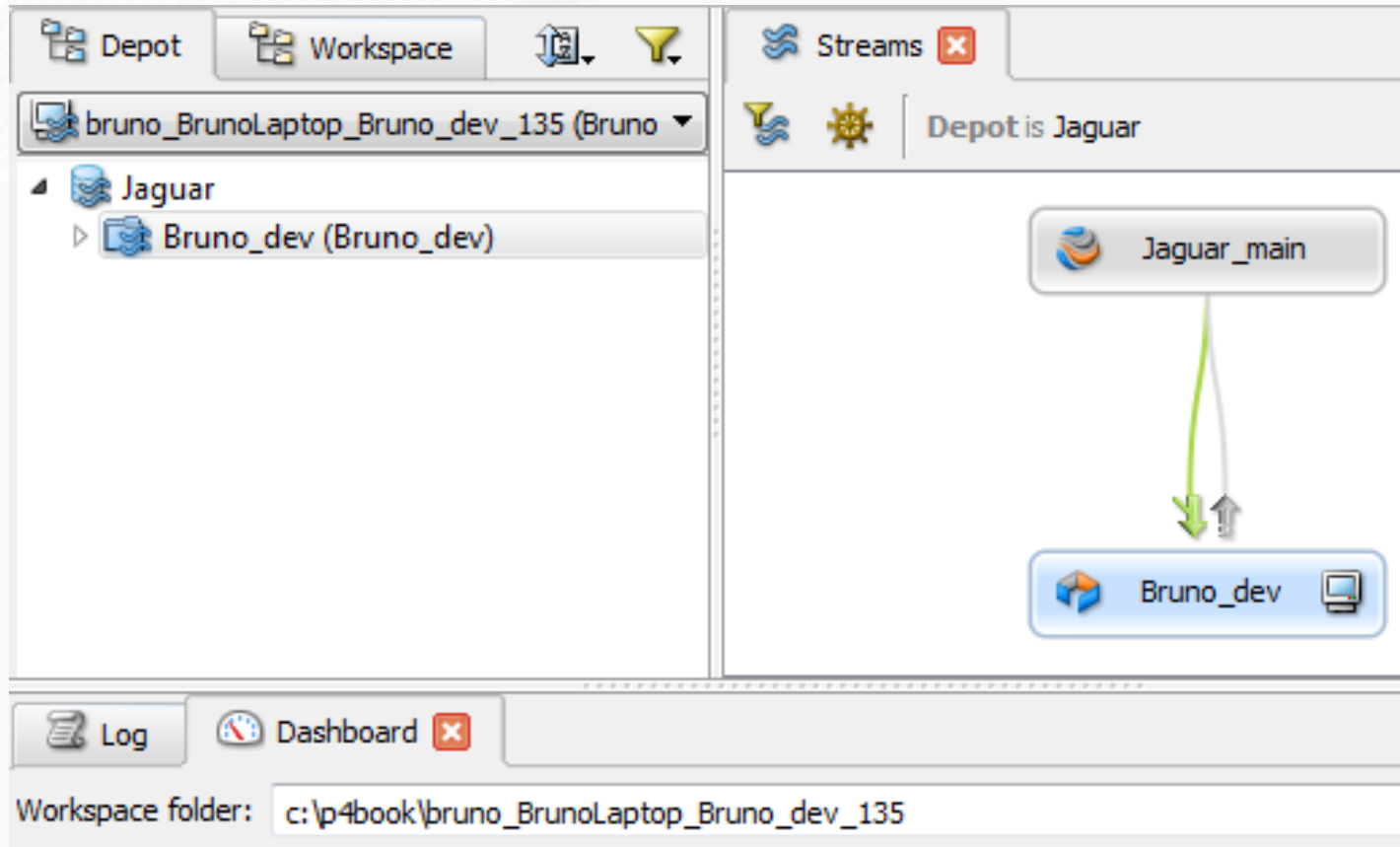
Depot Tree	Client Expression
▶  Jaguar	

☒ Switch to new workspace immediately

☐ Automatically get all revisions

OK Cancel Help

Changes requiring propagation



Depot Workspace

bruno_BrunoLaptop_Bruno_dev_135 (Bruno)

Jaguar

Bruno_dev (Bruno_dev)

Streams ×

Depot is Jaguar

Jaguar_main

Bruno_dev

Log Dashboard ×

Workspace folder: c:\p4book\bruno_BrunoLaptop_Bruno_dev_135

Dashboard last updated: 23/06/2013 15:34:24 [Update now](#)

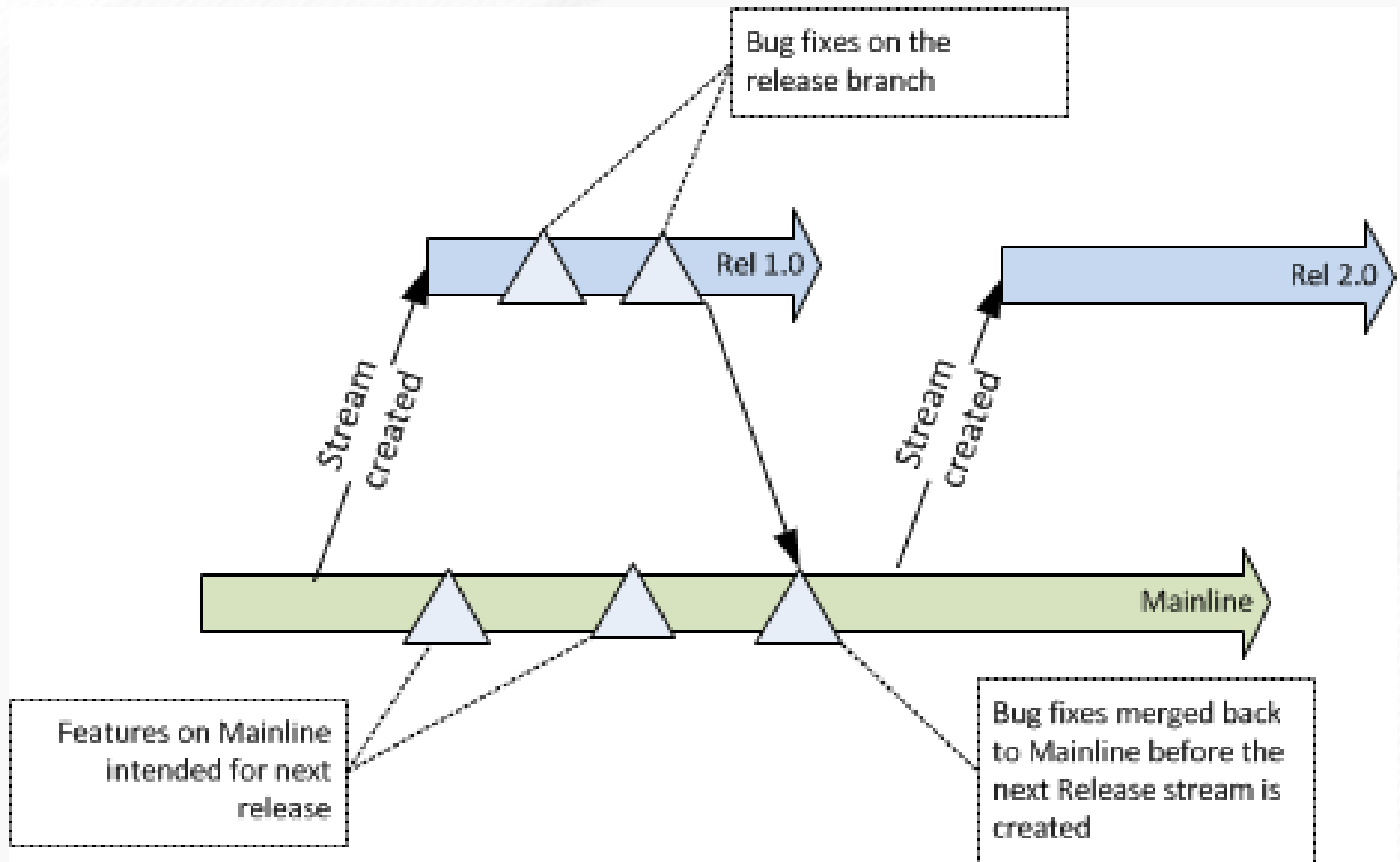
Tasks:

 [Merge from Jaguar_main \(1 change\)](#)

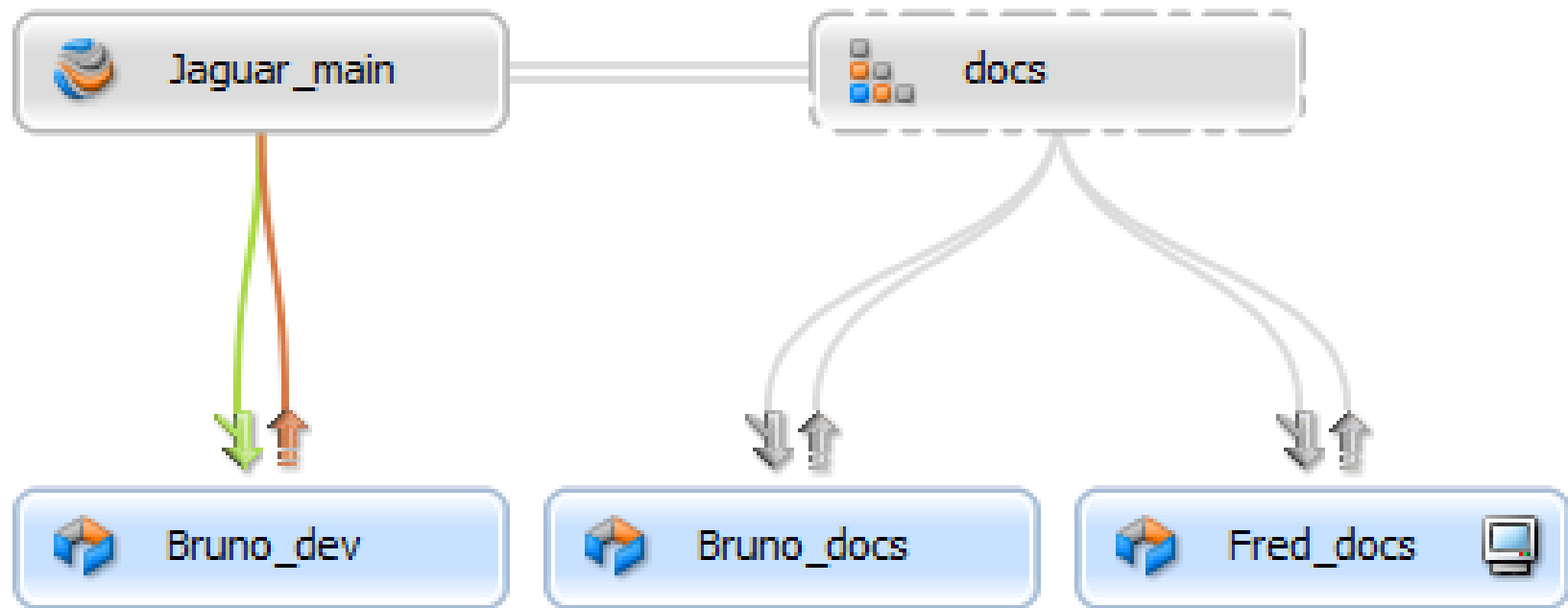
Folder status: All files are at the latest revision

Changelist	Potential Conflict	Da
...	-	

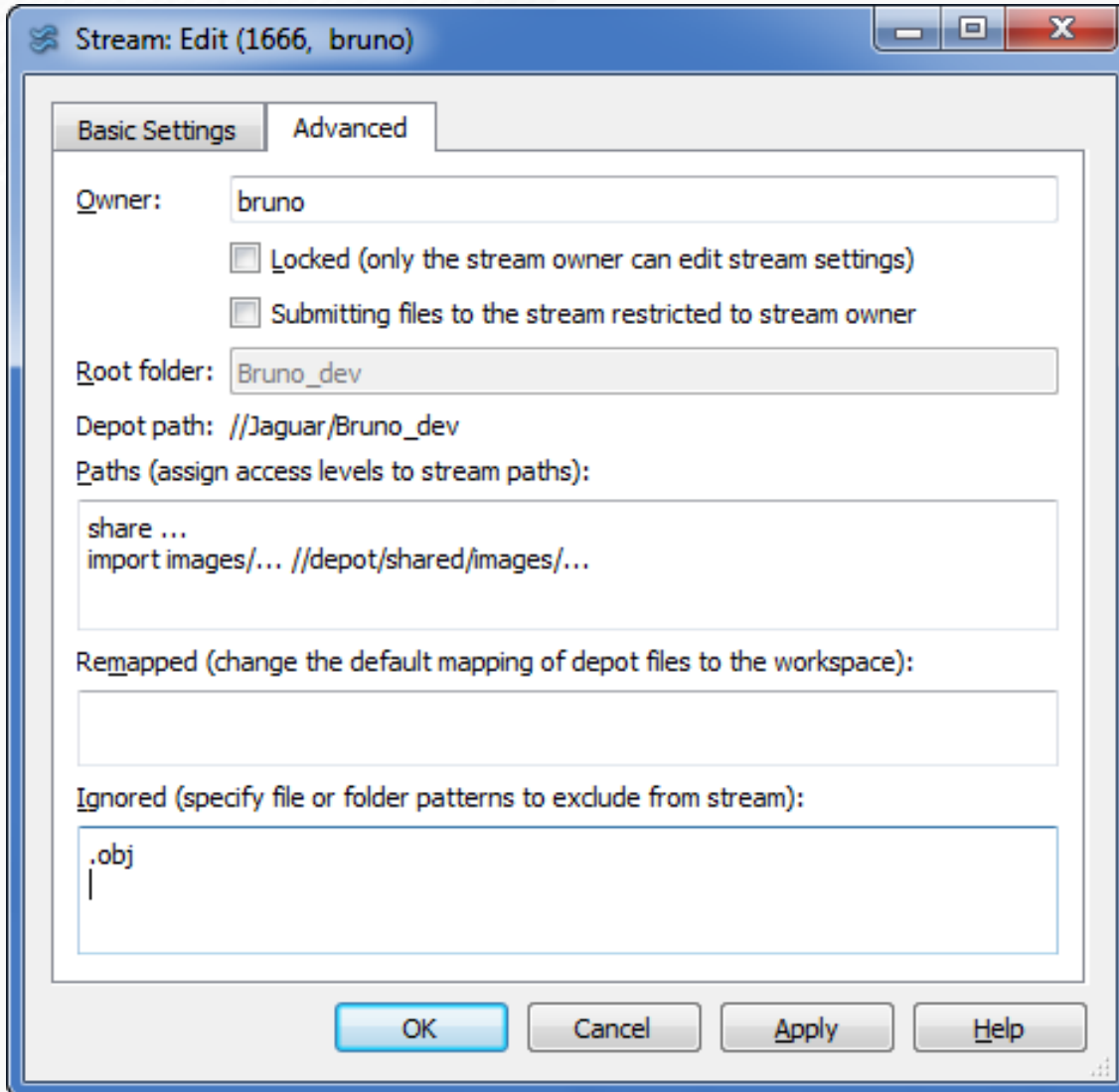
Release streams



Virtual streams



Stream details



Stream: Edit (1666, bruno)

Basic Settings Advanced

Owner:

☐ Locked (only the stream owner can edit stream settings)

☐ Submitting files to the stream restricted to stream owner

Root folder:

Depot path:

Paths (assign access levels to stream paths):

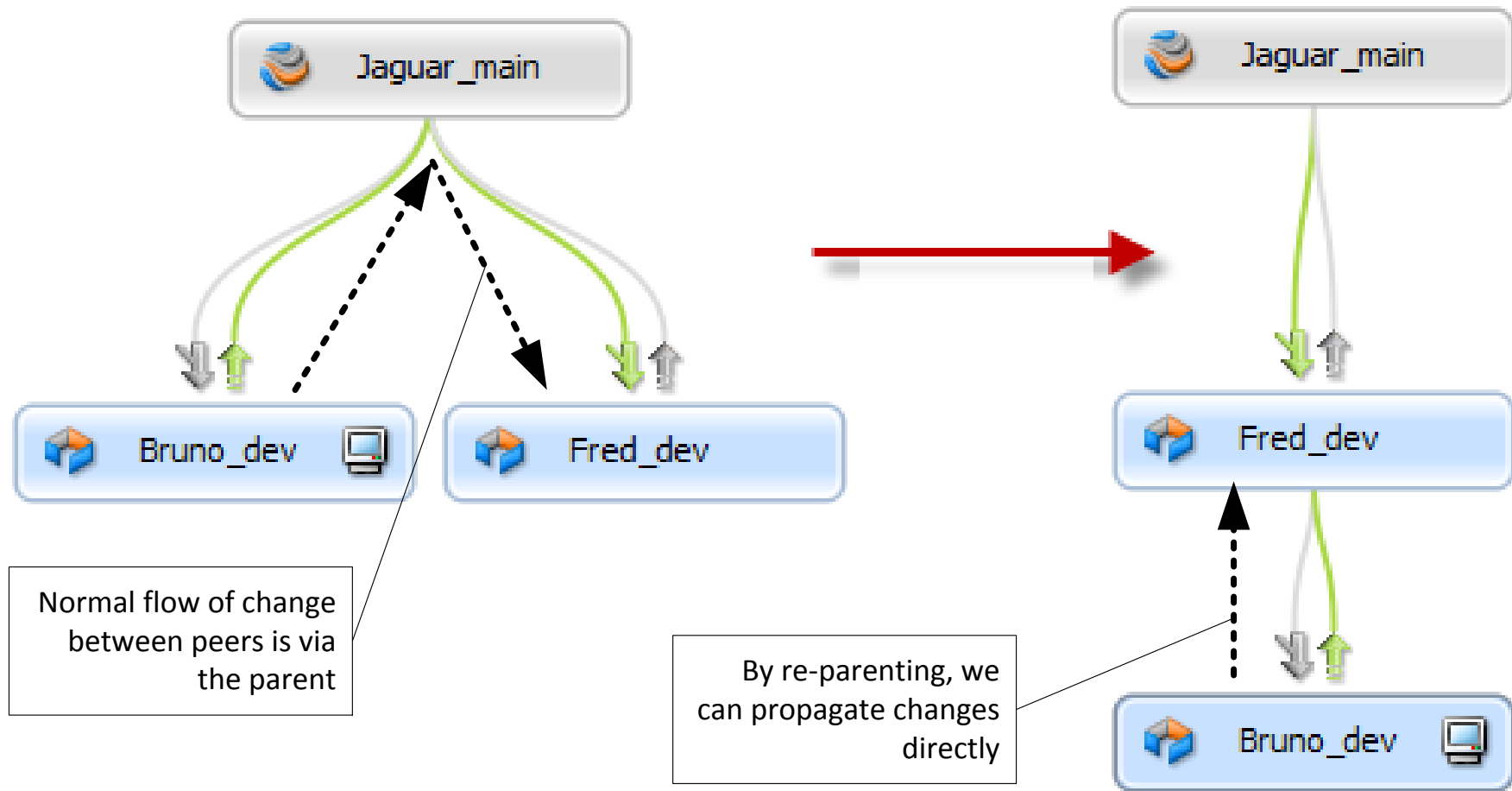
import images/... //depot/shared/images/...

Remapped (change the default mapping of depot files to the workspace):

Ignored (specify file or folder patterns to exclude from stream):

OK Cancel Apply Help

Propagating changes across hierarchy



New Actions in this Chapter

- **File → New → Stream**
- **Merge/Integrate**
- **Copy**

Introduction to Perforce for Users

Labels

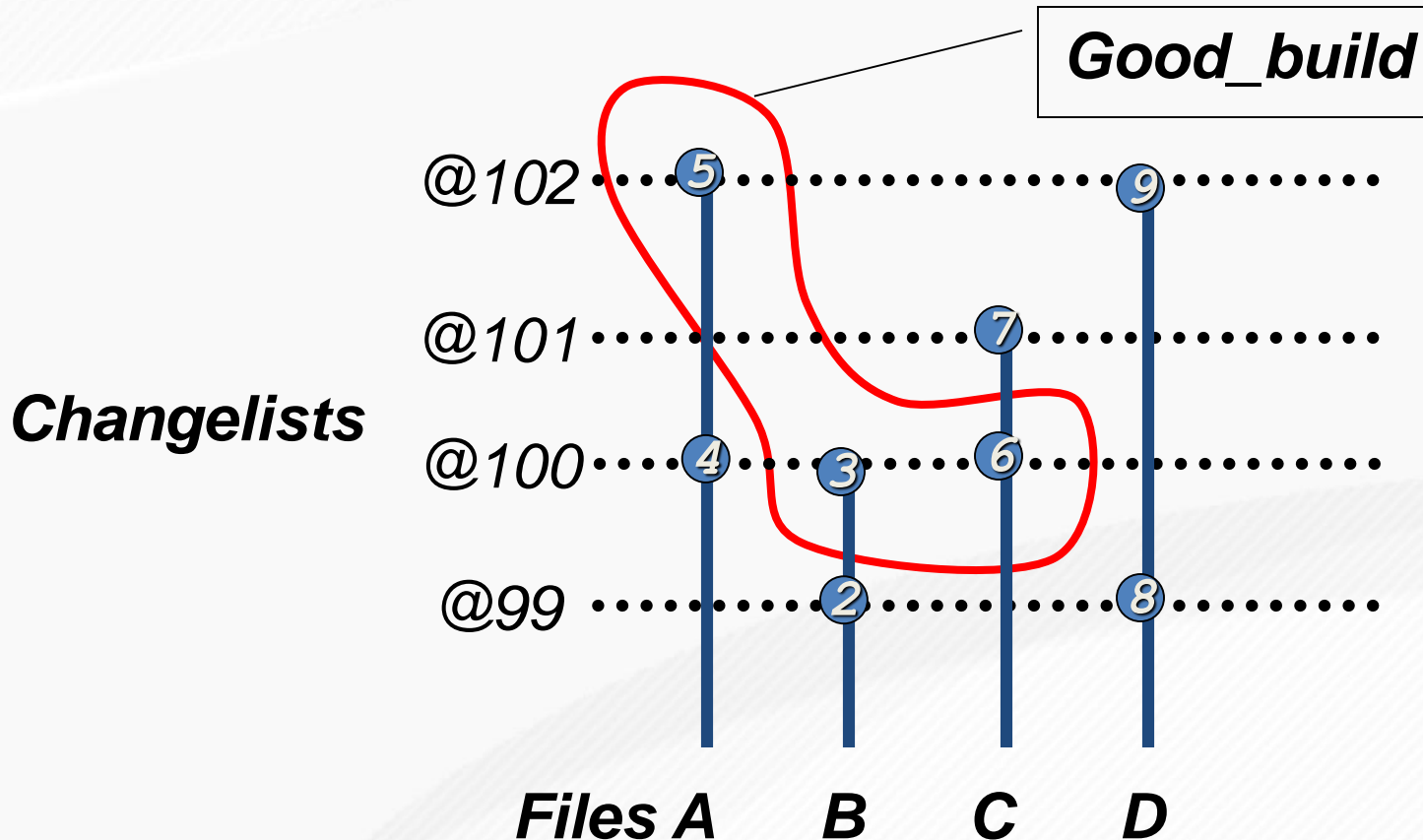
About Labels

- Labels vs. Changelist Numbers
- Creating a Label
- Tagging Files With a Label
- Locking a Label

Labels vs. changelist numbers

- *User-determined file set*
- *Can be changed*
- *Meaningful names*

Labels – cutting across changelists



Creating a label specification

live

Labels

- Add a description
- Select which files or paths can be tagged
- Can exclude files or paths

Tagging files with a label*live*

- Tag files using a revision specification
- Tag workspace file revisions
- Display the list of files tagged by a label

Use automatic labels

live

- Less data stored on your server
- Create a mnemonic between a changelist and a label name.

Updating labels

live

- A label can be removed from files
- A label can be updated with different file revisions
- A label can be updated with additional files

Getting tagged file revisions

live

- Examine the list of files tagged by the label
- Check workspace view
- **Get Revision...** and use the label name to specify the revisions

Showing labels

live

- Show all label specifications
- Filter by:
 - Folder or file name
 - Owner
 - String in label name

Locking a label

live

- Prevent changing the list of files tagged by a label
- The owner can unlock a label

New Actions in this Chapter

- `File → New → Label`
- `Context click →`
 - `Label →`
 - `Apply selected label to files →`
 - `Add a revision specifier`
 - `Remove selected label from files`
 - `Get Revision`
 - `Show the List of Files`
 - `Select label → Edit Label <labelname>`
 - `View → Labels`

Introduction to Perforce for Users

Job Tracking

Job Tracking

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

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

live

Jobs

- Add the job description and save the form
- Edit to add more information

Listing jobs

live

- List all jobs
- Filter by:
 - Bookmark
 - Keywords
 - Folder or file name
 - Filters are stored for future use

Job search examples

```
filter email
```

```
status|email
```

```
status=open user=edk
```

```
date>=1998/07/14
```

```
description=filter*
```

```
r2.1 beta.exe doesn't
```

Linking jobs to changelists

live



Users

- Add to the Job View: filter
- Jobs added automatically to default changelist
- Add a job to any numbered changelist



Pending



Submitted

Job fix reporting

live

- View the job status in the **Jobs** pane
- View a job to list associated changelists
- View a changelist to list associated jobs

New Actions in this Chapter

- File → New → Job
- View →
 - Jobs →
 - Context click on job →
 - Edit
 - Linked changelists display
 - Enter search string
 - Pending/Submitted Changelists → Expand changelist to see jobs
- Connection → Edit Current User
- Context click → Edit Pending/Submitted Changelist <nnn> → Add/Browse for jobs

Introduction to Perforce for Users

Code Review - Swarm

Section Contents

- Background to code review
- Overview and details of Swarm

•205

Build quality in

- Find bugs early
- Share knowledge within the team
 - Code tends to be neater, better documented, and better organized
- Consistency in design and implementation
- Better security considerations

•206

Code Collaboration to the Rescue



- ***Better Code***

- ***Pain-free Integrations***

- ***Faster Deploys***

Perforce Swarm

- Comprehensive code collaboration platform
- Flexible, integrates with your tool chain
- Seamless interaction via Perforce or Git
- Built for continuous delivery



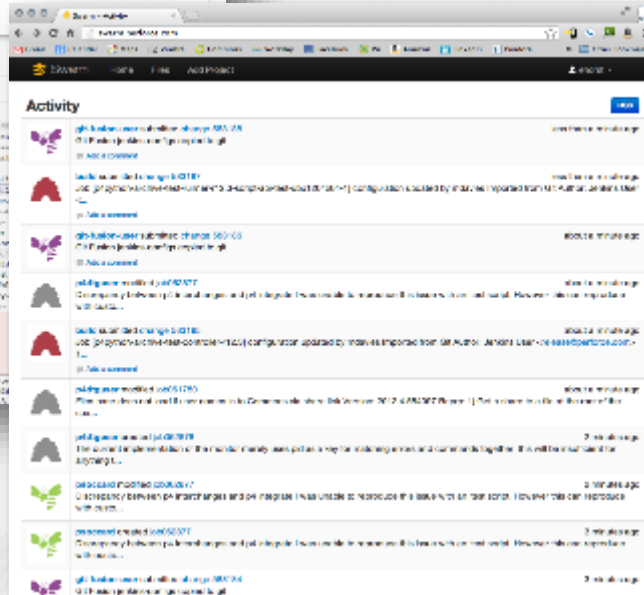
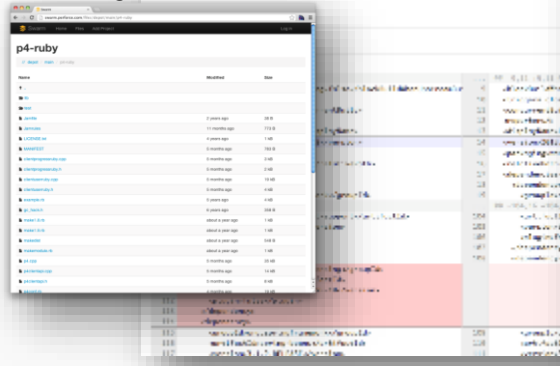
Swarm Highlights

•Code Review



•Projects

•Browsing



- Real-time code collaboration and review
- Inline comments and diff
- Integrates with rest of lifecycle
 - Bugs
 - Build
 - Code Analysis
- Supports Git and P4V users
- User Extensible
- Leverages Perforce versioning engine

Request a Code Review from within Perforce Versioning Engine...

•Request Review



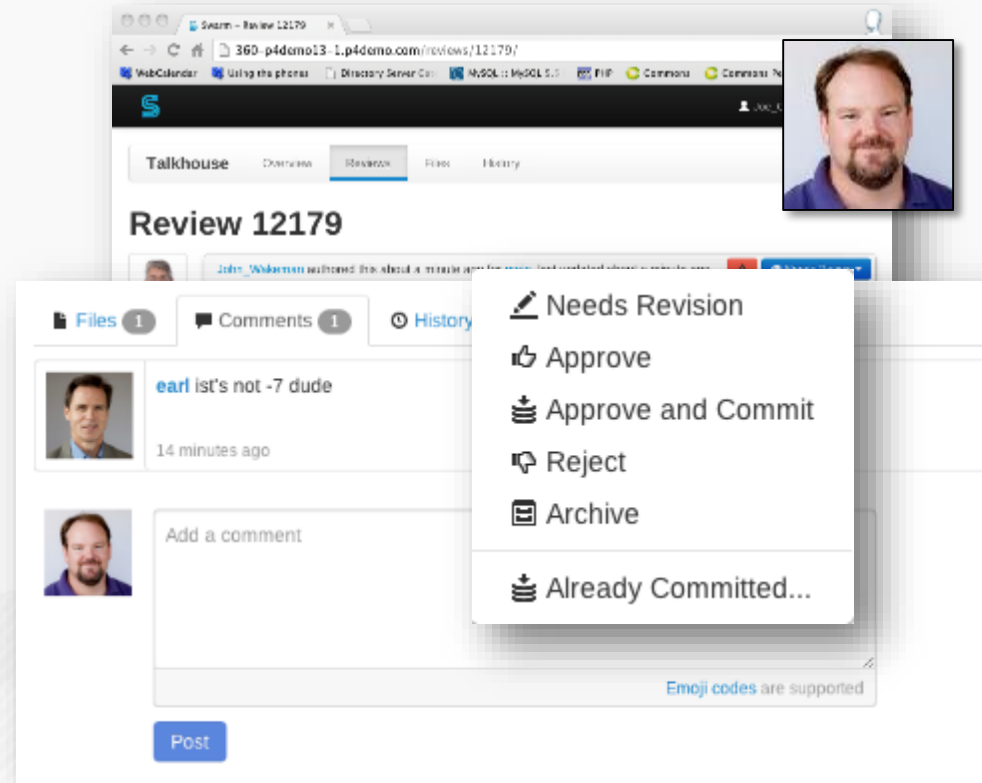
•Review Notification

From: John Wakeman <perforce@p4demo.com>
Subject: [Swarm] Review @12179 - @Joe_Coder please take a look
Date: September 26, 2013 1:00:22 PM PDT
To: joe_coder@360-p4demo13-1.p4demo.com , john_wakeman@360-p4demo13-1.p4demo.com
Reply-To: John Wakeman <john_wakeman@360-p4demo13-1.p4demo.com>

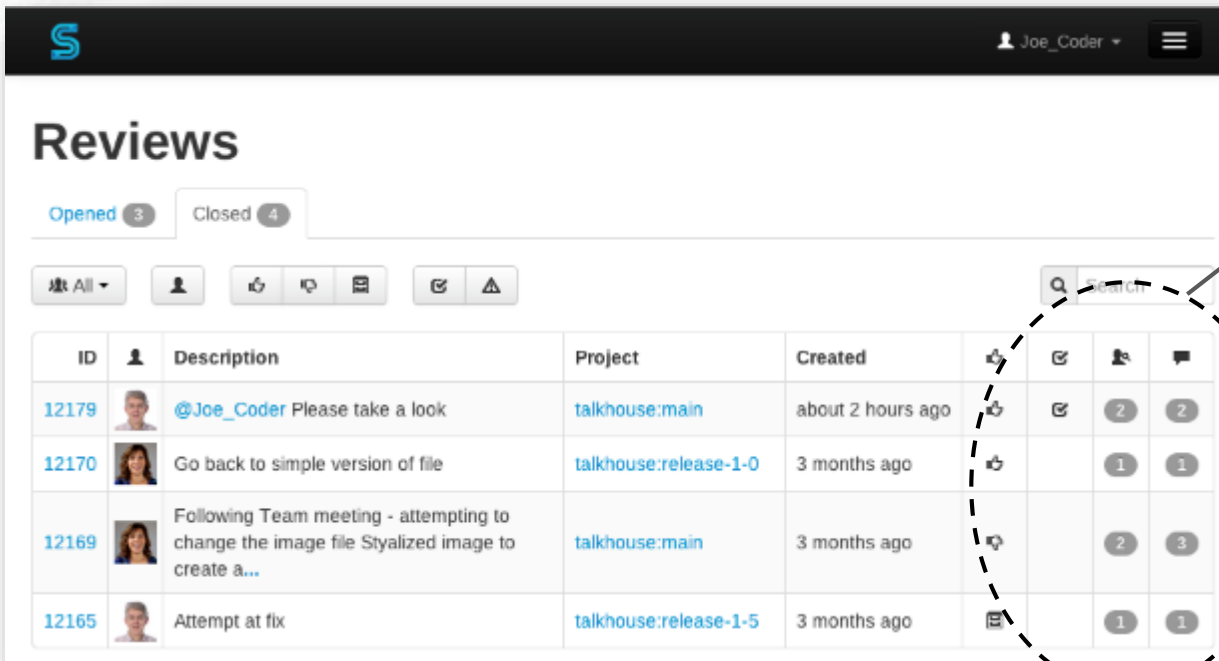
John_Wakeman requested review 12179 for talkhouse:main
 @Joe_Coder please take a look

Review, Comment, Approve...

- Link from email or within Swarm
- Comments
 - Inline in the “diff” view
 - See other comments
- Approve, Reject, etc.
 - Notification of status change



Status Dashboard



Reviews

Opened 3 Closed 4

All [User] [Like] [Dislike] [Comment] [Checkmark] [Warning]

Search

ID	Person	Description	Project	Created	Like	Dislike	Comment	Checkmark	Warning
12179	@Joe_Coder	Please take a look	talkhouse:main	about 2 hours ago	+	-	2	2	
12170		Go back to simple version of file	talkhouse:release-1-0	3 months ago	+	-	1	1	
12169		Following Team meeting - attempting to change the image file Stylized image to create a...	talkhouse:main	3 months ago	+	-	2	3	
12165		Attempt at fix	talkhouse:release-1-5	3 months ago	+	-	1	1	

- Review Status
- Results from other tools
 - Build
 - Code analysis
 - QA
- Ability to Test Run appropriate projects
- Everything is Clickable
 - Links to other tools

Finally...

Visit Perforce

<http://www.perforce.com>

and our Public Depot

<http://wiki.workshop.perforce.com>

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>