Perforce User Conference 2010

Shelf Life

Shelving in Perforce

Sven Erik Knop, Perforce Software

What’s in store?

- What is shelving?
- How does it work?
- Keeping track of shelves
- Use cases
  - Backup of modified files
  - Swap out current changes for something else
  - Code review
  - Transfer work to another workspace

What is shelving?

- You have opened files in your workspace
- You need to store a copy of your local changes
- Instead of creating a local copy:
    - Shelve the changes on the Perforce server
    - Shelved files can be accessed by other workspaces
    - Unshelve files locally to restore changes
    - Unshelve files into a different workspace to view or transfer changes

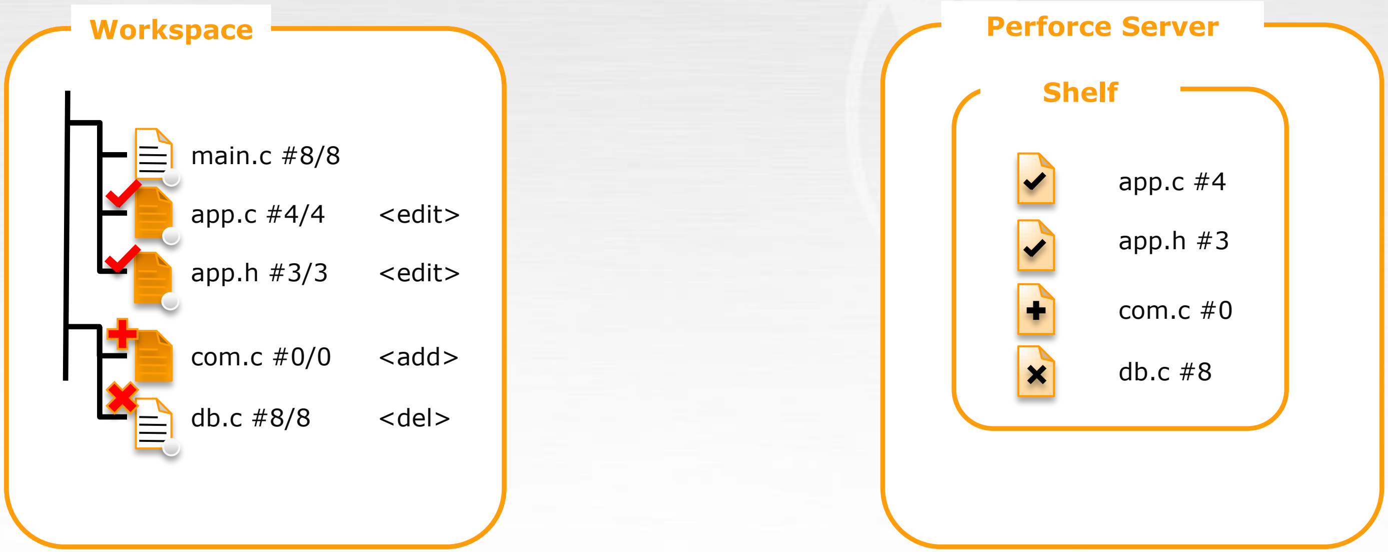
How does it work?

- A *shelf* in Perforce is a changelist
  - Internal status is “shelved”

- Changed files are transferred to the server
  - Stored as compressed single files in the depot

- Content of shelved files can be accessed
  - Syntax: ‘@=‘ and the shelve list number

Shelving



Workspace

main.c #8/8

app.c #4/4 <edit>

app.h #3/3 <edit>

com.c #0/0 <add>

db.c #8/8 <del>

Perforce Server

Shelf

app.c #4

app.h #3

com.c #0

db.c #8

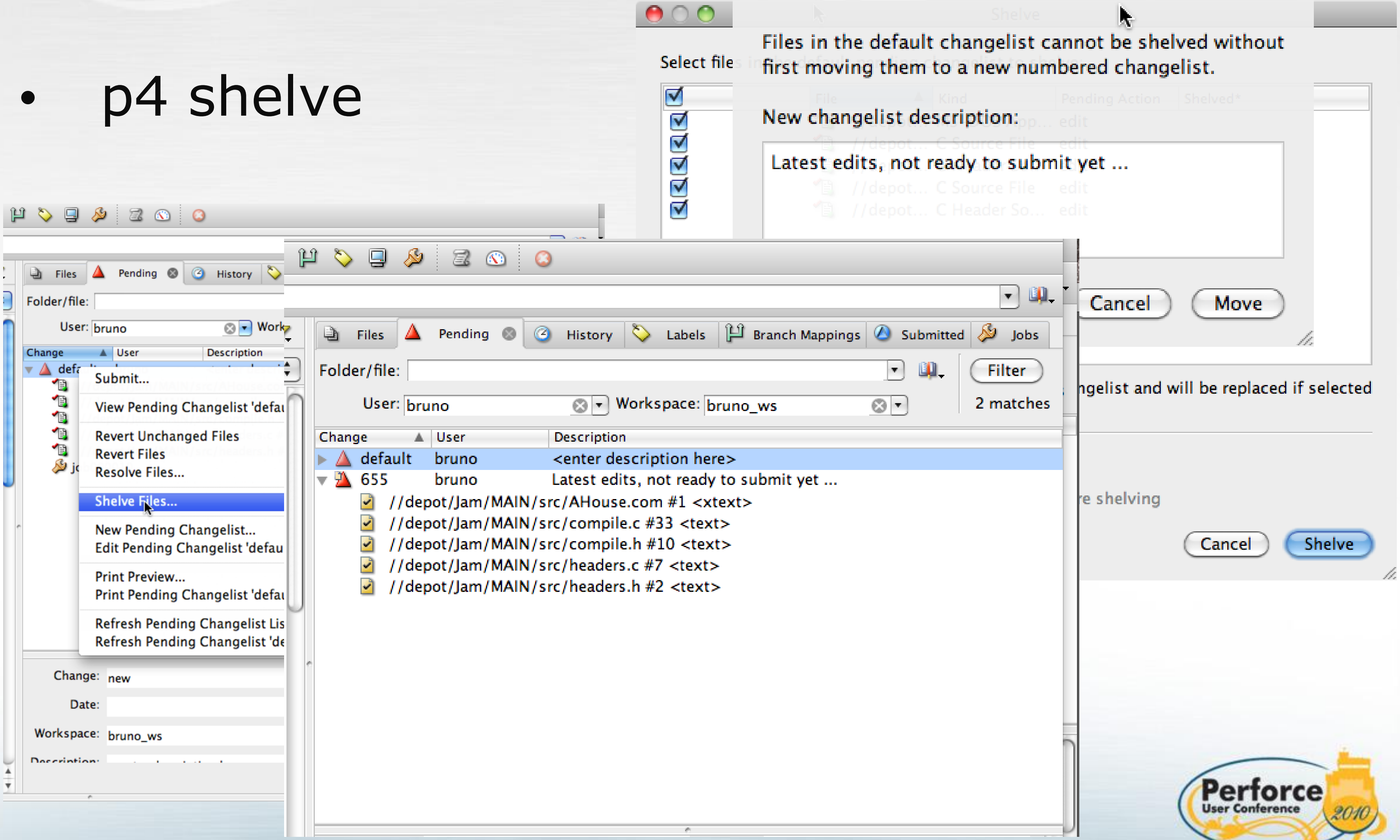
Open files

Shelve changes

Revert workspace (Optional)

Shelving commands

- p4 shelve



Files in the default changelist cannot be shelved without first moving them to a new numbered changelist.

New changelist description:

Latest edits, not ready to submit yet ...

Shelve
Select file
Cancel
Move
ngelist and will be replaced if selected
re shelving
Cancel
Shelve
Files
Pending
History
Labels
Branch Mappings
Submitted
Jobs
Filter
Submit...
View Pending Changelist 'defau
Revert Unchanged Files
Revert Files
Resolve Files...
Shelve Files...
New Pending Changelist...
Edit Pending Changelist 'defau
Print Preview...
Print Pending Changelist 'defau
Refresh Pending Changelist Lis
Refresh Pending Changelist 'de

Folder/file:

User: bruno Workspace: bruno_ws 2 matches

Change	User	Description
default	bruno	<enter description here>
655	bruno	Latest edits, not ready to submit yet ...

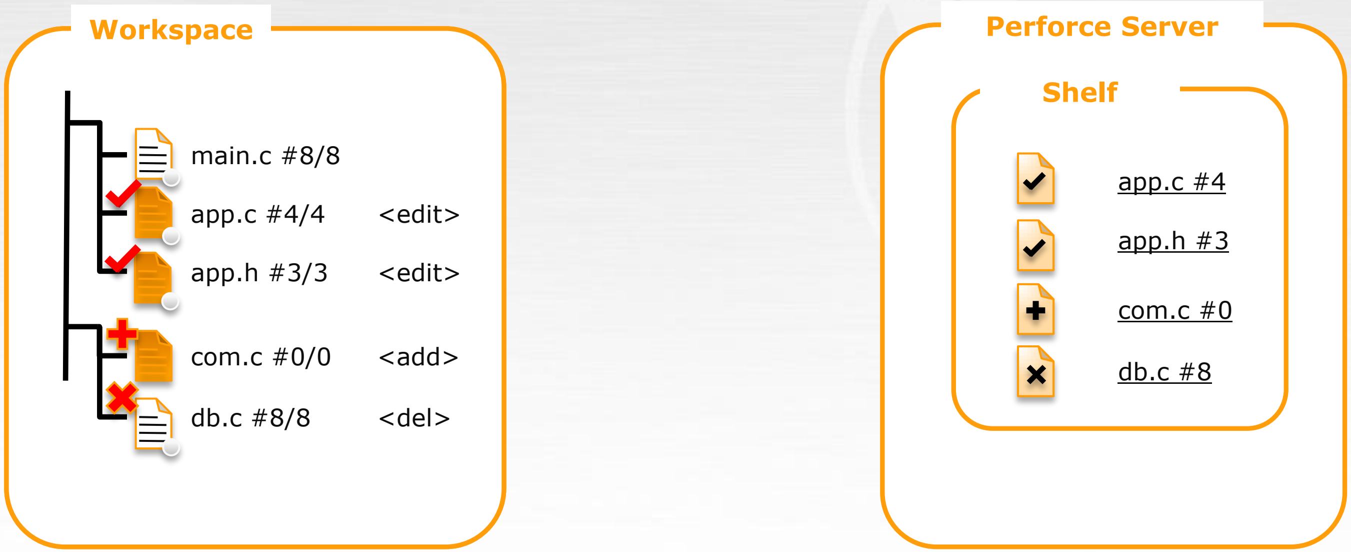
```
☑ //depot/Jam/MAIN/src/AHouse.com #1 <xtext>
☑ //depot/Jam/MAIN/src/compile.c #33 <text>
☑ //depot/Jam/MAIN/src/compile.h #10 <text>
☑ //depot/Jam/MAIN/src/headers.c #7 <text>
☑ //depot/Jam/MAIN/src/headers.h #2 <text>
```

Change: new

Date:

Workspace: bruno_ws

Unshelving



Workspace

main.c #8/8	
app.c #4/4	<edit>
app.h #3/3	<edit>
com.c #0/0	<add>
db.c #8/8	<del>

Perforce Server

Shelf

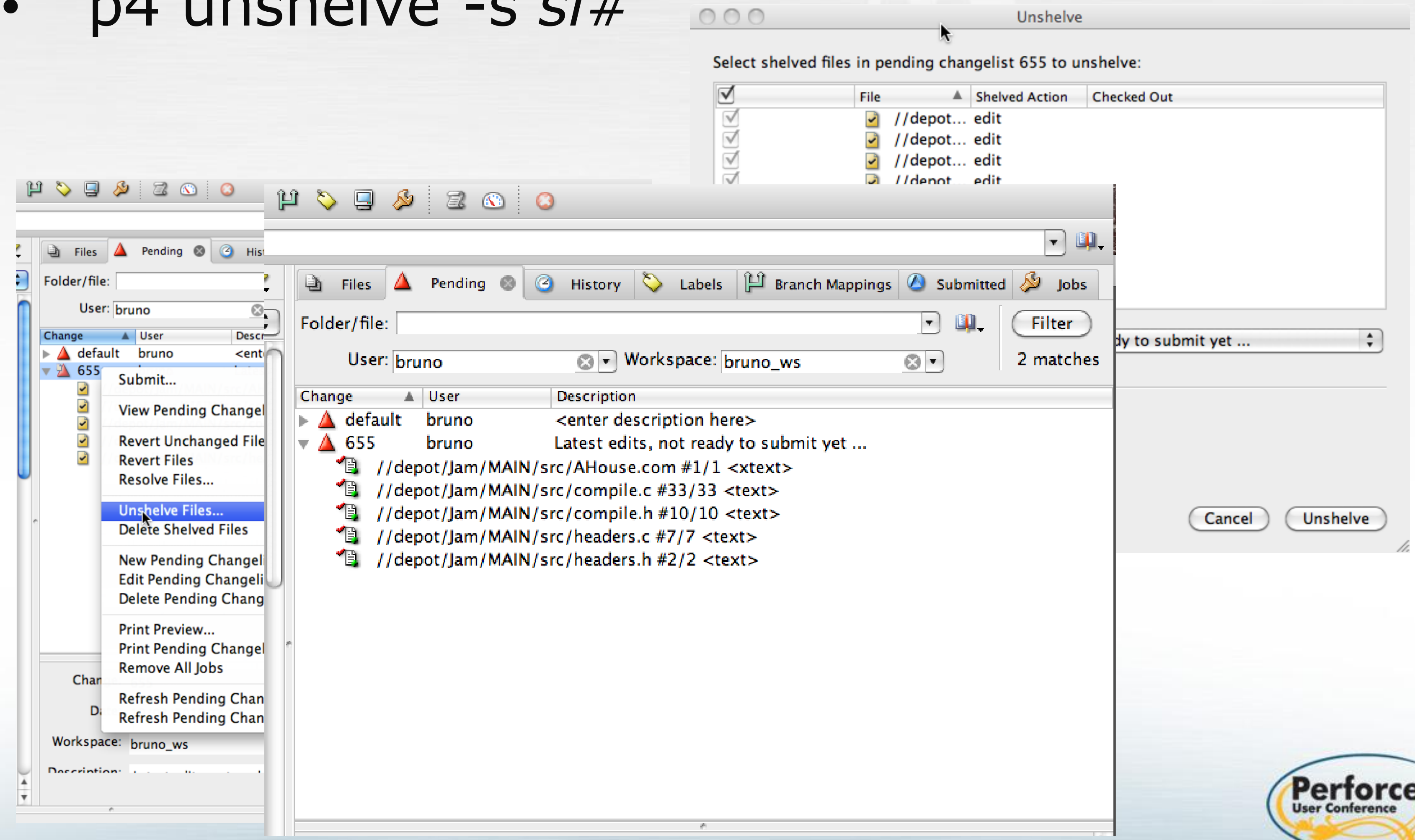
- ✓ app.c #4
- ✓ app.h #3
- + com.c #0
- ✗ db.c #8

Unshelve files

Delete shelf (optional: “p4 shelve -d”)

Unshelving commands

- p4 unshelve -s *sl#*



Unshelve

Select shelved files in pending changelist 655 to unshelve:

☑	File	Shelved Action	Checked Out
☑	//depot...	edit	
☑	//depot...	edit	
☑	//depot...	edit	
☑	//depot...	edit	

Folder/file:

User: bruno

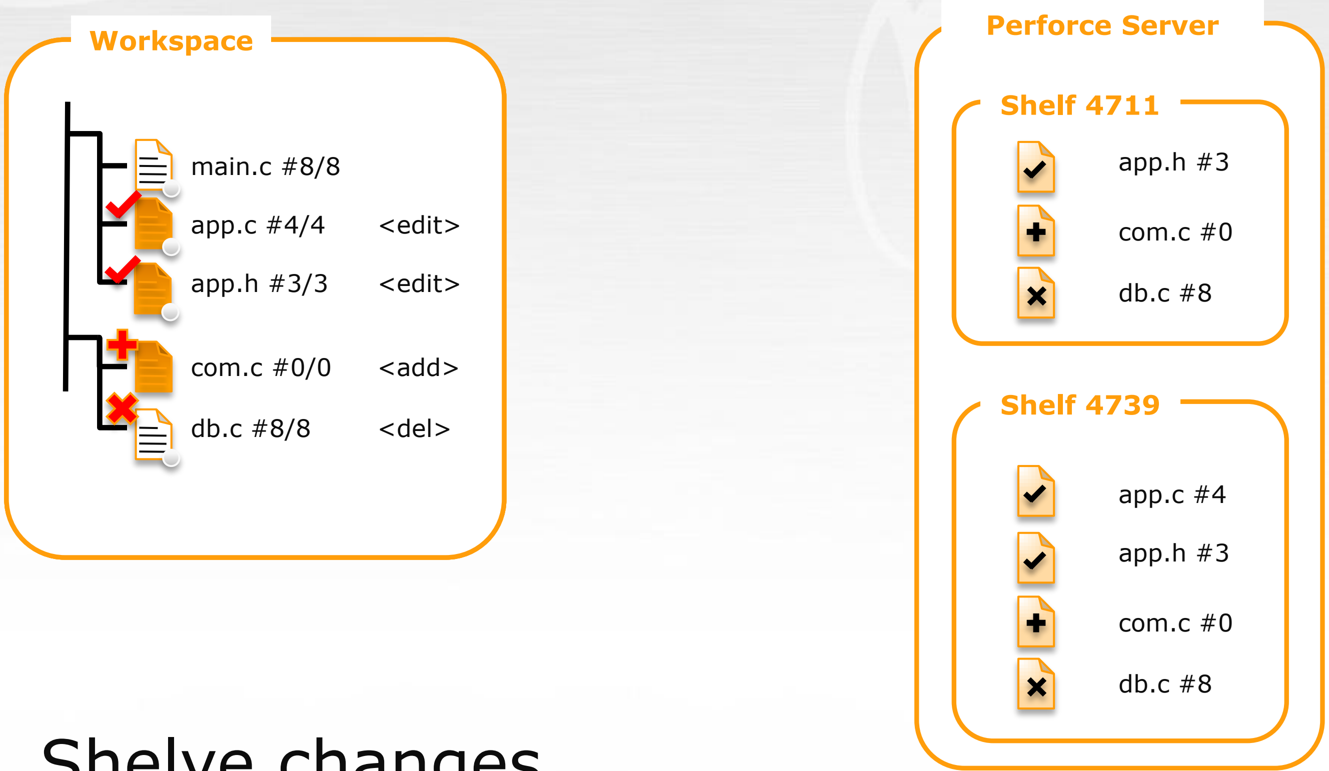
Workspace: bruno_ws

2 matches

Change	User	Description
default	bruno	<enter description here>
655	bruno	Latest edits, not ready to submit yet ...

```
//depot/Jam/MAIN/src/AHouse.com #1/1 <xtext>
//depot/Jam/MAIN/src/compile.c #33/33 <text>
//depot/Jam/MAIN/src/compile.h #10/10 <text>
//depot/Jam/MAIN/src/headers.c #7/7 <text>
//depot/Jam/MAIN/src/headers.h #2/2 <text>
```

Multiple shelves



Workspace

main.c #8/8

✓ app.c #4/4 <edit>

✓ app.h #3/3 <edit>

+ com.c #0/0 <add>

✗ db.c #8/8 <del>

Perforce Server

Shelf 4711

✓ app.h #3

+ com.c #0

✗ db.c #8

Shelf 4739

✓ app.c #4

✓ app.h #3

+ com.c #0

✗ db.c #8

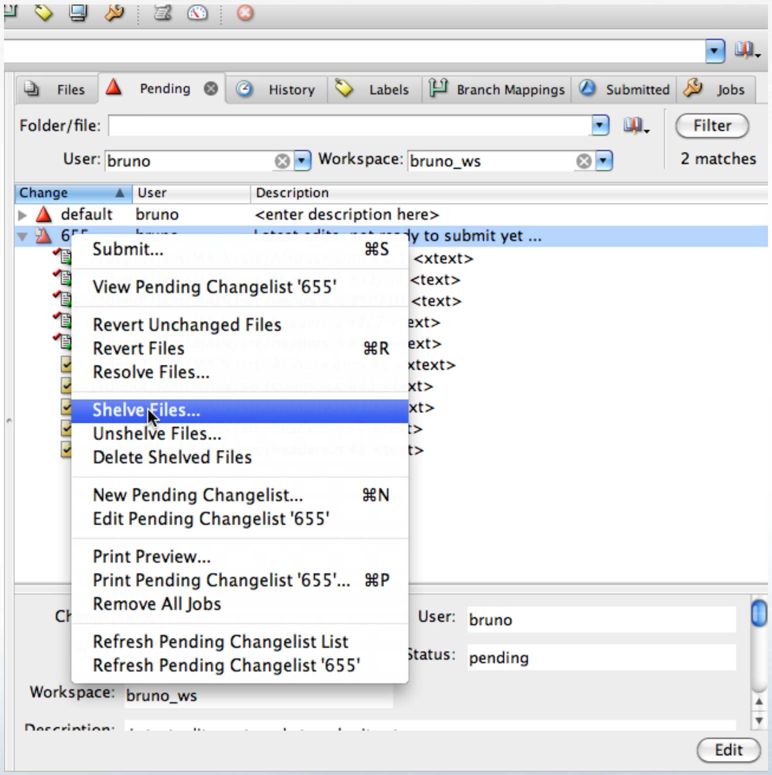
Shelve changes

Shelve further changes on different shelve

Restore previous shelf

Replacing files on a shelf

- p4 shelve -c *cl#* -f



Folder/file:

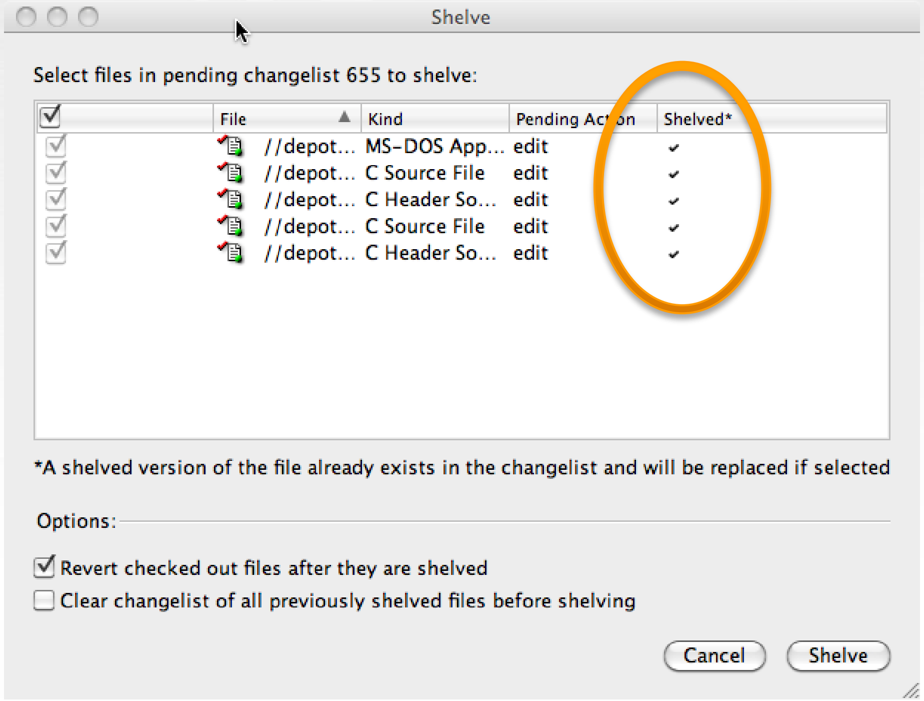
User: bruno Workspace: bruno_ws 2 matches

Change	User	Description
default	bruno	<enter description here>
655	bruno	Latest edits, not ready to submit yet ...

User: bruno

Status: pending

Workspace: bruno_ws



Select files in pending changelist 655 to shelve:

☑	File	Kind	Pending Action	Shelved*
☑	//depot...	MS-DOS App...	edit	✓
☑	//depot...	C Source File	edit	✓
☑	//depot...	C Header So...	edit	✓
☑	//depot...	C Source File	edit	✓
☑	//depot...	C Header So...	edit	✓

*A shelved version of the file already exists in the changelist and will be replaced if selected

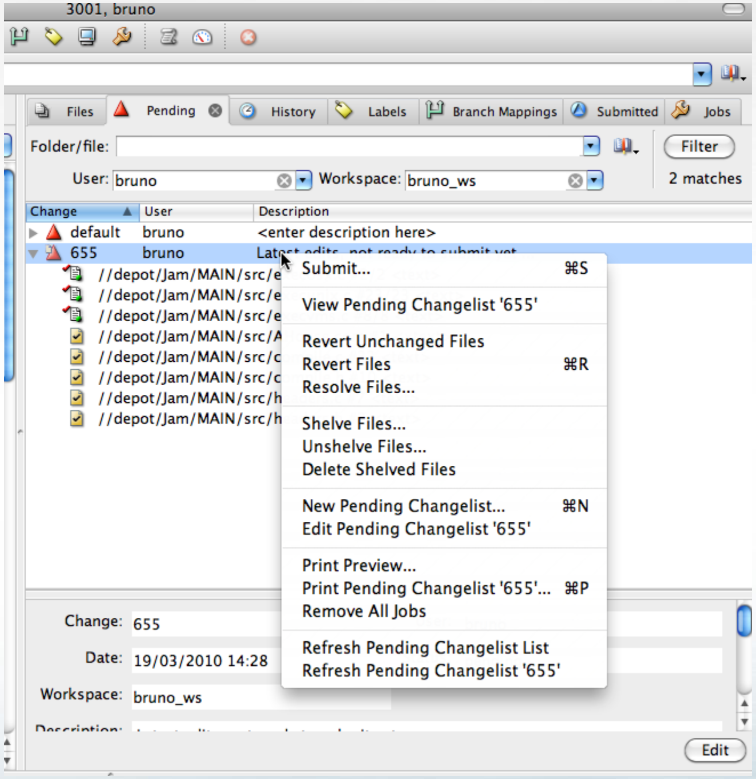
Options:

☑ Revert checked out files after they are shelved

□ Clear changelist of all previously shelved files before shelving

Replacing a whole shelf

- p4 shelve -c *cl#* -r



Folder/file:

User: bruno Workspace: bruno_ws

2 matches

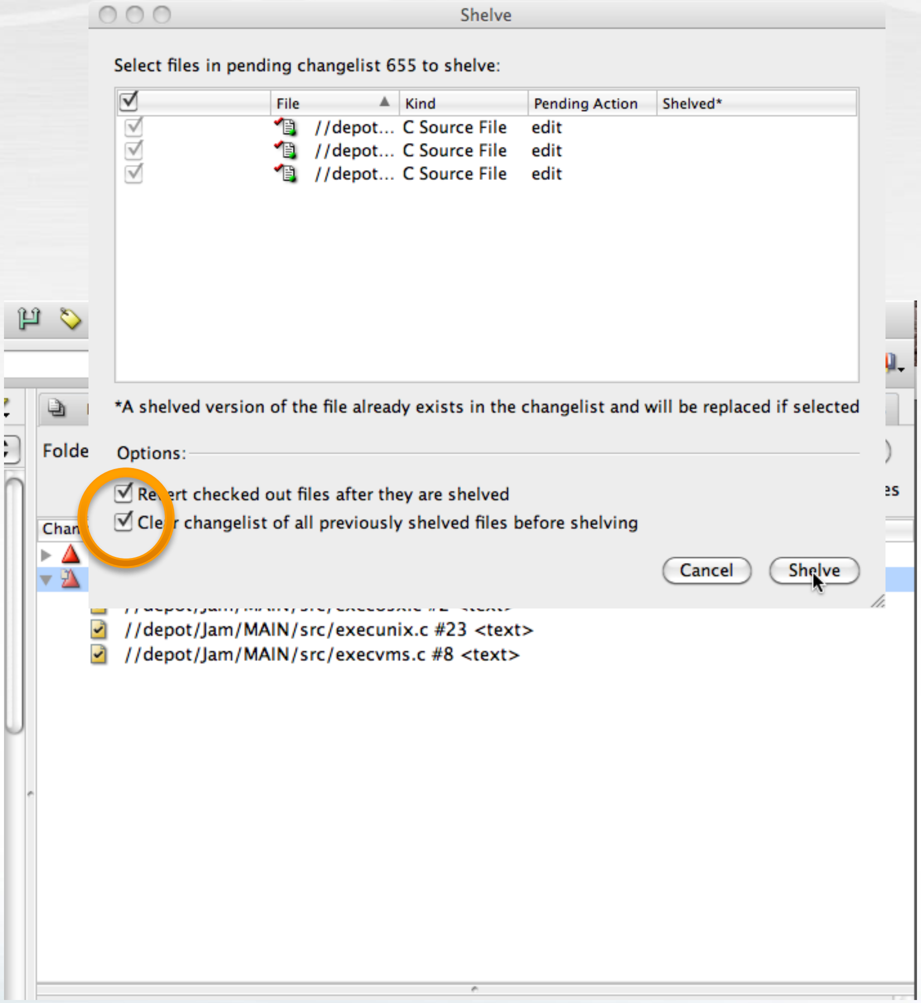
Change	User	Description
default	bruno	<enter description here>
655	bruno	Latest edits - not ready to submit yet

//depot/Jam/MAIN/src/e
//depot/Jam/MAIN/src/e
//depot/Jam/MAIN/src/e
//depot/Jam/MAIN/src/A
//depot/Jam/MAIN/src/c
//depot/Jam/MAIN/src/c
//depot/Jam/MAIN/src/h
//depot/Jam/MAIN/src/h

Change: 655

Date: 19/03/2010 14:28

Workspace: bruno_ws



Select files in pending changelist 655 to shelve:

☑	File	Kind	Pending Action	Shelved*
☑	//depot...	C Source File	edit	
☑	//depot...	C Source File	edit	
☑	//depot...	C Source File	edit	

*A shelved version of the file already exists in the changelist and will be replaced if selected

☑ Revert checked out files after they are shelved

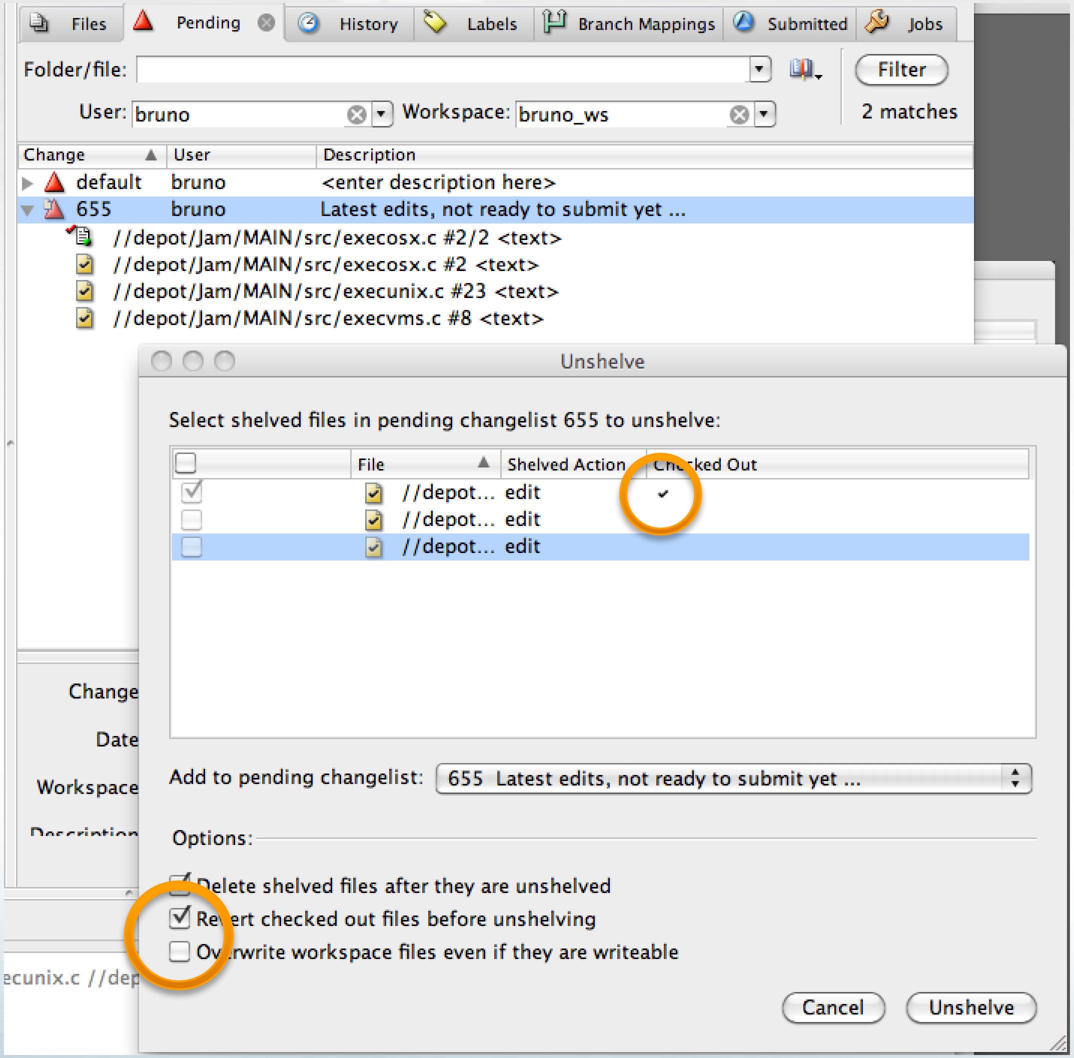
☑ Clear changelist of all previously shelved files before shelving

//depot/Jam/MAIN/src/execunix.c #23 <text>
//depot/Jam/MAIN/src/execvms.c #8 <text>

Unshelving details

- Workspace files need to be unopened

- Overwrite writeable files with “p4 unshelve -f”



Folder/file:

User: bruno Workspace: bruno_ws

2 matches

Change	User	Description
default	bruno	<enter description here>
655	bruno	Latest edits, not ready to submit yet ...

```
//depot/Jam/MAIN/src/execosx.c #2/2 <text>
//depot/Jam/MAIN/src/execosx.c #2 <text>
//depot/Jam/MAIN/src/execunix.c #23 <text>
//depot/Jam/MAIN/src/execvms.c #8 <text>
```

Unshelve

Select shelved files in pending changelist 655 to unshelve:

□	File	Shelved Action	Checked Out
☑	//depot...	edit	✓
□	//depot...	edit	
□	//depot...	edit	

Add to pending changelist: 655 Latest edits, not ready to submit yet ...

Options:

☑ Delete shelved files after they are unshelved

☑ Revert checked out files before unshelving

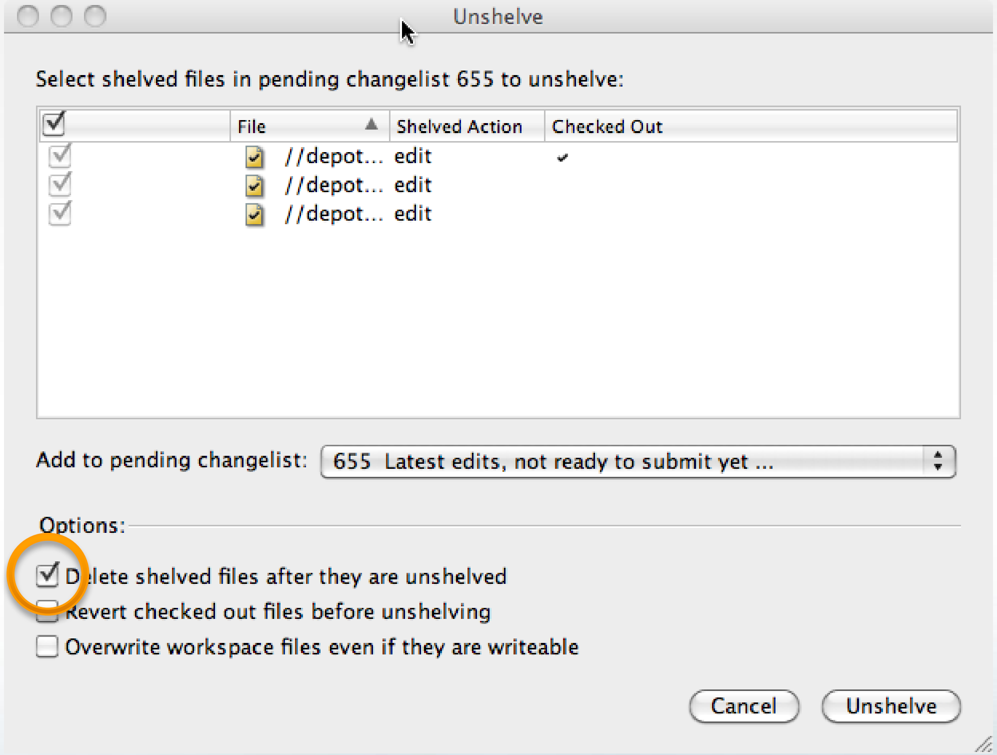
□ Overwrite workspace files even if they are writeable

Unshelving older edits

- Shelved list contains revision of shelved files
  - Files might have new revisions since shelving
- Unshelving restores and opens for edit previous #have revisions
- Upon submit or sync a *resolve* will be scheduled
- Resolve allows for merging changes

Deleting files on a shelf

- p4 shelve -d -c *cl#*



Unshelve

Select shelved files in pending changelist 655 to unshelve:

☑	File	Shelved Action	Checked Out
☑	//depot...	edit	✓
☑	//depot...	edit	
☑	//depot...	edit	

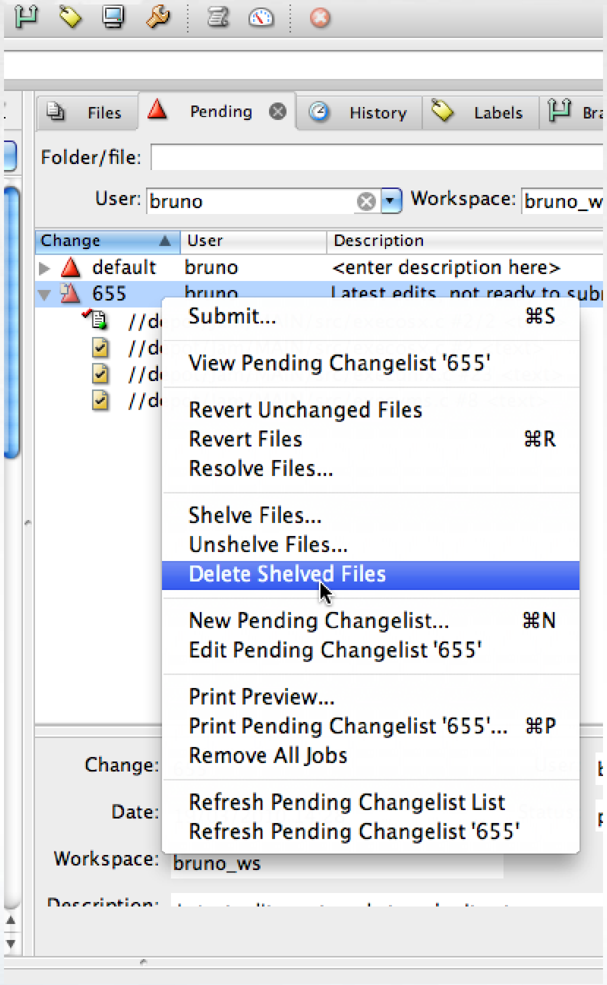
Add to pending changelist: 655 Latest edits, not ready to submit yet ...

Options:

☑ Delete shelved files after they are unshelved

□ Revert checked out files before unshelving

□ Overwrite workspace files even if they are writeable



Files
Pending
History
Labels
Bra

Folder/file:

User: bruno

Workspace: bruno_w

Change	User	Description
default	bruno	<enter description here>
655	bruno	Latest edits, not ready to subm
//d		
//d		
//d		
//d		

Submit... ⌘S
View Pending Changelist '655'
Revert Unchanged Files
Revert Files ⌘R
Resolve Files...
Shelve Files...
Unshelve Files...
Delete Shelved Files
New Pending Changelist... ⌘N
Edit Pending Changelist '655'
Print Preview...
Print Pending Changelist '655'... ⌘P
Remove All Jobs
Refresh Pending Changelist List
Refresh Pending Changelist '655'

Change:

Date:

Workspace: bruno_ws

Description:

Inventory: What is shelved?

- List all changelists that contain shelved files
  - p4 changes -s shelved

- Show shelving details of a shelved list
  - p4 describe -S *shelvedlist#*

Inventory: What is on a shelve?

- Show all shelved files of a shelved list
  - p4 files @=*shelvedlist#*

- Show difference between workspace and shelve
  - p4 diff @=*shelvedlist#*

- Print file content of a shelved file
  - p4 print @=*shelvedlist#*

Use cases

- Backup of a workspace
- Swapping projects
- Transferring changes between workspaces
- Review tool

Use case: Backup of a Workspace

- Situation
  - Local changes are not ready for submit yet
  - Changes need to be backed up
- Shelving provides
  - Backup of changes in the workspace
  - Ability to roll back intermediate changes
  - Ability to recover changes to separate client host

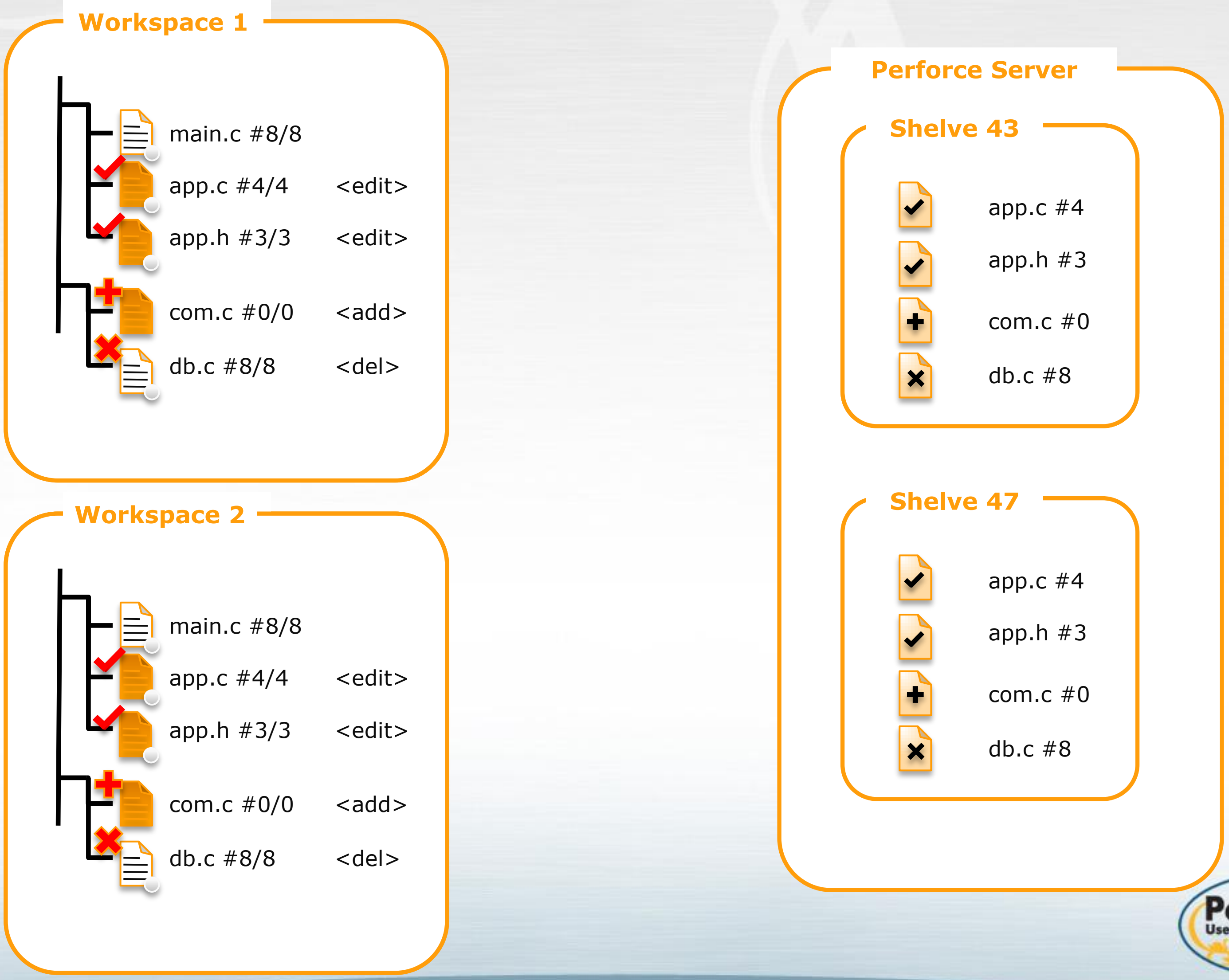
Use case: Swapping Projects

- Situation
  - Current project needs to be put on hold
  - Another project needs to modify the same files
- Shelving provides
  - Backup of the current changes
  - Several shelves can be used in parallel
  - Shelves can contain the same files with different content

Use case: Transferring Changes

- Situation
  - Changes need to be tested on different platforms
  - Not ready to be submitted
- Shelving provides
  - Ability to unshelve files to different workspaces
  - Files are opened in a separate changelist
- Additional consideration
  - Create a second shelve to transfer changes back

Transferring changes between clients



Workspace 1

	main.c #8/8	
✓	app.c #4/4	<edit>
✓	app.h #3/3	<edit>
+	com.c #0/0	<add>
✗	db.c #8/8	<del>

Workspace 2

	main.c #8/8	
✓	app.c #4/4	<edit>
✓	app.h #3/3	<edit>
+	com.c #0/0	<add>
✗	db.c #8/8	<del>

Perforce Server

Shelve 43

✓	app.c #4
✓	app.h #3
+	com.c #0
✗	db.c #8

Shelve 47

✓	app.c #4
✓	app.h #3
+	com.c #0
✗	db.c #8

Use case: Code Review

- Shelves can be used for code review
- File content can be accessed on the shelve
  - Or unshelved into the reviewers workspace

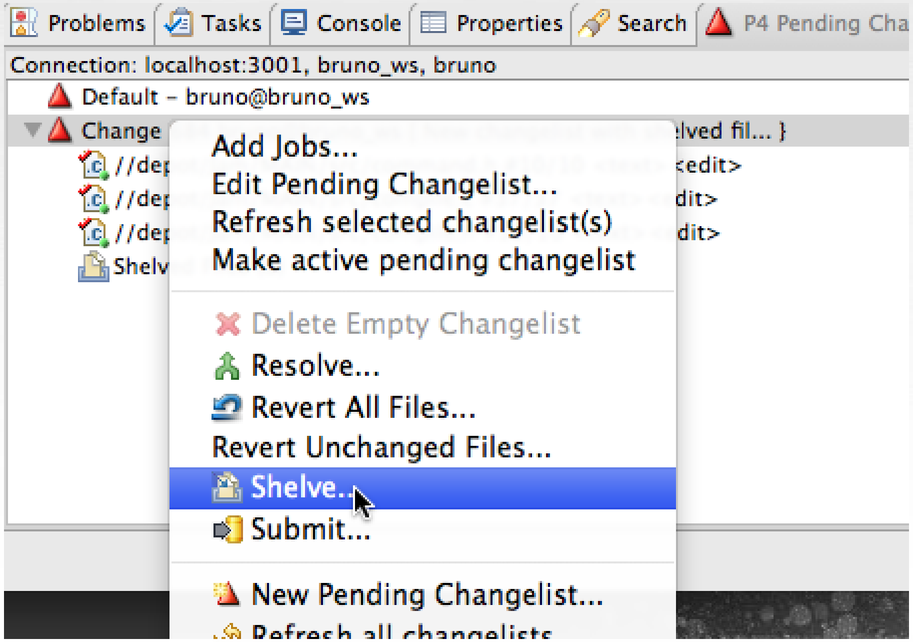
- Code review could be enforced using a trigger or the protection table
  - Reviewer would unshelve and submit changes

Shelf or private branch?

- Shelves cannot be shared
- Useful for short-term development
- Enables easier code review
- Possible alternative to sparse branches

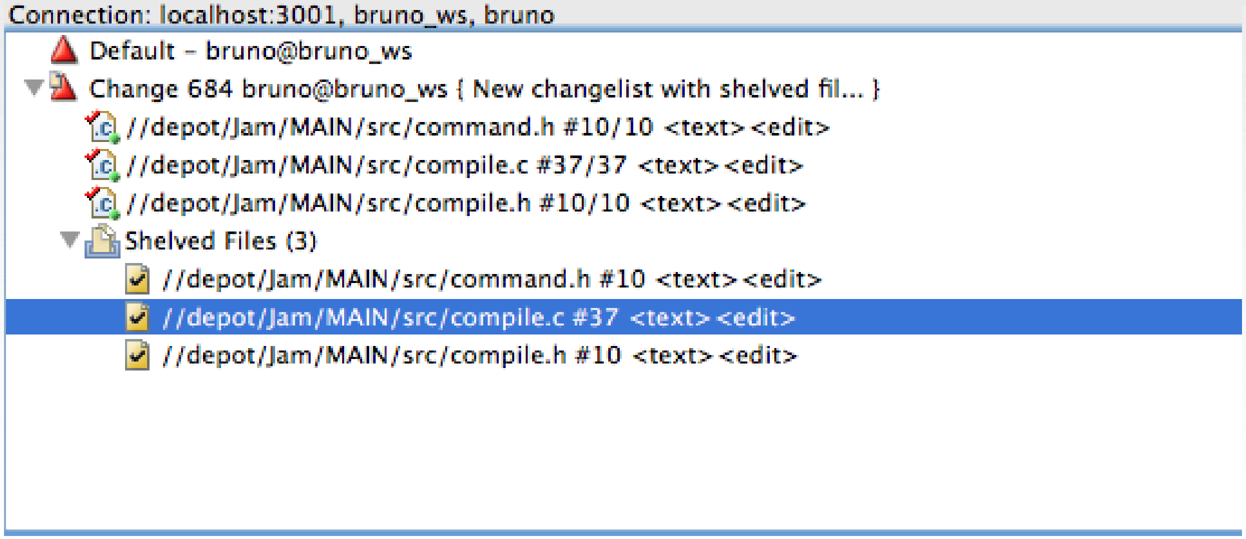
- Transient rather than persistent

Eclipse integration



Connection: localhost:3001, bruno_ws, bruno

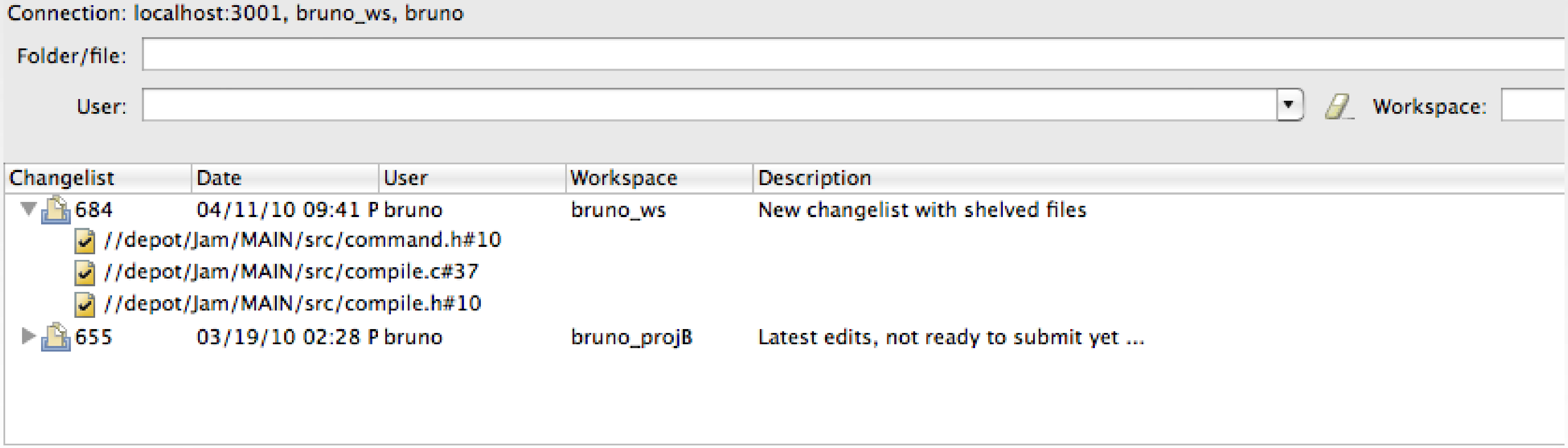
Eclipse integration



Connection: localhost:3001, bruno_ws, bruno

- Default - bruno@bruno_ws
- Change 684 bruno@bruno_ws { New changelist with shelved fil... }
  - //depot/Jam/MAIN/src/command.h #10/10 <text> <edit>
  - //depot/Jam/MAIN/src/compile.c #37/37 <text> <edit>
  - //depot/Jam/MAIN/src/compile.h #10/10 <text> <edit>
  - Shelved Files (3)
    - //depot/Jam/MAIN/src/command.h #10 <text> <edit>
    - //depot/Jam/MAIN/src/compile.c #37 <text> <edit>
    - //depot/Jam/MAIN/src/compile.h #10 <text> <edit>

Eclipse integration



Connection: localhost:3001, bruno_ws, bruno

Folder/file:

User: Workspace:

Changelist	Date	User	Workspace	Description
▼ 684	04/11/10 09:41 P	bruno	bruno_ws	New changelist with shelved files
✓ //depot/Jam/MAIN/src/command.h#10				
✓ //depot/Jam/MAIN/src/compile.c#37				
✓ //depot/Jam/MAIN/src/compile.h#10				
▶ 655	03/19/10 02:28 P	bruno	bruno_projB	Latest edits, not ready to submit yet ...

Shelving and permissions

- To create or update a shelf, **write** permission is required from the protection table
  - That is, effective **write** permission as defined by
    - p4 protects -m
- No **write** permission to the shelved files required
- Can be used to enforce reviewing

Shelving and Tunables

- dm.shelve.maxfiles
  - Limits the number of files that can be shelved in a single shelved list
  - Default 10M (10 million)

- dm.shelve.maxsize
  - Limits the size of each file to be shelved
  - Default 0 (unlimited)

Outlook: 2010.1 enhancements

- Shelving triggers

shelve-submit	Invoked when files are shelved
shelve-commit	Invoked after files are shelved
shelve-delete	Invoked when shelves are deleted

  - Can be used to monitor, regulate or prohibit shelving usage

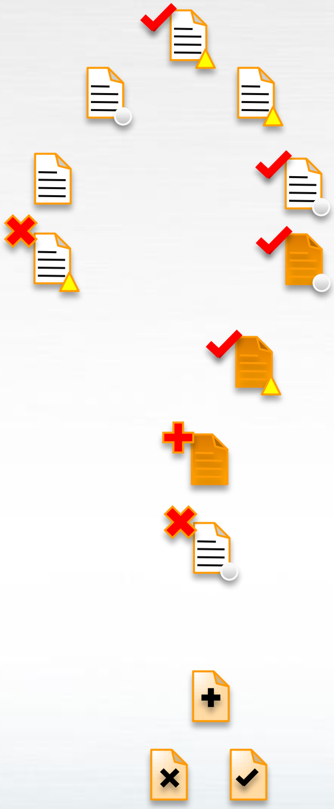
- p4 sizes -S
  - Displays size for shelved files only

Conclusion

- Shelving is an exciting new feature of Perforce
- Shelving has many different applications
  - Stashing changes away on the server
  - Transferring or sharing changes
  - Code review

- Perforce will continue to enhance shelving

Questions/Feedback?



Perforce User Conference 2010