



INTERSYSTEMS LEARNING SERVICES

InterSystems Change Control



InterSystems Change Control

COPYRIGHT NOTICE

© 2020 InterSystems Corporation, Cambridge, MA. All rights reserved.

InterSystems is a registered trademark of InterSystems Corporation.



InterSystems, InterSystems Caché, InterSystems Ensemble, InterSystems HealthShare, HealthShare, InterSystems TrakCare, TrakCare, InterSystems DeepSee, and DeepSee are registered trademarks of InterSystems Corporation.

InterSystems IRIS data platform, InterSystems IRIS for Health, InterSystems IRIS, InterSystems iKnow, Zen, and Caché Server Pages are trademarks of InterSystems Corporation.

All other brand or product names used herein are trademarks or registered trademarks of their respective companies or organizations.

This document contains trade secret and confidential information which is the property of InterSystems Corporation, One Congress Street, Boston, MA 02114, or its affiliates, and is furnished for the sole purpose of the operation and maintenance of the products of InterSystems Corporation. No part of this publication is to be used for any other purpose, and this publication is not to be reproduced, copied, disclosed, transmitted, stored in a retrieval system or translated into any human or computer language, in any form, by any means, in whole or in part, without the express prior written consent of InterSystems Corporation.

The copying, use and disposition of this document and the software programs described herein is prohibited except to the limited extent set forth in the standard software license agreement(s) of InterSystems Corporation covering such programs and related documentation. InterSystems Corporation makes no representations and warranties concerning such software programs other than those set forth in such standard software license agreement(s). In addition, the liability of InterSystems Corporation for any losses or damages relating to or arising out of the use of such software programs is limited in the manner set forth in such standard software license agreement(s).

THE FOREGOING IS A GENERAL SUMMARY OF THE RESTRICTIONS AND LIMITATIONS IMPOSED BY INTERSYSTEMS CORPORATION ON THE USE OF, AND LIABILITY ARISING FROM, ITS COMPUTER SOFTWARE. FOR COMPLETE INFORMATION REFERENCE SHOULD BE MADE TO THE STANDARD SOFTWARE LICENSE AGREEMENT(S) OF INTERSYSTEMS CORPORATION, COPIES OF WHICH WILL BE MADE AVAILABLE UPON REQUEST.

InterSystems Corporation disclaims responsibility for errors which may appear in this document, and it reserves the right, in its sole discretion and without notice, to make substitutions and modifications in the products and practices described in this document.

For Support questions about any InterSystems products, contact:

InterSystems WorldWide Response Center

Telephone: +1-617-621-0700
Tel: +44 (0) 844 854 2917
Email: support@InterSystems.com

ICC 410: CCR Transport – Perforce Basics



Objectives

- Explain what source control is and why it is useful.
- Explain the basic terminology associated with Perforce.
- Configure a P4V client.
- Perform basic actions using P4V such as:
 - Navigating depot, checking out files, and submitting changelists.
- Identify basic points of integration between InterSystems internal applications and Perforce.
- Examine file history using:
 - Change History, Time-lapse view, and Revision Graph.



Part 1: Source Control



What is Source Control?

- Database for flat file items.
 - Provides central storage for code.
 - Provides full versioning capabilities.
 - Prevents permanent deletion.
 - Maintains all history.



What is Source Control? (cont.)

- How we use it:
 - Provide answers to the questions about changes: Who, What, When, Where, Why and How?
 - Keep track of every version of every item used as part of a System's configuration or custom code.
 - Allow automated merging of changes between environments.
 - Provide snapshot of operating environment.
- Also know as SCM (Source Control/Code Management).



Example Source Control Software

- Perforce.
 - InterSystems Standard.
- Git.
- CVS (Concurrent Versions System).
- VSS (Visual Source Safe).
- SVN (Subversion).
- StarTeam.

PERFORCE



What is Perforce Helix?

- An easy-to-use source control solution.
- Main advantages of Perforce are:
 - It's fast!
 - Powerful Perforce Visual Client (P4V).
 - Excellent branching and merge capability.
 - Very low admin effort.
- Integrated with InterSystems System Portal & IDE's (Studio & Atelier).



Part 2: Perforce Architecture



Perforce

- Client-Server model.
- Single server.
 - Hosted at Cambridge HQ: `perforce.iscinternal.com:1666`.
 - For all CCR deployments.
 - Holds depots (discrete source DBs).
 - Replica server in Sydney (`auperforce.iscinternal.com`).
 - Only for local work in Sydney, most people should connect to Cambridge server.

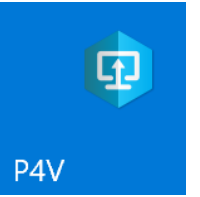


Perforce (cont.)

- Many clients.
 - All ISC users.
 - CCR online server.
 - “Connected” CCR Tier 1 environments.
 - CCR Tier 2 clients use the CCR online server to act as their client.



Client Software



- P4.
 - Command line interface (CLI) client to Perforce.
 - Very useful for scripting and automation.
 - P4V.
 - P4 CLI wrapped with visual client (desktop app).
 - Client provides advanced features.
 - Free, but connecting to Perforce server requires a license.
 - Perforce does give away free 5 license Perforce servers.
 - Licensed per user.
 - Available for Windows, Unix® and Linux platforms.
-



Perforce Glossary: Depot

- Depot.
 - The database of files on the server.
 - Can browse from a client
 - Sync files on demand.



Perforce Glossary: Client Workspace

- Client workspace.

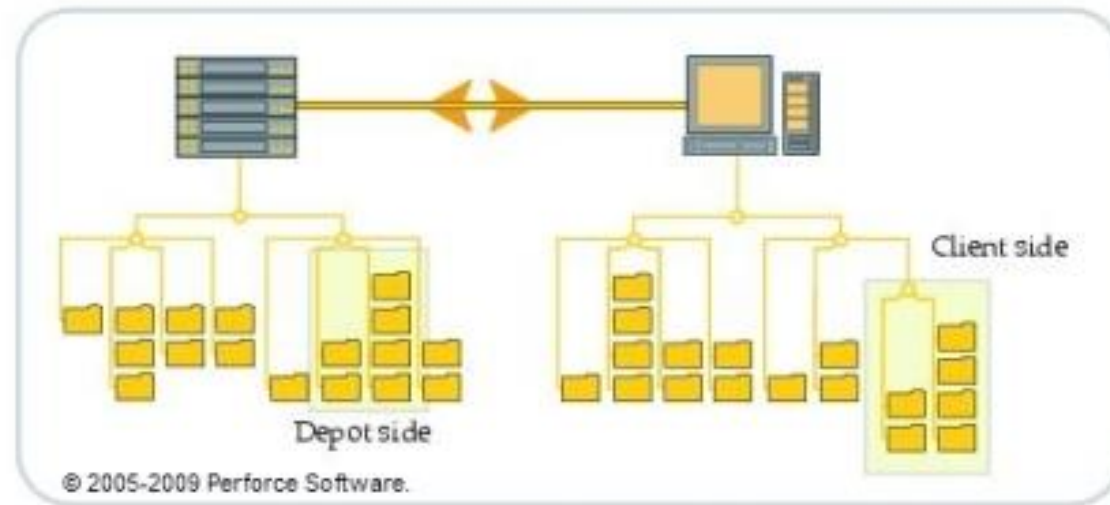


- A link between the depot and the local directory for working files.
- A client must have a workspace defined to check out files.
- The collection of local copies of files.
- Perforce server manages metadata on the version of each file in every workspace.
- Attributes include Name, Owner, Options, Host, Root Path, View path.



Client Workspace

- Files stored in a server depot are mapped and synced to a local client workspace.
- Don't modify file settings locally.
 - Never change a file to R/W through file system controls.



Quiz: Perforce Clients

Question:

Identify the Perforce client(s) from the list below. Select all that apply.

- A. P4V.
- B. Git.
- C. CCR online server.
- D. P4.
- E. Client workspace.



Quiz: Perforce Clients (cont.)

Answer:

A. P4V.

C. CCR online server.

D. P4.

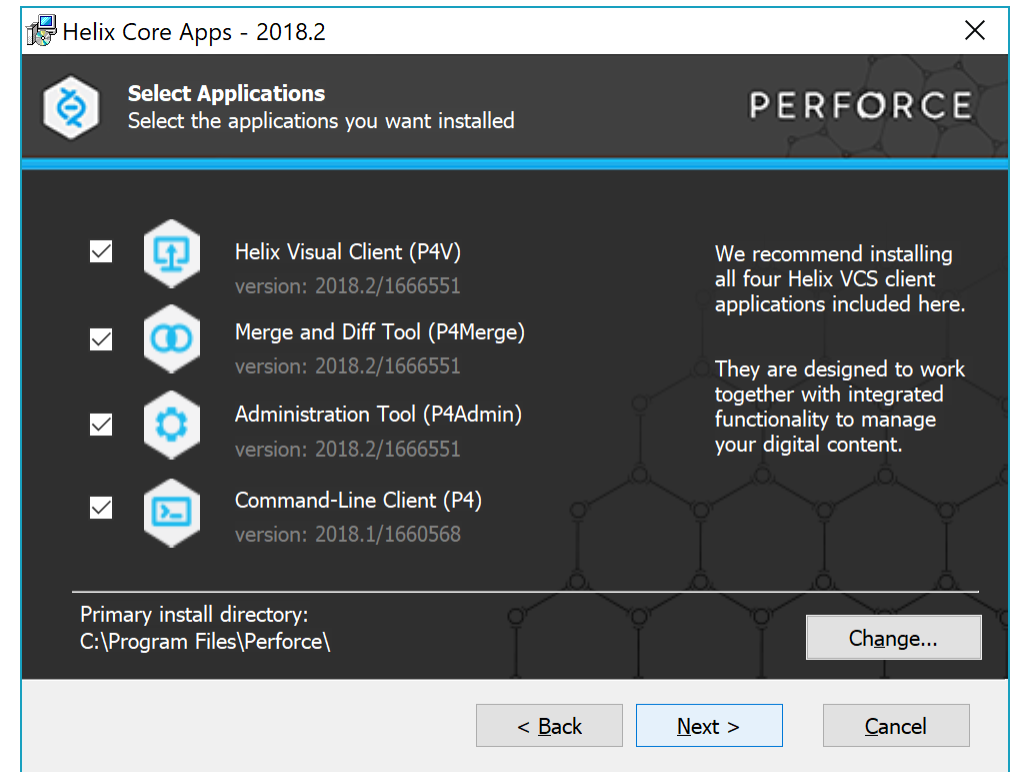
The client workspace is used by a client but isn't a client in and of itself.



Part 3: Installing and Configuring the P4V Client

How To: Download and Install the P4V Client

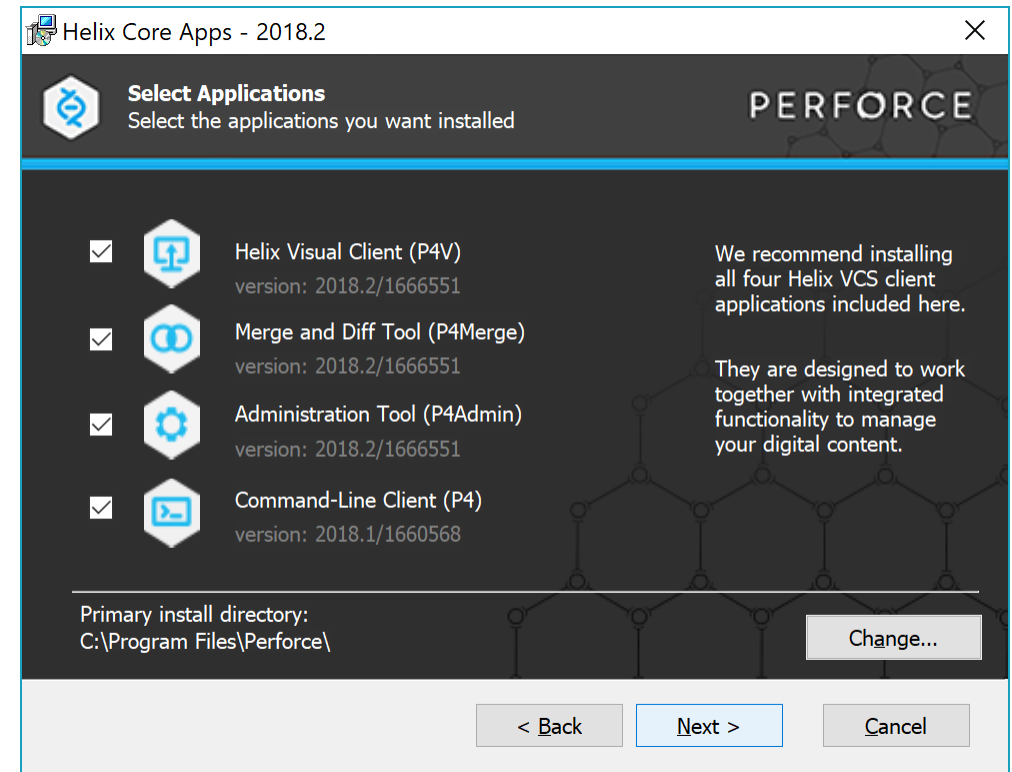
- Any users on the InterSystems internal network can install P4V to directly connect to the InterSystems Perforce server.



How To: Download and Install the P4V Client (cont.)

■ Installation:

1. Go to http://www.perforce.com/download/complete_list.
2. Select your client platform (e.g. Windows).
3. Select your client OS (e.g. Windows for 32-bit Intel (x86)).
4. Click "Accept and Download" for "P4V: Visual Client."
5. Run the executable.
6. Accept the default features and click "Next."



How To: Download and Install the P4V Client (cont.)

■ Installation:

7. Set the fields as follows:

- Server = `perforce.iscinternal.com:1666`.
- User Name = your Perforce user name (most likely your TRC user name).
- Text Editing Application = leave default, on Windows should be location of `notepad.exe`.

8. Click "Next."

9. Click "Install."

10. Wait for installation to complete, then click "Close."

Helix Core Apps - 2018.2

Client Configuration
Specify the settings for the client application(s)

PERFORCE

Server
perforce.iscinternal.com:1666 ?
Name or hostname and port number of the Perforce server that clients connect to. Omit hostname for connecting to a local server.

User Name
soukoctr
Perforce user name for connecting to a Perforce server.

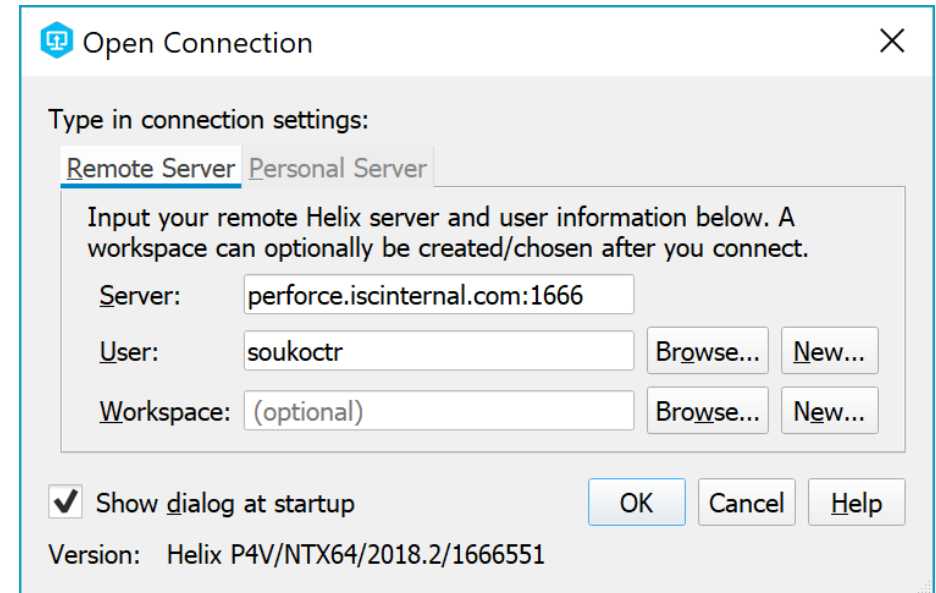
Set P4 Text Editor
C:\WINDOWS\SysWOW64\notepad.exe Browse...
Application used to open text based forms.

< Back Next > Cancel



How To: Configure the P4V Client

- These steps require a Perforce username. If you do not have one, please contact Tech Services (tech.help@intersystems.com).
1. Start the P4V client.
 2. In the Open Connection dialog box, Remote Server tab:
 1. Set Server to be `perforce.iscinternal.com:1666`.
 2. Set User field to your InterSystems SSO username.
 3. Click New next to the workspace field.



How To: Configure the P4V Client (cont.)

3. Enter your Perforce password to proceed (this should have been provided along with your username).

Open Connection

Type in connection settings:

Remote Server Personal Server

Input your remote Helix server and user information below. A workspace can optionally be created/chosen after you connect.

Server: perforce.iscinternal.com:1666

User: soukoctr Browse... New...

Workspace: (optional) Browse... New...

Show dialog at startup

OK Cancel Help

Version: Helix P4V/NTX64/2018.2/1666551



How To: Configure the P4V Client (cont.)

4. In the popup wizard from step 3:

1. Replace the default workspace name.

- Single user development environment (aka "Private BASE"):
"[*PerforceUsername*][_*MachineName*]" (e.g. bspead_BENSPEAD630).
- Multiple user development environment (aka "Shared BASE"):
"SHARED_[*MachineName*]" (e.g. SHARED_AIXDEV3).

2. Set workspace root to the local directory under which all source is stored (e.g. C:\Perforce).

3. Confirm that "Switch to new workspace immediately" is selected.

4. Confirm that "Automatically get all revisions" is cleared.

5. Click "OK" to finish creating the workspace.



How To: Configure the P4V Client (cont.)

5. Click “OK” in the Open Connection dialog to connect to that new workspace (the new name should auto-populate in the workspace field).



Quiz: Perforce Workspace

Question:

What does a workspace with a prefix of SHARED_* signify?

- A. Many users can use the workspace on their laptops for testing purposes.
- B. The workspace is in use on a development instance shared by multiple developers.
- C. The workspace is located on a public File Share.
- D. The workspace is generous with its toys.



Quiz: Perforce Workspace

Answer:

B. The workspace is in use on a development instance shared by multiple developers.

Only a single workspace can be used per DB namespace, and therefore the convention for multi-developer instances is to name it SHARED_<machineName>.



Exercise ICC410-1: Create a Workspace



Part 4: Perforce Concepts



Perforce Glossary: Branch

- Branch.
 - The folder structure of your Perforce depot.
 - Enables similar structures to represent each of the multiple environments in the System.



Perforce Glossary: Sync

- Sync.
 - Action that copies depot files to local workspace.
 - Download from server.
 - Synced files are ReadOnly.
 - Click Get Latest button with a file highlighted to Sync latest revision of the file to your workspace.
 - **IMPORTANT:** Perforce keeps track of the version of each file synced to every workspace.
 - Designated by filename#have.
 - Do not manually manipulate files outside of Perforce clients or the workspace will get confused.



File status icons



- The image and color of the file icons in P4V will indicate the status of that file.
- Hovering over the icon will tell you more information.
 - E.g., Who has the file checked out and where.



Exercise ICC410-2: Sync Files



Perforce Glossary: Changelist

- Collection of changed files within Perforce.
- Identified by a numerical ID.
- 2 types of changelists depending on where changes are located:
 -  Pending - changes are 'in progress' and only exist in user's workspace.
 -  Submitted - changes were applied to the depot, are visible to all users and cannot be altered.



Perforce Glossary: Job



- Logical identifier (aka 'Tag') representing a discrete project.
- May be attached to multiple changelists.
- Jobs are very important for integration with internal applications:
 - CCR IDs are used as jobs for work managed by CCR Online.
 - JIRA IDs are used for jobs for work managed by JIRA.
 - Devlog IDs are used for jobs for Database Platforms development work.



Quiz: Jobs and Changelists

Question:

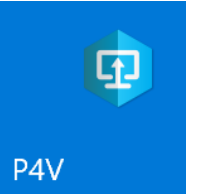
There is a one-to-one relationship between Perforce jobs and Perforce changelists. True or False?

Answer:

False. A changelist may have many jobs and a job may be assigned to many changelists; it is a many-to-many relationship.



How To: Navigate within P4V on Windows



Navigation	How To
Go To... <ul style="list-style-type: none">• Changelist• User• Workspace• Job	<ol style="list-style-type: none">1. [Ctrl]+g
Copy File Location	<ol style="list-style-type: none">1. Click on a file2. [Ctrl]+c
Highlight file within current tree (depot or workspace) whose path has been copied	<ol style="list-style-type: none">1. Click on any file2. [Ctrl]+v

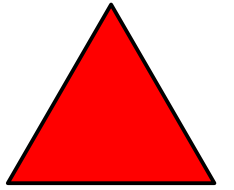


How To: Search the Depot

- 2 options:
 - Right-click branch -> Find File.
 - Be specific, searching a large depot takes time.
 - Ensure you are searching the depot, not your workspace.
 - p4 grep command.
 - Allows searching via REGEX.
 - NOTE: There is upper limit on number of files searched with p4 grep.
- Launch search from deepest location possible.
- Searching too broadly may return “no results” even when one exists.



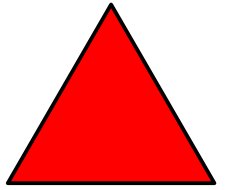
Working on Files: Developing



- Folders are an attribute of files within Perforce.
 - Therefore, can't have an empty folder in Perforce.
- Examples of changes against files:
 - Add.
 - Checkout (to change).
 - Delete.
 - Rename/Move.
 - Rollback.



Working on Files: Developing (cont.)



- All changes are associated with a pending changelist.
 - A “default” pending changelist exists for every user in every workspace.
 - Can have multiple pending changelists for multiple streams of work.





Working on Files: Checkout action

- Required to edit file in workspace.
- Changes file in workspace from read-only to read-write.
- Updates server-side file metadata.
 - Other clients can see who has a file checked out and where.
- Adds file to “default” pending changelist.
- How To:
 1. Select file to checkout.
 2. Click Checkout button in top toolbar.



Working on Files: Add Action

- Use to add a new file to the depot.
- Adds file to “default” changelist.
- How To:
 1. Save or move file to appropriate folder in workspace filesystem.
 2. In Perforce, open the workspace view and click Refresh in the upper left corner.
 3. In the workspace view, click file to add.
 4. Click “Add” button in top toolbar.
 5. Submit changelist.



Working on Files: Delete Action

- Use to “delete” a file in the depot.
 - Does not entirely delete file from server.
 - Still able to get any prior revision of the file.
- Adds file to “default” changelist to delete when changelist submitted.
- How To:
 1. In the workspace view, click file to delete.
 2. Click “Delete” button in top toolbar.
 3. Submit changelist.



Working on Files: Rename/Move

- Use to rename a file or move to a different folder.
 - Actually uses a delete and an add action.
- Adds a delete action on the old file and an add action for the new file to a new, saved, changelist.
- How To:
 1. In the workspace view, right-click file > Rename/Move.
 2. Fill out the popup window and click Save to Changelist. Either use:
 1. The top half of the popup if you are intending to rename the file.
 2. The bottom half of the popup if you are intending to move the file.
 3. Submit changelist.



RCS Keywords – Tagging the source of your source

- Files can embed an RCS keyword.
- Like a source control “watermark.”
- Automatically replaced by a specific string when a file is synced from Perforce.
- Filetype needs to be:
 - \$Id\$ and \$Header\$: “text+ko”.
 - All other keywords: “text+k”.

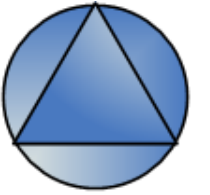


RCS Keyword Examples

Keyword	Expands To	Example
<code>\$Author\$</code>	User submitting the files.	<code>\$Author: bruno \$</code>
<code>\$Change\$</code>	Changelist number under which file was submitted.	<code>\$Change: 439 \$</code>
<code>\$Date\$</code>	Date of last submission (YYYY/MM/DD).	<code>\$Date: 2011/08/18 \$</code>
<code>\$DateTime\$</code>	Helix Server local date and time of last submission (YYYY/MM/DD hh:mm:ss).	<code>\$DateTime: 2011/08/18 23:17:02 \$</code>
<code>\$File\$</code>	Filename only, in depot syntax, without revision number.	<code>\$File: //depot/path/file.txt \$</code>
<code>\$Header\$</code>	Synonymous with <code>\$Id\$</code> .	<code>\$Header: //depot/path/file.txt#3 \$</code>
<code>\$Id\$</code>	Filename and revision number in depot syntax.	<code>\$Id: //depot/path/file.txt#3 \$</code>
<code>\$Revision\$</code>	Helix Server revision number.	<code>\$Revision: #3 \$</code>



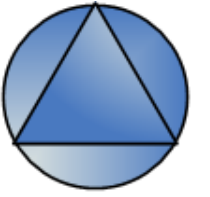
Working on Files: Submitting Your Changes



- Save Changelist.
 - Allows updating Description, Job, and use of the shelving feature.
 - Generates temporary changelist number that will be changed when submitted.
 - Does not update Perforce Server with changes.



Working on Files: Submitting Your Changes (cont.)



- Submit Changelist to apply changes.
 - Workspace file becomes the “head” revision in the depot.
 - Description required.
 - Make sure a job is attached when working with internal systems!
 - Best practice: Remove any unaltered files via “Revert Unchanged” before submitting.
 - Best practice: Diff changes when possible.
- Revert Changelist if want to discard all changes to files in changelist.



Quiz: Location of changes

Question:

Tim has a pending, saved changelist. He is not using the shelving feature. His laptop hard drive is destroyed. How can he recover the changelist and modified files in the changelist? Select all that apply.

- A. Using a full backup of his hard drive (if it exists).
- B. Using the Perforce server.
- C. Using what he remembers about the changes made to recreate those changes manually.



Quiz: Location of changes (cont.)

Answer:

- A. Using a full backup of his hard drive (if it exists).
- C. Using what he remembers about the changes made to recreate those changes manually.

By default, pending changelists only store files in the local workspace on local disk.

Take steps to ensure you don't have to recreate changes from scratch if the disk crashes!



Shelving



- Shelving stores copies of open files temporarily in the Perforce server without checking them in.
- Useful for:
 - Backing-up work in progress.
 - Code review.
 - Sharing code ahead of merges.
- Shelving rules:
 - File must be checked out to be shelved.
 - Changelist containing file must have a number.
 - Recall changelists are assigned a temporary number when saved.

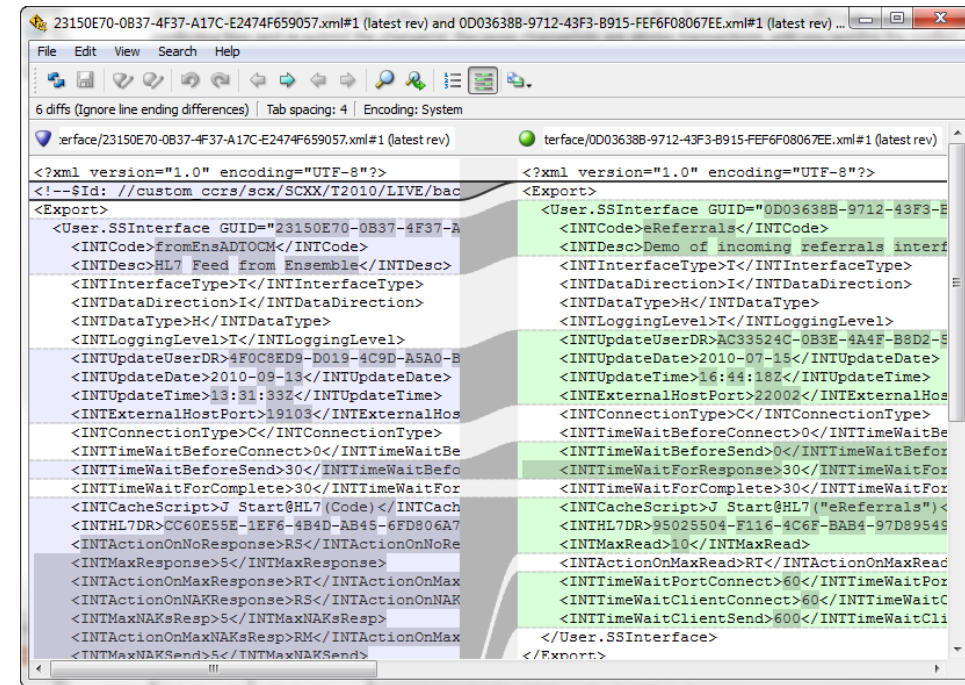


Exercise ICC410-3: Modify and Submit Files



Diff

- P4Merge.
 - Useful tool installed when you install P4V.
 - Can diff directories or individual files.
 - Can diff any revision of a file from:
 - Depot.
 - Local checked-out revision.
 - Can diff files not stored in Perforce.
 - Cannot diff binary files.
- “Diff against” menu option in P4V launches P4Merge.
 - Alternatively, drag and drop revisions in the History view for a file.



Ways to “Undo”

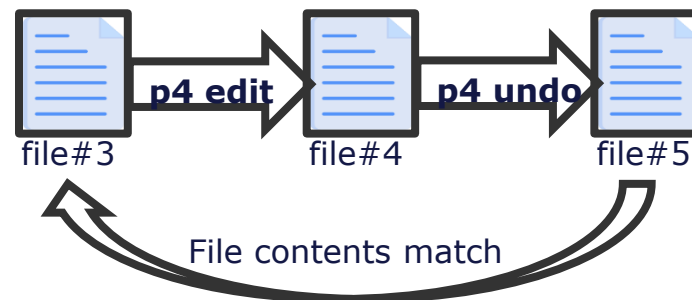
- Revert.
 - Removes checkout status without submitting changes.
 - No impact on depot files (as local checkouts have no affect until submission).
 - Local files return to their previous revision.
- Back out.
 - Removes specific changes made at a given changelist, date, or revision.
 - Allows you to keep changes made in subsequent revisions.



Ways to “Undo” (cont.)

- Undo

- Only Helix Server 2017.1+.
- Undo makes a new revision, with content equal to an earlier revision.
- Restores a file or folder back to a specified changelist, date, revision number, or label.
- Any changes made after that point in time are not retained in the file.
 - But they are retained in the revision history!



Exercise ICC410-4: Diff and Undo



Quiz: Undo

Question:

Take care when using “Undo” because the current revision will be lost in the version history when Undo rolls it back to an older revision. True or False?



Quiz: Undo

Answer:

False.

“Undo” will create a new revision of a file with content equal to an older revision... it does not remove subsequent revisions – they stay intact as part of the Revision History for future reference (or in case there is a need to Undo the Undo).

Best practice is to save a copy of the head revision before executing an “Undo” in case that code needs to be referenced later.

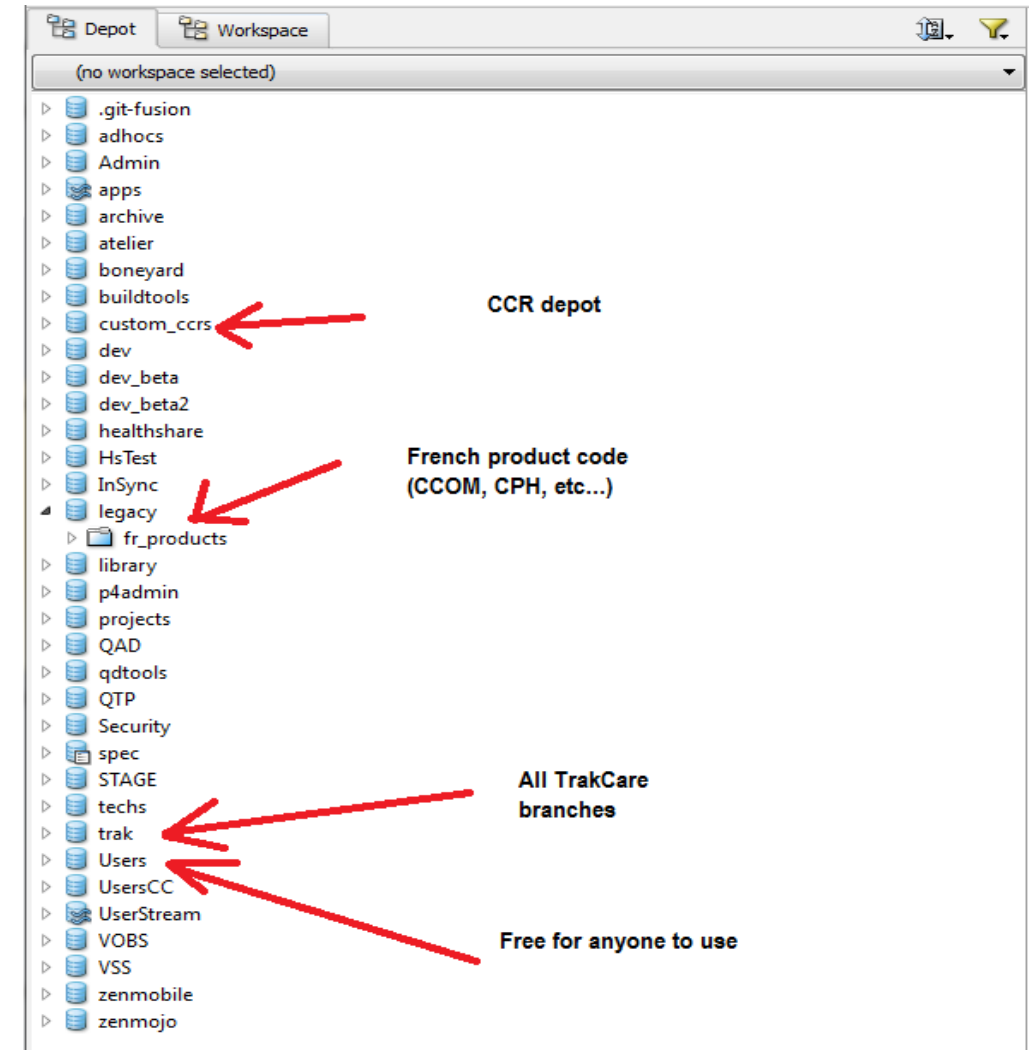


Part 5: Perforce at InterSystems



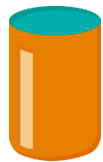
InterSystems Depot Organization

- You probably don't see all of them.
 - Depends on privileges of your Perforce account.
 - NOTE: Your Windows account and Domain Group are not tied to your Perforce account.
 - Tech Services manages rights within Perforce directly via Perforce Security Groups.



InterSystems IDE & Perforce Integration

- Managed by the `%Studio.SourceControl.ISC` “Studio Hooks” class.
 - Included in CCR Client Tools.
 - ‘Server-side Source Control’ because all source is managed via a “Shared” Perforce Workspace located on the file system of the InterSystems product instance.
- Easiest set up with CCR configuration routine `^%buildccr`.
- In practice: One namespace from one server = One branch in Perforce.

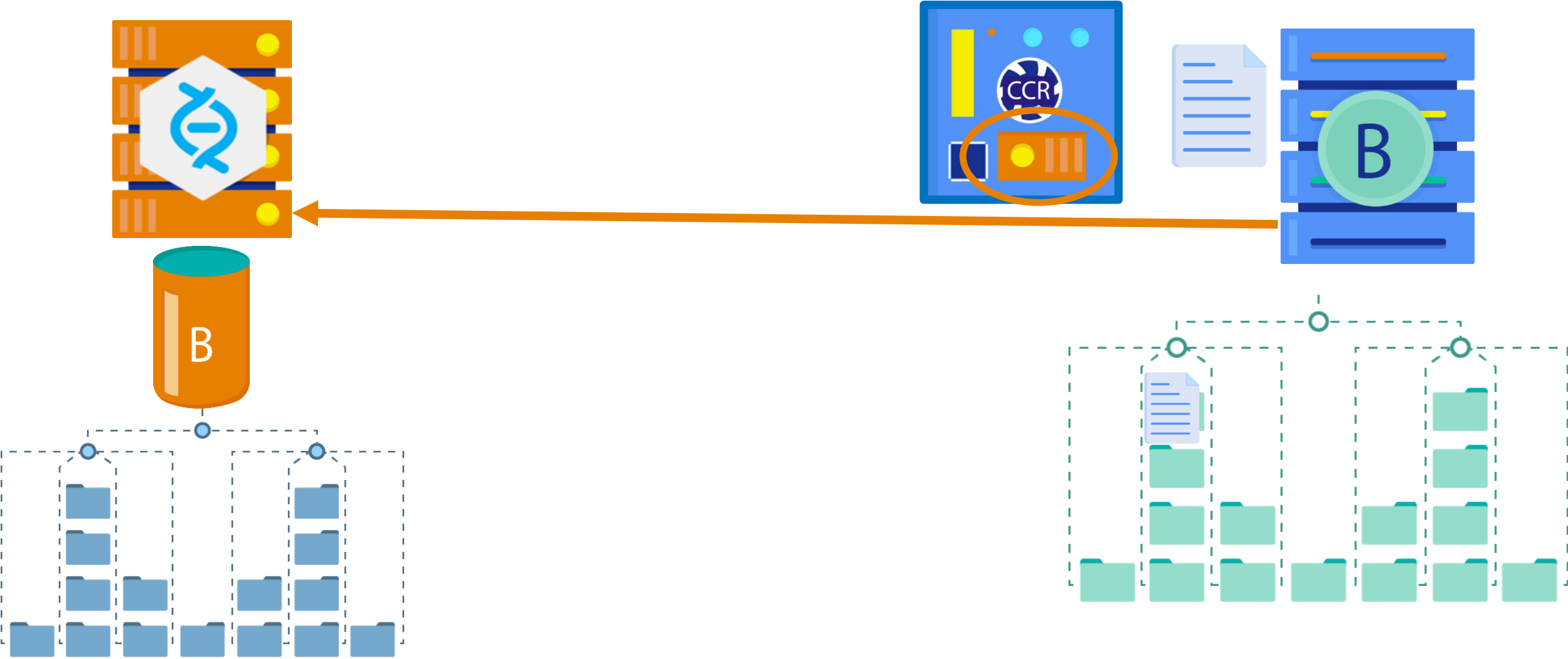


InterSystems IDE & Perforce Integration (cont.)

- Management Portal and Perforce usernames should match.
 - If not, custom configuration is needed to map usernames.
- Perforce “basic” commands emulated from IDE (Studio or Atelier).
 - Check-Out (p4 edit), Undo Check-Out (p4 revert), Check-In (p4 submit), Get Latest (p4 sync), Add to Source Control (p4 add).



Review: Sending Items from **Connected** BASE



CCR & Perforce

- There is a 1-to-1 relationship between a CCR and a P4 job.
 - But multiple changelists can be associated with a single job.
 - Possible to edit a job's changelists.
 - Don't forget to regenerate the ItemSet if you do!





CCR & Perforce (cont.)

- CCR Release Management feature uses P4 shelving.
 - You need to set up a local, personal P4 workspace to resolve any “release” issue.
 - Make sure this P4 workspace supports integration by including both codelines in its Workspace Mappings.
 - Always unshelve the files to a new changelist before attempting to resolve them.
 - Resolving conflicts requires checking out integrated files to perform sympathetic merges.



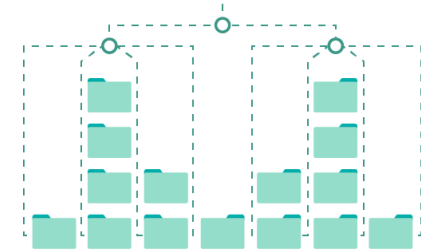
Review: CCR Branches

- Standard branch (folder) location for items stored within Perforce:
 - `//custom_ccrs/<Cc>/<Site>/<Sys>/<Env>/`
 - `custom_ccrs` – top level for CCR apps.
 - `<Cc>` – Country code.
 - `<Site>` – Site code. 
 - `<Sys>` – System code. 
 - `<Env>` – BASE/TEST/UAT/LIVE.



Review: CCR Branches (cont.)

- Example BASE branch for the CCR application:
 - `//custom_ccrs/us/ISCX/CCR/BASE/`.
- Local workspace on the Environment is of the same structure:
 - `\source\custom_ccrs\us\ISCX\CCR\BASE\`.



JIRA & Perforce

- There is a 1-to-1 relationship between a JIRA issue and a P4 job.
 - Can associate multiple changelists with a single job.
 - Possible to edit a job's changelists.
 - P4 job created once the JIRA is "In Progress."



JIRA & Perforce (cont.)

- TrakCare Product ecosystems:
 - Perforce, JIRA, FishEye & Jenkins.
 - Triggers perform checks on JIRA transition based on FishEye being able to retrieve changes in Perforce.
 - Jenkins uses the link between Perforce and JIRA to “tag” the JIRAs with the build number (if successful).



Quiz: Jobs at InterSystems

Question:

Select all of the systems that create Perforce Jobs based on application IDs to tie together changelists with a change record:

- A. CCR.
- B. iService.
- C. DevLog.
- D. JIRA.
- E. WRC.
- F. Turbo.



Quiz: Jobs at InterSystems

Answer:

A. CCR.

C. DevLog.

D. JIRA.

The other applications do not store things in Perforce.



Part 6: Moving Changes between Branches



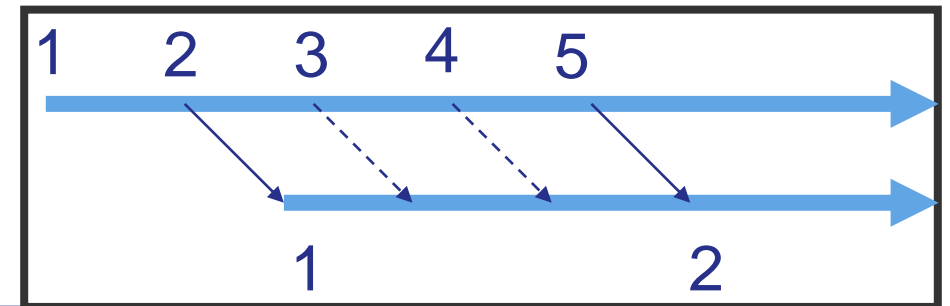
Perforce Glossary: Integrate, Resolve, Merge

- Integrate.
 - Movement of changes ('diff chunks') from one branch to another.
- Resolve.
 - When moving changes for an existing file, the differing revisions must be resolved.
 - Either revision can be retained, or the changes can be merged.
- Merge.
 - Diff chunks from source and target files can be selectively picked to determine which pieces from which file should be in the final result that is checked in.



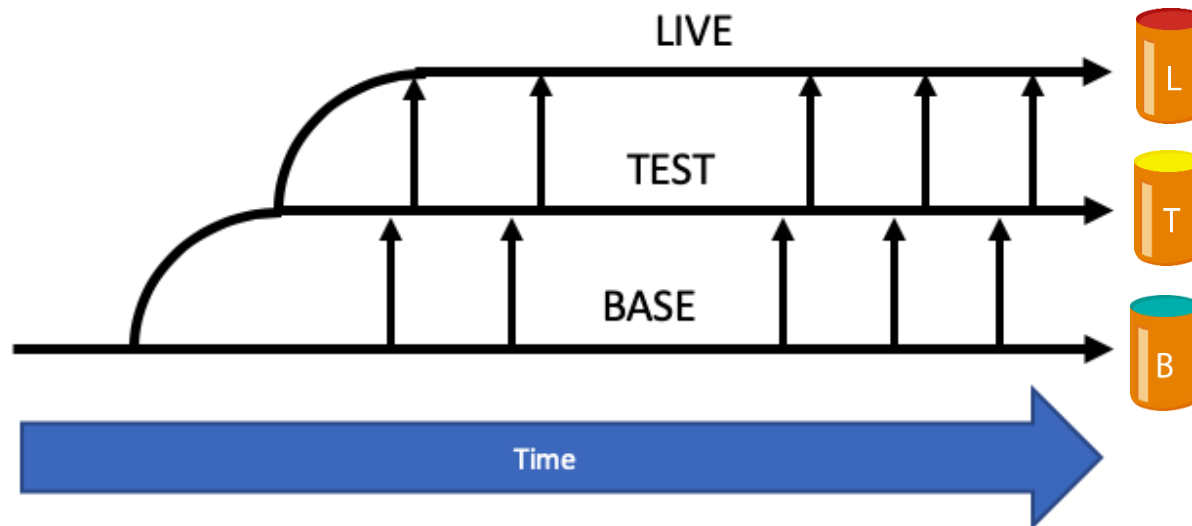
P4 Integrate

- Movement of changes (i.e., 'diff chunks') from one branch to another.
 - Source – the file doing the moving.
 - Target – place where the file is being moved; created if it doesn't exist.
 - Perforce keeps track of which versions have already been integrated between codelines/branches.
 - Won't look at them again in future integrations.



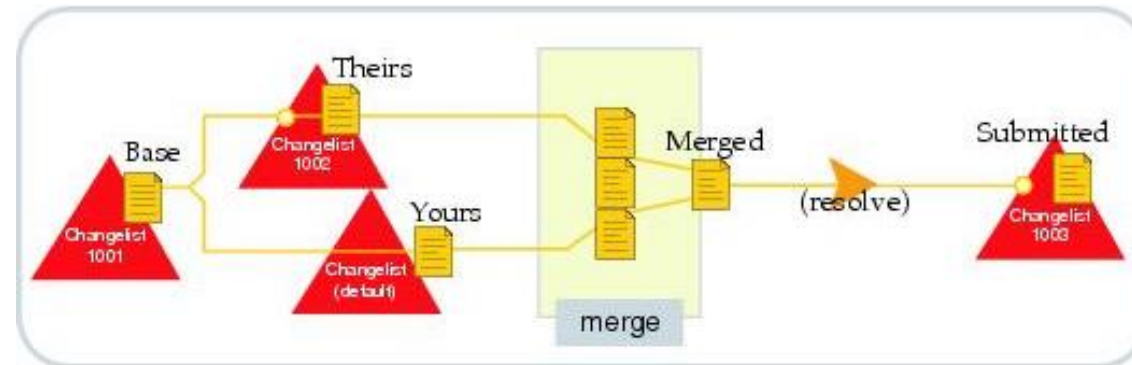
Codeline Management: Web Development Model

- Promotion common in web development.
 - The model is automated for environment branches via the use of CCR.
 - Changes only made manually in BASE and then integrated BASE → TEST → LIVE.



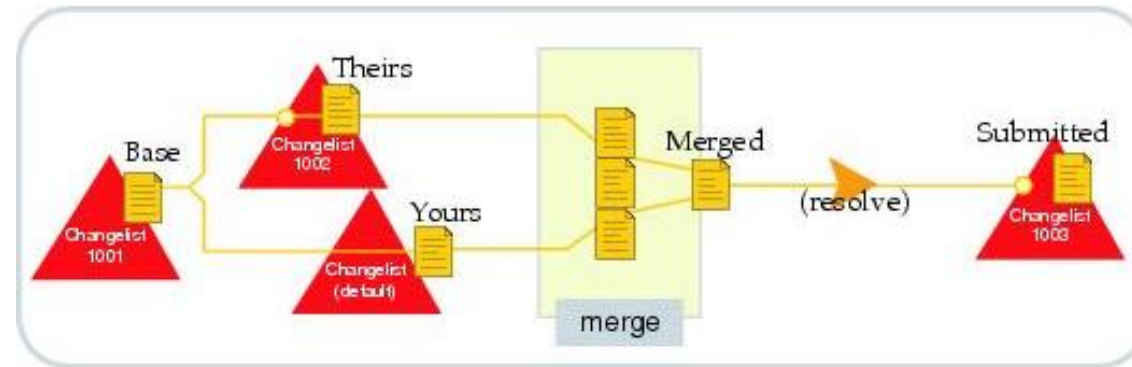
Resolve and Merge

- **Base:** The file revision, in conjunction with the source revision, used to help determine what integration changes should be applied to the target revision.
- **Theirs:** The revision in the depot with which the client file (your file) is merged when you resolve a file conflict. When you are working with branched files, theirs is the donor or **Source** file.



Resolve and Merge (cont.)

- **Yours:** The edited version of a file in your client workspace when you resolve a file. Also, the **Target** file when you integrate a branched file.
- P4Merge is the default tool used for merging files within Perforce.
- P4Merge training is a more advanced topic.



Exercise ICC410-5: Merge and Integrate



Quiz: Integration

Question:

Integrating from one file (source) to another (target) will always result in the head revision of the target file having content identical to that of the source file. True or False?

Answer:

False. "Accept Source" is one of the Resolve options when an integration takes place, but there is also "Accept Target" (bookkeeping integration), and "Accept Merged" which will only move diff chunks from the source into the target and not the complete file.



Part 7:

Tools for Developing and Maintaining Software



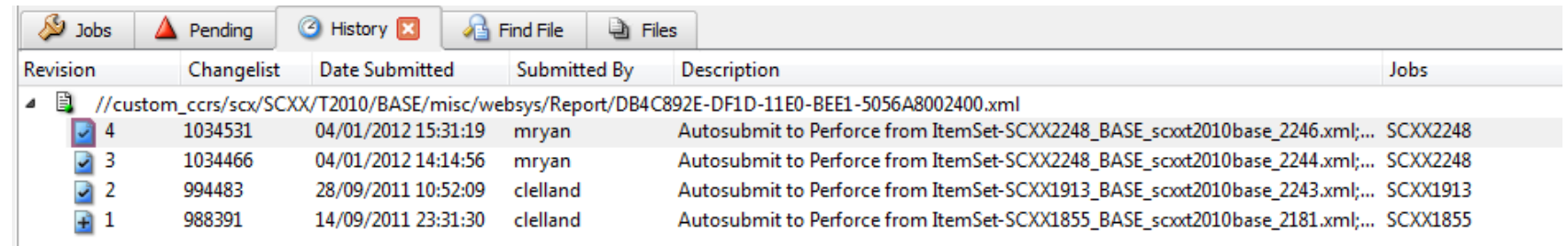
Perforce Glossary: Revision, Head Revision

- Revision.
 - The state of a file as it was checked in against a changelist.
- Head Revision.
 - The most current version of a file for a particular branch in the depot.



Change History

- File history.
 - How to: Click a file, then select History tab.
 - Submissions with timestamps.
 - Drag and drop comparison (Diff).
 - TIP: Jobs column is not shown by default but is very helpful!
 - Makes it clear what CCR/JIRA/Devlog was responsible for that revision.
 - Right-click on column headers and select "Jobs"; move column to the left for easier scanning.



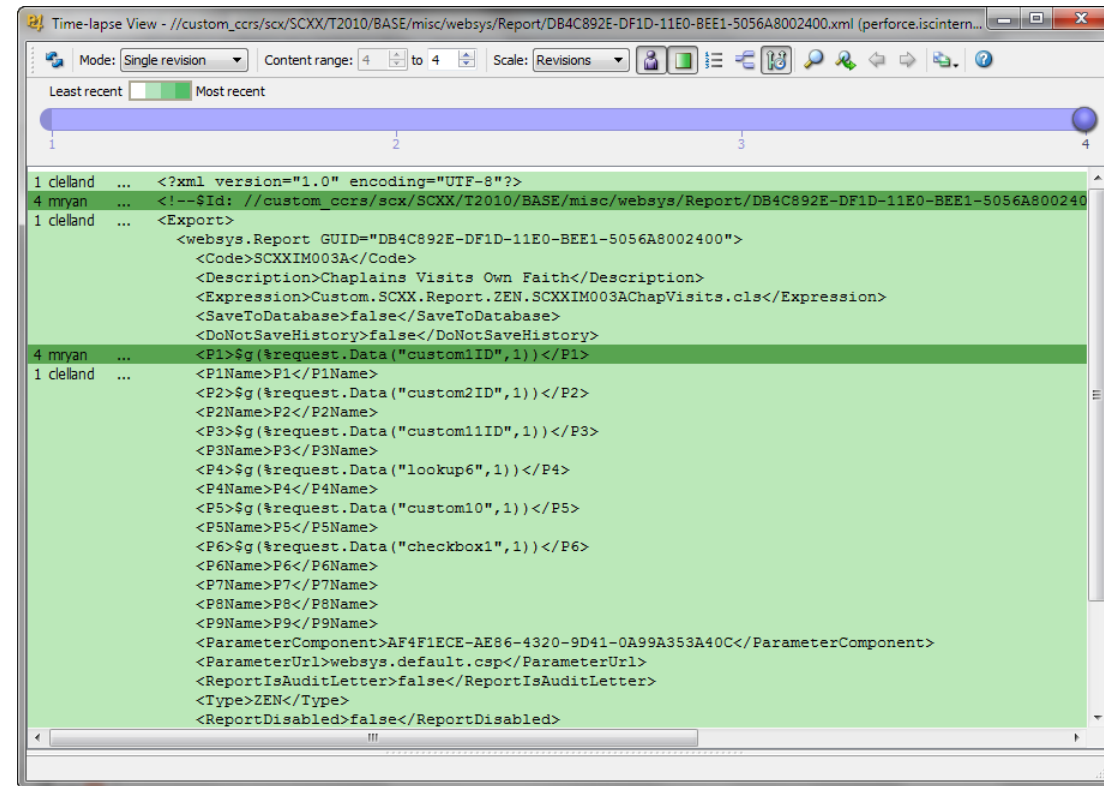
The screenshot shows a software window titled "History" with several tabs: "Jobs", "Pending", "History", "Find File", and "Files". The "History" tab is active, displaying a table with the following columns: Revision, Changelist, Date Submitted, Submitted By, Description, and Jobs. The table contains four rows of data, each representing a revision of a file. The file path is shown as expanded in the background.

Revision	Changelist	Date Submitted	Submitted By	Description	Jobs
4	1034531	04/01/2012 15:31:19	mryan	Autosubmit to Perforce from ItemSet-SCXX2248_BASE_scxxt2010base_2246.xml;...	SCXX2248
3	1034466	04/01/2012 14:14:56	mryan	Autosubmit to Perforce from ItemSet-SCXX2248_BASE_scxxt2010base_2244.xml;...	SCXX2248
2	994483	28/09/2011 10:52:09	clelland	Autosubmit to Perforce from ItemSet-SCXX1913_BASE_scxxt2010base_2243.xml;...	SCXX1913
1	988391	14/09/2011 23:31:30	clelland	Autosubmit to Perforce from ItemSet-SCXX1855_BASE_scxxt2010base_2181.xml;...	SCXX1855



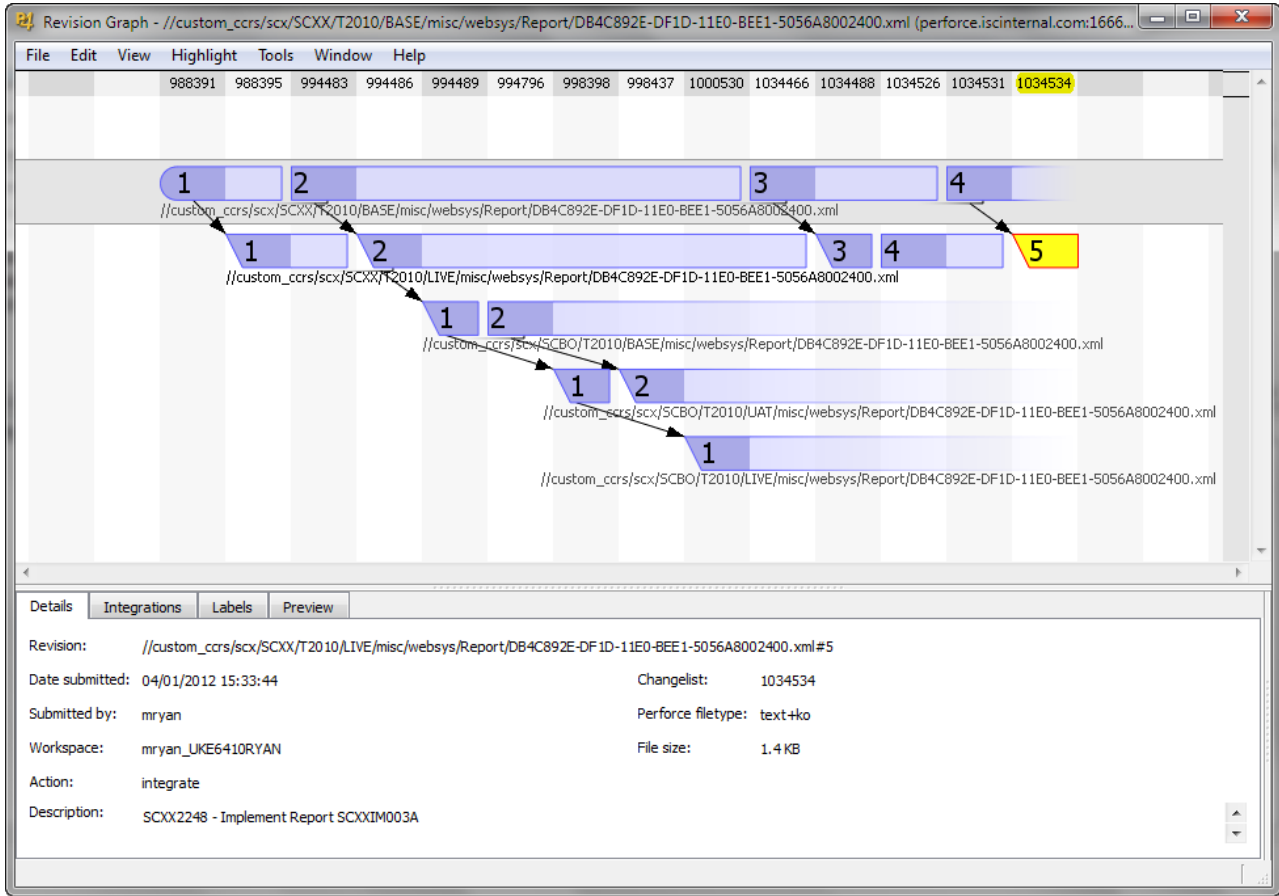
Time-lapse view

- How to: Right-click file and select “Timelapse” from top toolbar.
- Very useful tool for ‘archeology.’
- Enables digging through historical layers of the files to find when diff chunks were added.
 - Hopefully able to look up a linked job in CCR to identify why change made.



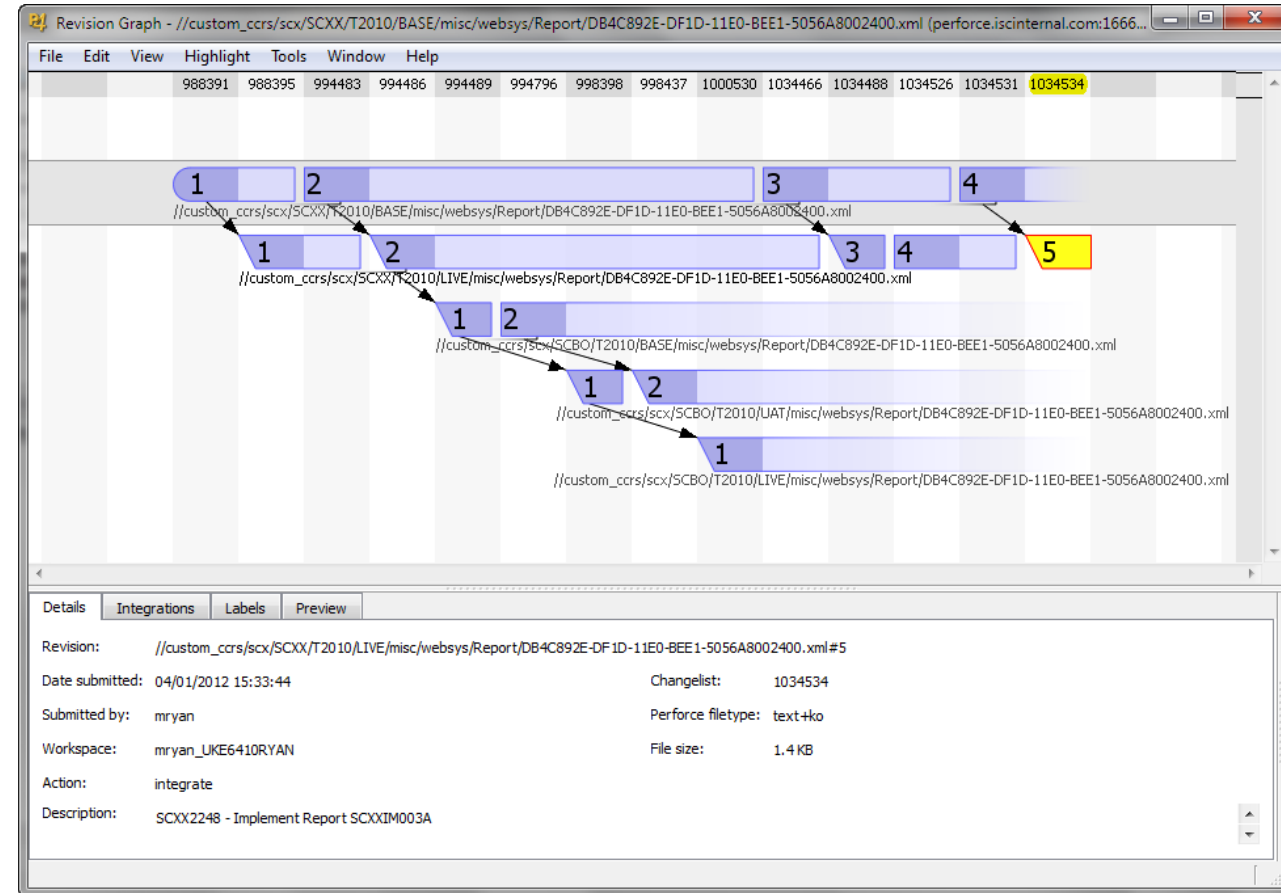
Revision Graph

- How to: Right-click file and select “Revision Graph” from top toolbar.
- Allows tracking integration history of different files between different branches.



Revision Graph (cont.)

- Required for determining if a change not yet progressed blocks the integration of another change.
- Powerful tool because any revision from any branch can be dragged onto any other revision of any branch to launch the diff tool.



Exercise ICC410-6: View Change History



Quiz: Historical Research

Question:

Which tools can be used to look “back in time” to determine historical details about a particular file or diff chunk in that file? Select all that apply.

- A. Revision Graph.
- B. Pending changelist.
- C. Time-lapse view.
- D. P4Merge.
- E. p4 grep.
- F. Change History.



Quiz: Historical Research (cont.)

Answer:

- A. Revision Graph.
- C. Time-lapse view.
- D. P4Merge.
- F. Change History.

Pending changelists aren't historical and p4 grep can help locate a file, but not provide historical details on a file.



Where to find information about Perforce

- Perforce client:
 - P4V help: Menu "Help > P4V Help [F1]".
 - p4 client help: Type p4 help at OS command line.
- Perforce website:
 - <http://www.perforce.com>.



Where to find information about Perforce (cont.)

- Perforce documentation:

- http://www.perforce.com/documentation/perforce_technical_documentation.
 - Introduction to Perforce & Getting started with P4V.
 - P4/P4V Cheat Sheet.
 - Perforce Command Reference, P4 User's Guide, P4 Administrator's Guide, etc...



Where to find information about Perforce (cont.)

- Other sources of documentation:
 - P4 knowledge database: <http://kb.perforce.com>.
 - P4 blog: <http://www.perforce.com/blog> (some useful tips and tricks).
 - P4 forums: <http://forums.perforce.com>.
 - Technical questions: <http://stackoverflow.com/questions/tagged/perforce>.
- Don't forget – Google can help you find material across all these sources!
- For InterSystems employees, the Perforce@intersystems.com mailing list is a great internal source of knowledge sharing.



Summary

- What are the key points for this module?

