INTERSYSTEMS LEARNING SERVICES

InterSystems Change Control

InterSystems Change Control

ICC 450: CCR Transport – Best Practices & Debugging Techniques

InterSystems
Creative data technology

Overview

- Additional changes past BASE phase.
- Revision History.
- Transport Logs.
- Manual integration within CCR.
- Perforce integration conflicts.
- Backing out.
- Fixing workflow issues.
- Updating Client Tools.

Part 1: Additional Changes Past BASE Phase

Additional Changes Past BASE Phase

- Cannot bundle and upload changes in or past BASE_Complete.
  - Changes not peer reviewed in BASE.
- To add changes past BASE_Complete either:
  - Use concept of catch-up CCR.
    - Strongly encouraged best practice.
    - Simplifies change record documentation.
  - Use backwards transition(s) to return to BASE.
    - Requires backout if past BASE phase.
    - Usually more work than catch-up CCR.

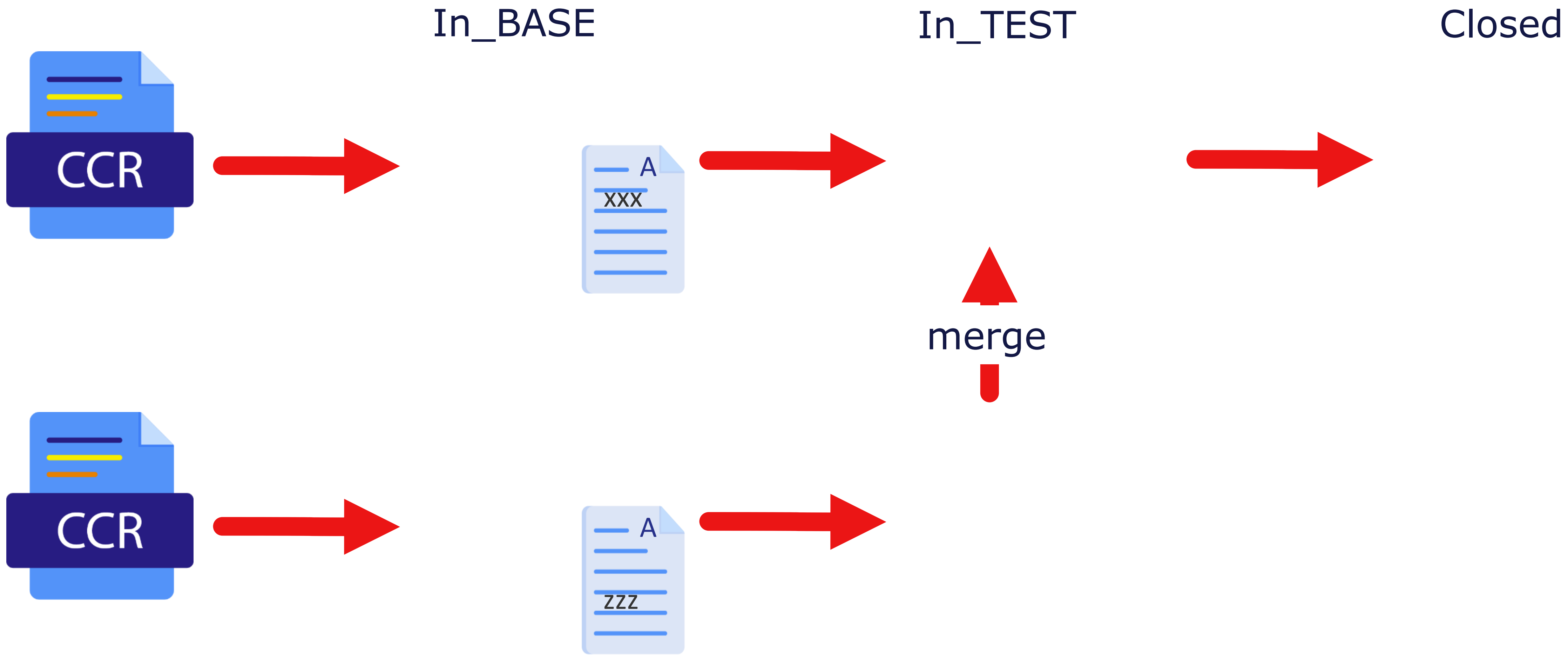
How To: Use Catch-up CCR

1. Create new CCR or clone original CCR and trim details as required.
   - Specify original CCR as Prerequisite CCR.
   - Use Title field to clarify it is a catch up CCR.
     - Example: “[Catch up to ISCX12345]: Fix typo in username label”
2. Progress to In_BASE state.
3. Make and upload changes.
   - Or upload changed items left behind.
4. Document, test, and progress catch-up CCR to same state as original CCR.

How To: Use Catch-up CCR (cont.)

5. Merge catch-up CCR into original CCR.
   - Source is catch-up CCR, target is original CCR.
   - Documentation and Perforce changelists from catch-up associated with original CCR.
6. Progress original CCR.
   - Will integrate all Perforce changes from original and catch-up CCR.

Example: catch-up CCR



Returning a CCR to BASE

- Backing out changes can cause merge conflicts.
  - Resolving merge conflicts time consuming.
    - Can introduce risk.
- Makes it more difficult to understand whether a change has ever made it to TEST or UAT environments.
  - Must use transition history instead of just noting state of the CCR.
- Do preview backout and verify no merge conflict before performing any backwards transition.

Quiz: Catch-up CCRs

Question:

Backout from TEST is always an easy, low-risk task.

True or False?

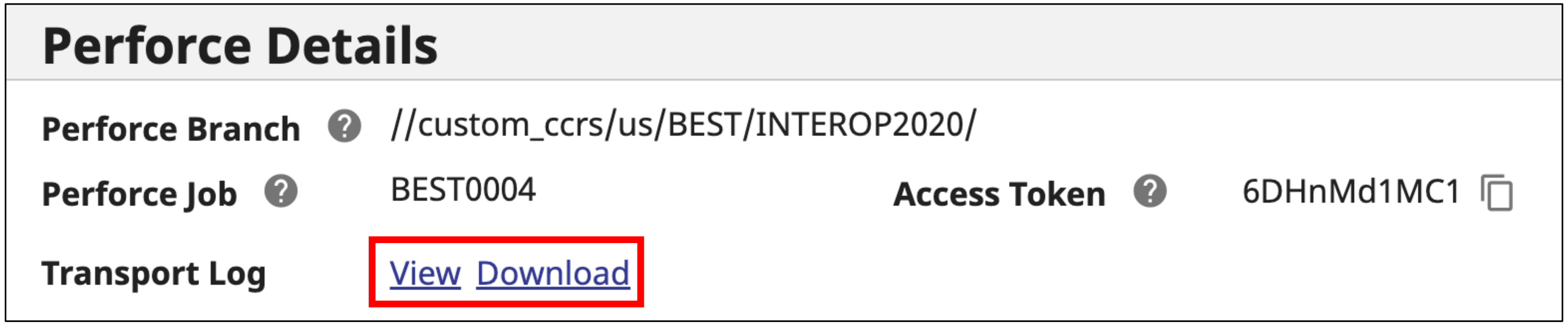
Answer:

False. Cannot easily backout if changes made in BASE on top of changes that need to be backed out.

Part 3: Using Transport Logs

Review: Transport Logs

- Contain record of Perforce and ItemSet activity for that CCR.
  - ItemSet uploads / commits.
  - Perforce integrations.
  - ItemSet creation for download.
  - ItemSet load log (uploaded from the client).
- To access Transport Log, click View or Download in Perforce Details.

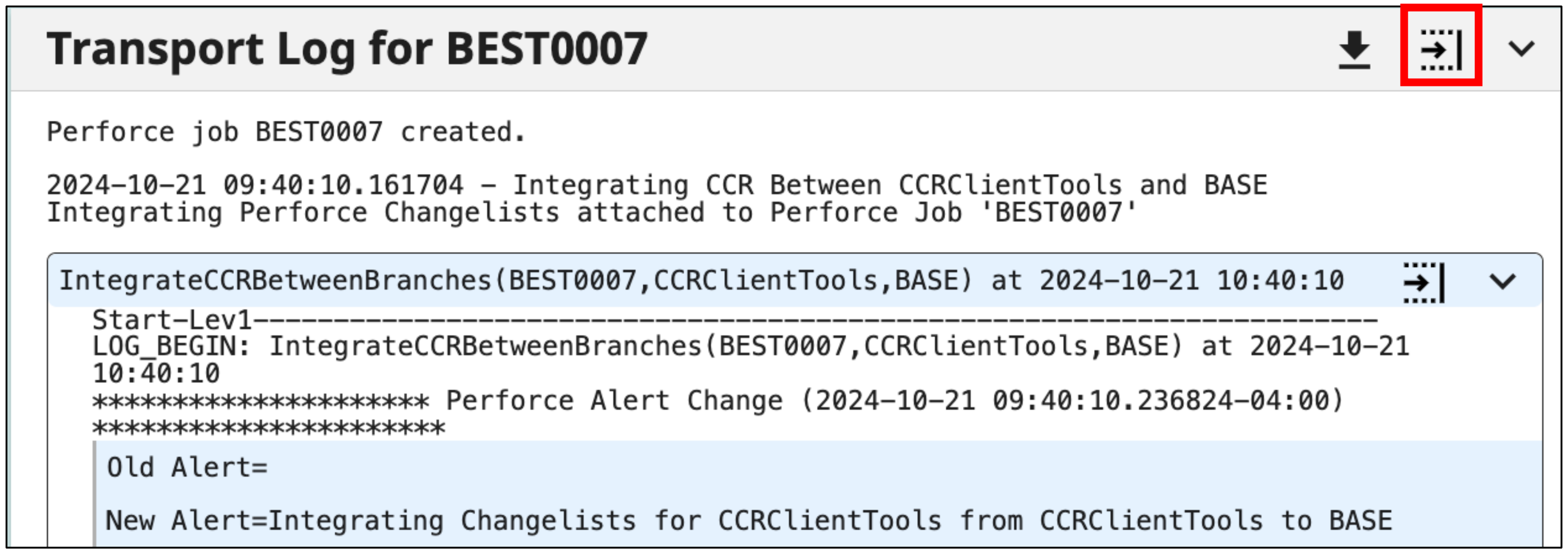


Perforce Details

Perforce Branch	//custom_ccrs/us/BEST/INTEROP2020/		
Perforce Job	BEST0004	**Access Token**	6DHnMd1MC1
Transport Log	View Download		

Viewing Transport Log

- Click jump to bottom button to access end of log.
  - Most recent entries at bottom.
- Reviewing log great way to better understand CCR automation.

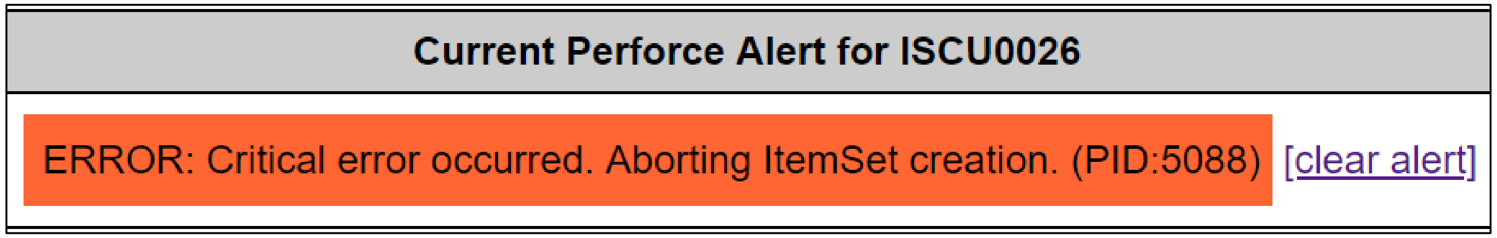


Transport Log for BEST0007

```
Perforce job BEST0007 created.

2024-10-21 09:40:10.161704 - Integrating CCR Between CCRClientTools and BASE
Integrating Perforce Changelists attached to Perforce Job 'BEST0007'

IntegrateCCRBetweenBranches(BEST0007,CCRClientTools,BASE) at 2024-10-21 10:40:10
  Start-Lev1--------------------------------------------------------------------
  LOG_BEGIN: IntegrateCCRBetweenBranches(BEST0007,CCRClientTools,BASE) at 2024-10-21
  10:40:10
  ********************* Perforce Alert Change (2024-10-21 09:40:10.236824-04:00)
  **********************
   Old Alert=

   New Alert=Integrating Changelists for CCRClientTools from CCRClientTools to BASE
```

Clear CCR Transport Alerts from Perforce Details

- Alerts in Perforce Details usually cleared through corrective action, such as successful integration.
- For alerts not cleared automatically, clear manually after corrective action completed:
  - Open transport log > click [clear alert].
- Cleared alerts:
  - Still available in transport log.
  - Inform other users that the alert is no longer an issue.



Current Perforce Alert for ISCU0026

ERROR: Critical error occurred. Aborting ItemSet creation. (PID:5088) [clear alert]

Quiz: Using Revision History and Transport Logs

Question:

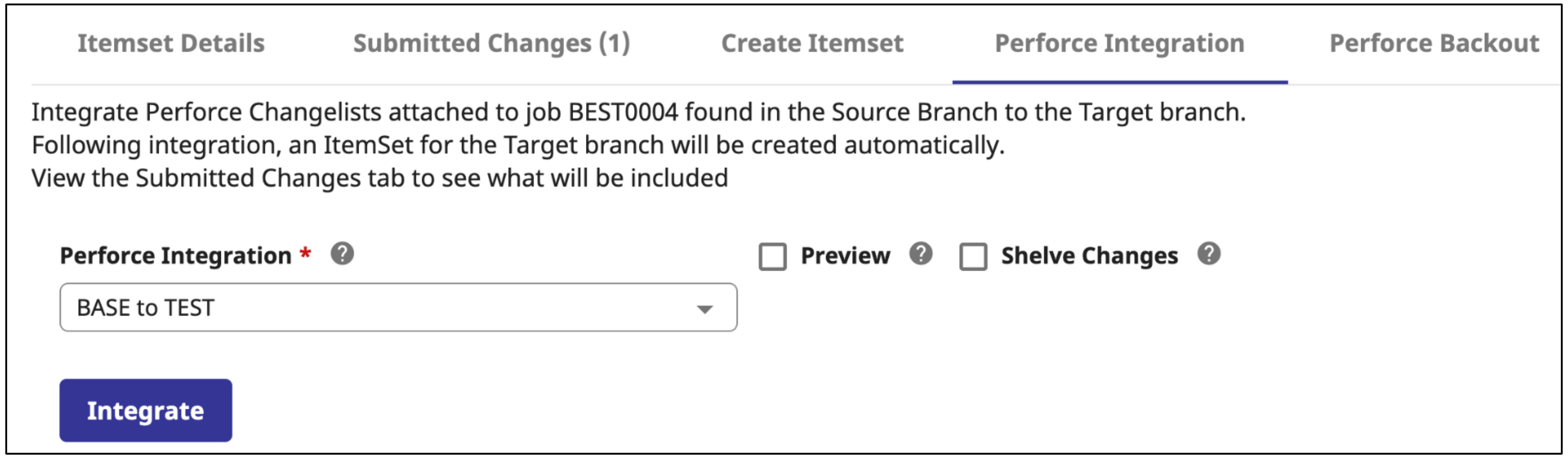
Transport Logs includes all logging which occurred by Perforce-related actions which occurred against that CCR or its ItemSets. True or False?

Answer: True.

Part 4: Manual Integration

How To: Manually Trigger Preview Integration

1. Perforce Details > Perforce Integration.
2. Verify menu for integration environments correct.
3. Select Preview checkbox.
   - Preview will not commit integration but rather return any predicted conflicts.
4. Click Integrate.



Integrate Perforce Changelists attached to job BEST0004 found in the Source Branch to the Target branch.
Following integration, an ItemSet for the Target branch will be created automatically.
View the Submitted Changes tab to see what will be included

Perforce Integration *: BASE to TEST

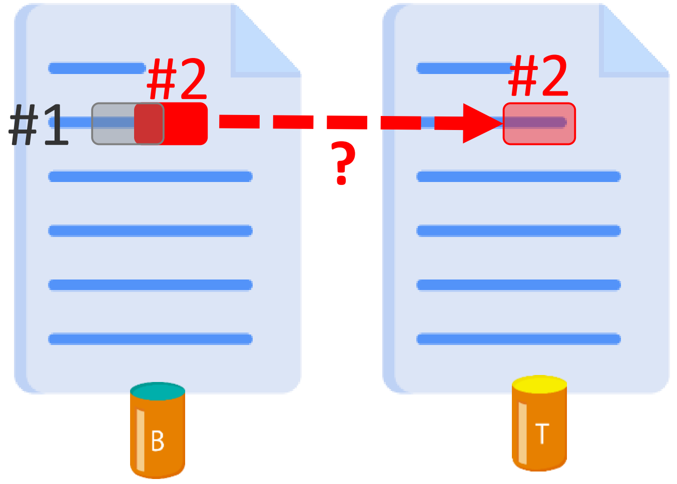
☐ Preview ☐ Shelve Changes

Part 5: Perforce Integration Conflicts

Quiz: Merge Conflicts

Question:

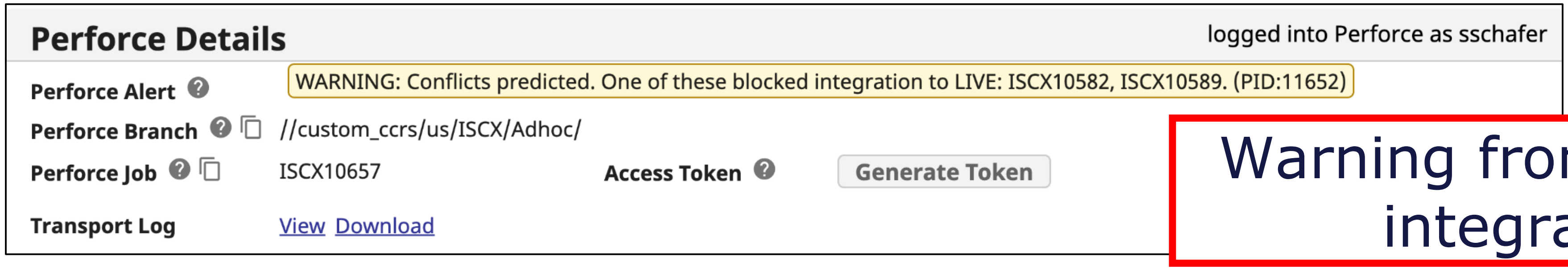
The following will cause a merge conflict. Change #1 is made but not progressed before #2 is made and progressed to TEST. True or False?



Answer: True.

Perforce Integration Failed Alert

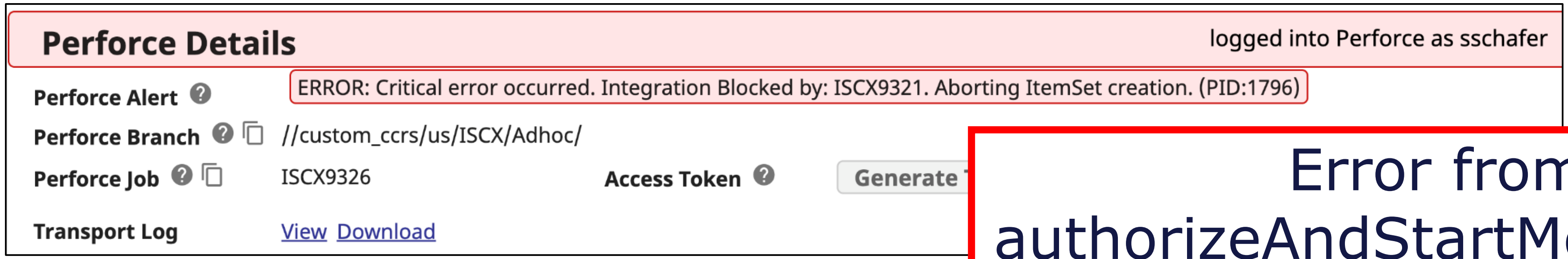
- When 1 CCR ID specified, that CCR is definitely the problem.
- When 2+ CCR IDs specified, at least 1 of them is the problem.



Perforce Details logged into Perforce as sschafer

Perforce Alert	WARNING: Conflicts predicted. One of these blocked integration to LIVE: ISCX10582, ISCX10589. (PID:11652)		
Perforce Branch	//custom_ccrs/us/ISCX/Adhoc/		
Perforce Job	ISCX10657	**Access Token**	Generate Token
Transport Log	View Download		

Warning from preview integration



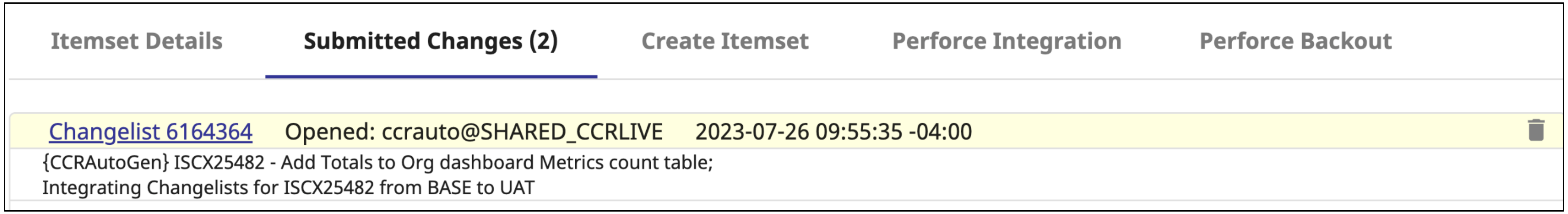
Perforce Details logged into Perforce as sschafer

Perforce Alert	ERROR: Critical error occurred. Integration Blocked by: ISCX9321. Aborting ItemSet creation. (PID:1796)		
Perforce Branch	//custom_ccrs/us/ISCX/Adhoc/		
Perforce Job	ISCX9326	**Access Token**	Generate
Transport Log	View Download		

Error from authorizeAndStartMoveToTEST

Perforce Integration Failed (cont.)

- Non-preview integrations shelved for InterSystems support.
  - Shelving = temporarily stored but not committed.
  - Can safely ignore shelved changelists.

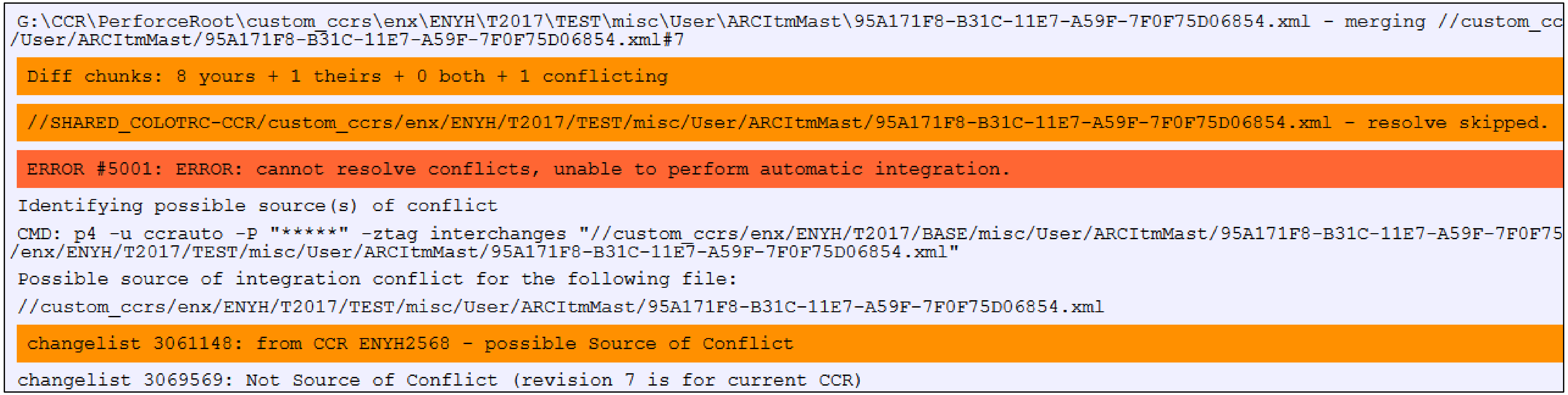


Changelist 6164364 Opened: ccrauto@SHARED_CCRLIVE 2023-07-26 09:55:35 -04:00

{CCRAutoGen} ISCX25482 - Add Totals to Org dashboard Metrics count table;
Integrating Changelists for ISCX25482 from BASE to UAT

Transport Log for Perforce Integration Errors

- Transport log contains additional information on perforce errors.
- Indicates number of conflicting diff chunks for Perforce integration errors.



```
G:\CCR\PerforceRoot\custom_ccrs\enx\ENYH\T2017\TEST\misc\User\ARCItmMast\95A171F8-B31C-11E7-A59F-7F0F75D06854.xml - merging //custom_cc
/User/ARCItmMast/95A171F8-B31C-11E7-A59F-7F0F75D06854.xml#7
Diff chunks: 8 yours + 1 theirs + 0 both + 1 conflicting
//SHARED_COLOTRC-CCR/custom_ccrs/enx/ENYH/T2017/TEST/misc/User/ARCItmMast/95A171F8-B31C-11E7-A59F-7F0F75D06854.xml - resolve skipped.
ERROR #5001: ERROR: cannot resolve conflicts, unable to perform automatic integration.
Identifying possible source(s) of conflict
CMD: p4 -u ccrauto -P "*****" -ztag interchanges "//custom_ccrs/enx/ENYH/T2017/BASE/misc/User/ARCItmMast/95A171F8-B31C-11E7-A59F-7F0F75
/enx/ENYH/T2017/TEST/misc/User/ARCItmMast/95A171F8-B31C-11E7-A59F-7F0F75D06854.xml"
Possible source of integration conflict for the following file:
//custom_ccrs/enx/ENYH/T2017/TEST/misc/User/ARCItmMast/95A171F8-B31C-11E7-A59F-7F0F75D06854.xml
changelist 3061148: from CCR ENYH2568 - possible Source of Conflict
changelist 3069569: Not Source of Conflict (revision 7 is for current CCR)
```

Merge Conflict: Solution

- Solution depends on whether:
  - Merge conflict discovered on preview integration.
  - AuthorizeAndStartMoveToXXXX transition.

Merge Conflict on Preview Integration: Solution

1. Edit CCR A to specify CCR B as prerequisite.
2. Progress CCR B ahead of CCR A in terms of phase.

Merge Conflict on authorizeAndStartMoveToXXXX: Solution

- For this example, merge conflict occurred when CCR A performed authorizeAndStartMoveToTEST.
  - CCR B specified as blocking CCR A.

1. Perform markIntegrationFailed transition on CCR A.
2. Edit CCR A to specify CCR B as prerequisite.
3. Progress CCR B.
4. Optional: Perform preview integration on CCR A to confirm all merge conflicts resolved.

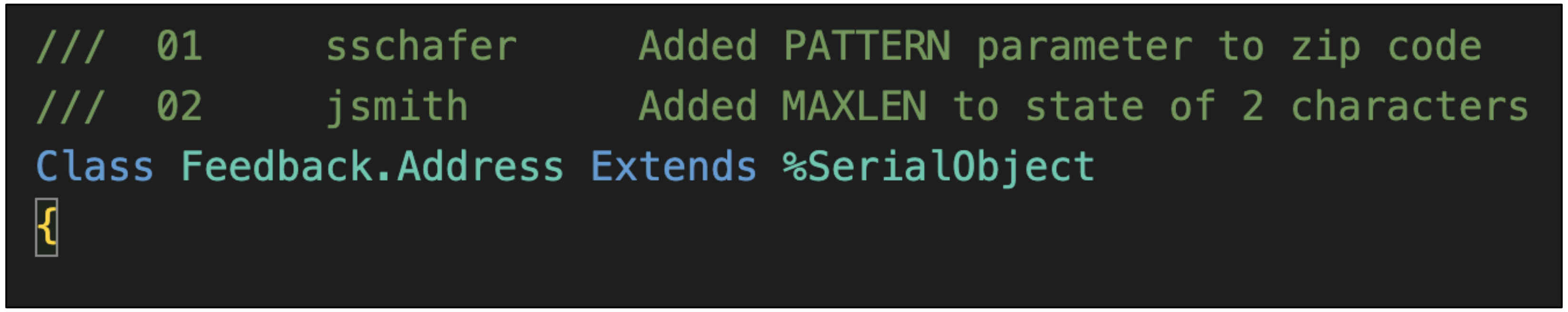
4. Progress CCR A.
   - CCR B must progress ahead of CCR A in terms of phase.

Circular Dependency

- Cause:
  - 2+ CCRs dependent on each other.
  - 2 CCRs both in In_BASE state and the following happens chronologically.
    - CCR A makes change.
    - CCR B makes overlapping change.
    - CCR A makes change that overlaps with a change in CCR B.
- Solution:
  - Merge CCRs.
  - To continue progress CCRs individually, manual intervention by InterSystems support might be possible.

Class Definition Descriptions and Merge Conflicts

- Do not use class descriptions for change documentation.
  - Class descriptions with change documentation can cause merge conflicts.
  - Keep all change documentation within CCR.
- Example of bad usage of class descriptions for change documentation:



```
///  01     sschafer     Added PATTERN parameter to zip code
///  02     jsmith       Added MAXLEN to state of 2 characters
Class Feedback.Address Extends %SerialObject
{
```

Progressing Edit before Add

- Cannot progress a CCR editing an item ahead of the CCR adding the same item.
- Error on preview integration:
  - WARNING: Prerequisite CCR detected.
  - Solution: Stop. Do not perform authorizeAndStartMoveToXXXX until progress CCR with add action to next Environment.
- Error on authorizeAndStartMoveToXXXX:
  - Critical error occurred. Another CCR has been detected as a prerequisite and marked as such. Aborting ItemSet creation.
  - Solution: Follow steps for merge conflict resolution.

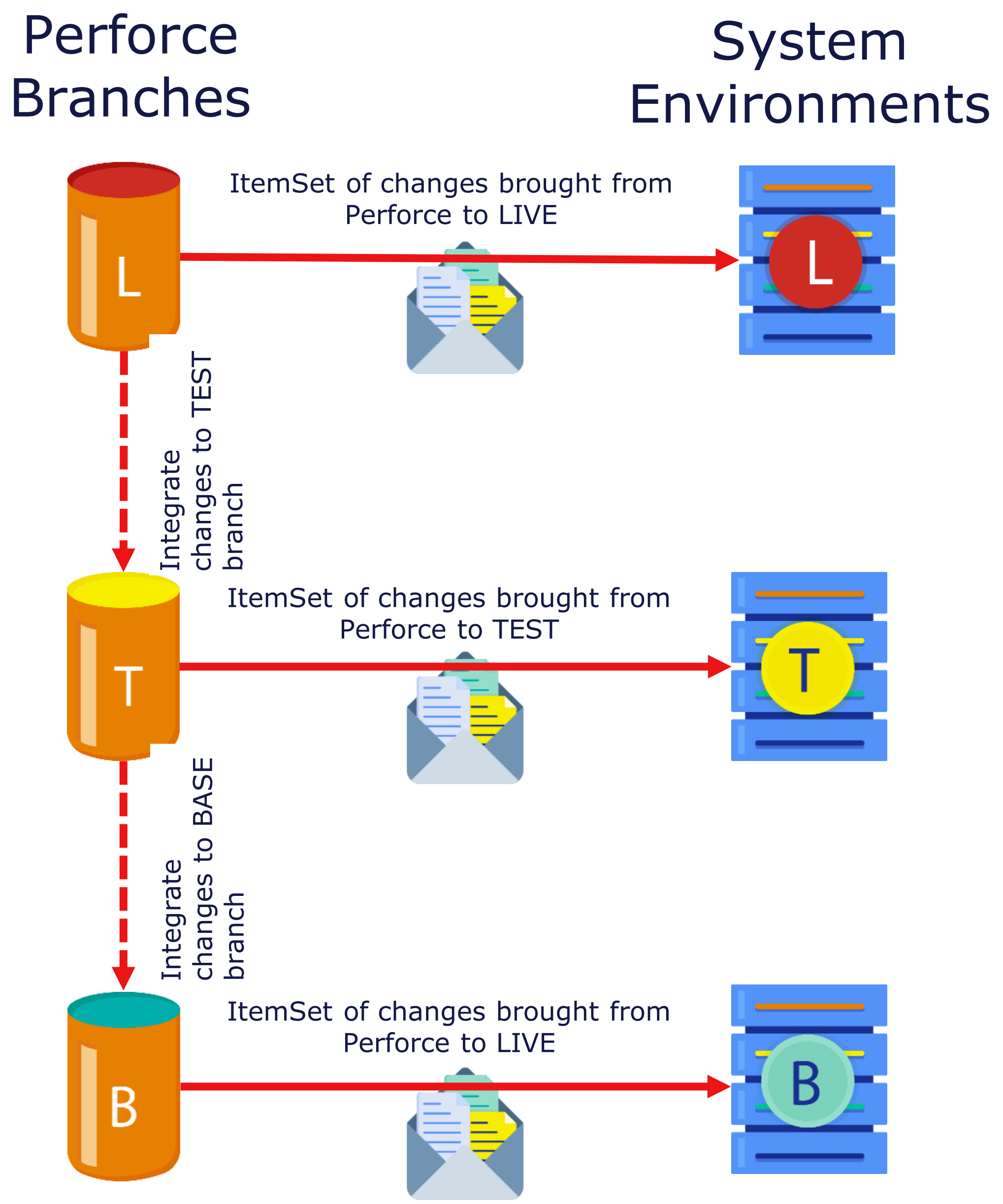
Progressing Edit before Add: Why It Cannot Be Allowed

- Integrating edit before add causes edit action to become branch action (similar to add action).
- Backing out branch action causes deletion.
- Therefore, allowing edit to progress first would cause unintentional deletion of item if backout necessary.
- Only CCR intending to add item should be able to delete item during backout.

Part 6: Backing Out

Backout Overview

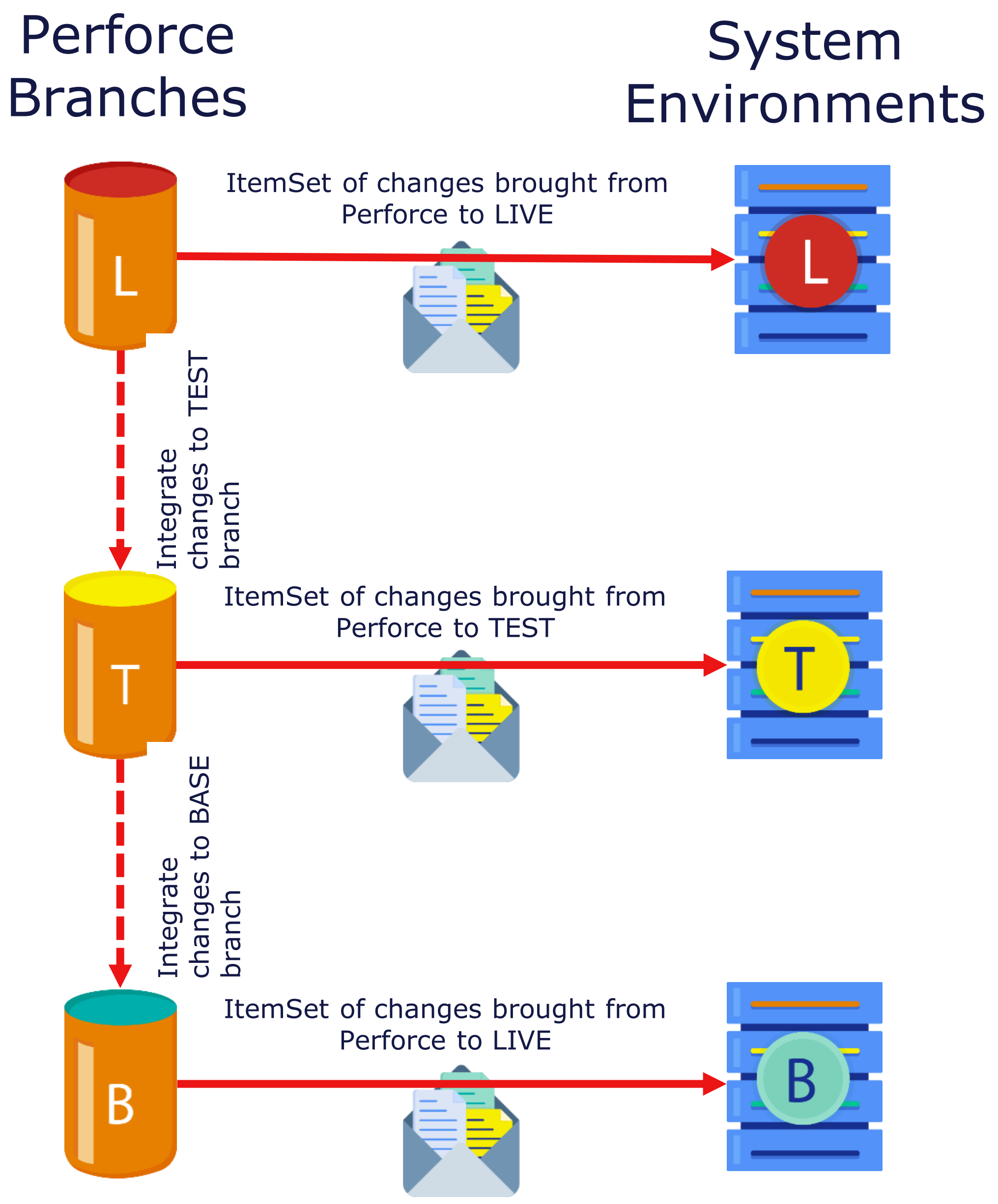
- Back out changelists from highest Perforce branch to which the change progressed.
- Integrate to other Perforce branches.
- Create ItemSets based on Perforce branches for each environment.
- Deploy ItemSets for each environment.



Perforce Branches
System Environments
L
ItemSet of changes brought from Perforce to LIVE
L
Integrate changes to TEST branch
T
ItemSet of changes brought from Perforce to TEST
T
Integrate changes to BASE branch
B
ItemSet of changes brought from Perforce to LIVE
B

Backout Overview (cont.)

- 3 ways to backout:
  - Automatically triggered.
  - Perforce Details section of CCR.
  - Direct Perforce intervention.
    - Covered in ICC460.


Perforce Branches
System Environments
L
ItemSet of changes brought from Perforce to LIVE
L
Integrate changes to TEST branch
T
ItemSet of changes brought from Perforce to TEST
T
Integrate changes to BASE branch
B
ItemSet of changes brought from Perforce to LIVE
B

Automatic Backout

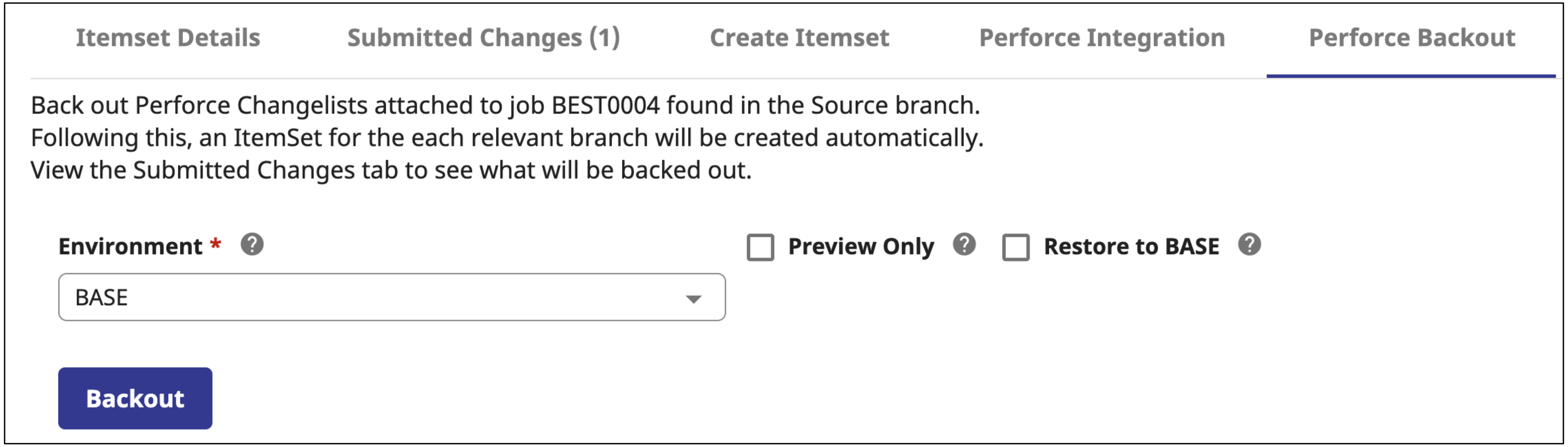
- CCR automatically backs out all changes from all environments for:
  - Tier 1 CCRs only.
  - cancel and changeSpec transitions.
- ItemSets for each affected environment generated if Perforce backout successful.
  - Deploy all generated ItemSets like any other ItemSet.
- Progress CCR through next transition after successful ItemSet deployment.

Back Out Without Automatic Backout

- For Tier 2 CCRs, see additional considerations in ICC615.
- For markValidationFailed transition on Tier 1 CCRs, must use manual controls in Perforce Details pane.
  - Successful backout using these controls generates ItemSets for all affected environments.
    - Make sure to deploy all ItemSets!
- In the event of backout errors, contact support.
  - They may need to intervene using p4v.

Manual Backout

- Perforce Details > Perforce Backout.



Back out Perforce Changelists attached to job BEST0004 found in the Source branch.
Following this, an ItemSet for the each relevant branch will be created automatically.
View the Submitted Changes tab to see what will be backed out.

Environment * : BASE

Preview Only [ ]   Restore to BASE [ ]

Manual Backout (cont.)

- Choose furthest Environment to which changes progressed
  - Default usually appropriate.
  - Automatically integrates backout to prior environments and creates ItemSets for all affected Environments.
    - Example: Choosing TEST will backout from and create ItemSets for TEST and BASE.
- Can choose Preview Only to check for merge conflicts.

Manual Backout: Restore to BASE Flag

- Cleared.
  - Removes changes related to this CCR from all environments.
  - Use for cancel or changeSpec.
- Selected.
  - Maintain changes related to this CCR in BASE.
  - Integrates backout changelist to BASE then backs it out to reintroduce change.
  - Use for markValidationFailed.
  - Still generates ItemSet that must be deployed.
    - Version information of items updated.

Quiz: Back out

Question:

Which 1 of the following is FALSE about backing out a CCR correctly?

A. Requires updating the corresponding Perforce branches.

B. Requires deploying ItemSets, including an ItemSet for BASE.

C. It is not necessary when cancelling a CCR that did not enter LIVE phase.

D. Requires performing transitions on a CCR.

E. Only users with adequate CCR knowledge should backout CCRs.

Quiz: Back out (cont.)

Answer:

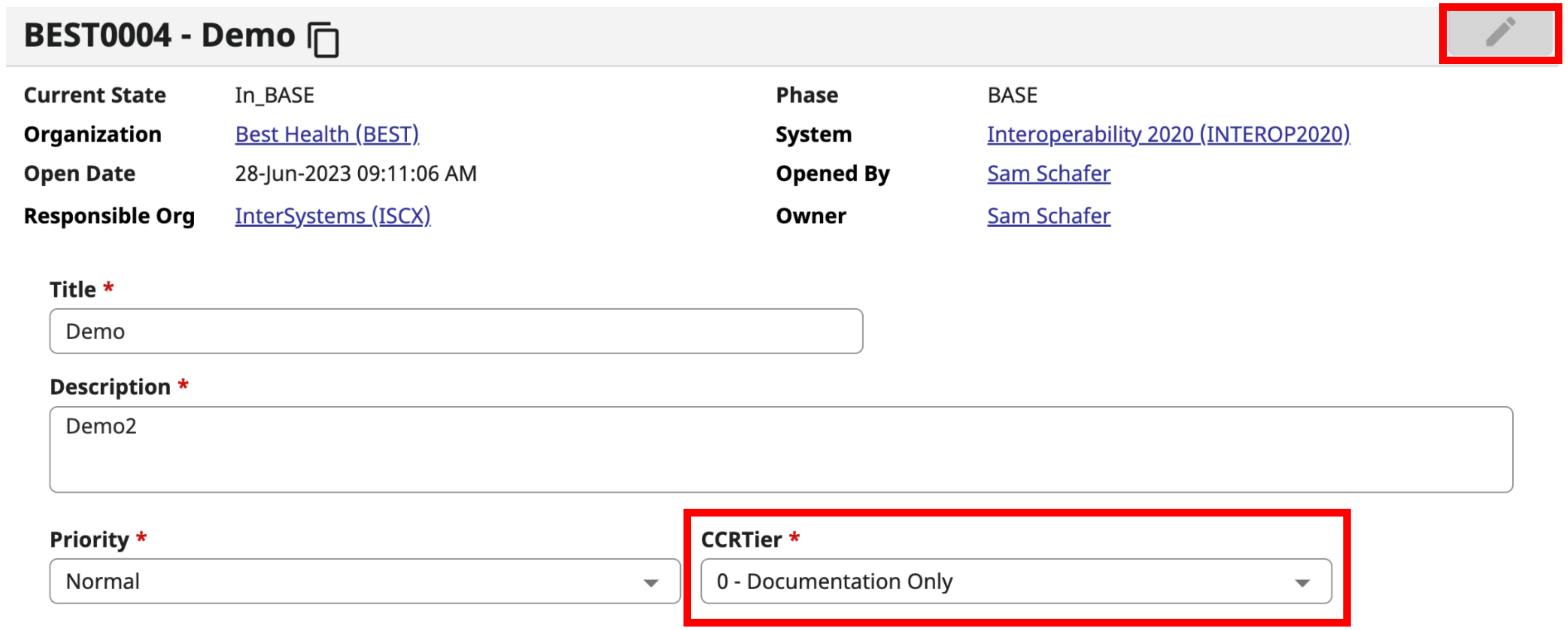
C. It is not necessary when cancelling a CCR that did not enter LIVE phase.

It is always necessary when cancelling a CCR.

Part 7: Fixing Workflow Issues

Reassigning Tier Level

- Workflow differs between CCR Tiers.
- Click pencil icon to fix CCR Tier if wrong Tier specified during creation.



BEST0004 - Demo

Current State	In_BASE	**Phase**	BASE
Organization	Best Health (BEST)	**System**	Interoperability 2020 (INTEROP2020)
Open Date	28-Jun-2023 09:11:06 AM	**Opened By**	Sam Schafer
Responsible Org	InterSystems (ISCX)	**Owner**	Sam Schafer

Title *
Demo

Description *
Demo2

Priority *
Normal

CCRTier *
0 - Documentation Only

Upload to Wrong CCR

- Perforce changelists associated with CCRs using Perforce Job field.
  - Job for each changelist set to ID of CCR.
- CCR application does not have functionality to change job of changelists.
  - Possible using p4v or Swarm.
    - Requires access to internal InterSystems network.
- Therefore, customers must contact InterSytems when specify wrong CCR ID on Bundle and Upload screen.

Quiz: Reassigning Tier Level

Question:

You have to be the CCR owner in order to change the Tier level. True or False?

Answer:

False.

There is no requirement to assign a CCR to one's self to edit it. The Tier can be edited and changed by anyone. However, it is probably best to have the owner make changes to CCRs.

Part 8: Additional Information

- Updating Client Tools.
- Leveraging revision history.
- Perforce alert: editing an item that does not exist in Perforce.

Maintaining CCR Client Tools

- Always update CCR Client Tools through same System.
  - Maintains accurate version history in Perforce.
- Best practice to configure separate System for %SYS namespace for:
  - Changes that affect entire instance such as hardware or memory allocations.
  - Maintaining CCR Client Tools.
- Alternatively, choose 1 existing System for such updates.
  - Example: HSCUSTOM on HealthShare systems.

How To: Update CCR Client Tools

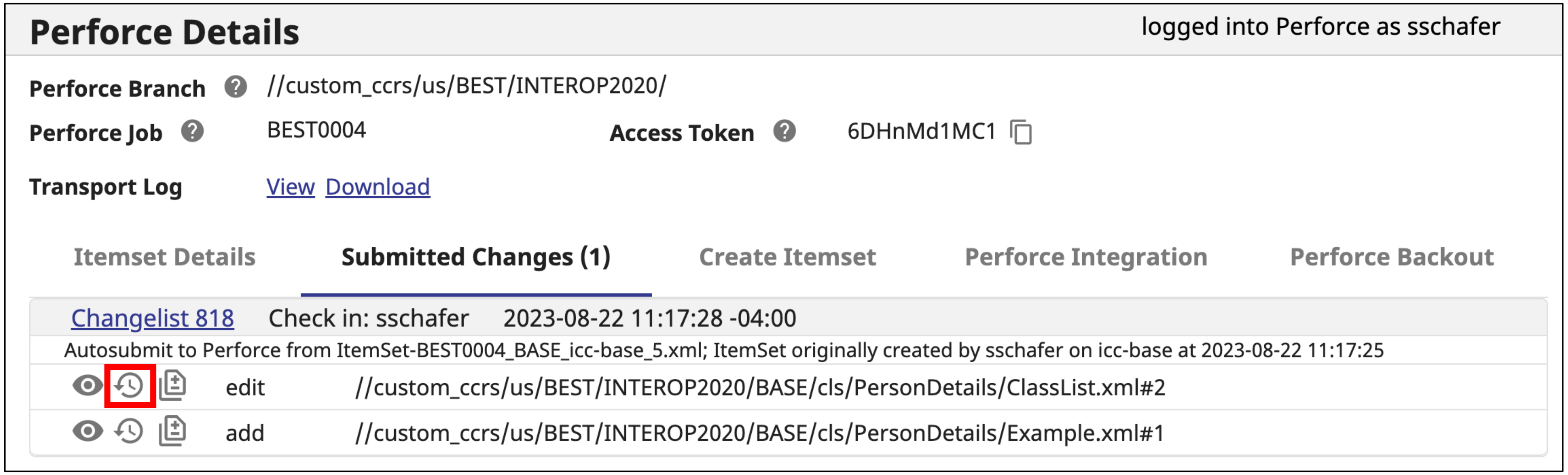
- Go to appropriate System Details page.
- Click Update Client Tools button.
  - Can only be actioned by Perforce users.
  - Creates Tier 1 CCR and integrates current client tools to BASE branch of System.
- Progress generated CCR through normal workflow.
  - Make sure to deploy generated ItemSet to BASE.

Revision History

- For file chosen, enumerates:
  - Each revision to item.
  - CCR ID associated with each changelist.
  - Whether that CCR is active, along with other CCR details.
  - Active = not in Closed, Merged, or Cancelled state.
- Useful for:
  - Identifying other active CCRs which may collide with current CCR.
  - Understanding how frequently an item is changed.
  - Spotting anomalies in the Revision History.
    - Example: changelist without job which could break future integration.

Access Revision History from CCR

1. Open CCR of interest.
2. Click Submitted Changes tab.
3. Click next to item of interest.



Perforce Details logged into Perforce as sschafer

Perforce Branch	//custom_ccrs/us/BEST/INTEROP2020/
Perforce Job	BEST0004
Access Token	6DHnMd1MC1
Transport Log	View Download

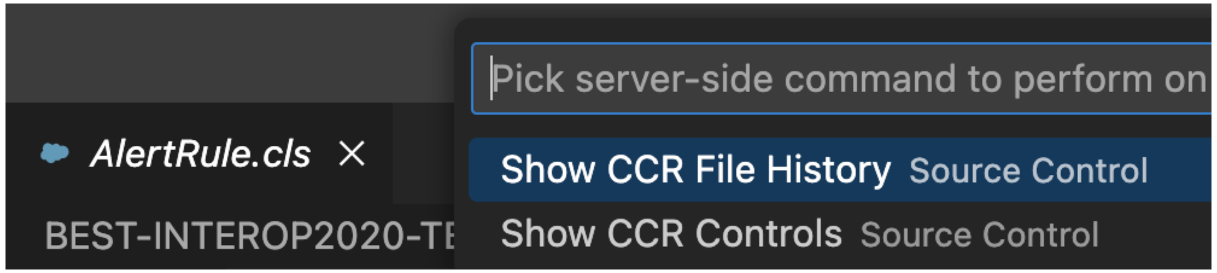
Changelist 818 Check in: sschafer 2023-08-22 11:17:28 -04:00

Autosubmit to Perforce from ItemSet-BEST0004_BASE_icc-base_5.xml; ItemSet originally created by sschafer on icc-base at 2023-08-22 11:17:25

edit	//custom_ccrs/us/BEST/INTEROP2020/BASE/cls/PersonDetails/ClassList.xml#2
add	//custom_ccrs/us/BEST/INTEROP2020/BASE/cls/PersonDetails/Example.xml#1

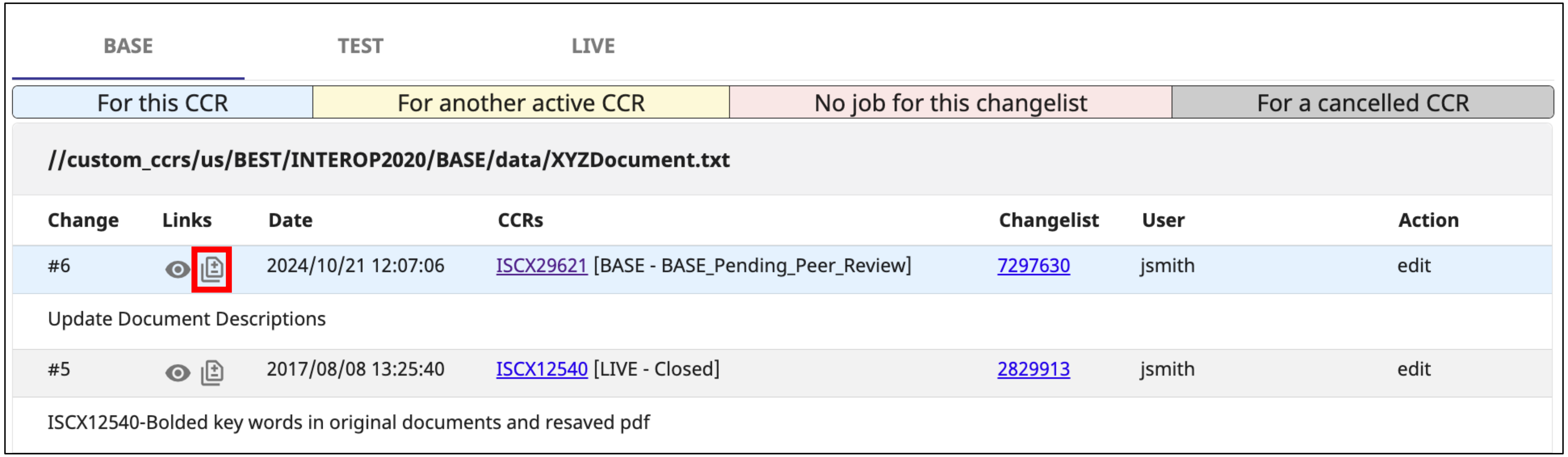
Access Revision History from IDE

1. Open file of interest in IDE.
2. Source Control menu > Show CCR File History.


AlertRule.cls
BEST-INTEROP2020-TE
Pick server-side command to perform on
Show CCR File History Source Control
Show CCR Controls Source Control

Revision History Functionality

- Color-coded, with the legend at top.
- Click diff icon to diff that revision versus previous revision.



BASE TEST LIVE

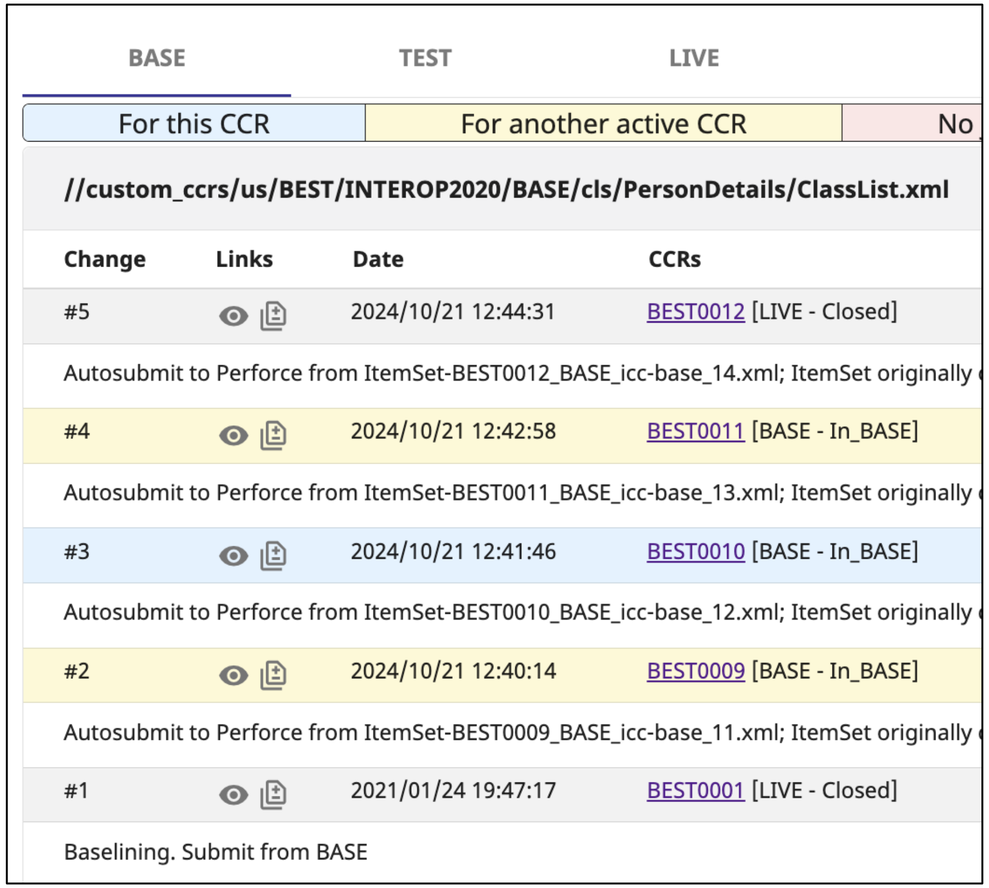
For this CCR	For another active CCR	No job for this changelist	For a cancelled CCR

//custom_ccrs/us/BEST/INTEROP2020/BASE/data/XYZDocument.txt

Change	Links	Date	CCRs	Changelist	User	Action
#6		2024/10/21 12:07:06	ISCX29621 [BASE - BASE_Pending_Peer_Review]	7297630	jsmith	edit
Update Document Descriptions						
#5		2017/08/08 13:25:40	ISCX12540 [LIVE - Closed]	2829913	jsmith	edit
ISCX12540-Bolded key words in original documents and resaved pdf						

Understanding Revision History

- Any active CCR that created revisions prior to your CCR might cause conflicts.
- Blue revision #3 is CCR used to access history.
- Closed CCRs will not cause conflicts.
- Possible BEST0009 blocks this CCR from progressing to TEST.
  - Would block if overlapping changes.
  - Note that CCR is in In_BASE state therefore has not yet progressed to TEST.
- Possible this CCR blocks BEST0011 from progressing to TEST.



//custom_ccrs/us/BEST/INTEROP2020/BASE/cls/PersonDetails/ClassList.xml

Change	Links	Date	CCRs
#5		2024/10/21 12:44:31	BEST0012 [LIVE - Closed]
Autosubmit to Perforce from ItemSet-BEST0012_BASE_icc-base_14.xml; ItemSet originally			
#4		2024/10/21 12:42:58	BEST0011 [BASE - In_BASE]
Autosubmit to Perforce from ItemSet-BEST0011_BASE_icc-base_13.xml; ItemSet originally			
#3		2024/10/21 12:41:46	BEST0010 [BASE - In_BASE]
Autosubmit to Perforce from ItemSet-BEST0010_BASE_icc-base_12.xml; ItemSet originally			
#2		2024/10/21 12:40:14	BEST0009 [BASE - In_BASE]
Autosubmit to Perforce from ItemSet-BEST0009_BASE_icc-base_11.xml; ItemSet originally			
#1		2021/01/24 19:47:17	BEST0001 [LIVE - Closed]
Baselining. Submit from BASE			

Quiz: Using Revision History and Transport Logs

Question:

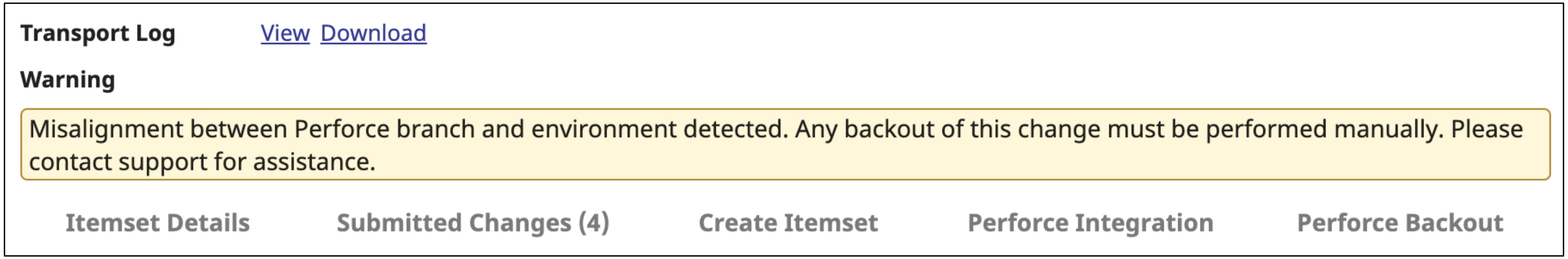
The ‘history’ link under show submitted changes has links to show diffs of the current version against every other version. True or False?

Answer: False.

Revision History only has diffs between versions, not the current version against all other versions. Within CCR, it is only possible to diff any version against any other version from the view page.

Editing Item that Does Not Exist in Perforce

- ItemSets identify whether changes are edits, adds, or deletes to items.
- Perforce changes edit to add if item does not exist in Perforce.
- Warning message appears on CCR.
  - Warning: Misalignment between Perforce branch and environment detected. Any backout of this change must be performed manually. Please contact support for assistance.



Transport Log View Download

Warning

Misalignment between Perforce branch and environment detected. Any backout of this change must be performed manually. Please contact support for assistance.

Editing Item that Does Not Exist in Perforce (cont.)

- Does not block progression of CCR towards Closed.
- Progressing CCR risky because cannot backout without knowledge of change.
  - Perforce has no version history to identify changes.
  - Backout would cause deletion from Perforce and Environment.
    - Automatic backout blocked.
- Contact support if warning shown on any CCR.
  - May need to create a new baseline.
  - Proper baseline prevents ever seeing this warning.

Summary

- What are the key points for this module?