



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 Memorial Drive, Cambridge, MA 02142, 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 450: CCR Transport – Best Practices & Debugging Techniques



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

- Reasons need to make or upload additional changes past BASE include:
 - Neglected to bundle and upload all changed items.
 - Minor typos or fixes necessary in code.
- Cannot bundle and upload changes against CCR in or past BASE_Complete.
 - Enforced in current client tool version (2019-Dec-12).
 - Reasons include:
 - Changes not peer reviewed in BASE.
 - If past authorizeAndStartMoveToTEST, changes not automatically integrated.



Additional Changes Past BASE Phase (cont.)

- Solution, either:
 - Backwards transition(s) to return to BASE.
 - Requires backout if past BASE phase.
 - Use concept of catch-up CCR.
 - No special type of CCR, only conceptual.
 - Ensures additional changes follow proper workflow, including peer review and integration.



Additional Changes Past BASE Phase (cont.)

- Similar to progressing changes, backing out changes can cause merge conflicts.
 - Resolving merge conflicts time consuming and, depending complexity, introduces risk.
 - Makes catch-up CCR more desirable.
- If do not want to use catch-up CCR:
 - Do preview backout and verify no merge conflict before performing any backwards transition.



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.
 - Consider using Title field to designate as catch up.
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.



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 2: Leveraging Revision History




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](#)

Itemset Details

Submitted Changes (1)




Create Itemset

Perforce Integration




Perforce Backout

[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

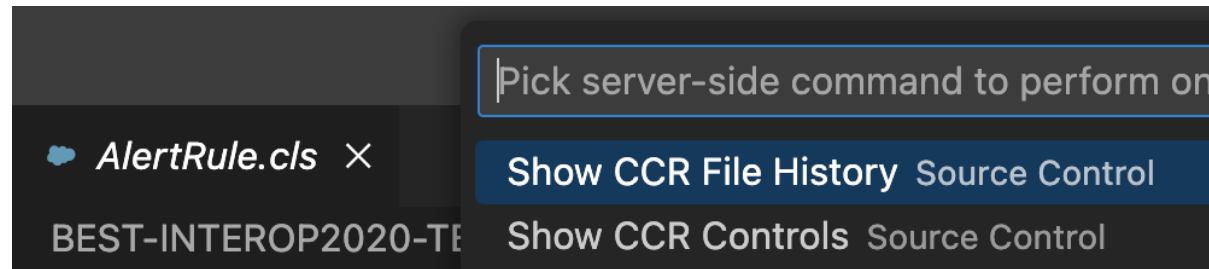


add //custom_ccrs/us/BEST/INTEROP2020/BASE/cls/PersonDetails/Example.xml#1



Access Revision History from IDE

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



Revision History Functionality

- Color-coded, with the legend at top.
- Click diff link to diff that revision versus previous revision.
- Links to CCR that changelist is associated with.
- Click BASE link to view revision history of corresponding item in BASE.

History of //custom_ccrs/us/ISCU/TESTSYS/TEST/cls/Simple01/Utils.xml

Legend:

For this CCR

For another active CCR

No job for this changelist

Currently in branch TEST. Go to branch: [BASE](#) [LIVE](#)

#2 change 208 integrate on 2019/02/25 11:50:03 by student01@SHARED_CCRTRAIN (text+ko) (diff)

CCR(s): [ISCU0120](#)

{CCRAutoGen} ISCU0120 - ICC460 CCR1: Write formatted time in all Simple##.Utils.PrintServerDateTime();
Integrating Changelists for ISCU0120 from BASE to TEST

#1 change 146 branch on 2019/02/24 21:02:26 by student01@SHARED_CCRTRAIN (text+ko) (diff)

CCR(s): [ISCU0019](#) [LIVE - Closed]

{CCRAutoGen} ISCU0019 - Add Simple##.Utils;
Integrating Changelists for ISCU0019 from BASE to TEST



Understanding Revision History

- Notice yellow revision #4 is CCR used to access history.
- Not concerned about revision #1.
 - Not highlighted.
 - Assuming proper procedures followed, should have left all branches in clean state.
- Possible #2 and #3 block #4 from progressing to TEST.
 - Would block if overlapping changes.
 - Pink, therefore active and not progressed all the way to LIVE.
 - Chronologically before #4 therefore overlapping changes must progress before #4.

#6 change 399 edit on 2019/02/25 12:03:38 by student01
CCR(s): [ISCU0122](#) [Merged], [ISCU0123](#) [Cancelled]
Autosubmit to Performe from ItemSet-ISCU0122_BASE_te

#5 change 348 edit on 2019/02/25 11:19:35 by student01
CCR(s): [ISCU0121](#) [In_BASE]
Autosubmit to Performe from ItemSet-ISCU0121_BASE_te

#4 change 297 edit on 2019/02/25 10:59:29 by student01
CCR(s): [ISCU0120](#)
Autosubmit to Performe from ItemSet-ISCU0120_BASE_te

#3 change 247 edit on 2019/02/25 10:24:52 by student01
CCR(s): [ISCU0070](#) [In_BASE]
Autosubmit to Performe from ItemSet-ISCU0070_BASE_te

#2 change 148 edit on 2019/02/25 08:59:41 by student01
CCR(s): [ISCU0020](#) [In_BASE]
Autosubmit to Performe from ItemSet-ISCU0020_BASE_te

#1 change 145 add on 2019/02/24 20:59:34 by student01
CCR(s): [ISCU0019](#) [LIVE - Closed]
Autosubmit to Performe from ItemSet-ISCU0019_BASE_te



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.




Part 3: Using Transport Logs



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] hyperlink to access end of log.
 - Typically most recent error at bottom.
- Reviewing log great way to better understand CCR automation.

[back to CCR](#)

SCGC11317 - COPD amend questionnaire

Current State	Pending_Manual_Move_To_LIVE	Phase	UAT
Organization	PMS Scotland NHS	System	T2014
Open Date	2018-04-26 06:39:34	Opened By	Young,Lisa
Responsible Organization	PMS Scotland NHS	Owner	Young,Lisa

Transport Log for SCGC11317 ([View in plaintext format](#)) [\[jump to bottom\]](#)

64.44 by Stephen Doherty (stephend_SCGC)

Load of file /trak/scgc/PRD2014/perforce/custom_ccrs/scx/SCGC/T2014/LIVE/misc/websys/DictionaryClass/03F23AE6-5F40-11E8-9851-F403E76E1375.xml

Loaded item: websys.DictionaryClass 2987 questionnaire.QCOPDQQ209 03F23AE6-5F40-11E8-9851-F403E76E1375

Load of file /trak/scgc/PRD2014/perforce/custom_ccrs/scx/SCGC/T2014/LIVE/misc/websys/DictionaryClass/058780C8-58EC-11E8-9EAA-F403E76E1375.xml

Loaded item: websys.DictionaryClass 2988 questionnaire.QCOPDQQ183 058780C8-58EC-11E8-9EAA-F403E76E1375

Load of file /trak/scgc/PRD2014/perforce/custom_ccrs/scx/SCGC/T2014/LIVE/misc/websys/DictionaryClass/07CD8E90-5F40-11E8-9851-F403E76E1375.xml

Loaded item: websys.DictionaryClass 2989 questionnaire.QCOPDQQ210 07CD8E90-5F40-11E8-9851-F403E76E1375

Load of file /trak/scgc/PRD2014/perforce/custom_ccrs/scx/SCGC/T2014/LIVE/misc/websys/DictionaryClass/0DE79C18-449B-11E8-BCDC-F403E76E1375.xml

ERROR #5001: Unable to save item \$lb("", "websys.DictionaryClass"), 0DE79C18-449B-11E8-BCDC-F403E76E1375

ERROR #5808: Key not unique: ClassName

ERROR #5001: Unable to load file /trak/scgc/PRD2014/perforce/custom_ccrs/scx/SCGC/T2014/LIVE/misc/websys/DictionaryClass/0DE79C18-449B-11E8-BCDC-F403E76E1375.xml

> ERROR #5001: Unable to save item \$lb("", "websys.DictionaryClass"), 0DE79C18-449B-11E8-BCDC-F403E76E1375

> ERROR #5808: Key not unique: ClassName

Load of file /trak/scgc/PRD2014/perforce/custom_ccrs/scx/SCGC/T2014/LIVE/misc/websys/DictionaryClass/0DED8D7A-58EC-11E8-9EAA-F403E76E1375.xml

Loaded item: websys.DictionaryClass 2991 questionnaire.QCOPDQQ184 0DED8D7A-58EC-11E8-9EAA-F403E76E1375

Load of file /trak/scgc/PRD2014/perforce/custom_ccrs/scx/SCGC/T2014/LIVE/misc/websys/DictionaryClass/166270F6-58EC-11E8-9EAA-F403E76E1375.xml

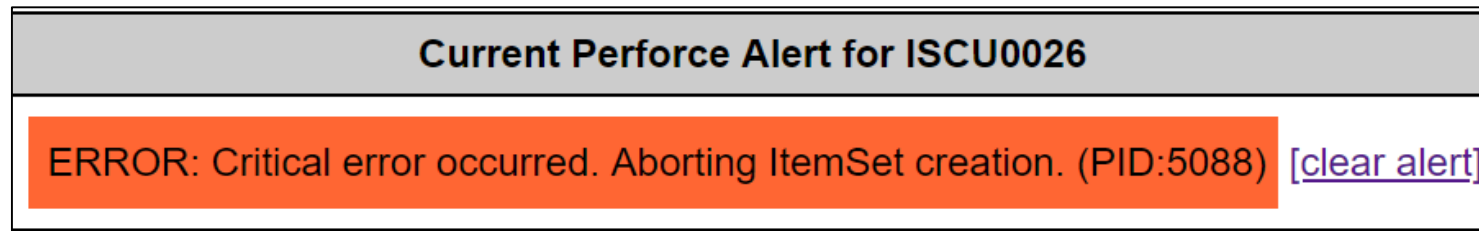
Loaded item: websys.DictionaryClass 2992 questionnaire.QCOPDQQ185 166270F6-58EC-11E8-9EAA-F403E76E1375

Load of file /trak/scgc/PRD2014/perforce/custom_ccrs/scx/SCGC/T2014/LIVE/misc/websys/DictionaryClass/1DDE5AD10-5A9B-11E8-9841-F403E76E1375.xml



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.



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.



Quiz: Perforce Alerts

Question:

It is an acceptable practice to leave Perforce Alerts displayed on a CCR and to learn to overlook them when they appear.
True or False?

Answer: False.

Perforce Alerts mean something unexpected occurred and the CCR Owner has the responsibility to understand what occurred and whether corrective action is required; engage Support if necessary to understand what took place.



Part 4: Manual Integration



Manually Integrate Changes

- 2 use cases:
 - Part of process of resolving merge conflict on `authorizeAndStartMoveToXXXX`.
 - Manually trigger preview before performing `authorizeAndStartMoveToXXXX`.



How To: Manually Integrate Changes

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

Itemset Details	Submitted Changes (1)	Create Itemset	Perforce Integration	Perforce Backout
<p>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</p>				
<p>Perforce Integration * ?</p>				
<p><input type="checkbox"/> Preview ? <input type="checkbox"/> Shelf Changes ?</p>				
<p>BASE to TEST</p>				
<p>Integrate</p>				



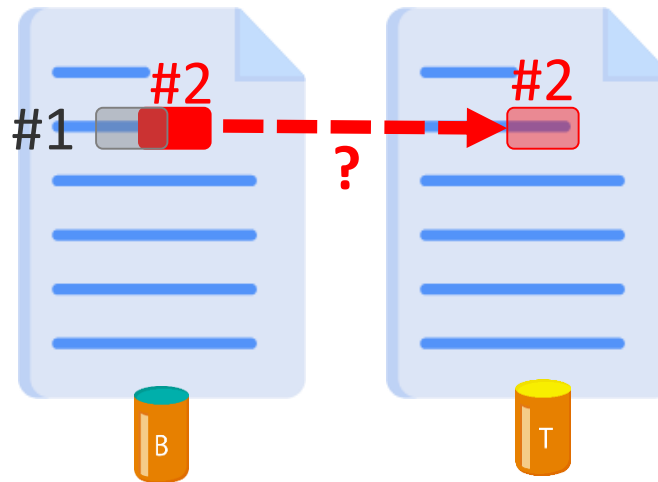
Part 5: Perforce Integration Conflicts



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.

Itemset Details	Submitted Changes (2)	Create Itemset	Perforce Integration	Perforce Backout
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_ccr
/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

- For this example, merge conflict occurred when CCR A performed `authorizeAndStartMoveToTEST`.
 - CCR B specified as blocking CCR A.
- 1. Edit CCR A to specify CCR B as prerequisite.
- 2. Progress CCR B.
- 3. Use Perforce controls on CCR A to perform manual integration from BASE to TEST.
 - Make sure to clear preview only box.
 - If this integration fails again, contact InterSystems support.



Merge Conflict: Solution (cont.)

4. Deploy generated ItemSet to TEST.
5. Progress CCR A and B normally.
 - CCR B must progress ahead of CCR A.



Merge Conflicts Causes

- Merge conflicts can occur due to:
 - A stale CCR that has sat for too long.
 - A cancelled CCR which wasn't properly backed out.
 - A circular dependency between two CCRs.
 - diff chunks #1, #3 from CCR-A and #2 from CCR-B.
 - Lack of proper baselining (discussed later in this course).



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

- Cause.
 - 2 different CCRs where 1 adds an item and the other edits the same item.
 - Will see an error if attempt to integrate the CCR with the edit before the add.



Progressing Edit before Add (cont.)

- Reason this is problematic:
 - 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.
- On error, Prerequisite CCR field of CCR with edit action automatically contains ID of CCR with add action.



Progressing Edit before Add (cont.)

- Error on preview integration:
 - WARNING: Prerequisite CCR detected.
 - Will not be able to perform authorizeAndStartMoveToXXXX until progress CCR with add action.
- Error on authorizeAndStartMoveToXXXX:
 - Critical error occurred. Another CCR has been detected as a prerequisite and marked as such. Aborting ItemSet creation.
 - Must manually integrate changes after progressing prerequisite CCR to next phase.



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.

Itemset Details

Submitted Changes (4)

Create Itemset

Perforce Integration

Perforce Backout



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.



Part 6: Backing Out

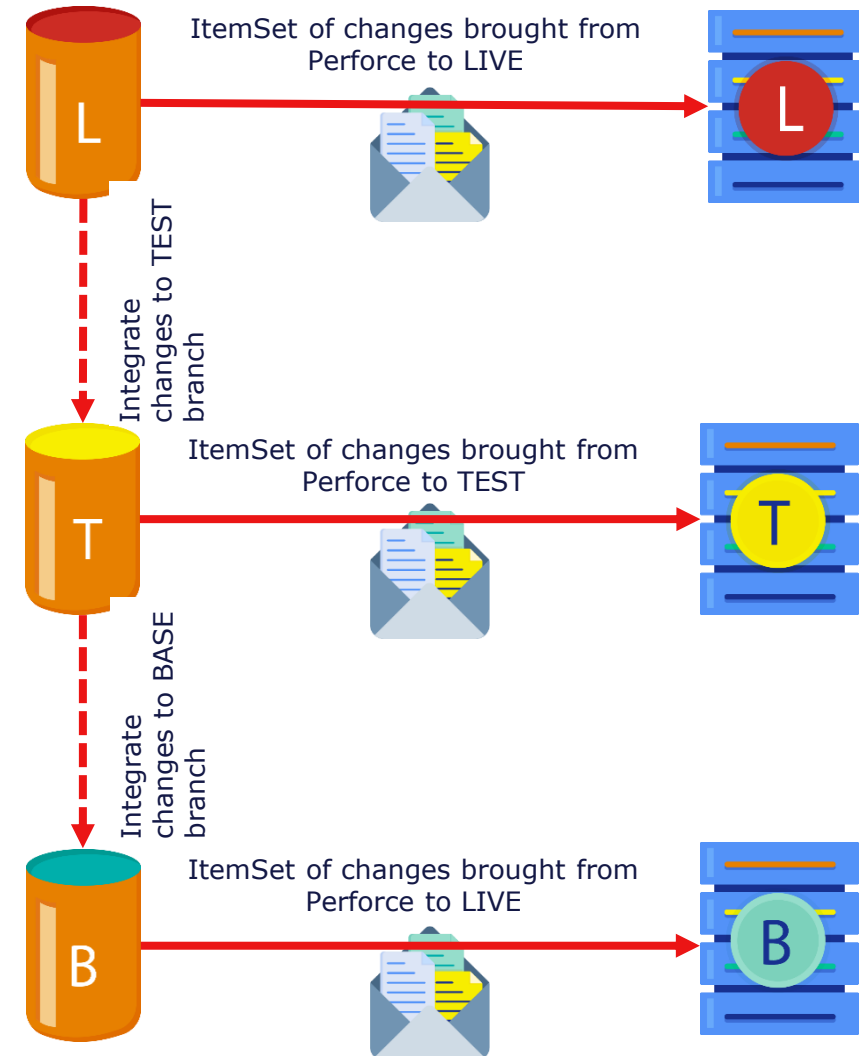


Backout Overview

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

Perforce Branches

System Environments

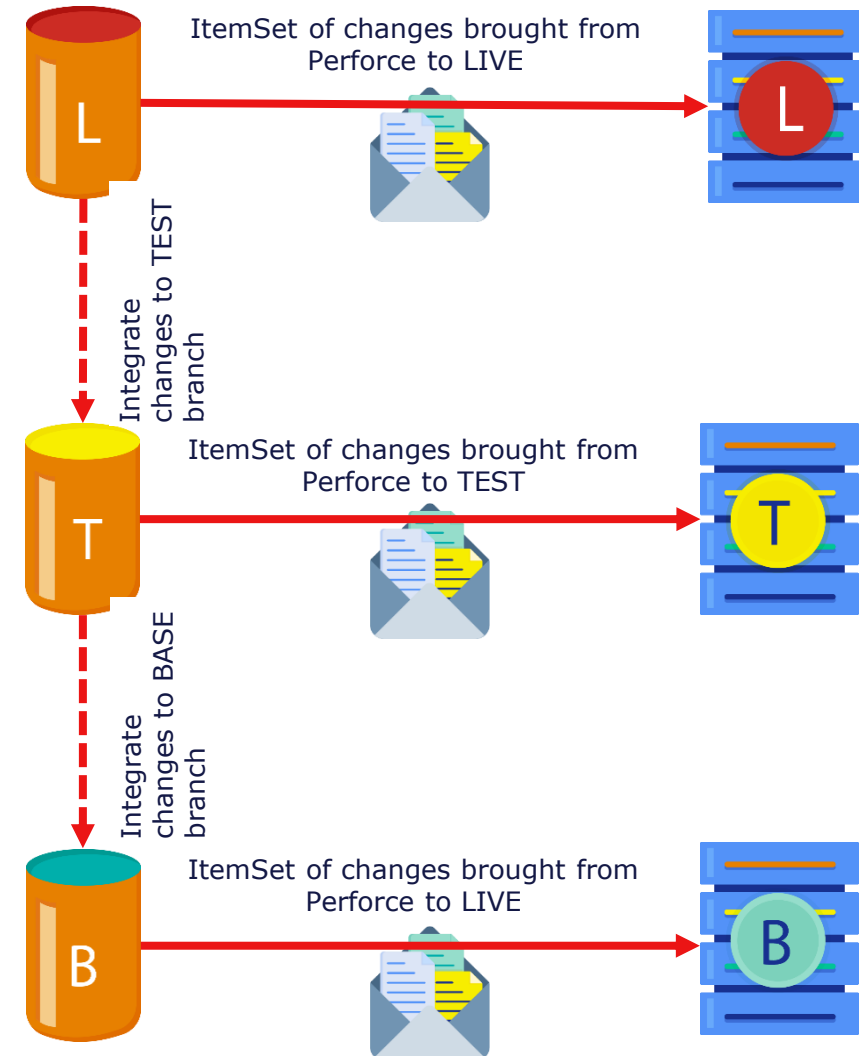


Backout Overview (cont.)

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

Perforce Branches

System Environments



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.**

Itemset Details

Submitted Changes (1)

Create Itemset

Perforce Integration

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 ?

Backout



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 InterSystems 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: Updating Client Tools



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.



Summary

- What are the key points for this module?

