INTERSYSTEMS LEARNING SERVICES

InterSystems Change Control

InterSystems
Creative data technology

InterSystems Change Control

COPYRIGHT NOTICE

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 100: Introduction to Change Control

InterSystems
Creative data technology

Objectives

- Explain the purpose and importance of change control.
- Explain the differences between change control and source control.
- Identify the 3 different types of changes.
- Identify the types of processes that involve change control.
- Identify the key performance indicators of change control.
- Identify ways to make change control processes successful.
- Articulate the need for and the value of change control for your own personal projects and workflows.

Part 1: What is Change Control?

Terms Related to “Change” at InterSystems

- Contract change.
  - Process to change a contract with another group.
- Customer change management.
  - Process to change a business process for our customers.
- Change request.
  - Process for an enhancement request.
- Change control.
  - Process for a change to system or environment.
- Consistent use of terminology key to effective communication.

*ITIL Change Control related definitions

- Change.
  - Addition, modification or removal of anything that could affect IT services.
- Change record.
  - Record containing the details of a change.
  - Each change record documents the lifecycle of a single change.
- *ITIL stands for “Information Technology Infrastructure Library.”
  - Set of detailed practices for IT service management.

Examples of a Change

- Change to:
  - Application configuration.
  - Disk layout.
  - Status of a service.
  - Application code.
  - Web server configuration.
  - User password.
  - User data.

Goal of Change Control

- “The goal of [change control]* is to establish **standard procedures** for managing change requests in an **agile** and **efficient manner** in an effort to drastically minimize the **risk** and **impact** a change can have on business operations.”
- *NOTE:
  - ITIL uses term “change management” to describe change control.
  - We are standardizing on term “change control” and will use it in place of “change management” in these courses.

Why Change Control?

- Technology, infrastructure and software requirements continuously changing.
- Solutions becoming more complex.
- InterSystems Hosted Solutions and Managed Services in specific regions cover full-service offerings and not merely software offerings.
- This is an important component of meeting SLAs.
- Part of ITIL and required for ISO20000.
  - TrakCare Support is being certified for ISO20000.
- Supports other processes already in use.

Main Benefits of Change Control

- Evaluates the risk involved in a change.
- Maintains records of changes.
  - Documented and tested implementation and backout plans.
- Provides accurate and timely information about the changes to be implemented.
- Formal and recorded review and approval process.

Main Benefits of Change Control (cont.)

- Ensures that changes are implemented with minimum disruption.
- Improves change prioritization.
- Adherence to compliance and standards.
- Determines the cost and benefit associated with a change.
- Improve quality and customer satisfaction.

Downsides of the Gut Feel Approach to Risk Analysis

1. This change is simple; it can’t fail…
2. I did that so many times…
3. It’s easy, anyone can do it…
4. Doesn’t require any planning…
5. It will only take 5 minutes…

1. FAILURE.
2. Circumstances change > FAILURE.
3. Inexperienced implementer > FAILURE.
4. Complications arise > FAILURE.
5. Unplanned complexity > FAILURE.

Downsides of the Gut Feel Approach to Risk Analysis (cont.)

6. It won’t impact anything else…
7. … I don’t need a rollback plan.

6. Unexpected interdependency > FAILURE.
7. Can’t rollback quickly > FAILURE.

There is no substitute for proper planning!

“Small Change” ≠ “Small Risk”…

OK, THERE IS A SMALL CHANGE...
RED BAG HAS THE SANDWICHES
GREEN BAG IS YOUR PARACHUTE

Quiz: Defining Change Control

Question:

Which of the following is the definition of “change control” for the purposes of this training?

A. The process by which a contract is changed.

B. Business process change for our customers.

C. Management of changes to systems or environments.

D. The process for handling enhancement requests.

Answer:

C. Management of changes to systems or environments.

Quiz: Use of Change Control

Question:
To what extent does your organization currently use change control?

A. We have change control processes and tools which work well for us.
B. We have a change control process in place but struggle to use it consistently.
C. We are working towards adopting change control processes.
D. We know we should put change control in place but it’s not currently a priority.
E. What is change control?

Part 2: Source Control

What is Source Control?

- Database for flat file Items.
  - Provides central storage for code and configuration.
  - Provides full versioning capabilities.
    - Maintains all history.
    - Prevents permanent deletion.
- How we use it:
  - Answer who, what, when, where, why and how for each change.
  - Maintain every version of every item of configuration or code for a system.
  - Allow automated merging of changes between environments.
  - Provide snapshot of operating environment.

Change Control with Source Control System

- Layering change control on top of a source control tool enables:
  - Versioning of configuration items.
  - Easier/automated rollout of changes to target environments.
  - Easier/automated rollback of changes when something goes wrong.
  - Further automation of value-add methodologies: Build, Test, Upgrade.

Change Control with Source Control System

- Change control process should cover both:
  - Changes that can exist in source control (Versioned).
  - Changes that cannot exist in source control (Documentation only).
  - With time, try to move Documentation-only changes to Versioned changes with improvements to tools used in the process.

Quiz: Change Control with Source Control

Question:

To what extent does your organization currently use source control?

A. We have source control in place and automated with our change control processes.

B. We have source control and hope to tie it in with change control workflows.

C. We are working towards adopting source control.

D. We know we should put source control in place but it’s not currently a priority.

E. What is source control?

Part 3:
Change Control: Real Life Examples

“Managing” without Change Control (Example 1)

- Serious questionnaire performance crisis.
- InterSystems Support spent several months in crisis mode tracking root cause.
  - Repeated assertions that no one had made any changes to the system.
  - Undocumented icon definition change interacted with other factors to cause crisis.

“Managing” without Change Control (Example 2)

- Configuration change booking restriction crisis.
  - TrakCare location list changed drastically from one day to the next.
    - Potential clinical risk moving emergency episodes between locations.
- Team worked in crisis mode for one full week before discovering root cause.
  - Customer made undocumented changes to booking restrictions that triggered a change in the location list.

Reaping the Benefits of Change Control (Example 1)

- Printing crisis at very large hospital.
  - 5pm change to an important report.
  - Crisis call received in the middle of the night that printing was broken.
  - Overnight on-call person found and backed out the change in 15 min.
    - No prior knowledge of that particular change.

Reaping the Benefits of Change Control (Example 2)

- Managing reapplication of changes during complex upgrades.
    - One very large site had over 300 changes which had to be reapplied to multiple environments as part of upgrade process.
    - Manually it would have taken 2 weeks to reapply, and mistakes were likely.
    - Changes reapplied from change control application in 2 hours with no issues.

Quiz: Personal Experience

Question:

How do you relate to the prior examples?

A. I have stories which align with the “Managing” without Change Control slides.

B. I have stories which align with the Reaping the Benefits of Change Control slides.

C. I have stories from both categories.

D. I couldn’t relate to either set of real-world examples.

Part 4: Types of Changes

Types of Changes (from ITIL)

- Emergency change.
  - “A change that needs to be evaluated, assessed and either rejected or approved in a short space of time.”
  - “Emergency change should be reserved for changes intended to repair an error in an IT service that is impacting the business to a high degree or to protect the organization from a threat.”
  - Example: rebooting a crashed server.

Types of Changes (from ITIL, cont.)

- Standard change.
  - “A pre-authorized change that is low risk, relatively common and follows a procedure or work instruction.”
  - “Standard changes are not required to follow the normal [change control] process and can be recorded in a different way.”
  - Example: resetting a user’s password.
- Normal change.
  - "A change that is not an emergency change or a standard change. Normal changes follow the defined steps of the [change control] process.”
  - Example: fixing a bug in integration logic.

Management of Standard Changes

- Catalog of standard changes.
  - Define specific criteria for when to execute the change.
  - Define the work instruction for the change.
- Regularly review and approve changes in catalog.
  - Some changes, such as data fixes, may require review more frequently.
  - Define the review and approval process.
- Can “demote” to a normal change as needed.
  - When predefined criteria aren’t met.
  - When predefined work instruction insufficient.

Quiz: Identifying Standard Changes

Which of the following could be Standard Changes?

A. Scheduled network maintenance. ✓

B. Clear CSPGateway cache. ✓

C. A specific SQL select statement. ✓

D. Patching. ✓

E. One-time data fix. ✗

F. Custom development change. ✗

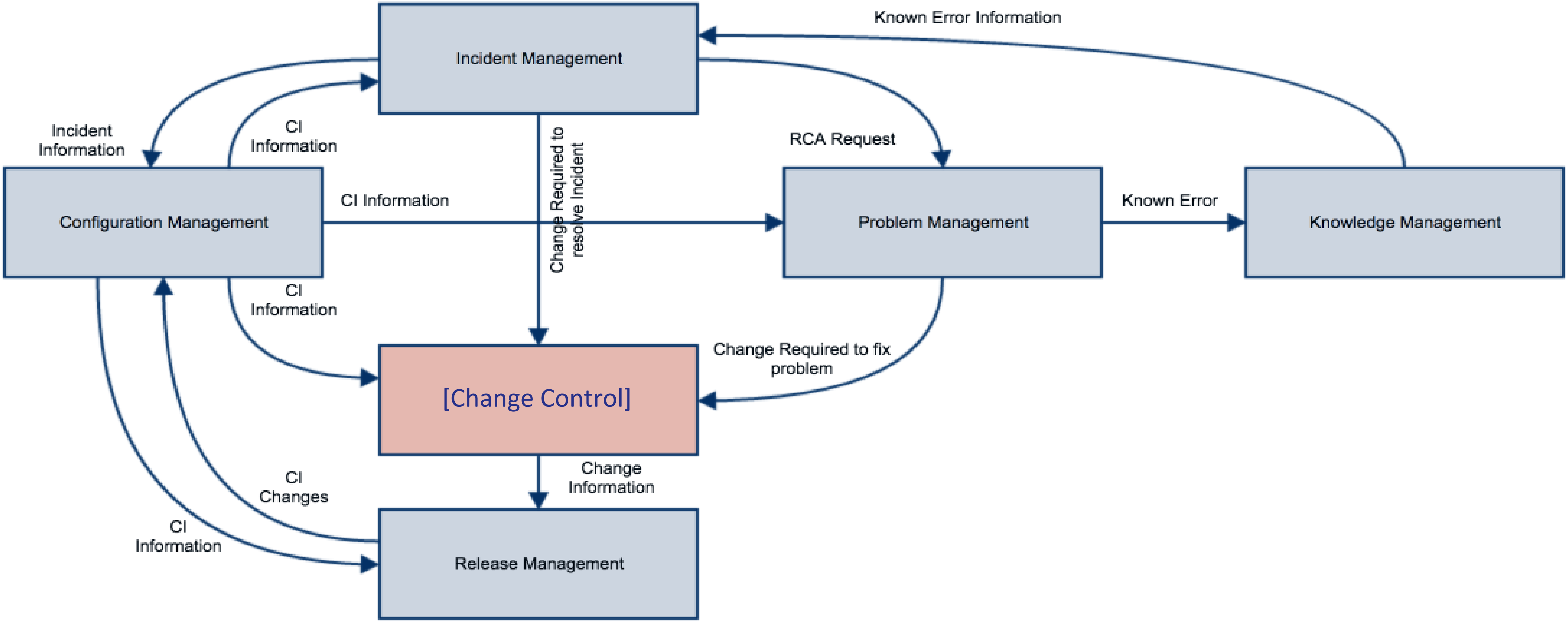
Low-risk, regularly occurring, well understood, repeatable

Part 5: Change Control Processes

Objective of Change Control Process

- Ensure that changes are recorded, then:
  - Evaluated.
  - Authorized.
  - Prioritized.
  - Planned.
  - Tested.
  - Implemented.
  - Documented.
  - Reviewed.
- … all in a controlled manner.

Interaction with other Processes



Top five indicators of a poor change control process

- Unauthorized changes.
- Unplanned outages.
- A low change success rate.
- A high number of emergency changes.
- Delayed project implementations.

Change Control Key Performance Indicators (KPIs)

- With proper change control, over time you should see:
  - Increase in number of successful changes implemented.
  - Reduction in the number of service disruptions.
  - Reduction in unauthorized changes.
  - Decrease in average time to implement a change.
  - Decrease in number of disruptions (incidents, problems) caused by failed changes.
  - Increase in ratio of planned vs. unplanned changes.
  - Decrease in ratio of normal vs. standard changes.

Roles and Responsibilities (ITIL)

- Change initiator, owner, implementer (could all be the same person).
- Change reviewer, approver.
- Change manager.
  - Review and approve minor normal changes and standard changes.
  - Analyze change records to identify trends.
  - Make sure change control processes are respected.
- CAB/eCAB ((Emergency) Change Advisory Board).
  - Advisory committee for major or significant normal changes.

Roles and Responsibilities (cont.)

- At a minimum, have a different implementor and reviewer.
  - Promotes at least 2 people understanding system.
    - Only 1 person understanding a system is high risk!
  - Helps catch errors (can’t peer review your own work).

Quiz: Change Control Benefits

Question:

Which of the following are benefits of proper change control use? Select all that apply.

A. Reduction in unauthorized changes. ✓

B. Low change success rate. ✗

C. Reduction in the number of service disruptions. ✓

D. High number of emergency changes. ✗

E. Documented history of system evolution. ✓

Part 6: Tips for Success

Tips to make Change Control Successful

- Avoid excessive bureaucracy.
  - Make it easy to raise and track changes.
- Have clear procedures defined for all types of changes.
- Communicate processes clearly and make sure they are well understood.
- Appoint change control manager.
  - Responsible for overseeing change control processes within your organization.
  - Empower them to ensure compliance with those policies.

Tips to make Change Control Successful (cont.)

- Build processes on top of source control wherever possible.
  - Enables automated application of normal changes and automated rollback.
- Automate standard changes.
  - Reduces risk.
- 'Little and often’.
  - Avoid large amounts of changes or changes that sit in development for a long period of time.
- Understand the value of change control.

Summary

- What are the key points for this course?

ICC 110:
Change Control at InterSystems

InterSystems

Creative data technology

Objectives

- Name the tools related to change control which are used by InterSystems and explain which teams use them.
- Explain the importance of change control across the lifecycle of InterSystems products.
- Identify change control related milestones within the InterSystems ARIES implementation methodology.

Review:
Terms Related to “Change” at InterSystems

- Contract change.
  - Process to change a contract with another group.
- Customer change management.
  - Process to change a business process for our customers.
- Change request.
  - Process for an enhancement request.
- Change control.
  - Process for a change to system or environment.
- Consistent use of terminology key to effective communication.

Part 1: InterSystems Change Control Tools

Change Control Tools within InterSystems

- Perforce Helix
- TrakCare Change Control (TCC)
- JIRA
- Change Control Record (CCR)

Change Control Tools: Perforce Helix

- Commercial Version Control System (VCS).
- Versions InterSystems Products and Implementation Projects.
- Customers engaging in InterSystems change control processes will receive Perforce Licenses.
- Enterprise Perforce server located in Cambridge, MA.

Change Control Tools:
TrakCare Change Control (TCC)

- Functional layer of TrakCare.
- Interface for all configuration that is programmatically imported into TrakCare or exported out of TrakCare.
  - On the file system, TrakCare configuration stored as XML files.
    - Format called ElementXML.
- Implemented within the TrakCare product and updated via patch, adhoc and release process.

Change Control Tools: JIRA

- Commercial workflow application.
- Tracks all HealthShare and TrakCare product changes.
- Internal to InterSystems network.
  - Not customer accessible.
- Manages versioning and flow of product code changes.
  - Integration with Perforce and InterSystems Data Platform Source Control Hooks.
- Manages versioning and flow of TrakCare product configuration changes.
  - Integration with Perforce and TrakCare Change Control (TCC).

Change Control Tools:
Change Control Record (CCR)

- Custom workflow application built on our own technology.
- Tracks all customizations to InterSystems products installed around the world.
- Manage versioning and flow of onsite custom code changes.
  - Integration with Perforce and InterSystems Data Platform Source Control Hooks.
- Manage versioning and flow of onsite custom configuration changes.
  - Integration with Perforce and TrakCare Change Control (TCC).

Change Control Tools: Change Control Record (cont.)

CCR

- Transports all changes performed on customer’s environments back to InterSystems Perforce server.
  - Uses CCR ItemSet framework.
- Full support of normal changes and standard changes.
- Multiple Tiers and workflow configuration options.
  - Very adaptable to meet the specific needs of the phase of the project.

Quiz: Change Control Tools

Question:

For each of the following, match the usage description to the appropriate tool: Perforce Helix, TrakCare Change Control (TCC), JIRA, or Change Control Record (CCR).

A. Tracks HealthShare and TrakCare product changes.

B. Version Control System for product code and site customizations.

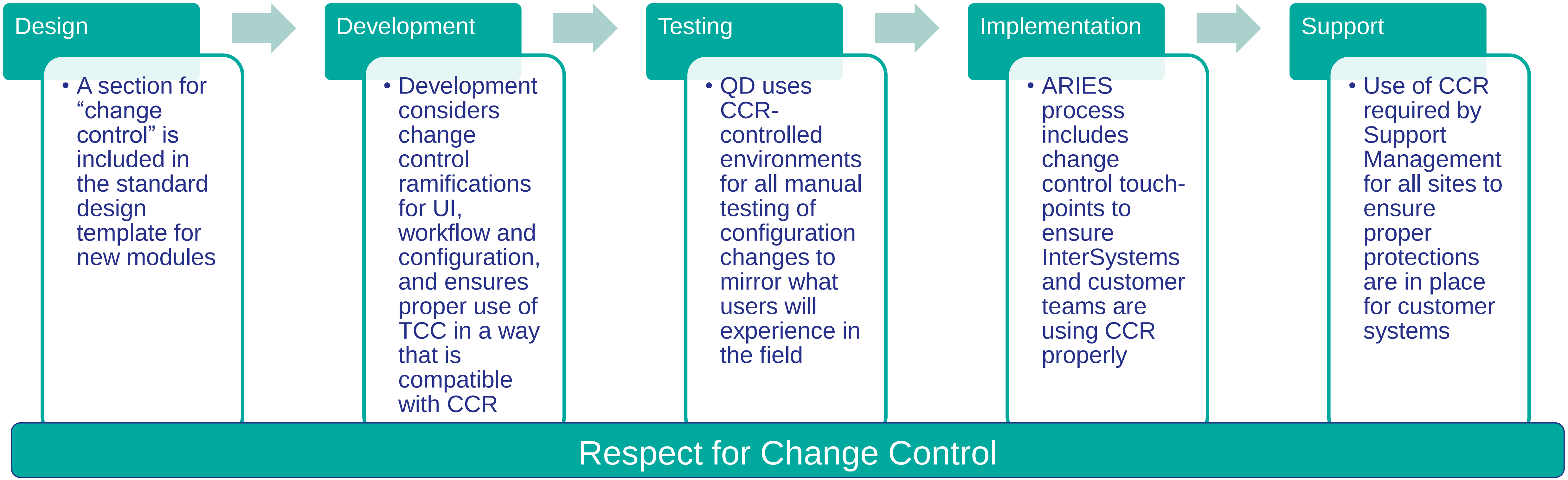
C. “Interface” for TrakCare configuration.

D. Tracks implementation-specific customization.

Part 2: InterSystems Products and the Culture of Change Control

Change Control and the TrakCare Lifecycle

- Change control considered at all points along product lifecycle.
  - Smooth implementation and business as usual experience relies on TCC and CCR being fully functional.



Design

- A section for "change control" is included in the standard design template for new modules

Development

- Development considers change control ramifications for UI, workflow and configuration, and ensures proper use of TCC in a way that is compatible with CCR

Testing

- QD uses CCR-controlled environments for all manual testing of configuration changes to mirror what users will experience in the field

Implementation

- ARIES process includes change control touch-points to ensure InterSystems and customer teams are using CCR properly

Support

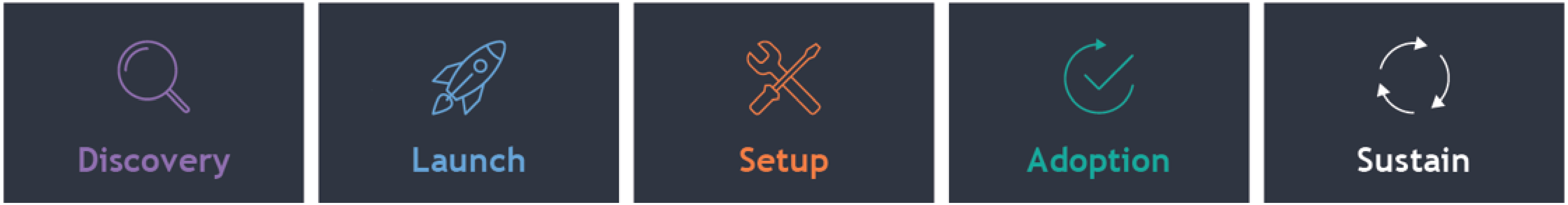
- Use of CCR required by Support Management for all sites to ensure proper protections are in place for customer systems

Respect for Change Control

Part 3: Change Control and ARIES

Introduction to ARIES

- InterSystems customer engagement methodology.
  - Designed to achieve efficient, fast and successful customer experience.
    - During the sales and delivery cycle of an enterprise solution.
    - Through to routine operational running of InterSystems products.
- ARIES Stages:



Discovery

Launch

Setup

Adoption

Sustain

ARIES and Change Control

- Change control leveraged at key points within ARIES Stages.
- Launch.
  - Initialize CCR with customer and system details.
  - Introduce customer to the importance of change control and CCR.
- Setup.
  - Change control should be used to record all changes during the setup stage.
  - Move from BASE-only change control workflow to BASE>TEST for Validation.

ARIES and Change Control

- Adoption.
  - Role-specific CCR training as part of larger ARIES train the trainer strategy.
- Sustain.
  - BAU operational policies and procedures will include CCR usage details.
  - TrakCare Support relies on change control on a day-to-day basis.
- It is critical to foster a culture of change control through an implementation project.

Quiz: Use of CCR within ARIES

Question:

At what ARIES Stage is CCR first enabled for use by InterSystems and customer project teams?

A. Discovery.

B. Launch.

C. Setup.

D. Adoption.

E. Sustain.

Answer:

C. Setup.

Part 4: Change Control and the Customer’s Responsibilities

Tips for Customers to Succeed

- Create culture of change control within your organization.
  - Understand the importance of change control processes.
  - Ensure appropriate training on change control processes for everyone working on systems.
- Promptly report any issue found within InterSystems source control hook functionality, CCR or TCC to your support team.
- See ICC100: Introduction to Change Control for additional tips, especially:
  - Appoint a change control manager.

Quiz: Customer Responsibilities

Question:

What role within the Customer’s organization is responsible for ensuring compliance with change control related policies?

A. Implementation manager.

B. Architect.

C. Peer reviewer.

D. Change control manager.

E. Project manager.

Quiz: Customer Responsibilities

Answer:

D. Change control manager.

For more about roles in change control, see ICC100.

Summary

- What are the key points for this course?

ICC 200: Introduction to CCR

InterSystems
Creative data technology

Overview

- What is CCR?
- What is Perforce?
- Environments and Systems.
- Basic navigation within CCR.
- CCR Transport Basics.
- CCR Tiers.
- CCR workflow vocabulary.
- Little and often principle.
- Resources.

Objectives

- Explain what the CCR application is and why it’s important.
- Describe why CCR uses Perforce.
- Describe the following terms.
  - CCR Tier 0, 1, and 2.
  - CCR Systems and environments.
  - CCR Transport.
- Define the concept of little and often.
- Access the CCR online application and perform basic navigation.
- Access additional resources for learning and questions.

Change Control Record Application (CCR)

- Custom workflow application built on InterSystems IRIS.
  - Application centrally hosted by InterSystems.
- Manages documentation, versioning, and flow of changes to application implementations.
  - Change = anything that could impact application functionality.
- “CCR” refers to application and individual change records.
- Many workflow options.
- Robust security model.
  - Only InterSystems employees can view records and code from other organizations.

Perforce Helix

- Commercial Version Control System (VCS).
  - Database for flat file items.
    - Central storage for code and configuration.
  - Provides full versioning capabilities.
- CCR uses Perforce to:
  - Version exportable changes (code and configuration).
  - Automate merging of changes through workflow.
  - Complete copy of code in Environment.
- Customers using CCR receive Perforce licenses.
  - Only able to access Perforce through CCR application.

CCR Functionality

- Allow thorough documentation.
  - Fields for user to complete and automatically generated audit trails.
- Interact with Perforce on behalf of user.
- Facilitate movement of changes between Perforce and environments.
- Provide formal and recorded peer review.
- Provide formal and recorded authorization to move changes.

CCR Scope of Use

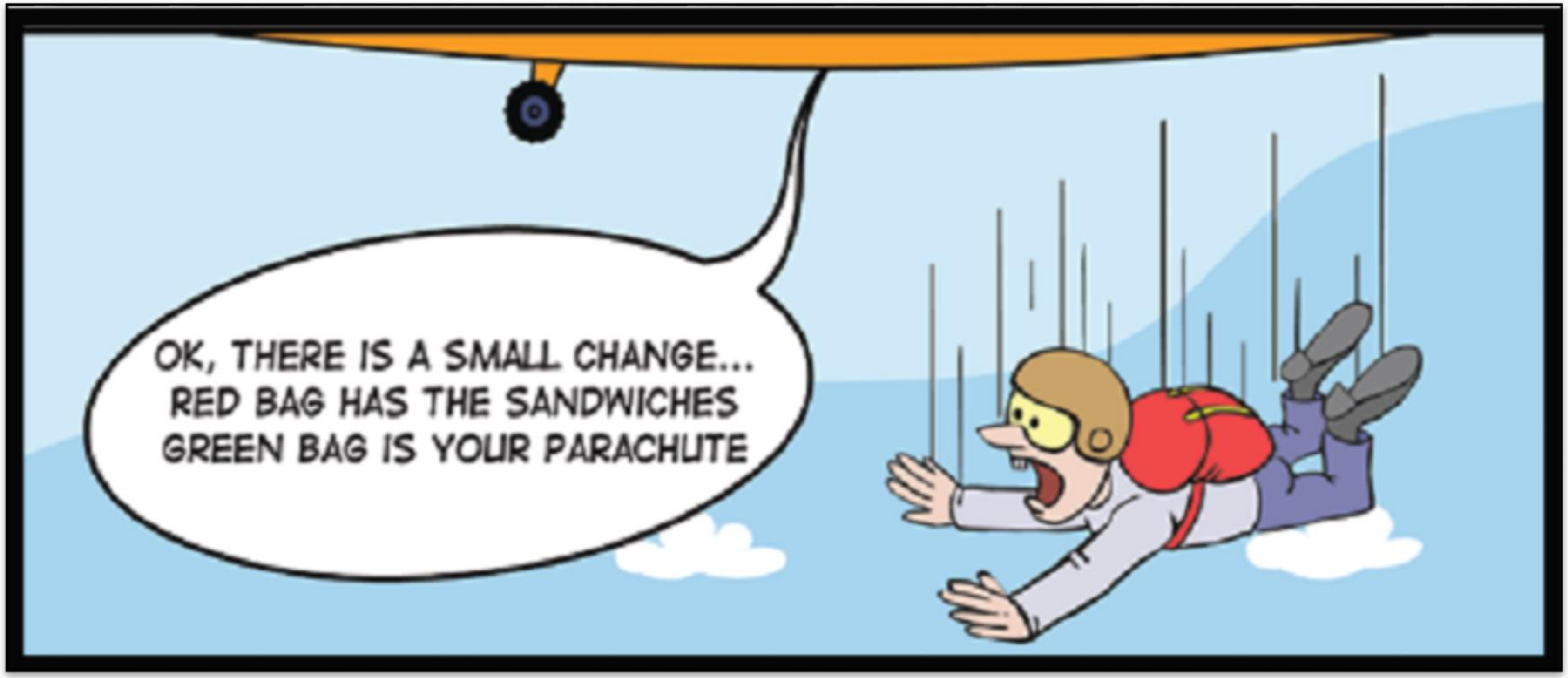
- Can be used in any phase of project.
  - Early development with one environment to multi-environment workflow for business as usual changes.
- Users include anyone making, managing, or searching for changes made to an implementation.
  - Developers, project managers, support personnel.
- Organizations that use CCR include:
  - InterSystems.
  - Implementation partners / third party consultants.
  - End user customers.

CCR First Principles and Benefits of Using CCR

- CCR First Principles guide functionality and benefits of CCR.
  - Repeatable.
    - Identify how to replicate changes in a new environment.
    - Catch missed implications and verify best practices followed.
    - Prevent unauthorized changes.
  - Revertible.
    - Ensure implementation of change allows for change to be undone.
  - Discoverable.
    - Identify recent changes during crisis.
- CCR functionality enables these 3 principles.

Small Change ≠ Small Risk

- All changes should go through CCR process.
  - Including those involving non-exportable items.


OK, THERE IS A SMALL CHANGE...
RED BAG HAS THE SANDWICHES
GREEN BAG IS YOUR PARACHUTE

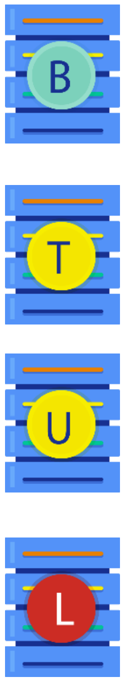
Part 1: CCR Environments and Systems

CCR Environment

- Hardware and software that provides standalone copy of functionality for an application.
- Example: one namespace of IRIS for Health with application code and configuration, on server running Red Hat.

4 Possible Primary CCR Environments

- BASE.
  - Initial environment where all changes originate.
    - Exception: LIVE-only Systems.
- TEST.
  - Testing environment for validation.
- UAT.
  - Secondary testing environment.
- LIVE.
  - Production environment.

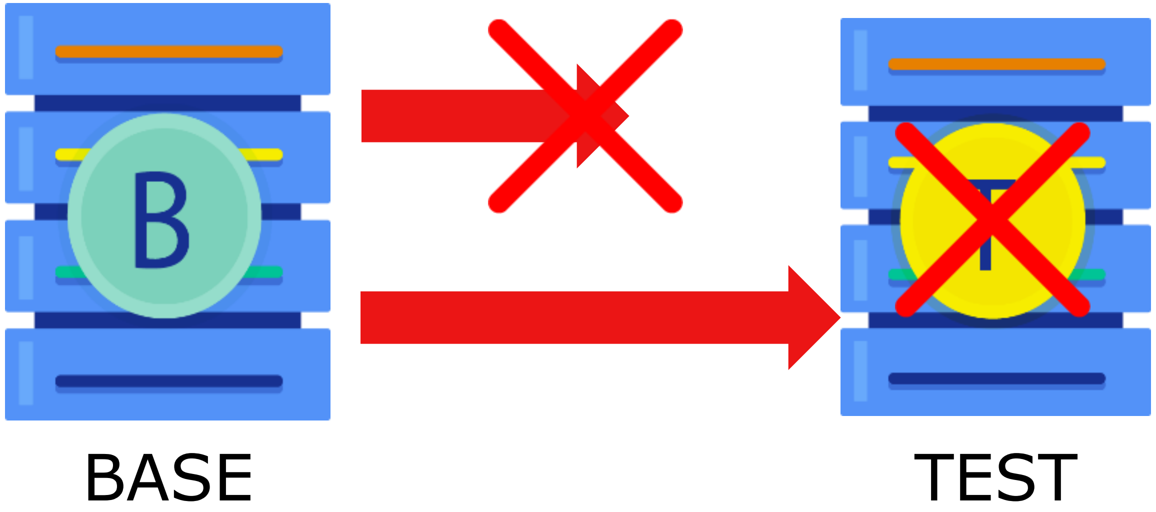


Possible Environments in System

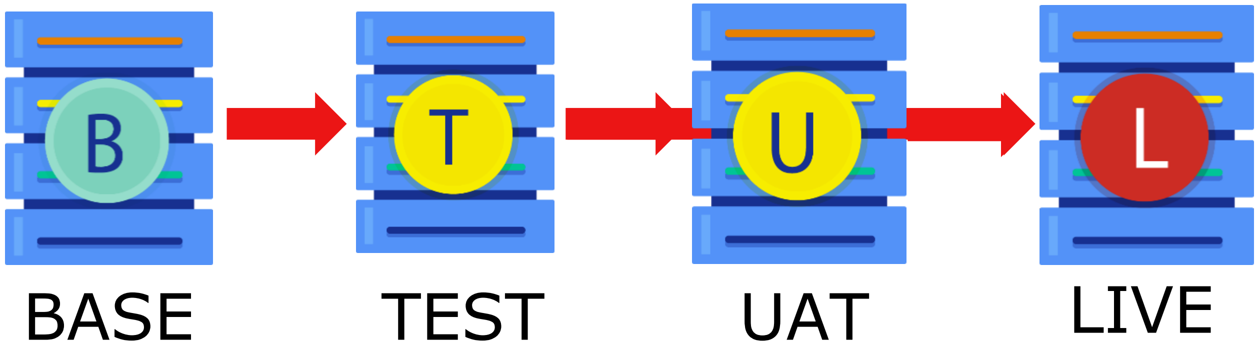
- 1 environment:
  - BASE or LIVE only.
- 2+ environments:
  - 1 must be BASE.
- Most common to have BASE, TEST, and LIVE.
- Non-BASE environments in multi-environment workflows:
  - Locked from manual changes outside CCR process.
  - Require customer authorization to receive changes.

Why multiple environments?

- Process of moving changes is tested.
  - Moving a change can fail.
- Functionality tested.
  - Isolating changed items helps verify functionality.
  - The change can fail to meet requirements or specifications.

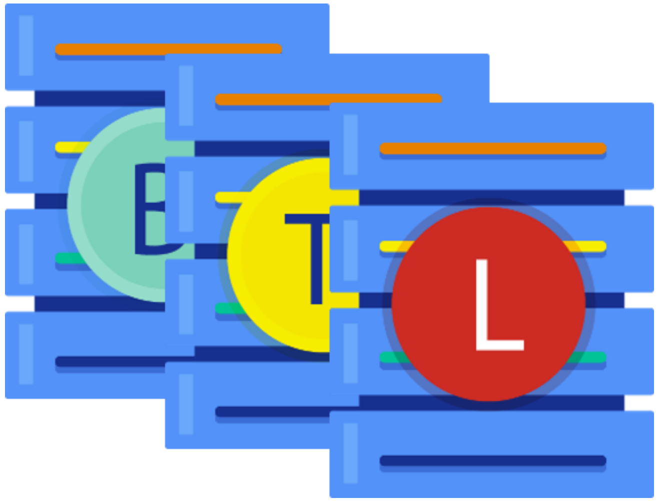


Change Flow through Environments



CCR System

- All environments included in workflow for particular application.
- Systems can be defined to have 1, 2, 3 or 4 Primary Environments.



Quiz: CCR Environments

Question:

What are the three most common CCR Primary Environments called?

A. SCRATCH. ✗

B. BASE. ✓

C. TEST. ✓

D. TRAIN. ✗

E. UAT. ✗

F. LIVE. ✓

Quiz: CCR Environment Usage

Question:

In which CCR environment must all changes originate if there are 3 environments in the System?

A. TEST.

B. BASE.

C. LIVE.

Answer:

B. BASE. All changes take place in BASE first prior to being moved to other environments in the CCR System.

Part 2: Introduction to CCR Application

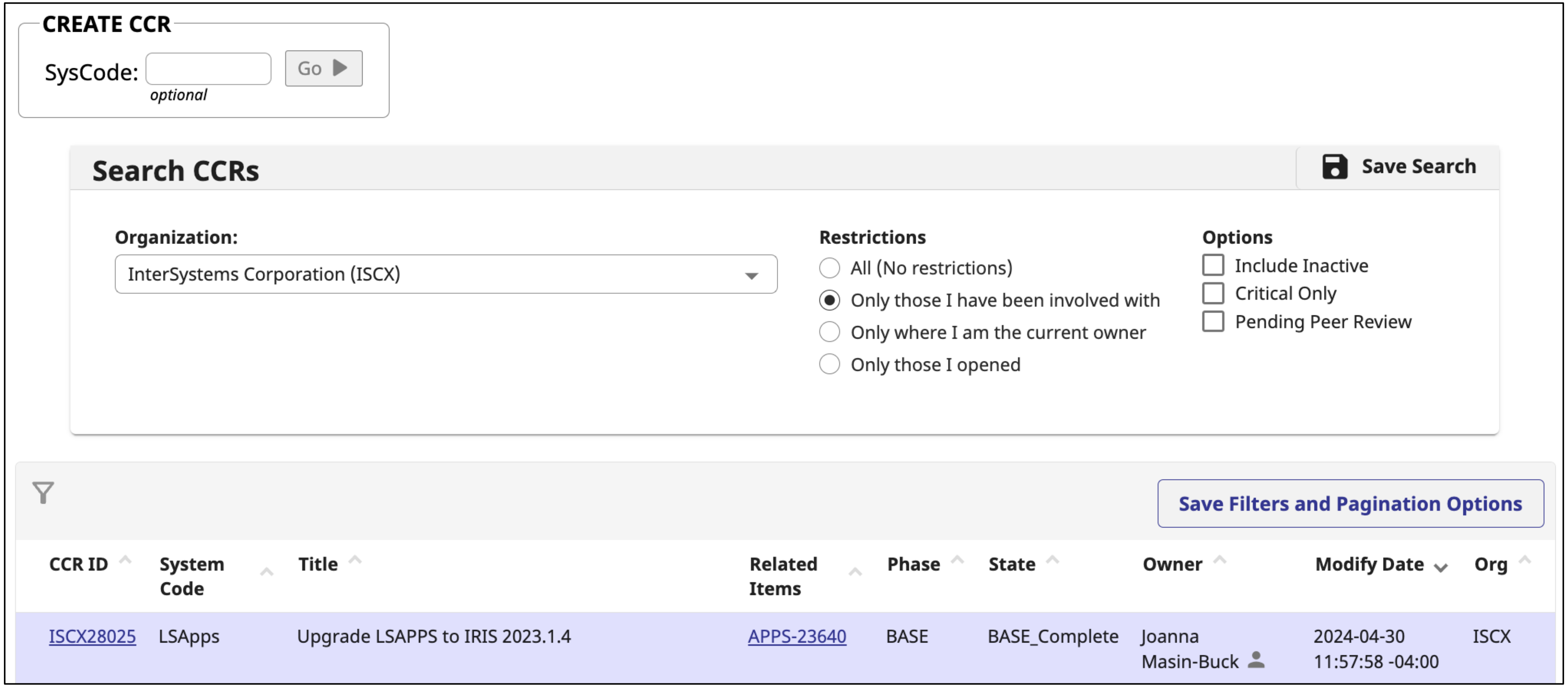
Log In

- ccr.intersystems.com.
- Use InterSystems SSO credentials to login.
  - Same as iService and learning.intersystems.com.

Demo

- Home page.
- Side navigation.
- System Details export.

CCR Home Page



CREATE CCR

SysCode: ________ *optional* Go ▶

Search CCRs

Save Search

Organization: InterSystems Corporation (ISCX)

Restrictions

○ All (No restrictions)
● Only those I have been involved with
○ Only where I am the current owner
○ Only those I opened

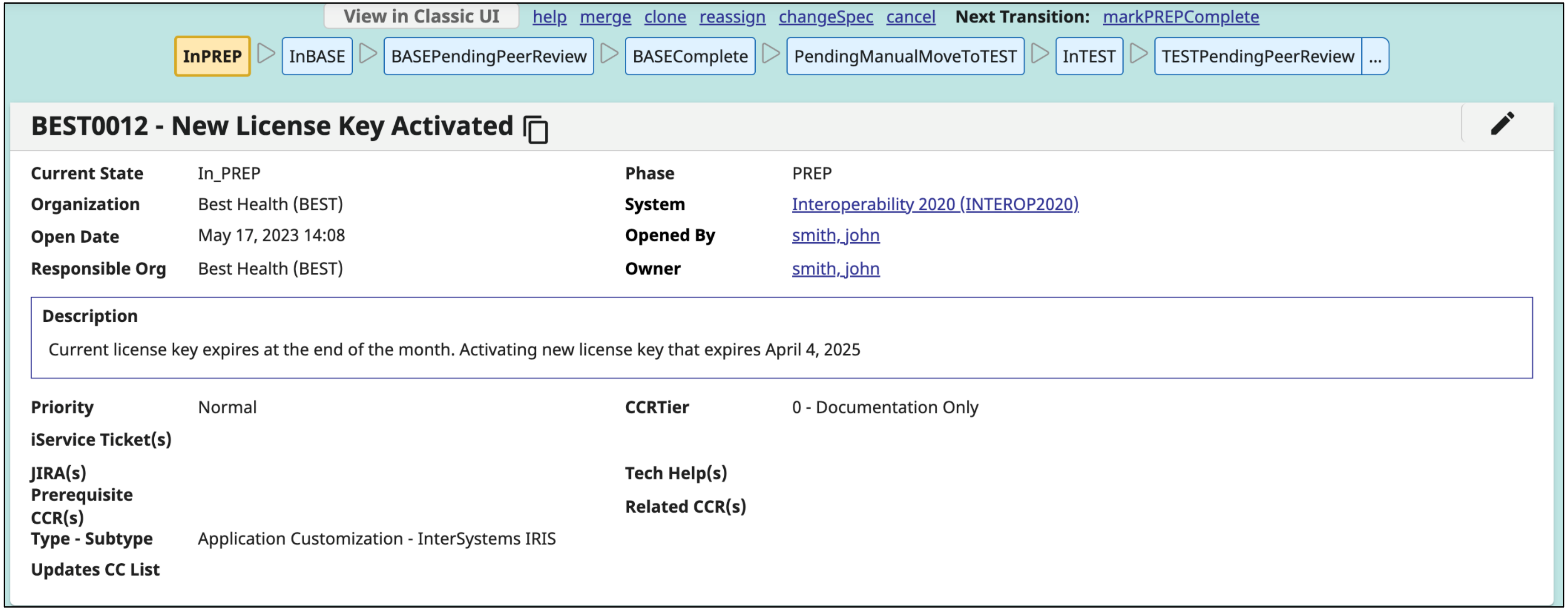
Options

□ Include Inactive
□ Critical Only
□ Pending Peer Review

Save Filters and Pagination Options

CCR ID	System Code	Title	Related Items	Phase	State	Owner	Modify Date	Org
ISCX28025	LSApps	Upgrade LSAPPS to IRIS 2023.1.4	APPS-23640	BASE	BASE_Complete	Joanna Masin-Buck	2024-04-30 11:57:58 -04:00	ISCX

CCR Record



BEST0012 - New License Key Activated

Current State	In_PREP	**Phase**	PREP
Organization	Best Health (BEST)	**System**	Interoperability 2020 (INTEROP2020)
Open Date	May 17, 2023 14:08	**Opened By**	smith, john
Responsible Org	Best Health (BEST)	**Owner**	smith, john

Description

Current license key expires at the end of the month. Activating new license key that expires April 4, 2025

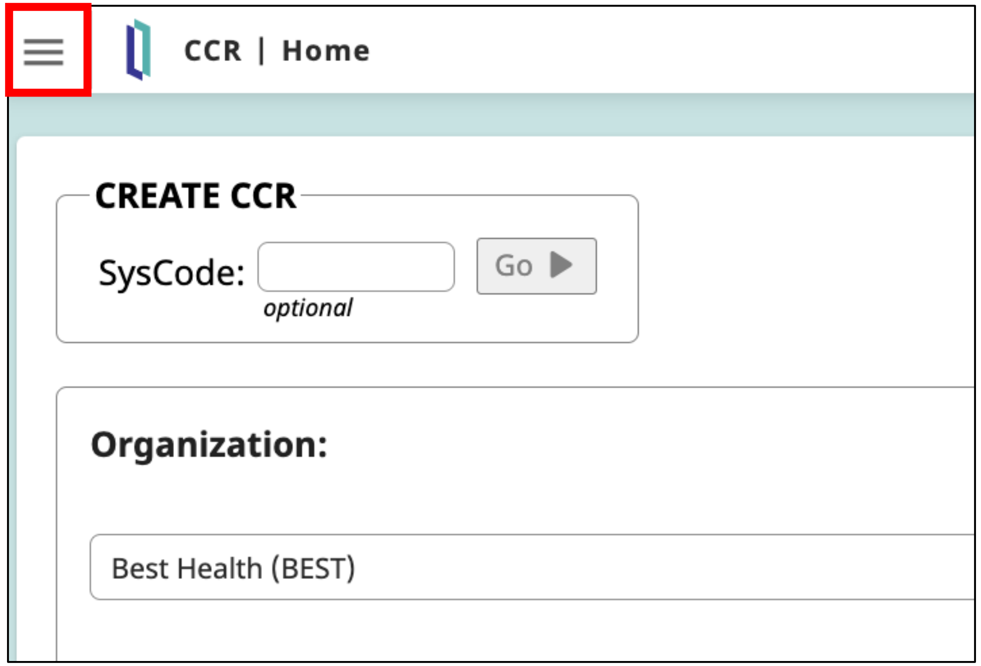
Priority	Normal	**CCRTier**	0 - Documentation Only
iService Ticket(s)			
JIRA(s)		**Tech Help(s)**	
Prerequisite CCR(s)		**Related CCR(s)**	
Type - Subtype	Application Customization - InterSystems IRIS		
Updates CC List			

CCR Home Page (cont.)

- Capabilities:
  - Create CCR.
  - Show CCRs for all Systems across Organization.
- Notable filters:
  - Restriction of Only those I have been involved with.
  - Option to Include Inactive.
    - Closed, cancelled, and merged CCRs not shown by default.
  - Search based on Title or Description.

Navigation Menu

- Available from top-left corner of every page.



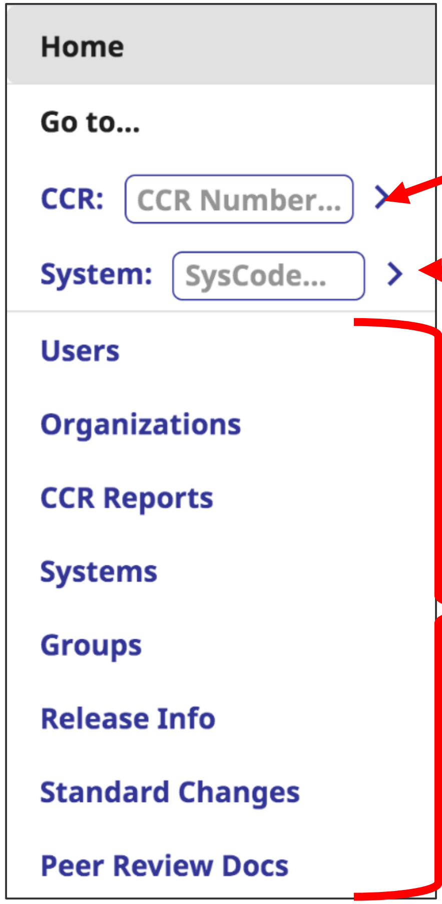
CREATE CCR

SysCode: *optional*

Organization:

Best Health (BEST)

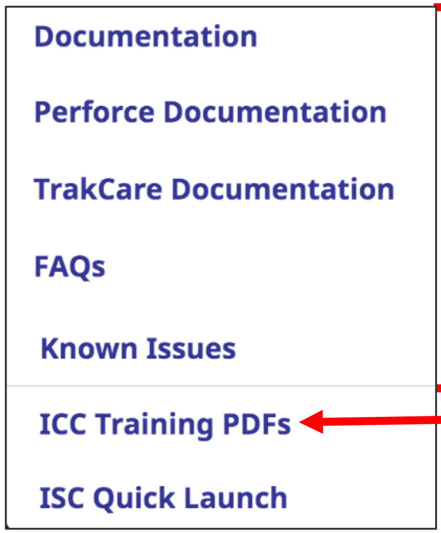
Navigation Menu (cont.)



Navigate to CCR: enter CCR ID

Navigate to System: enter SysCode or SiteCode/SysCode.

Navigation links.

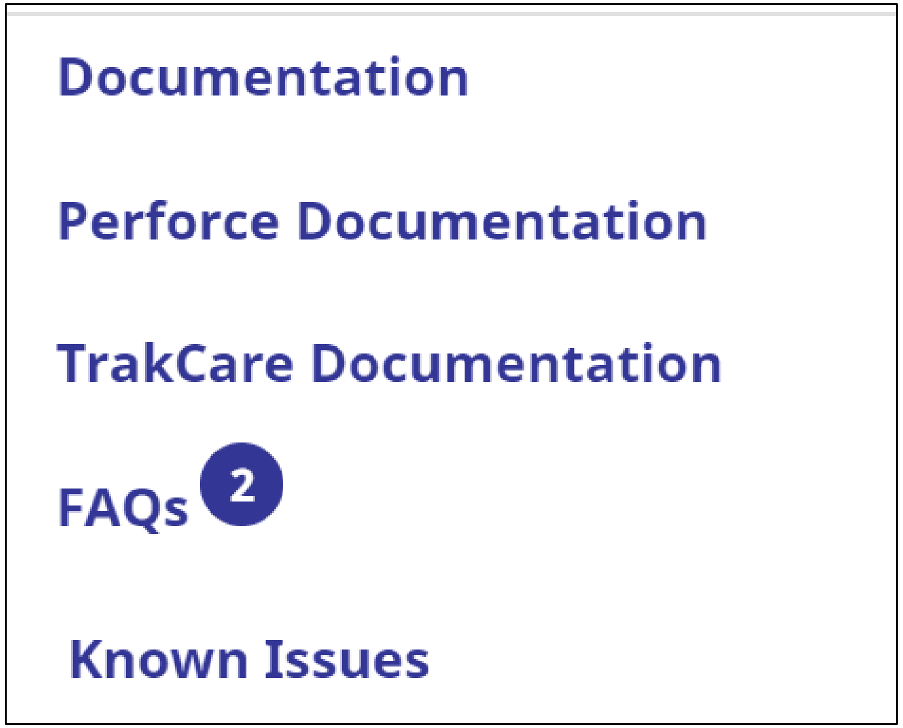


Resources.

Link to PDFs of all PowerPoints.

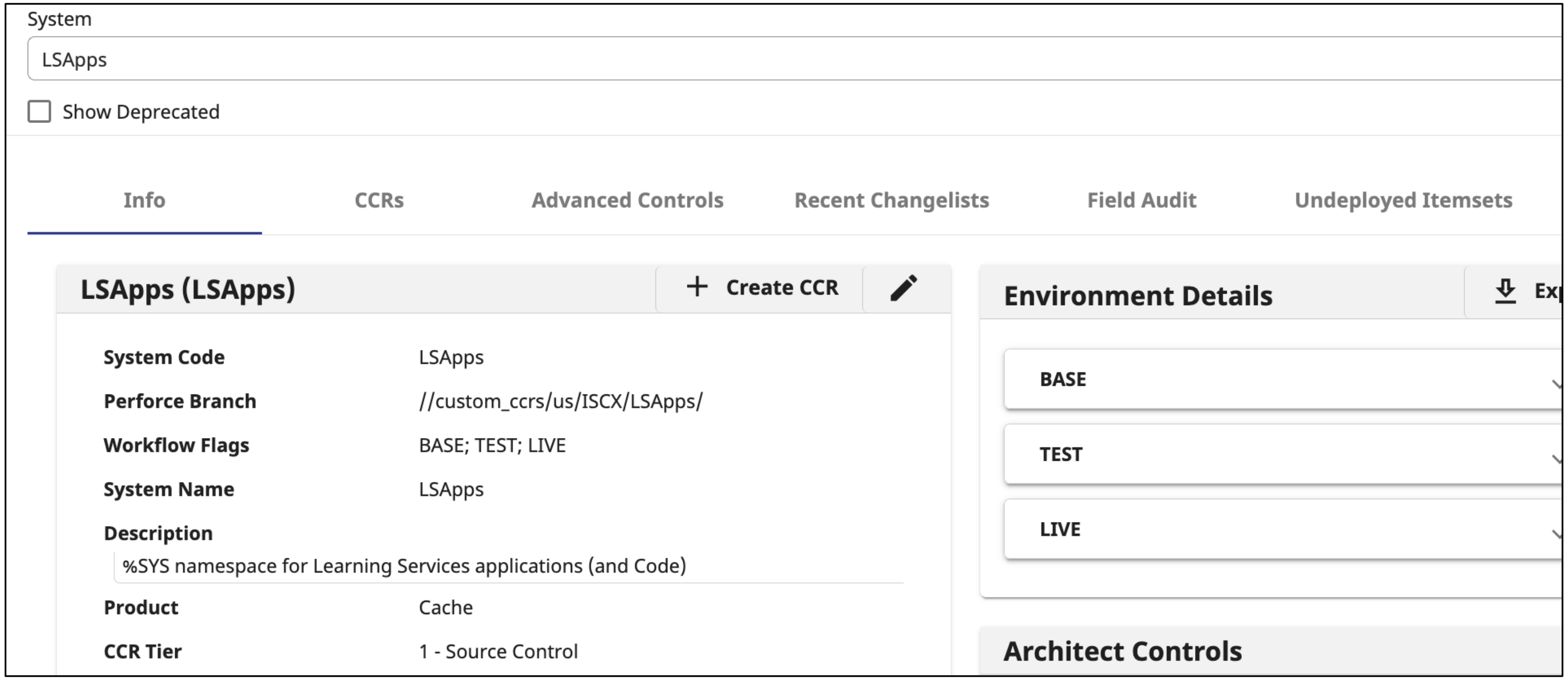
CCR FAQ

- Number of unread FAQ shown in navigation menu.
  - Great resource for delta training!
- Notification disappears after viewing CCR FAQs page.
- Any InterSystems employee can update or create FAQ.



CCR System Details Page

- Contains configuration details for each CCR System.
- Menu > Systems.



System

LSApps

□ Show Deprecated

LSApps (LSApps)

System Code	LSApps
Perforce Branch	//custom_ccrs/us/ISCX/LSApps/
Workflow Flags	BASE; TEST; LIVE
System Name	LSApps
Description	%SYS namespace for Learning Services applications (and Code)
Product	Cache
CCR Tier	1 - Source Control

Environment Details

Architect Controls

Part 3: Introduction to CCR Usage Tiers

CCR Transport – What does it do?

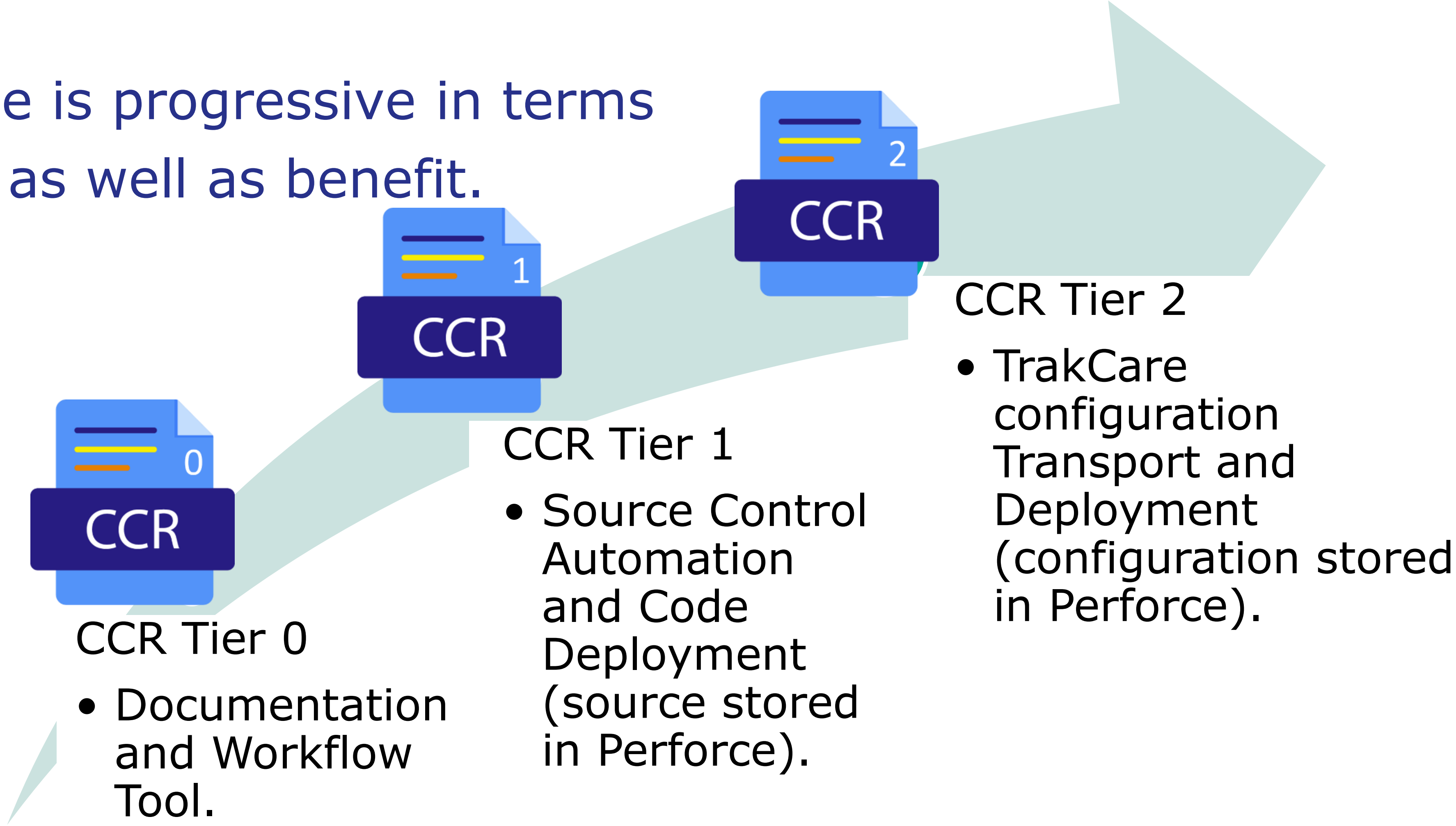
- Mechanism for moving changes between environments, CCR, and Perforce.
- Updates in BASE flagged as changed and exported.
- Exported items transported from BASE into Perforce (source control).
- Items transported out of source control into next environment.

CCR Usage Tiers

- There are different tiers of CCR usage for a project.
  - Provides flexibility for many types of applications to use CCR.
- Projects should determine their appropriate Tier of usage.
- Documentation and training applicable to a Tier should be fully understood prior to enabling that Tier.

CCR Usage Tiers (cont.)

- Tiered usage is progressive in terms of complexity as well as benefit.



CCR Tier 0

- Documentation and Workflow Tool.

CCR Tier 1

- Source Control Automation and Code Deployment (source stored in Perforce).

CCR Tier 2

- TrakCare configuration Transport and Deployment (configuration stored in Perforce).

Example Tier 0 Changes

- File system path changes.
- Memory settings.
- User creation.
- Patching.
- Upgrades.
- Any manual change to the environment or System.

Example Tier 1 Changes

- Classes.
- Routines.
- Reports.
- CSP Pages.
- JavaScript.
- Integration logic.
- DTL.
- Productions.

Example Tier 2 Changes

- TrakCare Code Tables
  - Layouts.
  - Security groups.
  - Workflow.

Quiz: CCR Tiers and Source Control

Question:

At which CCR Tier(s) are changes versioned and stored in Perforce? Select one.

A. CCR Tier 0.

B. CCR Tier 1.

C. CCR Tier 2.

D. CCR Tiers 0 and 1.

E. CCR Tiers 1 and 2.

F. CCR Tiers 0, 1 and 2.

Quiz: CCR Tiers and Source Control (cont.)

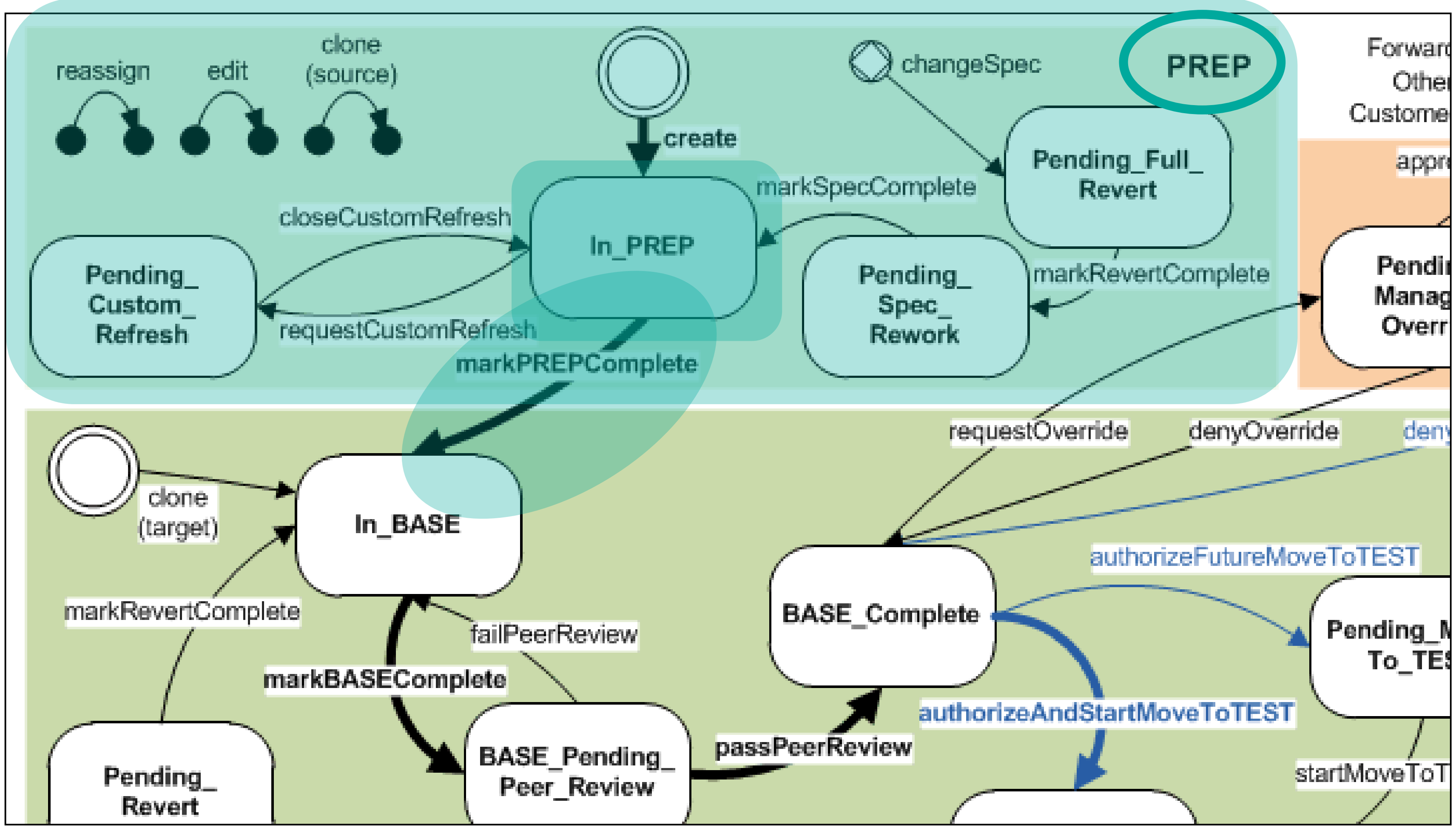
Answer:

E. CCR Tiers 1 and 2.

Part 4: CCR Workflow Overview

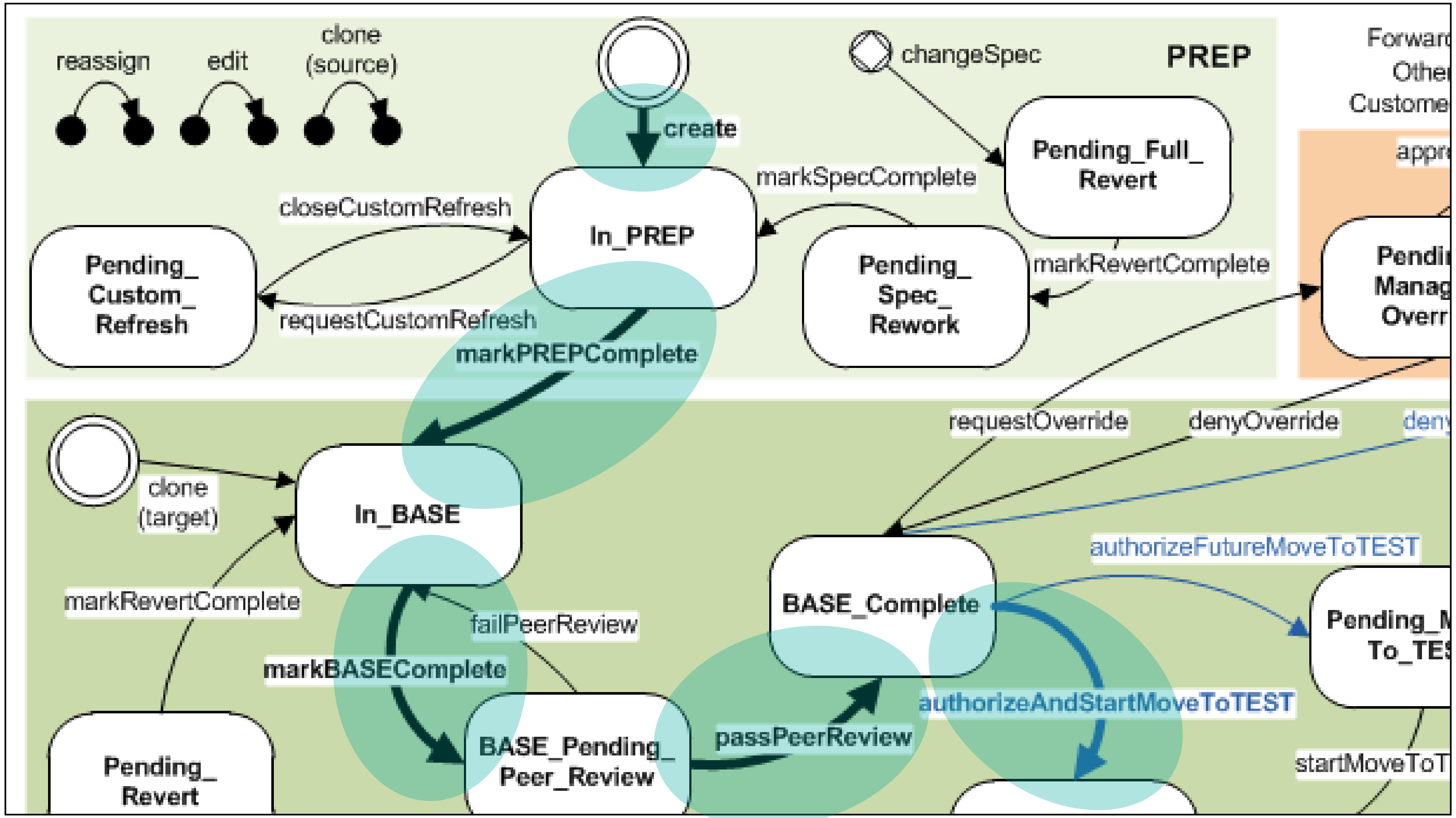
CCR Workflow Vocabulary

- State = current position within workflow.
    - In_PREP.
- Phase = collection of states.
    - PREP.
- Transition = move between CCR states.
    - markPREPComplete.



CCR Workflow Vocabulary

- Default workflow refers to typical workflow for most CCRs in System.



Part 5: Principle of Little and Often

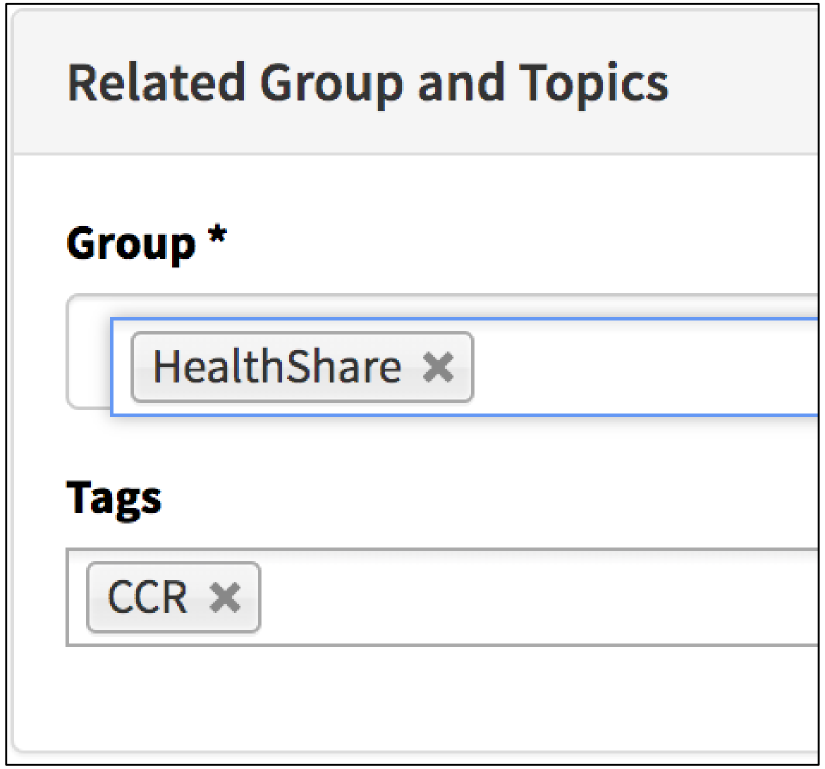
Principle of Little and Often

- Critical to successful usage of CCR.
- Minimize number of changes in 1 CCR.
  - All or none of the changes in a CCR progress through the workflow.
    - Difficult to isolate 1 change to progress separately.
  - Reduces the probability of blocked workflow due to dependencies.
  - Enables clear documentation of the impact of a CCR.
    - Better titles and descriptions due to more narrowly defined changes.
- Progress CCRs quickly (but safely).
  - Do not allow CCRs to remain mid-workflow for too long (weeks).
  - Minimize time between starting change and completing in all environments.

Part 6: Resources

Developer Community

- community.intersystems.com.
- Please post your questions so that others may benefit from answer too.
- When creating post, enter group corresponding to InterSystems product and CCR for tag.



Related Group and Topics

Group *

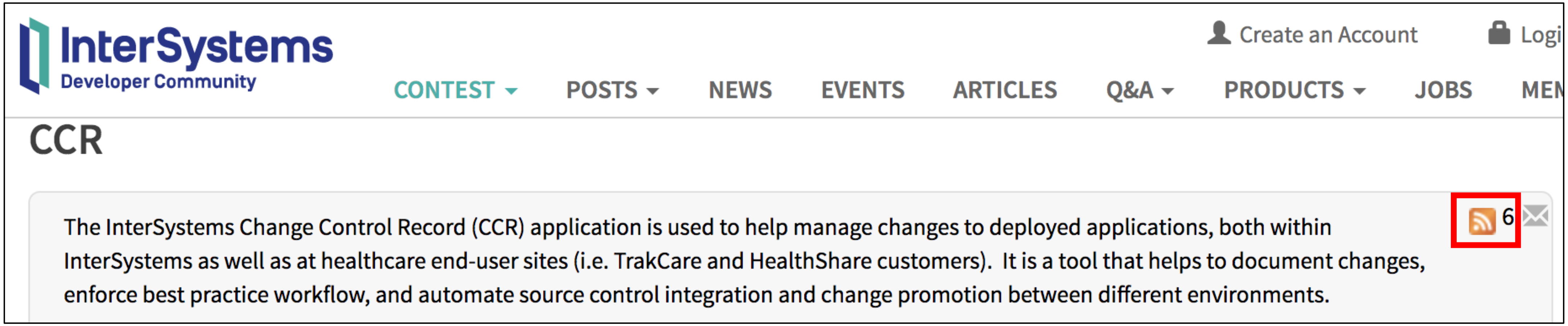
HealthShare ✕

Tags

CCR ✕

Developer Community (cont.)

- To subscribe to posts with CCR tag:
  - Go to community.intersystems.com/tags/ccr.
  - Click envelope icon on right.



CCR

The InterSystems Change Control Record (CCR) application is used to help manage changes to deployed applications, both within InterSystems as well as at healthcare end-user sites (i.e. TrakCare and HealthShare customers). It is a tool that helps to document changes, enforce best practice workflow, and automate source control integration and change promotion between different environments.

Learning Resources

- All Course Content PDF.
  - Merge of all PDFs of all training slides.
  - Also available from navigation menu.
- InterSystems Change Control (ICC) online learning.
  - Go to learning.intersystems.com and search for CCR.
  - Growing catalog of courses available, new courses announced on Developer Community.
  - No exercises available online.

Learning Resources (cont.)

- CCR Online FAQs.
  - Major functionality changes and updates posted here.
- CCR Online Documentation.
  - See full workflow diagram under State Diagram tab.

Additional InterSystems-Internal Resources

- For InterSystems Employee use:
- ccrusers@InterSystems.com internal mailing list for CCR questions
  - All users of CCR should join this mailman list to be kept up to date with CCR-related announcements, enhancements, outage notifications, etc.
  - InterSystems users are expected to keep their customer in the loop on new enhancements, planned outages, etc.

Additional InterSystems-Internal Resources

- perforce@InterSystems.com internal mailing list for Perforce questions.
- iknow/sites/AppS/CCR Application Project site on iKnow.

Summary

- What are the key points for this module?

Demonstration

- Overview of using CCR to document and progress a change.

ICC 300:
Introduction to CCR Tier 0

InterSystems
Creative data technology

Objectives

- Explain at a high level what CCR Tier 0 means and why it is important.
- Identify whether a deeper dive into CCR Tier 0 training material would be beneficial.

Part 1:
CCR Tier 0: Documentation and Workflow

Review: Example types of changes for each CCR Tier

- CCR Tier 0:
  - File system path changes.
  - Memory settings.
  - User creation.
  - Patching.
  - Upgrades.
  - Any ‘manual’ change to the environment or System.

- CCR Tier 1:
  - Classes.
  - Routines.
  - Reports.
  - CSP pages.
  - JavaScript.
  - Integration logic.
  - DTL.
  - Productions.

Review: Example types of changes for each CCR Tier (cont.)

- CCR Tier 2:
  - Layouts
  - Security groups.
  - Code table changes.
  - Workflow.

CCR Tier 0: The Fundamentals

- A “CCR Tier 0” Change Control Record is for “documentation purposes only.”
  - No record of the change is maintained in source control.
- A solid understanding of CCR Tier 0 usage and best practices is a prerequisite for using CCR at higher Tiers!
  - The CCR Workflow is best first learned on a “CCR Tier 0” record which is simpler to understand than Tier 1 and Tier 2 CCRs (these require the use of “CCR Transport”).
  - Except for extra steps related to CCR Transport, the workflow of Tier 0 CCRs is identical to Tier 1 and Tier 2 CCRs.

CCR Tier 0: The Fundamentals (cont.)

- A “CCR Tier 0” change is implemented *manually* in all environments.
  - Likewise, if the change needs to be backed out that is a completely manual task.
- Standard changes should all be CCR Tier 0 changes.
  - Standard changes should never involve ‘custom code’ which is stored in source control.

CCR Tier 0: The Audience

- The courses focused on CCR Tier 0 (ICC3nn) are a foundation for everything else related to proper CCR usage; therefore, much of the material is a prerequisite for other courses.
- Anyone who is involved with CCR should become familiar with CCR Tier 0 best practices.

CCR Tier 0: The Audience (cont.)

- Tier 0 material relates to:
  - The default workflow for CCR and the purpose and best practices around the data collected as part of that workflow.
  - Non-default workflow paths and when they should be used.
  - Project Management (merging, grouping, PM reports).
  - Initial System definition.
  - Peer review best practices.
- CCR Tier 0 reference/training material covers everything that doesn’t relate to “source.”

Quiz: CCR Tier 0 Applicability

Question:

CCR Tier 0 training material is not applicable to sites and users who will be spending most of their time using CCR Tier 1 or CCR Tier 2. True or False?

Answer:

False.

A thorough understanding of CCR Tier 0 principles is a requirement for all CCR users.

Areas for Further Study for CCR Tier 0

- ICC310 – Usage Basics.
  - Proper use of Default CCR workflow and fields.
- ICC320 – Configuring System Details.
- ICC330 – Auxiliary Tools and Transitions.
  - Peer Review details.
  - Additional Possible Transitions.
- ICC340 – Project Management.
  - CCR Groups and Reports.
- ICC350 – Standard Changes.

Summary

- What are the key points for this module?

ICC 310:
CCR Tier 0 – Usage Basics

InterSystems
Creative data technology

Objectives

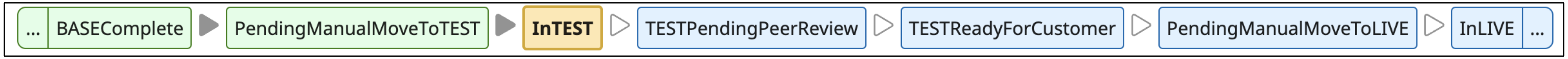
- Define terms in CCR workflow.
- Create and progress Tier 0 CCRs.
- Perform tasks at appropriate times during CCR workflow.
- Explain importance and usage of each CCR documentation field.

Workflow Diagram

- Shows all possible states, phases, transitions for BASE > TEST > UAT > LIVE workflow.
  - Slides assume BASE > TEST > LIVE workflow.
- Also known as:
  - State diagram.
  - Big scary diagram.
- Available from within CCR application.
  - Menu > Documentation > State Diagram.

Workflow Diagram (cont.)

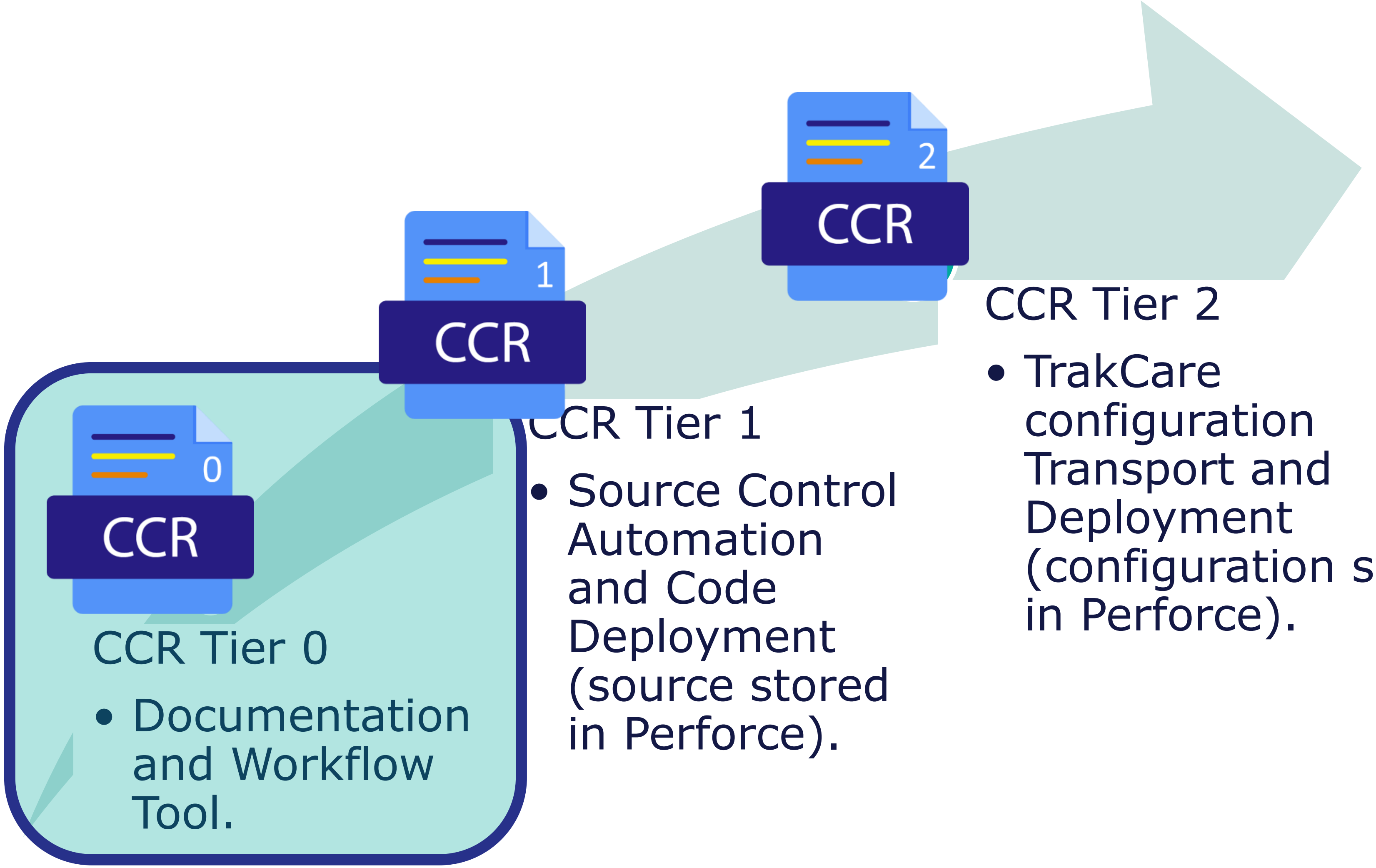
- States in workflow for your particular system available at the top of every CCR.



Part 1: The Fundamentals of a CCR

Review: CCR Usage Tiers

This course focuses on CCR Tier 0



CCR Tier 0

- Documentation and Workflow Tool.

CCR Tier 1

- Source Control Automation and Code Deployment (source stored in Perforce).

CCR Tier 2

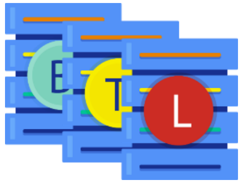
- TrakCare configuration Transport and Deployment (configuration stored in Perforce).

Key Terms

- Organization.
  - Customer organization owns application and CCR System.
  - All users belong to an organization based on their employer.
  - InterSystems employees (ISCX) can create CCRs for any System.
  - Members of Organization that owns System can create CCRs for that System.
- Responsible Organization.
  - Organization responsible for progressing and implementing a CCR.
  - Every CCR has one Responsible Organization.
    - Organization of user that created the CCR.
  - Only members of Responsible Organization can edit the CCR.

Key Terms (cont.)

- SiteCode.
  - 4 character code for an organization.
    - ISCX for InterSystems.

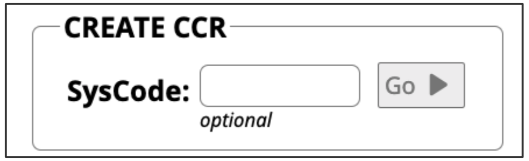


- System.
  - Collection of environments forming a CCR progression.
  - One organization can have many systems.
- SysCode.
  - Unique, unchanging short name for system.

Part 2: CCR PREP Phase

Create a CCR

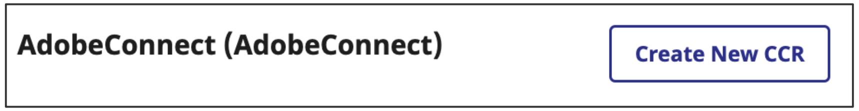
- CCR Homepage > Create CCR.



CREATE CCR

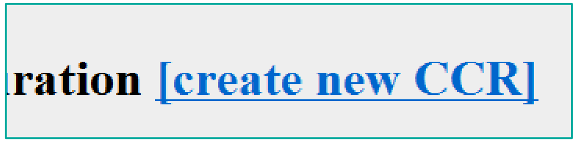
SysCode:
optional

- System Details Page > Create New CCR.



AdobeConnect (AdobeConnect)

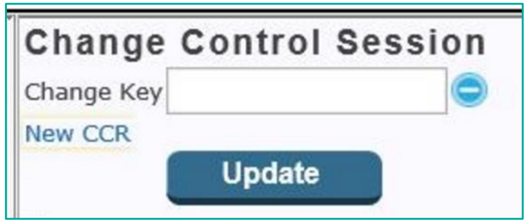
- VS Code - ObjectScript > Source Control menu > Show CCR Controls > [create new CCR].



ration [create new CCR]

Create a CCR (cont.)

- TrakCare Tools > Change Session > New CCR.



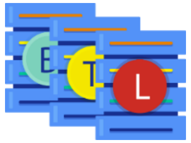
Change Control Session

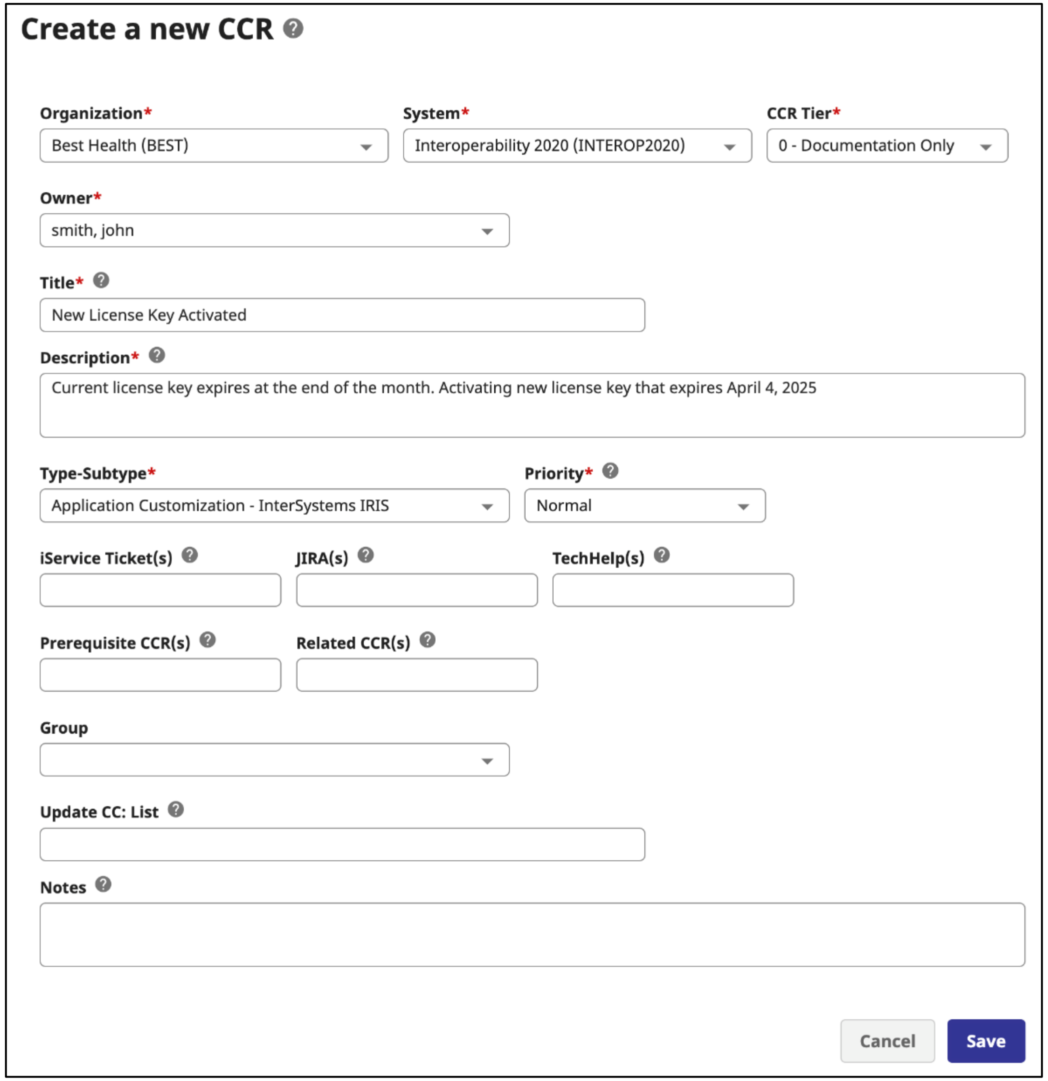
Change Key

- ISCLaunch > [Ctrl]+[Win]+H: type <SiteCode>/<SystemCode>.

Transition: create

- Ensure the proper System is selected.
- Include a useful Title and Description.
  - Think of this as a ‘release note’ which might be sent to users (use ‘present tense’ - change in LIVE).
- Type is typically Application Customization.
  - Has no effect on functionality or reporting.





Create a new CCR

Organization*	System*	CCR Tier*
Best Health (BEST)	Interoperability 2020 (INTEROP2020)	0 - Documentation Only

Owner*: smith, john

Title*: New License Key Activated

Description*: Current license key expires at the end of the month. Activating new license key that expires April 4, 2025

Type-Subtype*: Application Customization - InterSystems IRIS

Priority*: Normal

iService Ticket(s):

JIRA(s):

TechHelp(s):

Prerequisite CCR(s):

Related CCR(s):

Group:

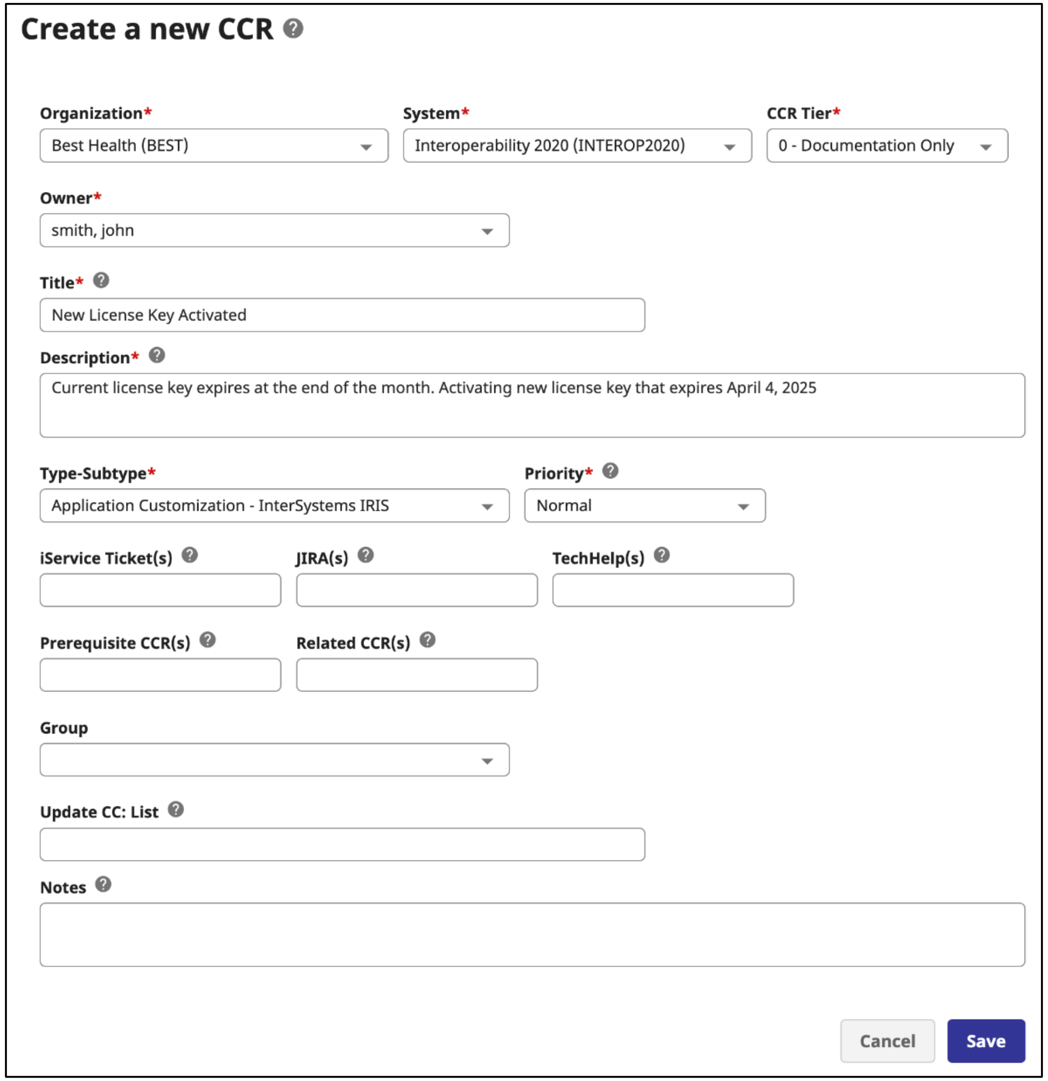
Update CC: List:

Notes:

Destination: In_PREP

Transition: create (cont.)

- Link related ticket(s).
  - TRC, JIRA, TechHelp and/or Prodlog.
    - Contact InterSystems to make JIRA field link to non-InterSystems server.
- Prerequisite CCR must precede this CCR.
- Related CCR is just a logical link.
- Click Save once finished.



Create a new CCR

Organization*: Best Health (BEST)

System*: Interoperability 2020 (INTEROP2020)

CCR Tier*: 0 - Documentation Only

Owner*: smith, john

Title*: New License Key Activated

Description*: Current license key expires at the end of the month. Activating new license key that expires April 4, 2025

Type-Subtype*: Application Customization - InterSystems IRIS

Priority*: Normal

iService Ticket(s):

JIRA(s):

TechHelp(s):

Prerequisite CCR(s):

Related CCR(s):

Group:

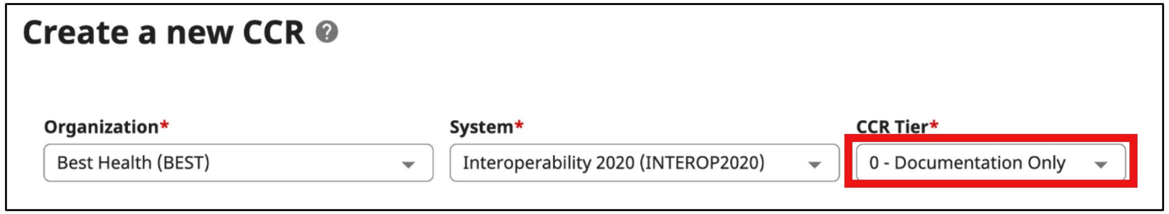
Update CC: List:

Notes:

Destination: In_PREP

Specifying the CCR Tier Setting

- System level.
  - Maximum tier of any CCR.
  - Default tier of new CCRs.
  - Example: Tier 1 system can have Tier 0 and Tier 1 CCRs.
- CCR Record level.
  - Tier for that CCR.
  - Defaults to System Tier.
  - Change from default if appropriate.
  - Determines enforced rules and workflow.



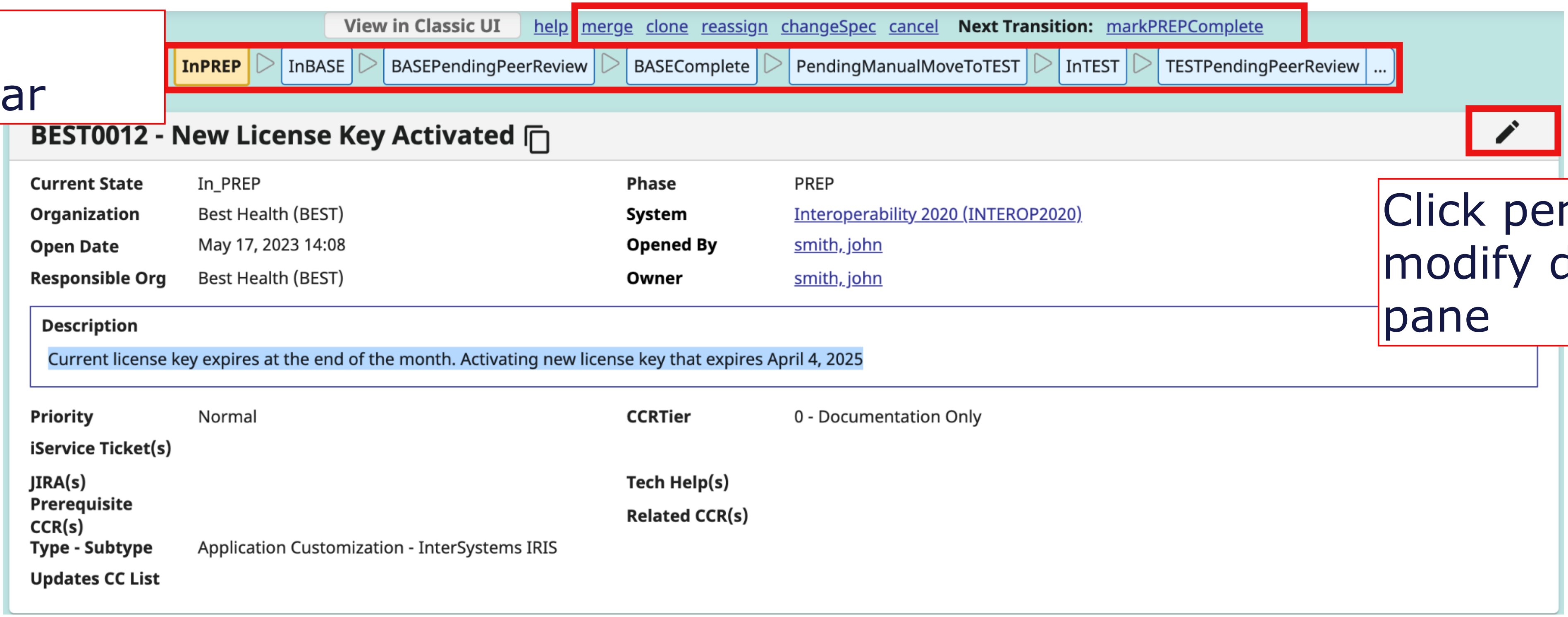
Create a new CCR

Organization*: Best Health (BEST)

System*: Interoperability 2020 (INTEROP2020)

CCR Tier*: 0 - Documentation Only

Review: CCR Record


View in Classic UI
help
merge clone reassign changeSpec cancel

Next Transition: markPREPComplete

InPREP ▷ InBASE ▷ BASEPendingPeerReview ▷ BASEComplete ▷ PendingManualMoveToTEST ▷ InTEST ▷ TESTPendingPeerReview ...

BEST0012 - New License Key Activated

Current State	In_PREP	**Phase**	PREP
Organization	Best Health (BEST)	**System**	Interoperability 2020 (INTEROP2020)
Open Date	May 17, 2023 14:08	**Opened By**	smith, john
Responsible Org	Best Health (BEST)	**Owner**	smith, john

Description

Current license key expires at the end of the month. Activating new license key that expires April 4, 2025

Priority	Normal	**CCRTier**	0 - Documentation Only
iService Ticket(s)			
JIRA(s)		**Tech Help(s)**	
Prerequisite CCR(s)		**Related CCR(s)**	
Type - Subtype	Application Customization - InterSystems IRIS		
Updates CC List			

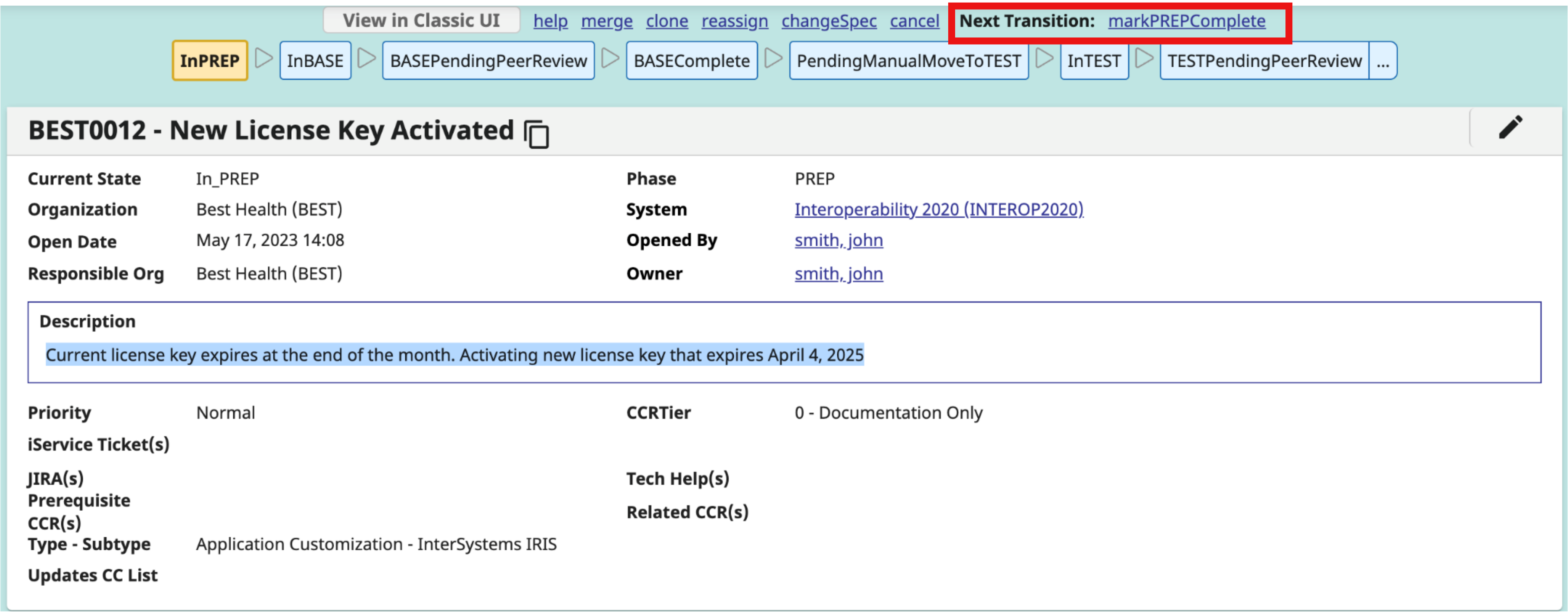
Available Transitions

Workflow progress bar

Click pencil to modify details in pane

CCR Navigation

- Next Transition(s)
  - Move CCR to next state
  - Default workflow is right-most transition.



BEST0012 - New License Key Activated

Current State	In_PREP	**Phase**	PREP
Organization	Best Health (BEST)	**System**	Interoperability 2020 (INTEROP2020)
Open Date	May 17, 2023 14:08	**Opened By**	smith, john
Responsible Org	Best Health (BEST)	**Owner**	smith, john

Description

Current license key expires at the end of the month. Activating new license key that expires April 4, 2025

Priority	Normal	**CCRTier**	0 - Documentation Only
iService Ticket(s)			
JIRA(s)		**Tech Help(s)**	
Prerequisite CCR(s)		**Related CCR(s)**	
Type - Subtype	Application Customization - InterSystems IRIS		
Updates CC List			

Quiz: Who Can Create a CCR

Question:

CCR users of which of the following organizations can create CCRs for a system owned by Today Robot?

A. Only InterSystems.

B. Only Today Robot.

C. InterSystems and Today Robot.

D. Any organization using CCR.

Quiz: Who Can Create a CCR (cont.)

Answer:

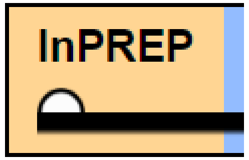
C. InterSystems and Today Robot.

Keep in mind the user that creates the CCR should also implement and progress the CCR. The Responsible Organization for the CCR will be the organization of the user that created the CCR.

Exercise ICC310-1

Create a CCR

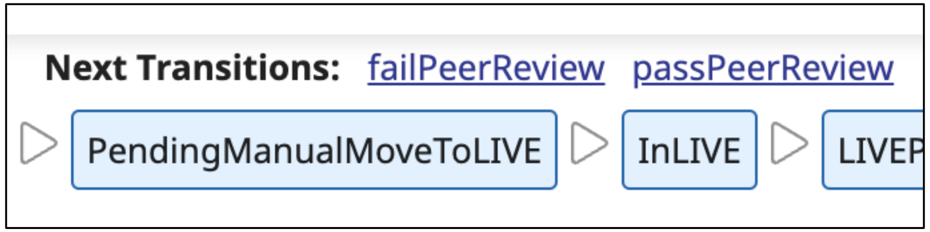
State: In_PREP



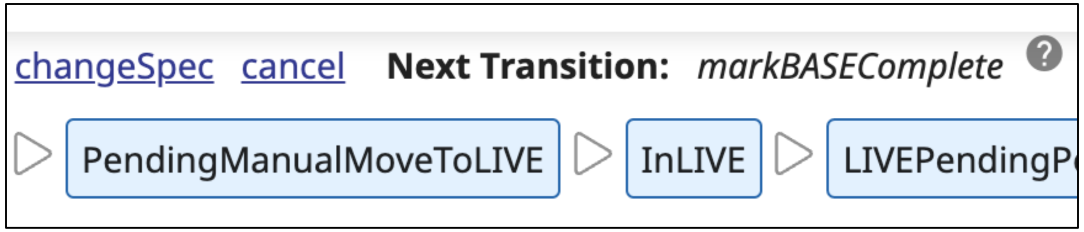
- Determine and document expected modifications.
- Do not make any changes to BASE yet.
- Fields to consider:
  - Modified Items: What are the specific things being changed, added or removed?
    - Class definitions.
    - Data.
  - Impacted Areas: What are the areas of the System which could be impacted by the things that were modified?
    - Ability for student50 to login.
    - Appearance of a button.
    - Will be starting point for writing testing plan later.

Performing Transitions

- Transitions found in upper-right corner when viewing CCR record.
- Right-most link moves CCR forward on default path towards closure.
- Cannot perform transition shown in italics with ?.
    - Hover over ? for reason.



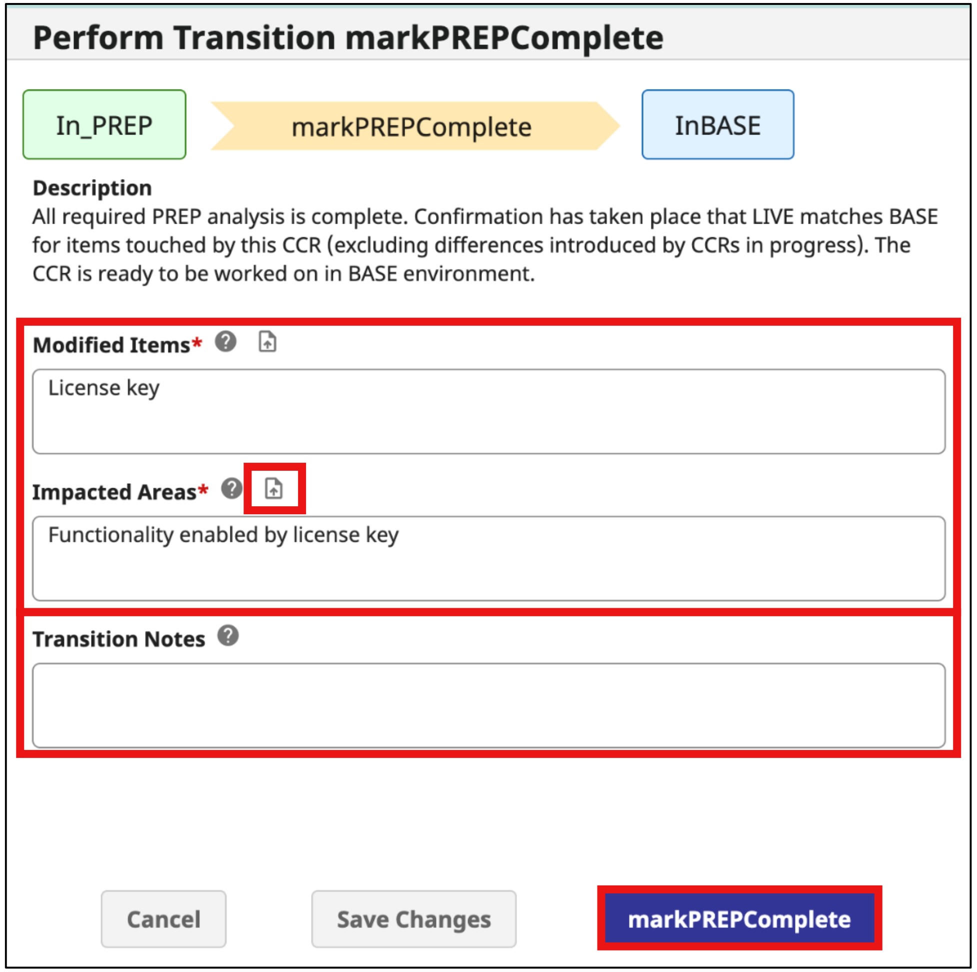
Next Transitions: failPeerReview passPeerReview



changeSpec cancel **Next Transition:** *markBASEComplete* ?

Performing Transitions (cont.)

- Required fields listed.
  - Ensure complete and accurate.
- Attachments may be added.
  - Automatically versioned.
- Transition Notes usually optional.
  - Exception: transitions that move CCR backwards.
    - Example: failPeerReview or markAcceptanceFailed.
- Click the transition button to finish.



Perform Transition markPREPComplete

Description
All required PREP analysis is complete. Confirmation has taken place that LIVE matches BASE for items touched by this CCR (excluding differences introduced by CCRs in progress). The CCR is ready to be worked on in BASE environment.

Modified Items*
License key

Impacted Areas*
Functionality enabled by license key

Transition Notes

Transition: markPREPComplete

- Default transition for moving CCR forward, out of In_PREP.
  - Transition CCR once basics of the change thought through.
- Able to complete or modify required fields after clicking markPREPComplete.
- Will throw error if Modified Items and Impacted Areas are identical.

Destination: In_BASE

Quiz: Initiating a Transition

Question:

Which of the following is NOT in the process for performing the markPREPComplete transition?

A. Click the markPREPComplete transition link to initiate the transition.

B. Complete the Modified Items and Impacted Areas fields, if necessary.

C. Click the markPREPComplete button to finalize the transition.

D. Click Edit to make markPREPComplete transition link visible.

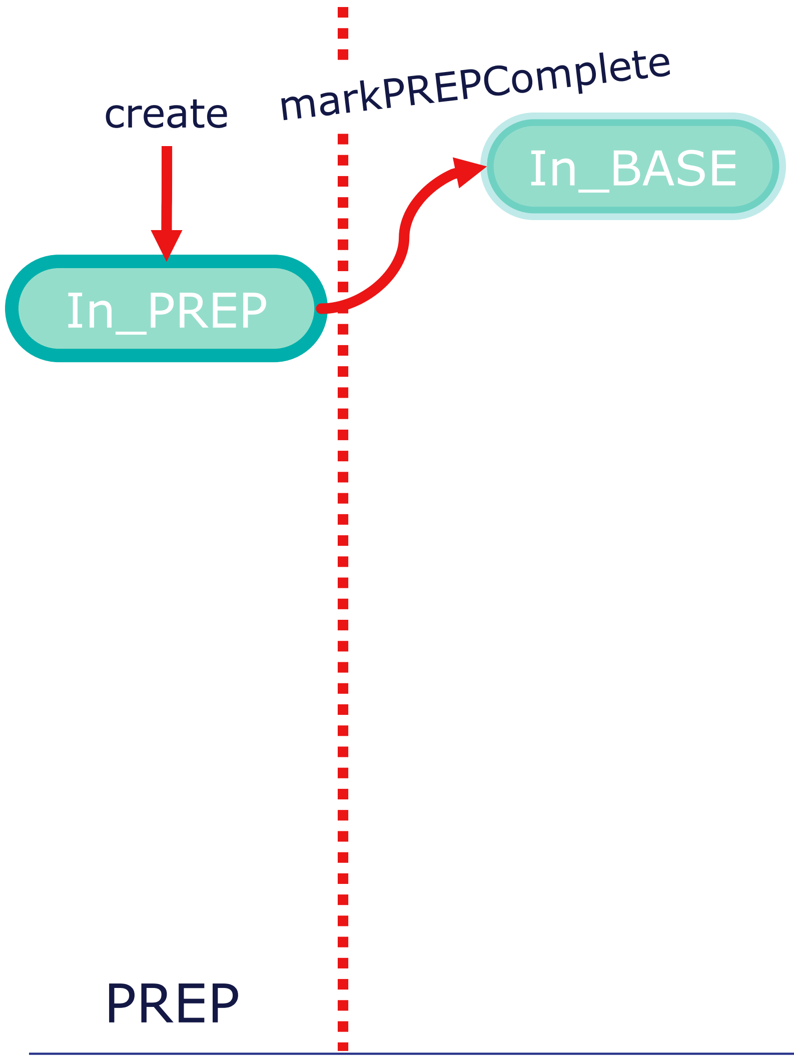
Quiz: Initiating a Transition (cont.)

Answer:

D. Click Edit to make markPREPComplete transition link visible.

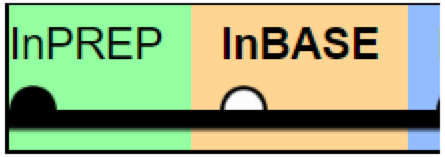
The edit icons in a CCR allow you to modify fields for that section of the CCR but have nothing to do with transitions. The transition links are always available in the navigation links at the top of the CCR.

Reviewing the CCR Journey…



Part 3: CCR BASE Phase (Part 1)

State: In_BASE

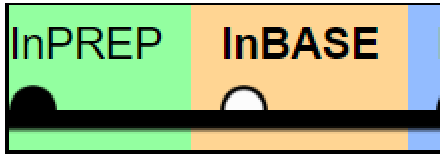


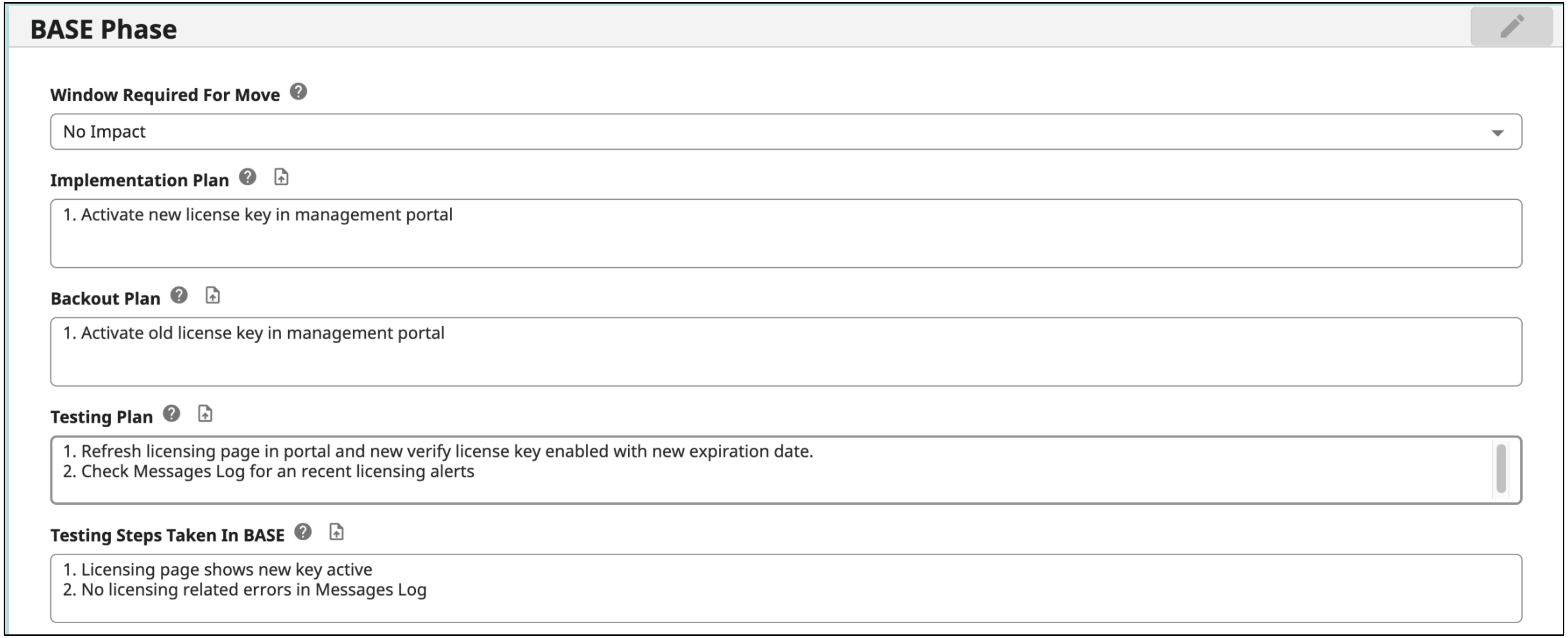
- Required fields from PREP Phase available and editable.
- New BASE Phase pane appears.
- Tasks:
  - Documentation.
  - Make change in BASE.
  - Test change in BASE.

Documentation: 3 Planning Fields

- Make sure that Implementation Plan, Backout Plan and Testing Plan thorough and clear.
  - Other people should be able to follow them and perform all steps.
  - Organization should set best practices for level of detail necessary.
    - Example: can you assume knowledge of how to navigate management portal?
  - Could link to any appropriate procedure document instead of duplicating documentation.
- Plans should apply to all environments, indicating exceptions as necessary.

Documentation: 3 Planning Fields (cont.)





BASE Phase

Window Required For Move

No Impact

Implementation Plan

1. Activate new license key in management portal

Backout Plan

1. Activate old license key in management portal

Testing Plan

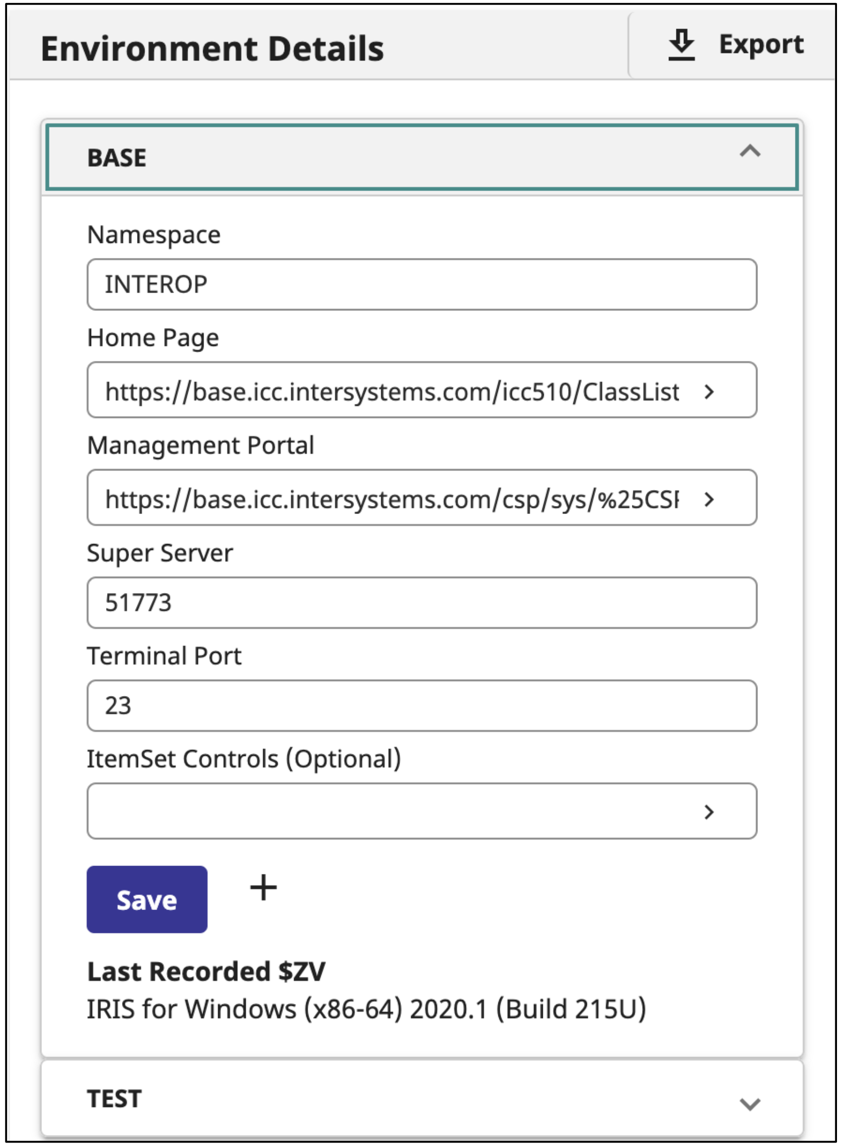
1. Refresh licensing page in portal and new verify license key enabled with new expiration date.
2. Check Messages Log for an recent licensing alerts

Testing Steps Taken In BASE

1. Licensing page shows new key active
2. No licensing related errors in Messages Log

Tokens

- Use tokens in URLs to dynamically link to correct environment.
  - Only use in 3 planning fields.
- <env> = current phase.
  - <env>.myapp.mydomain.com = base.myapp.mydomain.com in BASE.
- <homepage> = Home Page field on System definition.
- <smp> = Management Portal field on System definition.



Namespace: INTEROP

Home Page: https://base.icc.intersystems.com/icc510/ClassList

Management Portal: https://base.icc.intersystems.com/csp/sys/%25CSI

Super Server: 51773

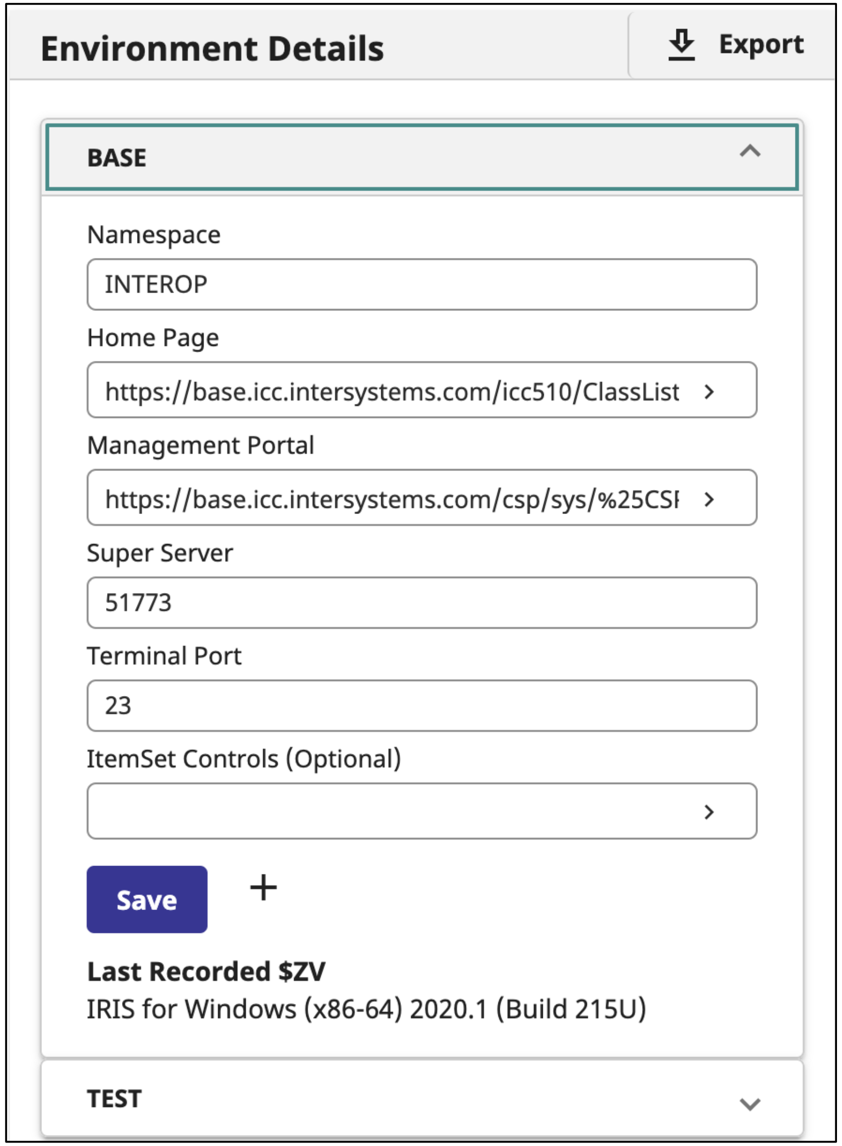
Terminal Port: 23

ItemSet Controls (Optional):

Last Recorded $ZV
IRIS for Windows (x86-64) 2020.1 (Build 215U)

Tokens (cont.)

- <smpPrefix> = domain portion of Management Portal field.
  - <smpPrefix> for screenshot = https://base.icc.intersystems.com/ in BASE.



Environment Details

BASE

Namespace: INTEROP

Home Page: https://base.icc.intersystems.com/icc510/ClassList

Management Portal: https://base.icc.intersystems.com/csp/sys/%25CSI

Super Server: 51773

Terminal Port: 23

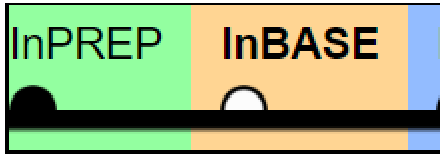
ItemSet Controls (Optional):

Last Recorded $ZV

IRIS for Windows (x86-64) 2020.1 (Build 215U)

TEST

Testing Steps Taken In BASE



- Document results of testing in Testing Steps Taken In BASE.
- Include evidence of testing:
  - Link to pages used for testing in BASE.
    - Do not use <env> token because evidence specific to BASE.
  - Upload attachments.
    - Examples: Screenshots, log files.
  - Helps peer reviewer verify results of testing.

Transition: markBASEComplete

- Perform markBASEComplete transition when:
  - Implementation, testing, and documentation complete.
  - Ready for peer review.

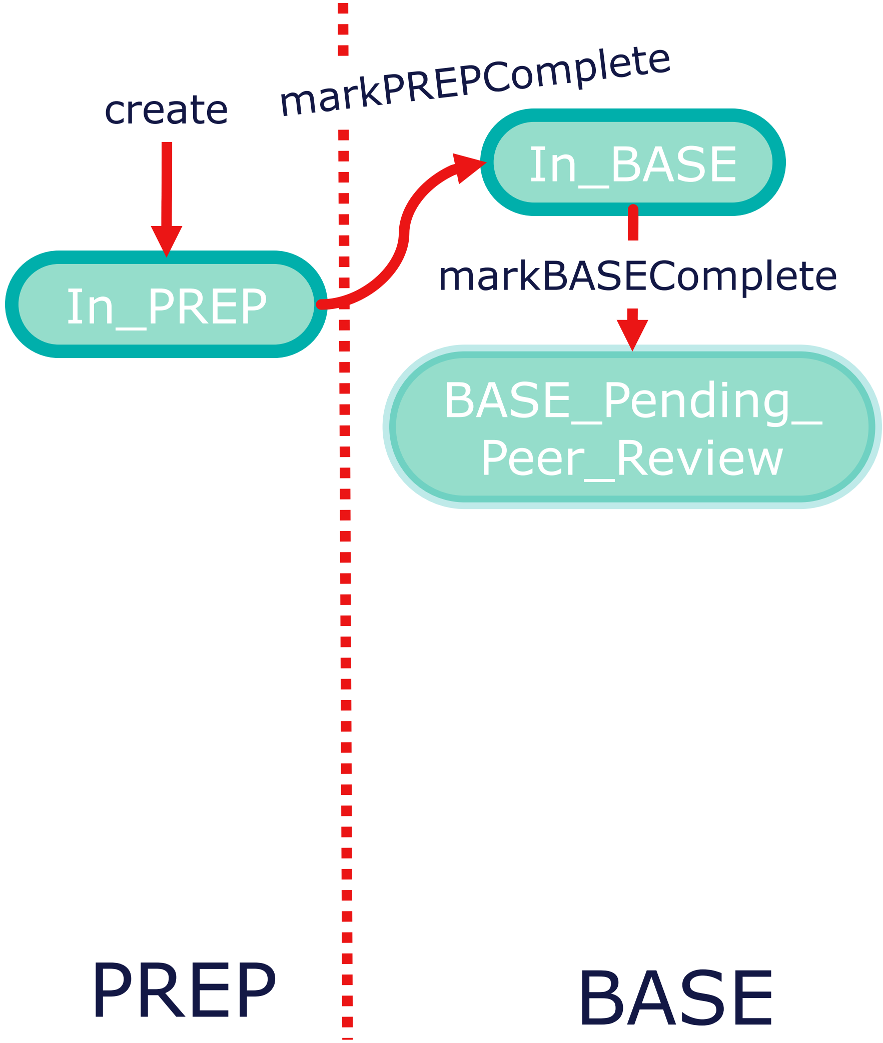
Destination: BASE_Pending_Peer_Review

Transition: markBASEComplete (cont.)

- Window Required for Move set to indicate impact on LIVE users during implementation:
    - No Impact: Users will not notice change during implementation; deploy anytime.
    - Out of Hours: Users may have minor inconvenience; deploy when fewest users are active.
    - Downtime: No users can be on the system during implementation; plan for downtime window.

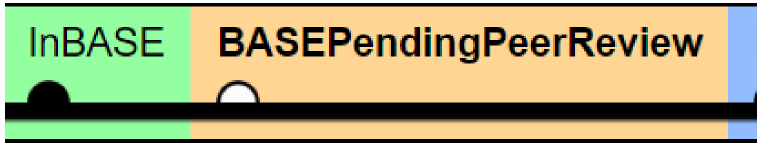
Destination: BASE_Pending_Peer_Review

Reviewing the CCR Journey…



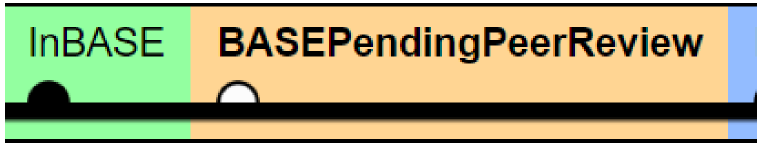
Part 4: Peer Reviews

State: BASE_Pending_Peer_Review



- CCR automatically reassigned to default peer reviewer for.
  - See ICC330 for details on default peer reviewers.
- passPeerReview transition not active for user who performed markBASEComplete.
  - Any other user from responsible organization can perform passPeerReview.

State: BASE_Pending_Peer_Review (cont.)



- Transition options:
  - failPeerReview – CCR should be amended for any number of reasons.
    - Sent back to owner.
    - Returns to In_BASE state.
  - passPeerReview – functionality, documentation and testing look complete and CCR can move forward.

Peer Reviews

- Vital part of workflow.
  - Increases quality of changes and documentation.
  - Reduces chance of errors.
  - Ensures at least 1 other person aware of change.
- May seem like overkill for trivial changes, but has saved sites from major crises.

Peer Review Best Practices

- Peer reviewer:
  - Should feel able to implement change based on documentation in CCR if original implementor unavailable.
  - Equally responsible for bad changes as person who made change.
- Ideally want as thorough of peer review as possible including verifying:
  - Clear and thorough documentation.
  - Ramifications of change.

Peer Reviewer Guidelines for BASE

- Title and Description clearly communicate nature and extent of change.
- Appropriate links to other systems and CCRs.
- Modified Items covers changes in Implementation Plan.
- All impacted areas identified.
- Implementation, Testing, and Backout plans complete and sufficient.
- Appropriate window required for move.
- Appropriate evidence in Testing Steps Taken in BASE.

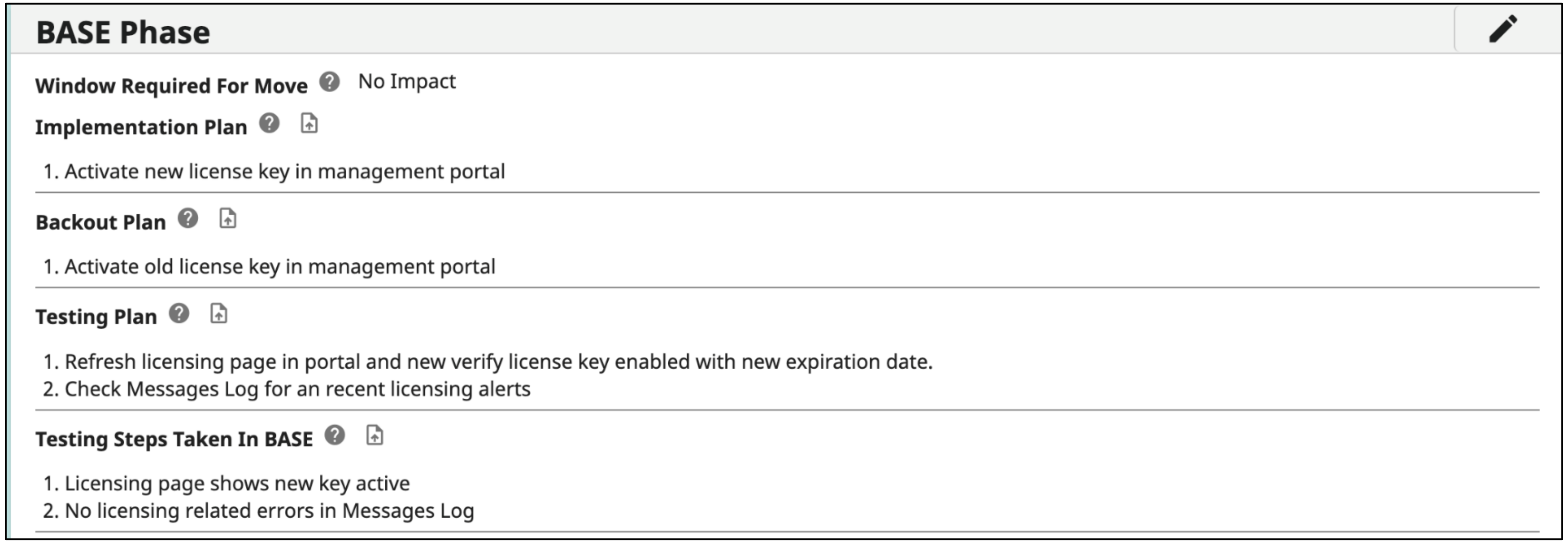
Backwards Transition: failPeerReview

- Not a terminal action.
  - The CCR is moved backwards to In_BASE to get further work.
  - CCR is cumulative, so only the specific fix to the issue is required.
- Don’t be afraid to fail a peer review.
  - Do not hold on to a CCR while owner makes quick corrections.
    - This only makes it confusing as to who is responsible for the change.
  - As a courtesy, it may be a good idea to let the owner know it is coming back and why prior to performing a failPeerReview.
- Transition Note required before confirming failPeerReview transition.

Destination: In_BASE

Transition: passPeerReview

- Review all fields carefully before performing passPeerReview.
- Optionally supply transition notes before confirming the passPeerReview transition.



BASE Phase

Window Required For Move No Impact

Implementation Plan

1. Activate new license key in management portal

Backout Plan

1. Activate old license key in management portal

Testing Plan

1. Refresh licensing page in portal and new verify license key enabled with new expiration date.
2. Check Messages Log for an recent licensing alerts

Testing Steps Taken In BASE

1. Licensing page shows new key active
2. No licensing related errors in Messages Log

Destination: BASE_Complete

Quiz: Peer Review

Question:

A peer reviewer can edit the documentation in a CCR. True or False?

Answer:

True.

However, it is a best practice for peer reviewers to only make minor changes, such as fixing a typo so that the implementer learns how to properly document changes and to make sure the implementor is aware of any major changes to the plans.

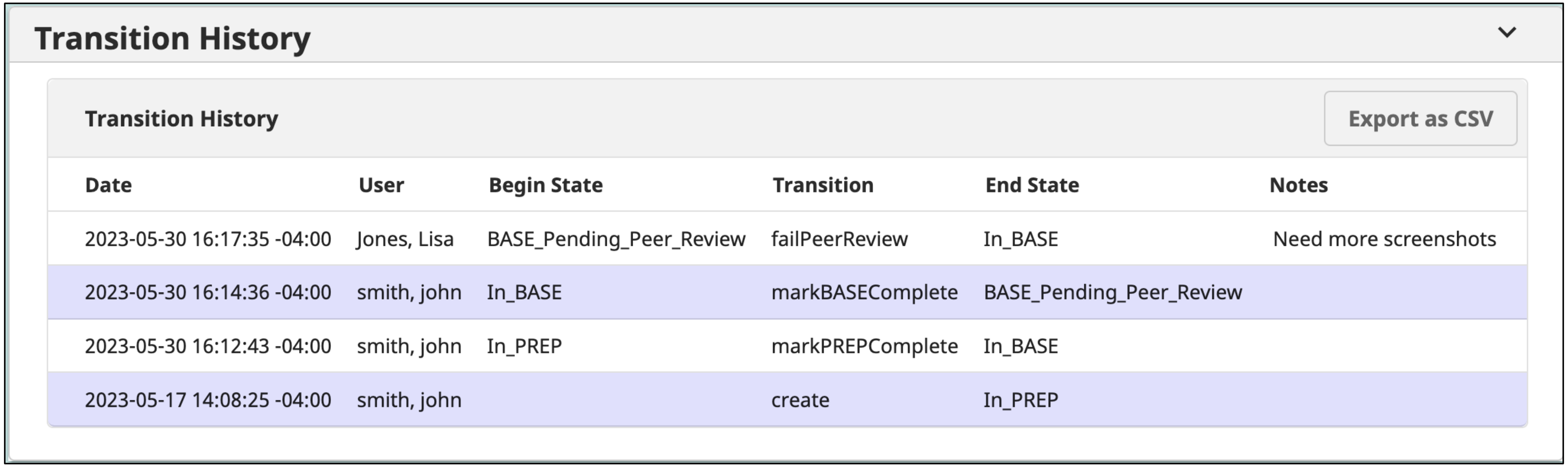
Exercise ICC310-2 and ICC310-3

Document, implement, test, and peer review.

Part 5: CCR BASE Phase (Part 2)

CCR Transition History

- List of transitions for this CCR, including transition details and any Transition Notes.
- Useful for debugging.

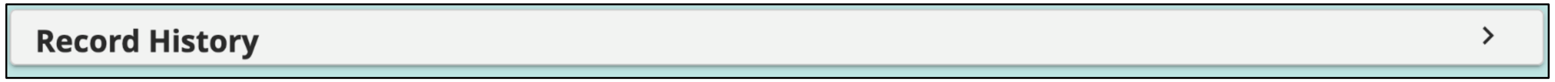


Transition History

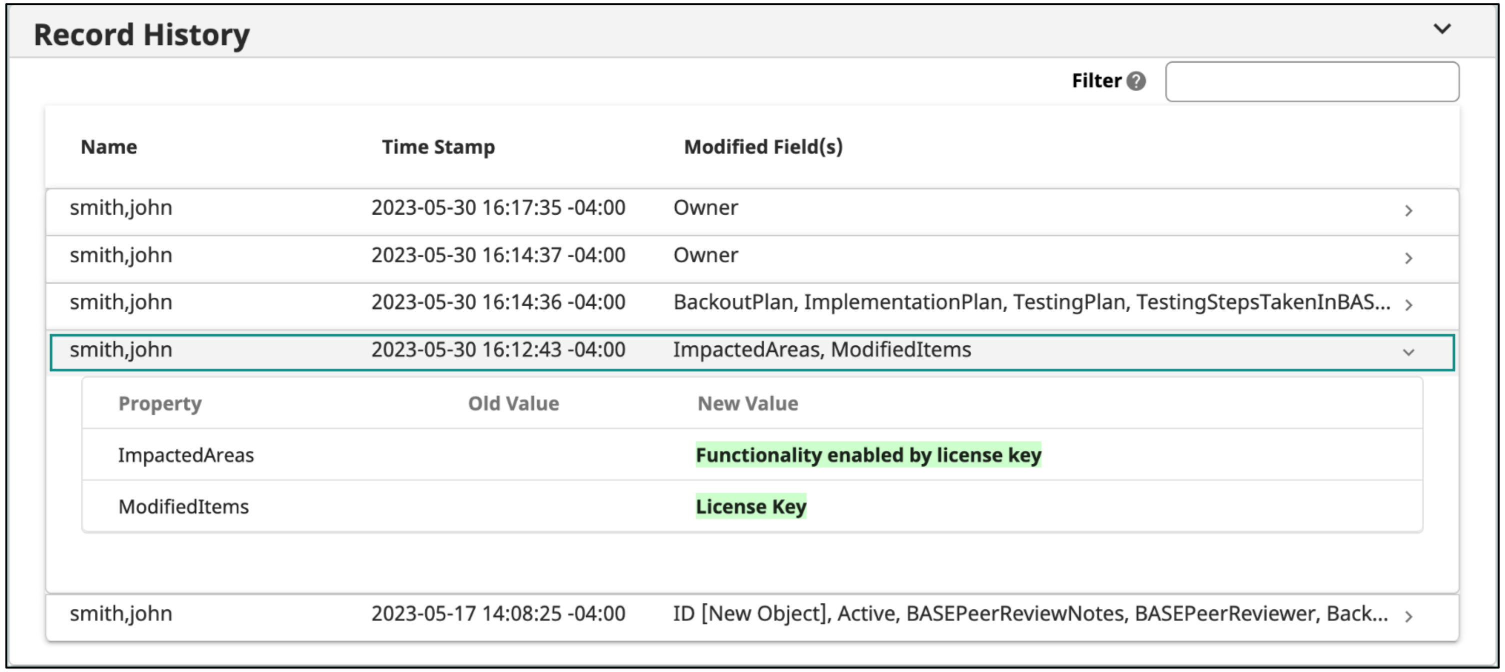
Date	User	Begin State	Transition	End State	Notes
2023-05-30 16:17:35 -04:00	Jones, Lisa	BASE_Pending_Peer_Review	failPeerReview	In_BASE	Need more screenshots
2023-05-30 16:14:36 -04:00	smith, john	In_BASE	markBASEComplete	BASE_Pending_Peer_Review	
2023-05-30 16:12:43 -04:00	smith, john	In_PREP	markPREPComplete	In_BASE	
2023-05-17 14:08:25 -04:00	smith, john		create	In_PREP	

Record History

- Collapsed by default.
- History of changes to fields.



Record History



Record History

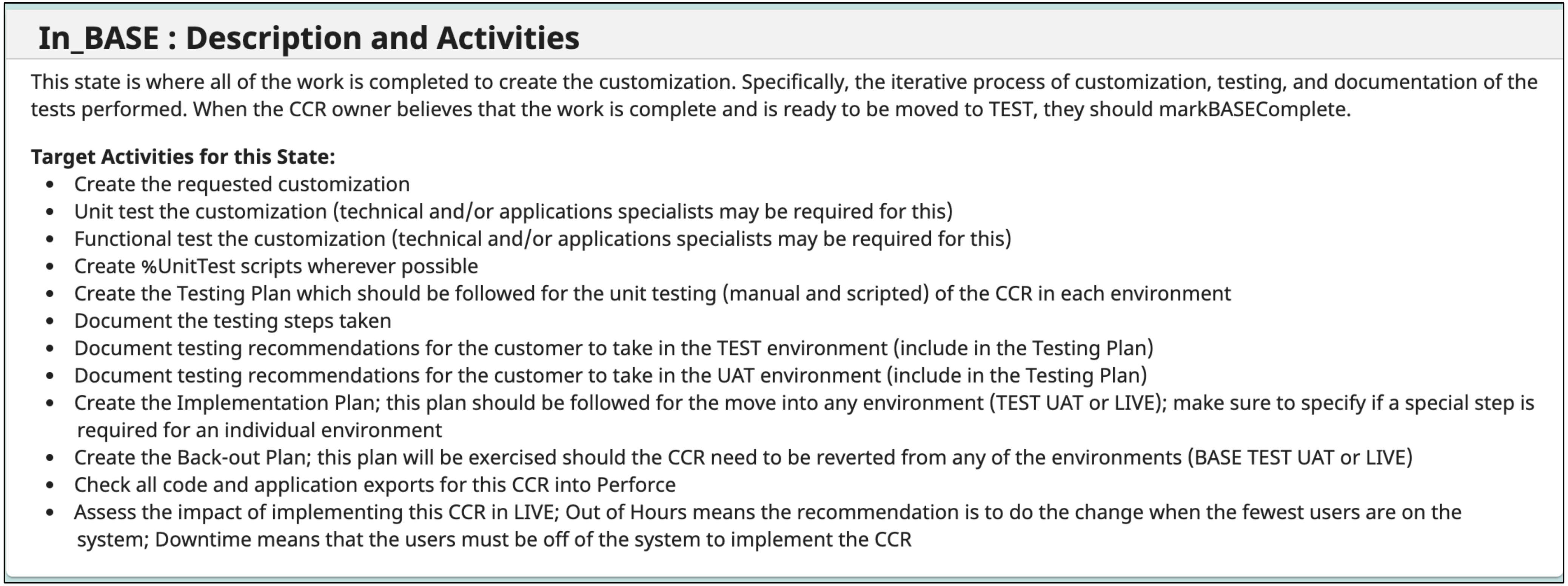
Filter

Name	Time Stamp	Modified Field(s)
smith,john	2023-05-30 16:17:35 -04:00	Owner
smith,john	2023-05-30 16:14:37 -04:00	Owner
smith,john	2023-05-30 16:14:36 -04:00	BackoutPlan, ImplementationPlan, TestingPlan, TestingStepsTakenInBAS...
smith,john	2023-05-30 16:12:43 -04:00	ImpactedAreas, ModifiedItems
smith,john	2023-05-17 14:08:25 -04:00	ID [New Object], Active, BASEPeerReviewNotes, BASEPeerReviewer, Back...

Property	Old Value	New Value
ImpactedAreas		**Functionality enabled by license key**
ModifiedItems		**License Key**

Current State Documentation with CCR Record

- <state_name> Description and Activities.
  - In-place help documentation for what to do next.



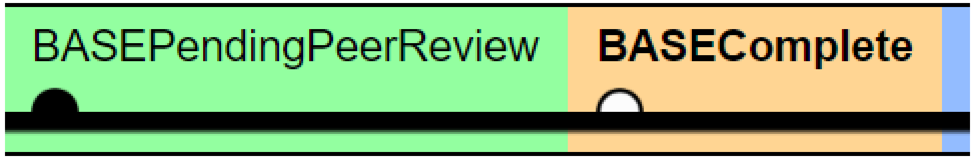
In_BASE : Description and Activities

This state is where all of the work is completed to create the customization. Specifically, the iterative process of customization, testing, and documentation of the tests performed. When the CCR owner believes that the work is complete and is ready to be moved to TEST, they should markBASEComplete.

Target Activities for this State:

- Create the requested customization
- Unit test the customization (technical and/or applications specialists may be required for this)
- Functional test the customization (technical and/or applications specialists may be required for this)
- Create %UnitTest scripts wherever possible
- Create the Testing Plan which should be followed for the unit testing (manual and scripted) of the CCR in each environment
- Document the testing steps taken
- Document testing recommendations for the customer to take in the TEST environment (include in the Testing Plan)
- Document testing recommendations for the customer to take in the UAT environment (include in the Testing Plan)
- Create the Implementation Plan; this plan should be followed for the move into any environment (TEST UAT or LIVE); make sure to specify if a special step is required for an individual environment
- Create the Back-out Plan; this plan will be exercised should the CCR need to be reverted from any of the environments (BASE TEST UAT or LIVE)
- Check all code and application exports for this CCR into Perforce
- Assess the impact of implementing this CCR in LIVE; Out of Hours means the recommendation is to do the change when the fewest users are on the system; Downtime means that the users must be off of the system to implement the CCR

State: BASE_Complete



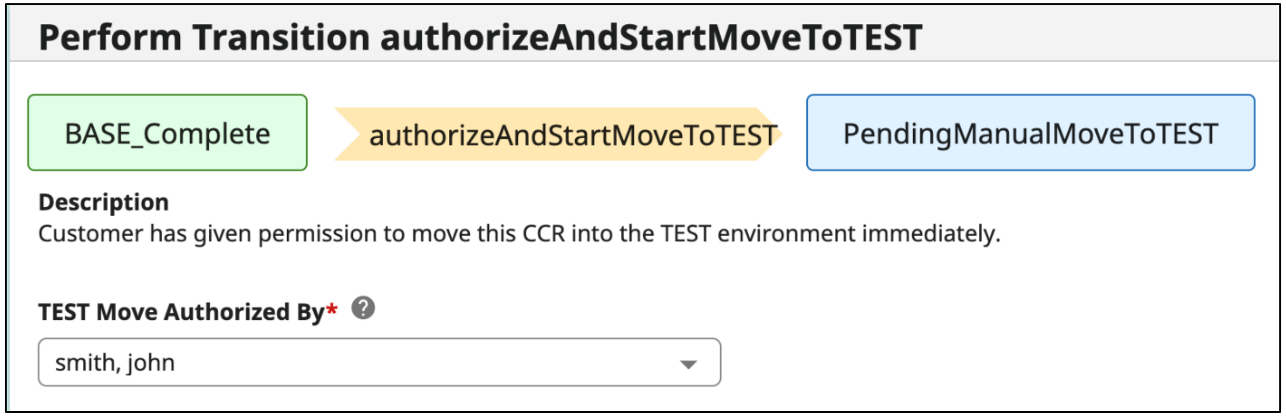
- Specifying, planning, thinking and initial doing complete!
  - Remainder of CCR workflow follows easily based on documentation.
- Must receive authorization before moving CCR and associated changes to TEST.

Authorization

- Authorization can either be:
  - Direct.
    - Must ask someone for authorization to move any CCR.
  - Express.
    - Do not ask for permission to move every change.
    - Have designated person to specify on every CCR as granting authorization.

Authorization: Who Can Do It?

- During transition that requires authorization, authorizer chosen from drop-down menu.
  - By default, includes all users from Organization that owns System.
  - Can restrict list to specific users.
    - Create an iService ticket to request initializing list.



Perform Transition authorizeAndStartMoveToTEST

Description
Customer has given permission to move this CCR into the TEST environment immediately.

TEST Move Authorized By*

smith, john

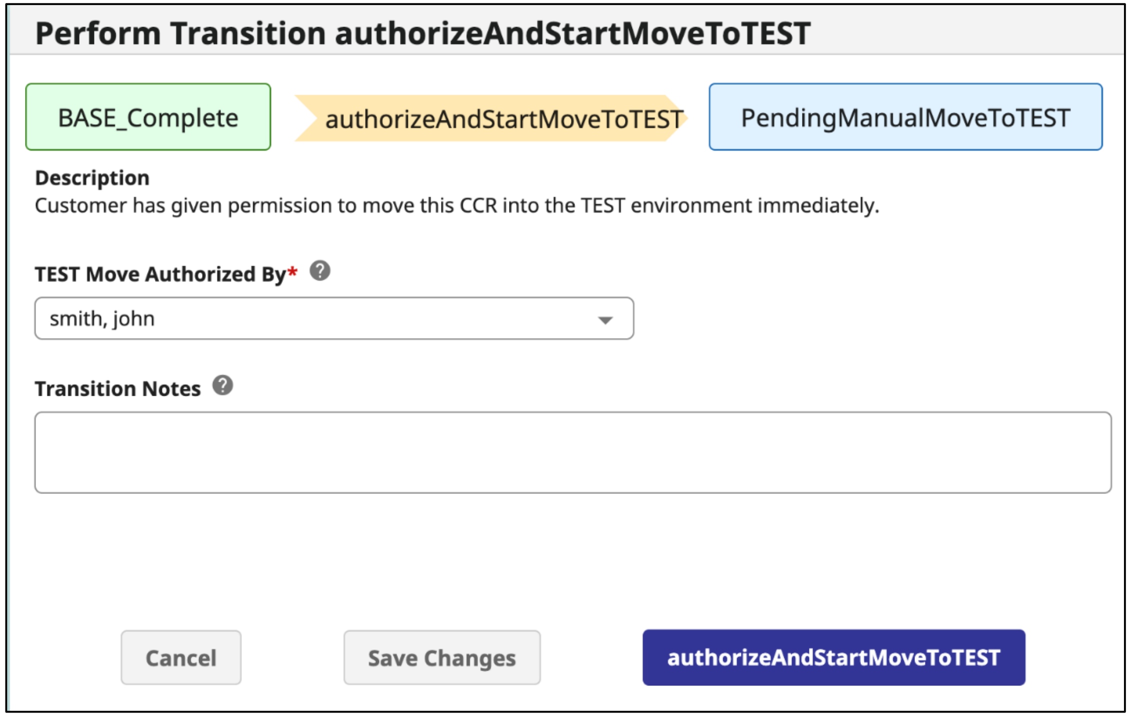
State: BASE_Complete (cont.)

BASEPendingPeerReview BASEComplete

- 3 transition options:
  - markAcceptanceFailed.
    - Returns to In_BASE state.
    - Owner should follow same procedure as if it were a failPeerReview.
  - authorizeFutureMoveToTEST.
    - Rarely used.
    - Should progress changes quickly.
  - authorizeAndStartMoveToTEST.

Transition: authorizeAndStartMoveToTEST

- Only perform when able to immediately follow Implementation Plan!
  - Triggers workflow automations in Tier 1 and Tier 2 CCRs.
- Required field: Test Move Authorized By.
- Optional field: Transition Notes.
  - Specify how received authorization.



Perform Transition authorizeAndStartMoveToTEST

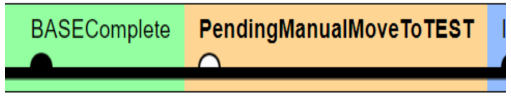
Description
Customer has given permission to move this CCR into the TEST environment immediately.

TEST Move Authorized By* : smith, john

Transition Notes

Destination: Pending_Manual_Move_To_TEST

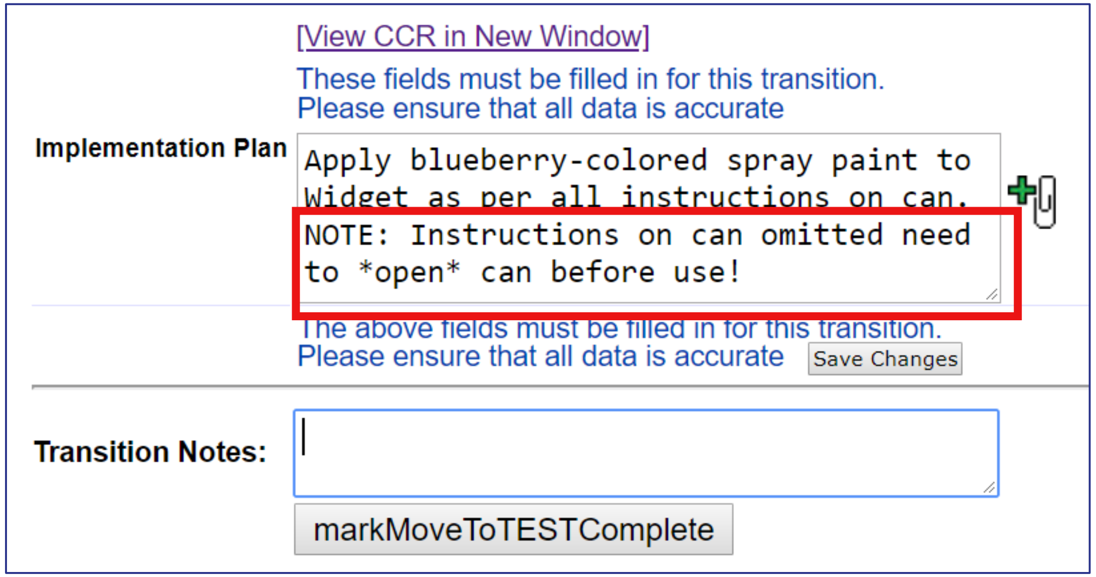
State: Pending_Manual_Move_To_TEST



- Complete all steps according to Implementation Plan.
- Update CCR fields with any surprises or deviation from plan.
- Perform markMoveToTESTComplete transition once Implementation Plan is complete and before starting Testing Plan.

Transition: markMoveToTESTComplete

- Double-check Implementation Plan does not require additional clarification.



[View CCR in New Window]

These fields must be filled in for this transition.
Please ensure that all data is accurate

Implementation Plan

```
Apply blueberry-colored spray paint to
Widget as per all instructions on can.
NOTE: Instructions on can omitted need
to *open* can before use!
```

The above fields must be filled in for this transition.
Please ensure that all data is accurate

Transition Notes:

Destination: In_TEST

Quiz: Authorization

Question:

For a system owned by Today Robot, by default who will be listed in the dropdown for the list of authorizers during authorizeAndStartMoveToTEST?

A. Only TodayRobot project managers.

B. Only InterSystems project managers.

C. Any InterSystems users.

D. Any TodayRobot users.

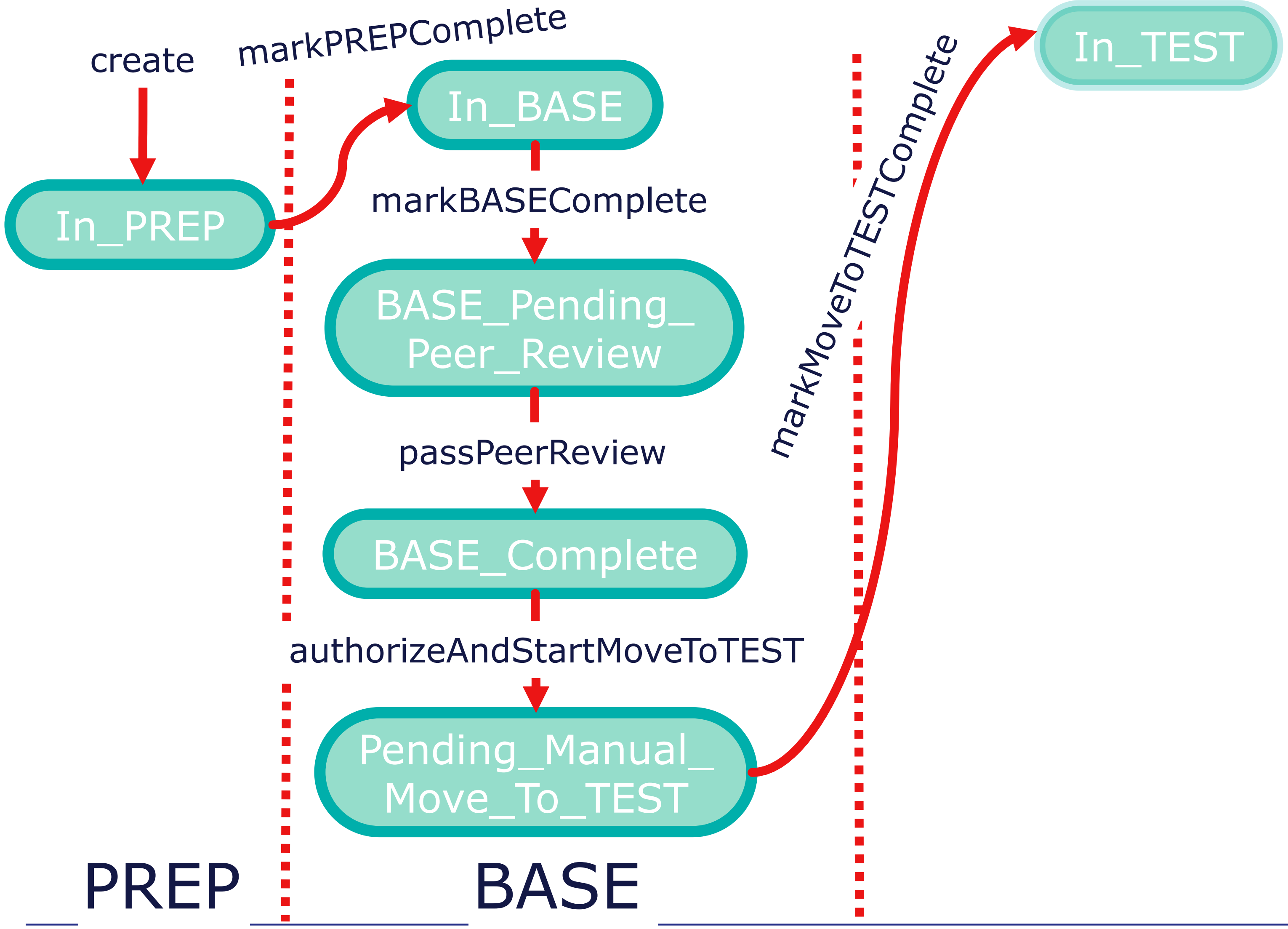
Quiz: Authorization

Answer:

D. Any Today Robot users.

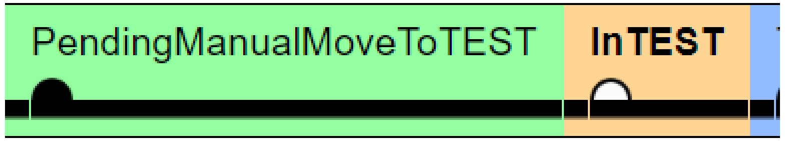
Any user from the organization that owns the system (in this case Today Robot) will appear in the list of possible authorizers. Policy at an organization may restrict who should be specified but that is not enforced. The user specified will receive an email notification.

Reviewing the CCR Journey…



Part 6: CCR TEST Phase

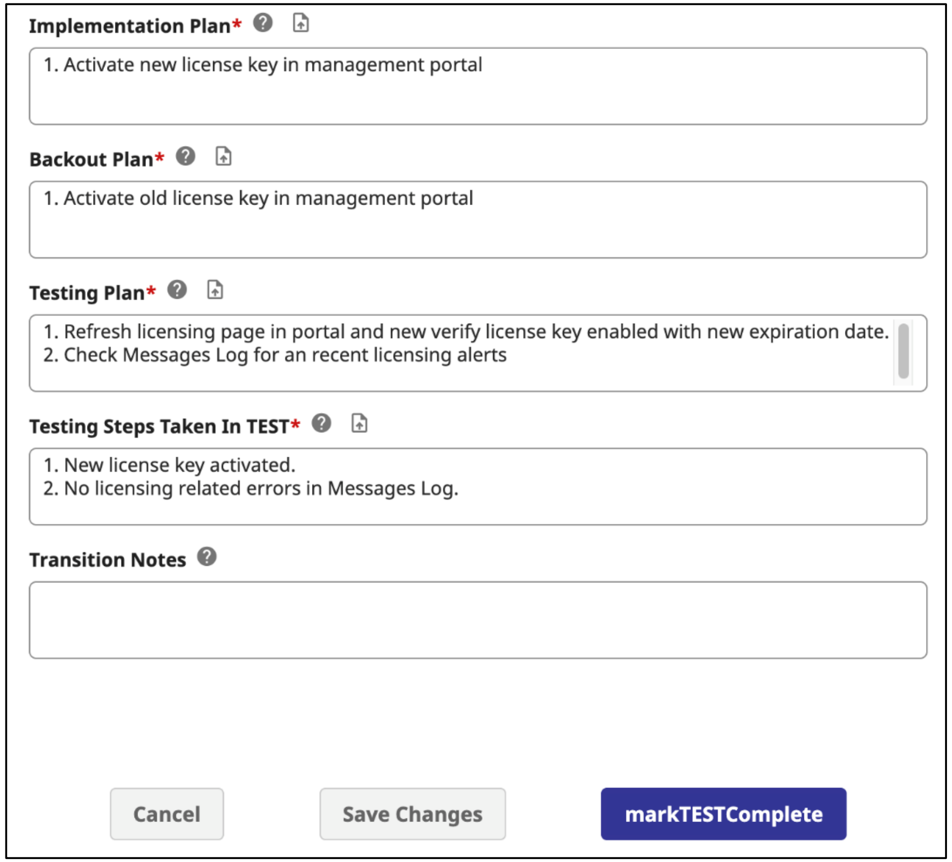
State: In_TEST



- The CCR has been fully implemented in TEST.
- There are only actions for the Owner to perform:
  - Run the Testing Plan in the TEST Environment.
  - Document the results of the testing in Testing Steps Taken in TEST.
    - Remember to use attachments for logs, screenshots, output files, etc; clear evidence of success will make it easier for the Peer Reviewer!
- 2 transition options:
  - markValidationFailed.
  - markTESTComplete.

Transition: markTESTComplete

- Ensure Testing Steps Taken in TEST are complete.
- Ensure no updates required to other fields.
- Finalize transition to proceed.



Implementation Plan*

1. Activate new license key in management portal

Backout Plan*

1. Activate old license key in management portal

Testing Plan*

1. Refresh licensing page in portal and new verify license key enabled with new expiration date.
2. Check Messages Log for an recent licensing alerts

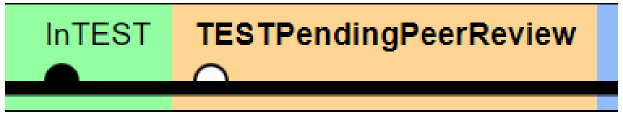
Testing Steps Taken In TEST*

1. New license key activated.
2. No licensing related errors in Messages Log.

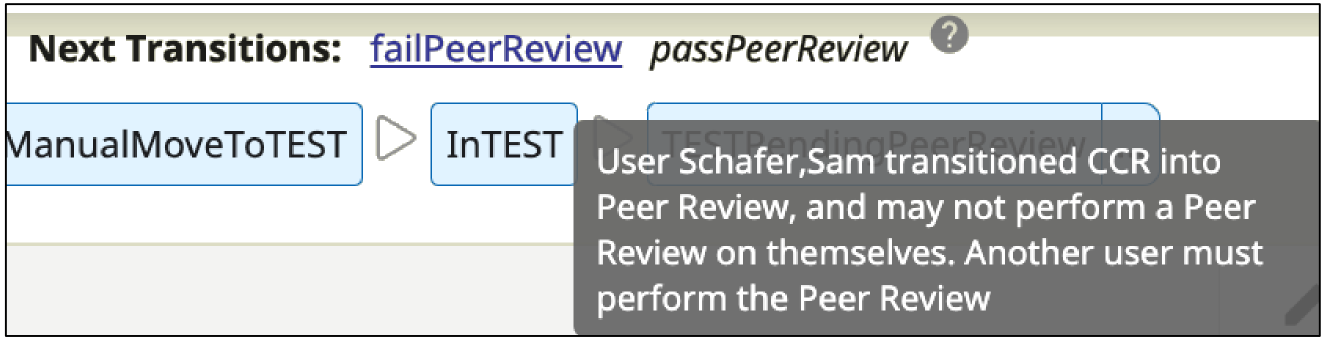
Transition Notes

Destination: TEST_Pending_Peer_Review

State: TEST_Pending_Peer_Review



- CCR automatically reassigned to default Peer Reviewer.
  - Remember, anyone from same organization other than user who performed the markTESTComplete transition can peer review.
- Peer Reviewer Guidelines for TEST:
  - Ensure Testing Steps Taken In TEST field contains sufficient evidence that the Testing Plan was thoroughly followed.
- Same options as BASE_Pending_Peer_Review.
  - failPeerReview.
    - Returns to In_TEST state.
  - passPeerReview.



Next Transitions: failPeerReview *passPeerReview*

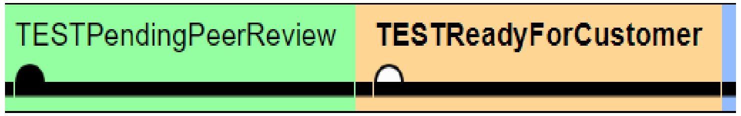
User Schafer,Sam transitioned CCR into Peer Review, and may not perform a Peer Review on themselves. Another user must perform the Peer Review

Transition: passPeerReview

- Review all fields carefully before performing passPeerReview.
- Optionally able to supply transition notes before confirming the passPeerReview transition.

Destination: TEST_Ready_For_Customer

State: TEST_Ready_For_Customer



- Customer has option to examine the change.
- Customer has 3 transition options:
  - markAcceptanceFailed.
    - Returns to In_TEST state.
    - Owner should follow same procedure as if it were a failPeerReview.
  - authorizeFutureMoveToLIVE.
  - authorizeAndStartMoveToLIVE.

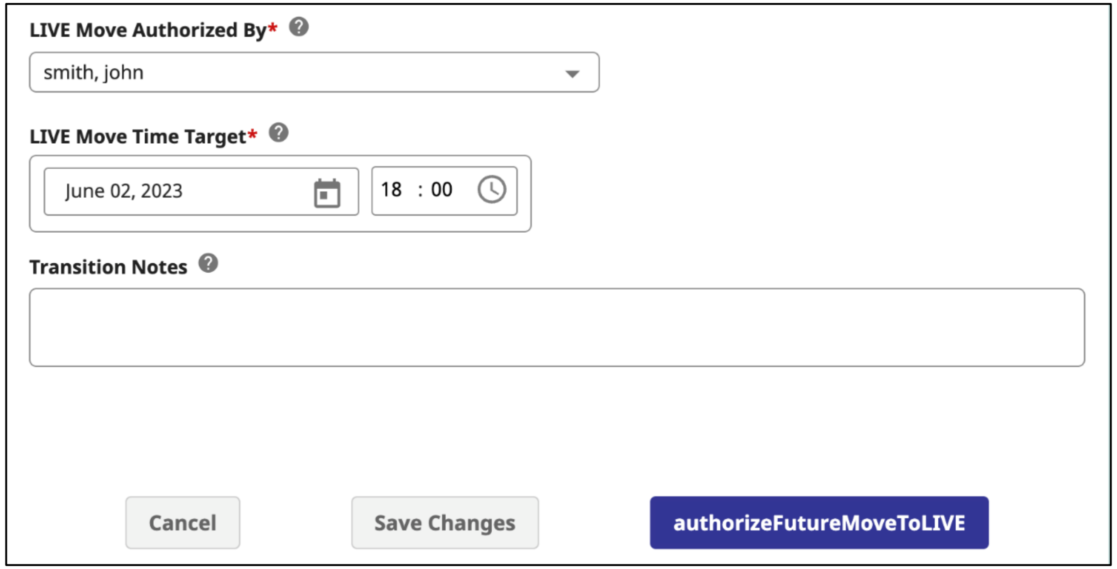
Transition: authorizeFutureMoveToLIVE

- Sometimes change cannot go into LIVE immediately.
- Use authorizeFutureMoveToLIVE to specify authorization received but for a specified date/time.
  - Still requires manual move at date/time!
- Future moves can also be specified for TEST or UAT.

Destination: Pending_Move_To_LIVE

Transition: authorizeFutureMoveToLIVE (cont.)

- Specify LIVE Move Time Target.
  - Can download calendar object.
  - Reminder email sent to owner and owner’s manager (if owner is an InterSystems employee) 1 and 3 days before LIVE Move Time Target.



LIVE Move Authorized By*

smith, john

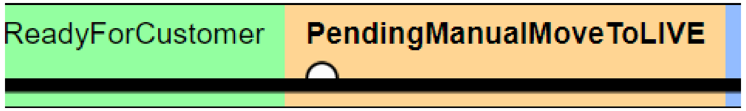
LIVE Move Time Target*

June 02, 2023 | 18 : 00

Transition Notes

Destination: Pending_Move_To_LIVE

State: Pending_Move_To_LIVE


ReadyForCustomer PendingManualMoveToLIVE

- Holding state for CCR that has been authorized to move to LIVE but can’t be moved until a later time.
- Perform startMoveToLive transition when ready.
  - Moves CCR to Pending_Manual_Move_To_LIVE state.

Transition: authorizeAndStartMoveToLIVE

- If a Future move isn’t needed, then use authorizeAndStartMoveToLIVE to move the CCR immediately.
- Bypasses Pending_Move_To_LIVE state.
- Make sure to note who provided authorization for the move to LIVE.
  - TIP: The Transition Notes can used to indicate how and when authorization was communicated.
    - E.g. email, Teams, phone call, in person, etc.

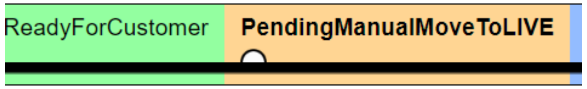
Destination: Pending_Move_To_LIVE

Transition: authorizeAndStartMoveToLIVE (cont.)

- Don’t perform this transition until you are ready to perform the Implementation Plan immediately!

Destination: Pending_Move_To_LIVE

State: Pending_Manual_Move_To_LIVE



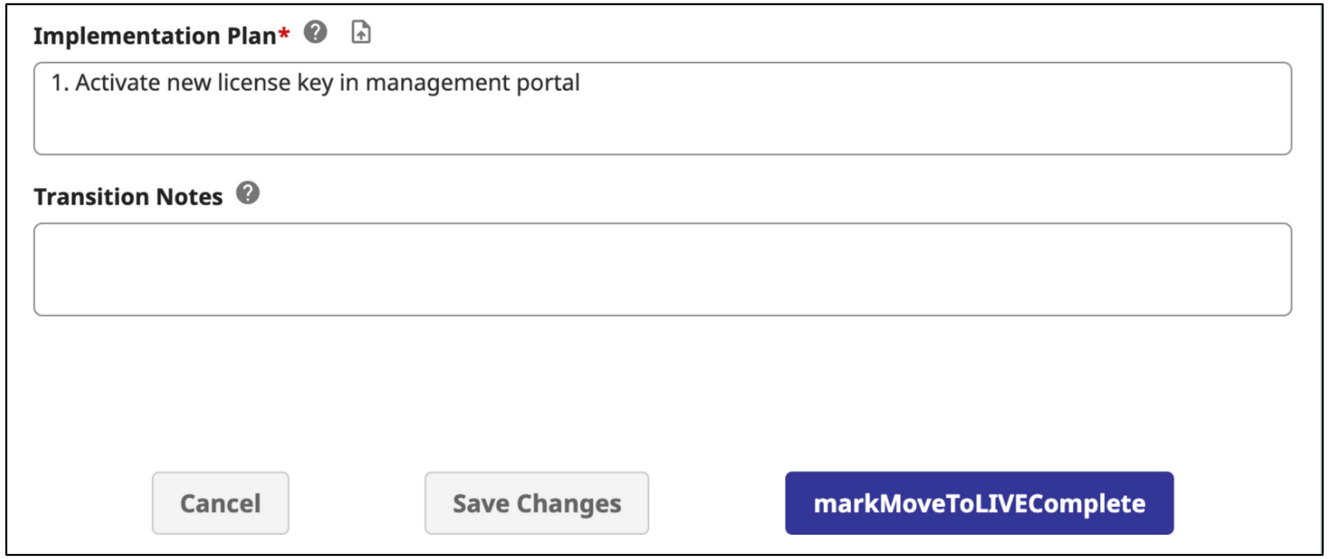
ReadyForCustomer

PendingManualMoveToLIVE

- Complete all steps according to Implementation Plan.
- Note Implementation Plan, Backout Plan, and Testing Plan move forward to the pane corresponding with current Phase.

Transition: markMoveToLIVEComplete

- Verify one last time that Implementation Plan is completely accurate.
- Perform markMoveToLIVEComplete transition once Implementation Plan complete but before Testing Plan started.
  - Moves CCR to LIVE phase.



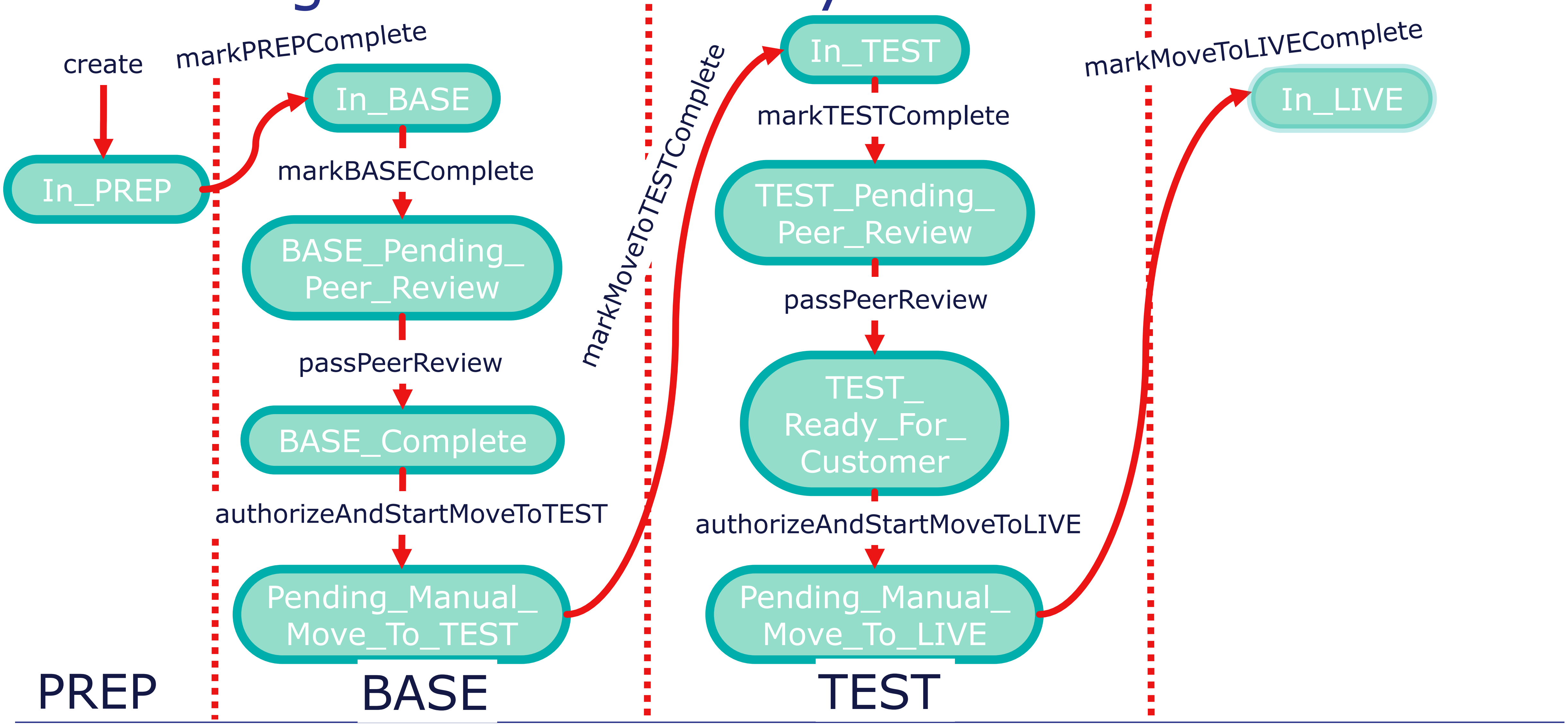
Implementation Plan*

1. Activate new license key in management portal

Transition Notes

Destination: In_LIVE

Reviewing the CCR Journey…



Quiz: TEST_Ready_For_Customer

Question:
Assuming a BASE > TEST > LIVE workflow, which of the following is the purpose of TEST_Ready_For_Customer?

A. Holding state while the implementor awaits approval to move change to TEST.
B. Holding state while the implementor awaits approval to move change to LIVE.
C. Holding state while the implementor tests the change in TEST.
D. Holding state while the implementor makes the change in the customer’s TEST environment.

Quiz: TEST_Ready_For_Customer (cont.)

Answer:

B. Holding state while the implementor awaits approval to move change to LIVE.

Approval to move to TEST is obtained during the BASE_Complete state. Implementation in TEST occurs during Pending_Manual_Move_To_TEST and testing for TEST environment takes places in the In_TEST state.

Quiz: markAcceptanceFailed

Question:

Using the workflow diagram, which of the following is true about markAcceptanceFailed when performed in the TEST_Ready_For_Customer state?

A. The CCR moves to the In_TEST state.

B. All changes must be backed out of all environments.

C. The CCR must be cancelled.

D. The CCR moves to the In_BASE state.

Quiz: markAcceptanceFailed (cont.)

Answer:

A. The CCR moves to the In_TEST state.

Nothing must be backed out. markAcceptanceFailed means the customer does not approve of the change and additional work is necessary. The implementer may decide whether to perform markValidationFailed and back out changes, or progress an additional catch-up CCR. It is usually preferable to progress a catch-up CCR (see ICC330 and ICC450).

Quiz: Scheduling a Move

Question:
If the customer wants a particular CCR to be moved into the LIVE environment at 6:00 PM on Friday and it is currently Tuesday, what is the right way to manage this timing in CCR?

A. Leave the CCR in TEST_Ready_For_Customer until Friday at 6:00 PM.
B. Use the authorizeFutureMoveToLIVE transition and indicate a LIVE Move Time Target of Friday at 6:00 PM.
C. Use the authorizeAndStartMoveToLIVE transition to record authorization immediately, but don’t run the Implementation Plan until 6:00 PM on Friday.

Quiz: Scheduling a Move (cont.)

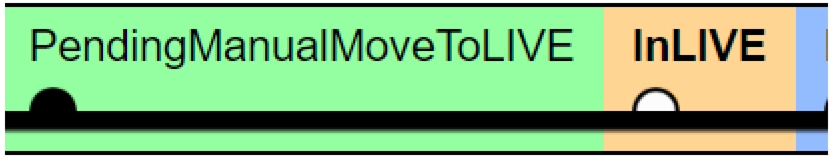
Answer:

B. Use the authorizeFutureMoveToLIVE transition and indicate a LIVE Move Time Target of Friday at 6:00pm.

A risks missing the move window, and C wrongly indicates the move started early.

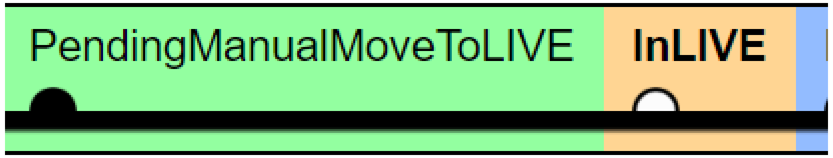
Part 7: CCR LIVE Phase

State: In_LIVE



- All that remains is executing the Testing Plan in LIVE.
  - Do quickly, if possible, to ensure the functionality is complete and as expected.
  - There may be some parts of the Testing Plan which should not be run in LIVE.
    - Make sure to look for any specific instructions for LIVE and follow them carefully.
    - E.g.: “LIVE: Do not create test patient data”.

State: In_LIVE (cont.)



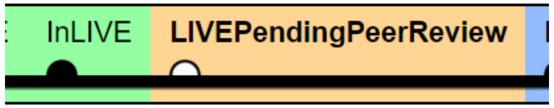
- Collect documentation and evidence of the testing to attach to the CCR.
- 2 transition options:
  - markValidationFailed.
    - Moves to Pending_Revert state as previously discussed.
    - Hopefully not necessary at this point!
  - markLIVEComplete.

Transition: markLIVEComplete

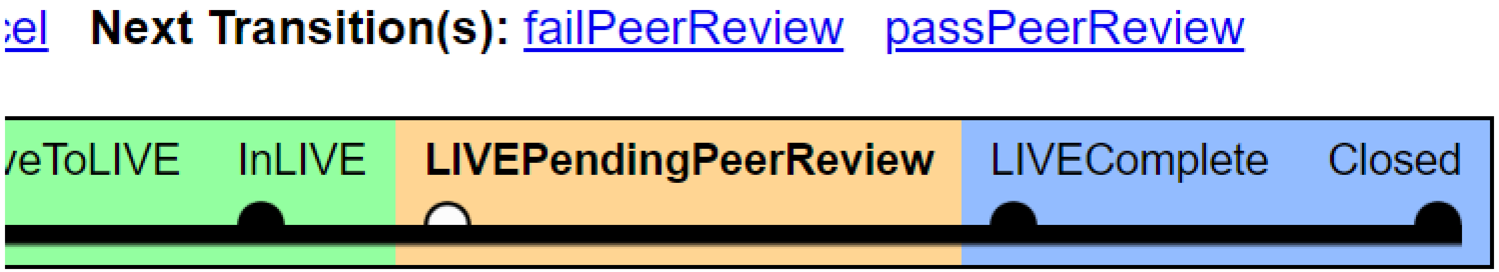
- Record all testing steps and evidence in Testing Steps Taken in LIVE field.
- Take a last opportunity to review all three plan fields; ensure nothing is missing!
  - CCRs can be ‘cloned’ to repeat the same work again.
    - This can save a lot of time!
  - CCRs will be referenced again in the future.
    - Your future self will thank you for leaving a well documented CCR!!

Destination: LIVE_Pending_Peer_Review

State: LIVE_Pending_Peer_Review



- Final peer review!
- Peer reviewer guidelines for LIVE:
  - Ensure Testing Steps Taken In LIVE field contains sufficient evidence of the Testing Plan being thoroughly followed.

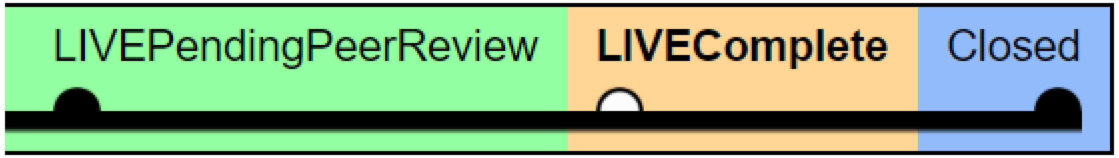

:el **Next Transition(s):** failPeerReview passPeerReview

Transitions: passPeerReview

- Review all fields carefully before performing passPeerReview.
- Optionally able to supply transition notes before confirming the passPeerReview transition.

Destination: LIVE_Complete

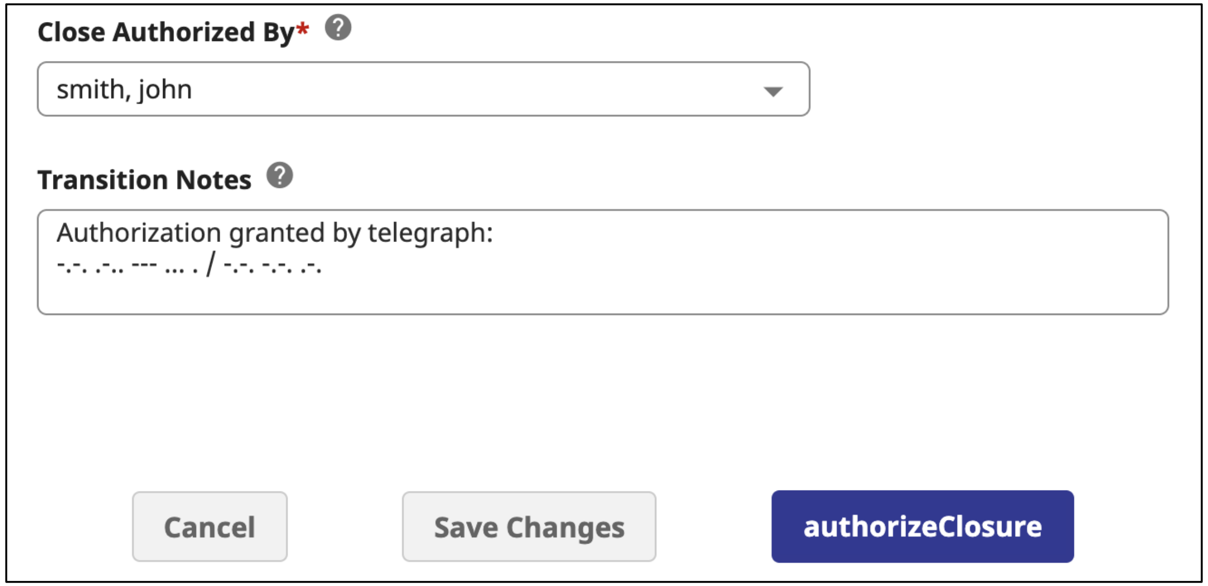
State: LIVE_Complete



- The last step: Get authorization from the System’s owner (typically the customer) to close the CCR.
  - Everything else is complete.

Transition: authorizeClosure

- Customer confirms functionality in LIVE and gives permission to close the CCR.
- Remember Transition Notes can be used to indicate how authorization was given.



Close Authorized By*

smith, john

Transition Notes

Authorization granted by telegraph:
-.-. .-.. --- ... . / -.-. -.-. .-.

Destination: Closed

State: Closed

- CCR is Closed and no longer active.
- Details remain for future reference.
- Edit icons still active in case corrections need to be made.
- Clone transition can be used to easily replicate the documentation if the change is required again.
  - This could be for the same System or for a different System.

Quiz: Implementing Changes

Question:

For a BASE > TEST > LIVE workflow, in which THREE of the following states should you implement your change? Use the workflow diagram for help.

A. In_PREP.

B. In_BASE.

C. In_TEST.

D. In_LIVE.

E. Pending_Manual_Move_To_TEST.

F. Pending_Manual_Move_To_LIVE.

Quiz: Implementing Changes (cont.)

Answer:

B. In_BASE.
E. Pending_Manual_Move_To_TEST.
F. Pending_Manual_Move_To_LIVE.

Make and test changes in BASE while in the In_BASE state. Since making a change in BASE is often an iterative process of making a change and testing, the workflow does not require 2 states. The rest of the workflow has different states for implementing and testing the change. Beyond BASE, only make changes to an environment in its corresponding Pending_Manual_Move_To_XXXX state.

Quiz: Testing Changes

Question:

For a BASE > TEST > LIVE workflow, in which THREE of the following states should you test the change and document the results of testing? Use the workflow diagram for help.

A. In_BASE.

B. In_TEST.

C. In_LIVE.

D. BASE_Complete.

E. Pending_Manual_Move_To_LIVE.

F. LIVE_Complete.

Quiz: Testing Changes (cont.)

Answer:

A. In_BASE.

B. In_TEST.

C. In_LIVE.

Test the change in every environment and before peer review so that the peer reviewer can verify the documentation.

Quiz: Transition Notes

Question:

Transition notes are always optional. True or False?

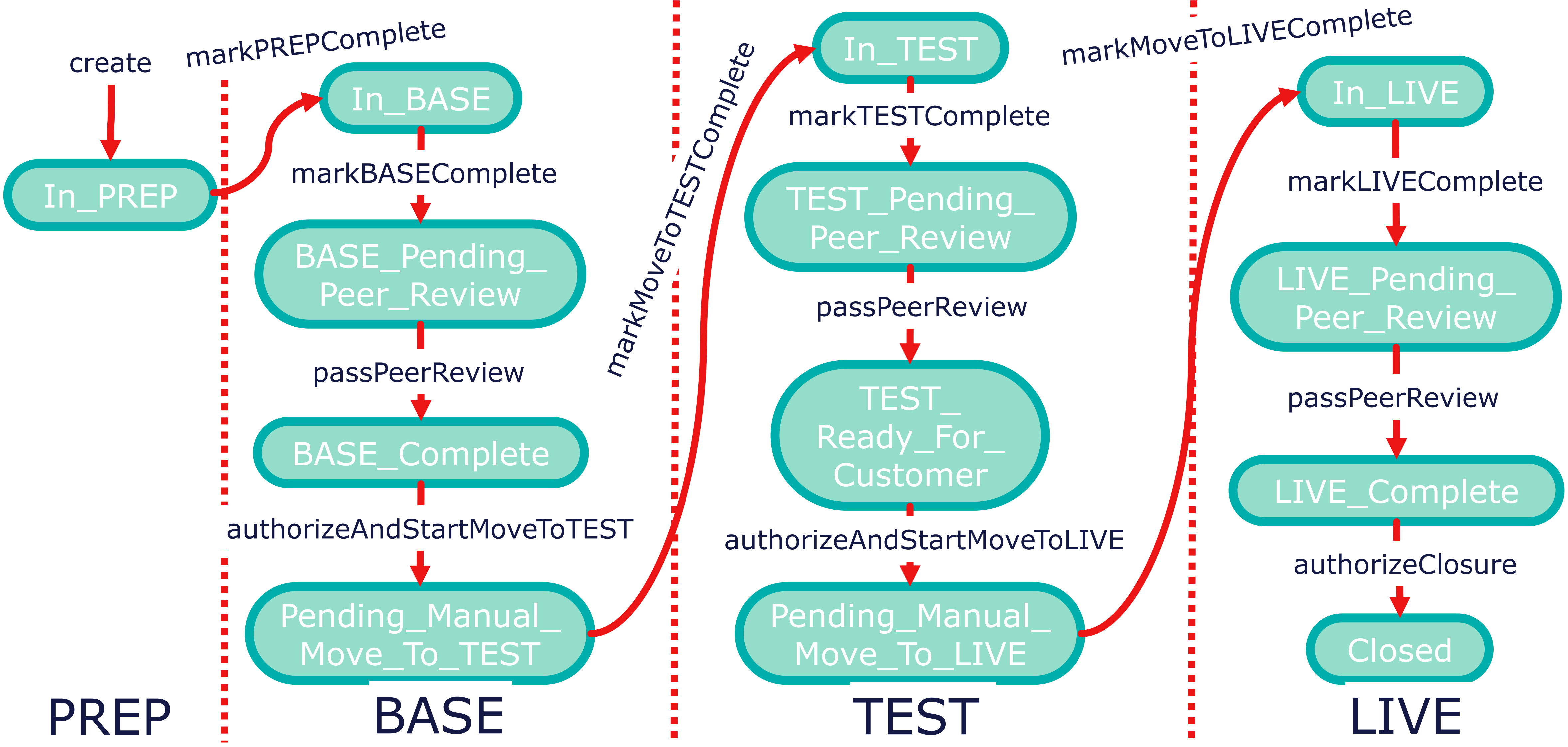
Answer:

False. Transition Notes are only required if the transition sends the CCR ‘backwards.’

Exercise ICC310-4

Progressing CCR to Closed

Reviewing the CCR Journey…



create

In_PREP

PREP

markPREPComplete

In_BASE

markBASEComplete

BASE_Pending_ Peer_Review

passPeerReview

BASE_Complete

authorizeAndStartMoveToTEST

Pending_Manual_ Move_To_TEST

BASE

markMoveToTESTComplete

In_TEST

markTESTComplete

TEST_Pending_ Peer_Review

passPeerReview

TEST_ Ready_For_ Customer

authorizeAndStartMoveToLIVE

Pending_Manual_ Move_To_LIVE

TEST

markMoveToLIVEComplete

In_LIVE

markLIVEComplete

LIVE_Pending_ Peer_Review

passPeerReview

LIVE_Complete

authorizeClosure

Closed

LIVE

Summary

- What are the key points for this module?

ICC 320: CCR Tier 0 – Configuring System Details

InterSystems

Creative data technology

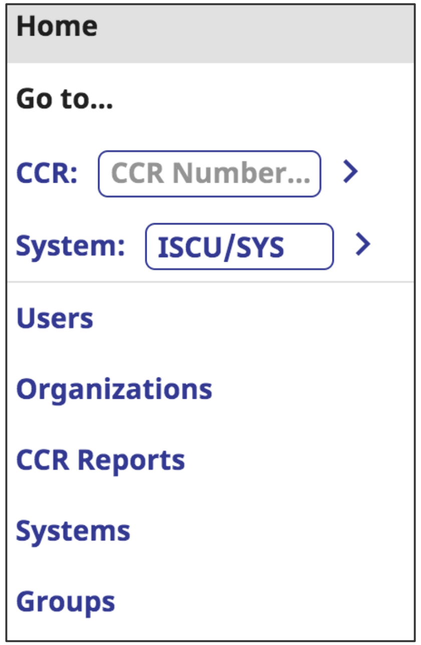
Objectives

- Identify the use for the various settings and controls on the System Details page.
- Demonstrate how maintaining Environment Details for a System saves time and effort.
- Show how to control the workflow for CCRs of a given System.
- Explain what benefits can be achieved by assigning primary and secondary architects within CCR.

Part 1: General System Settings

Navigation to System Details

- To navigate to System Details in Angular UI, either:
  - Enter SiteCode/SysCode for System field in side navigation menu.
  - Go to Systems and use autocomplete field on System Details page.



Home

Go to...

CCR: CCR Number...

System: ISCU/SYS

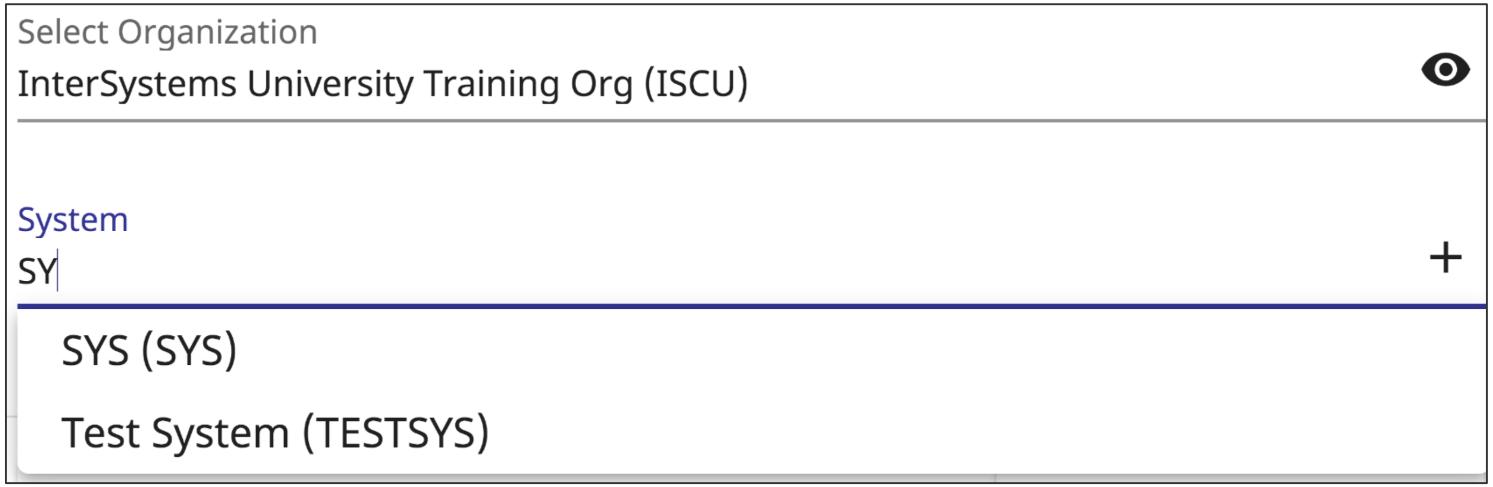
Users

Organizations

CCR Reports

Systems

Groups



Select Organization
InterSystems University Training Org (ISCU)

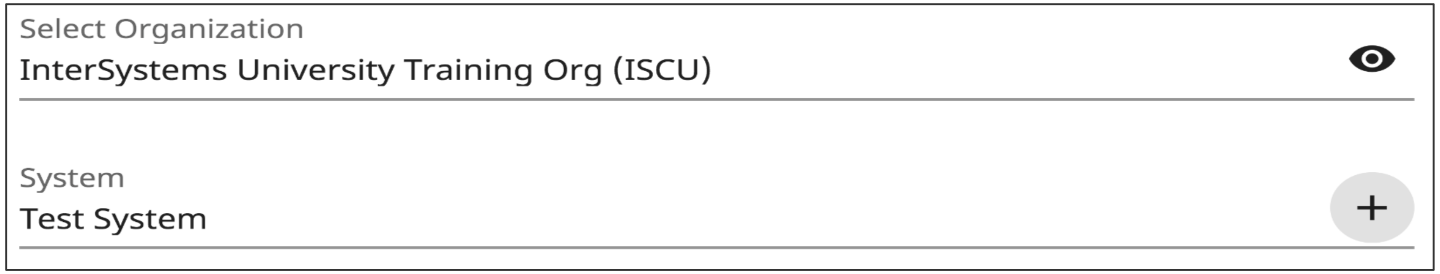
System
SY

SYS (SYS)

Test System (TESTSYS)

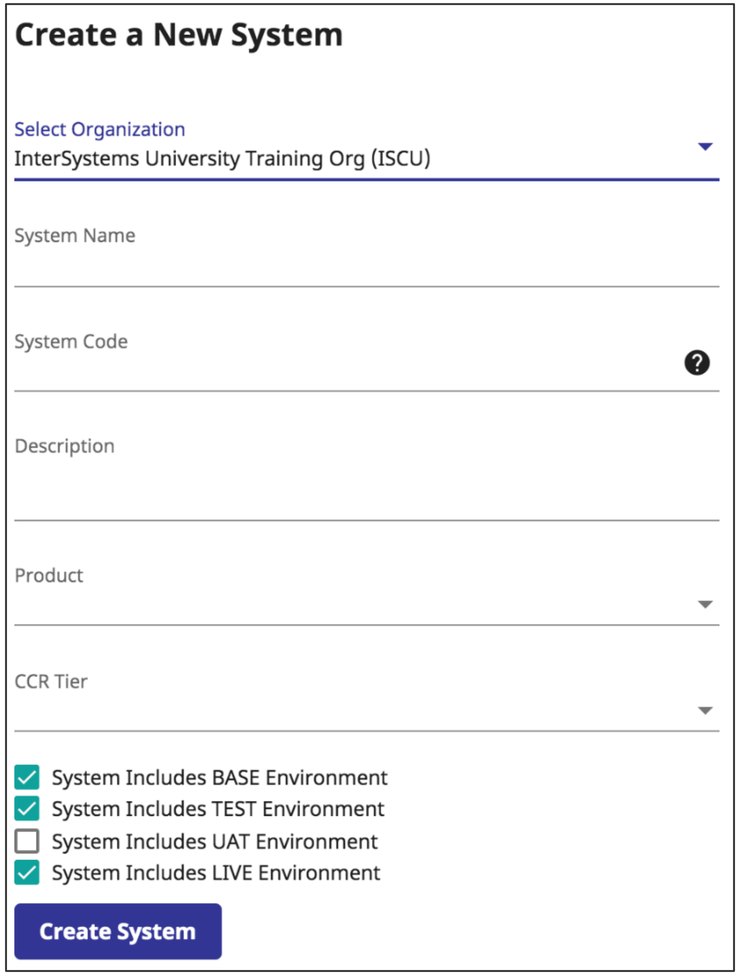
Create New System

- Go to Systems page.
- Select the organization from the dropdown.
  - Contact CCR development team if Organization not listed.



Select Organization
InterSystems University Training Org (ISCU)

System
Test System



Create a New System

Select Organization
InterSystems University Training Org (ISCU)

System Name

System Code

Description

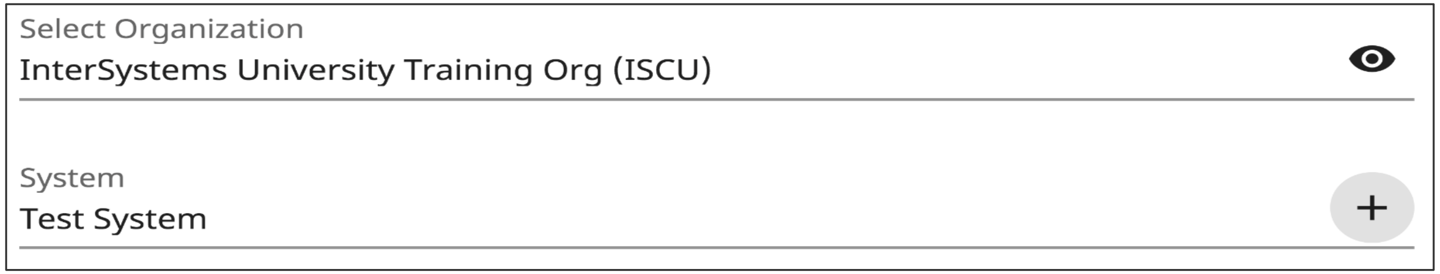
Product

CCR Tier

☑ System Includes BASE Environment
☑ System Includes TEST Environment
□ System Includes UAT Environment
☑ System Includes LIVE Environment

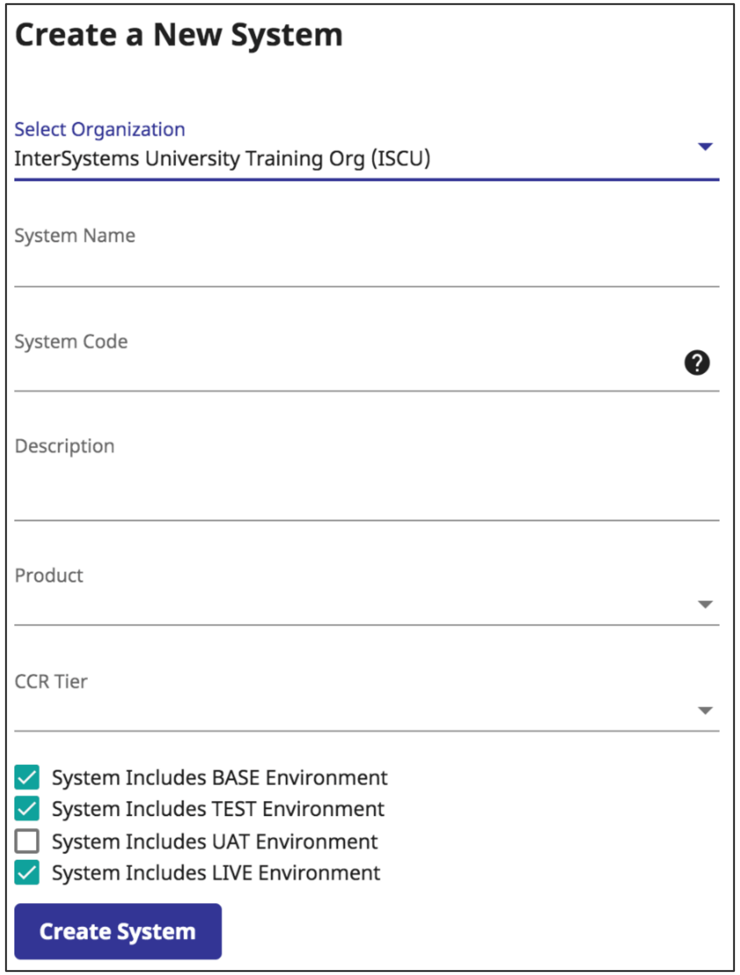
Create New System (cont.)

- Click + in System Field.
- Complete form and click Create System.
  - Cannot change System Code later.



Select Organization
InterSystems University Training Org (ISCU)

System
Test System



Create a New System

Select Organization
InterSystems University Training Org (ISCU)

System Name

System Code

Description

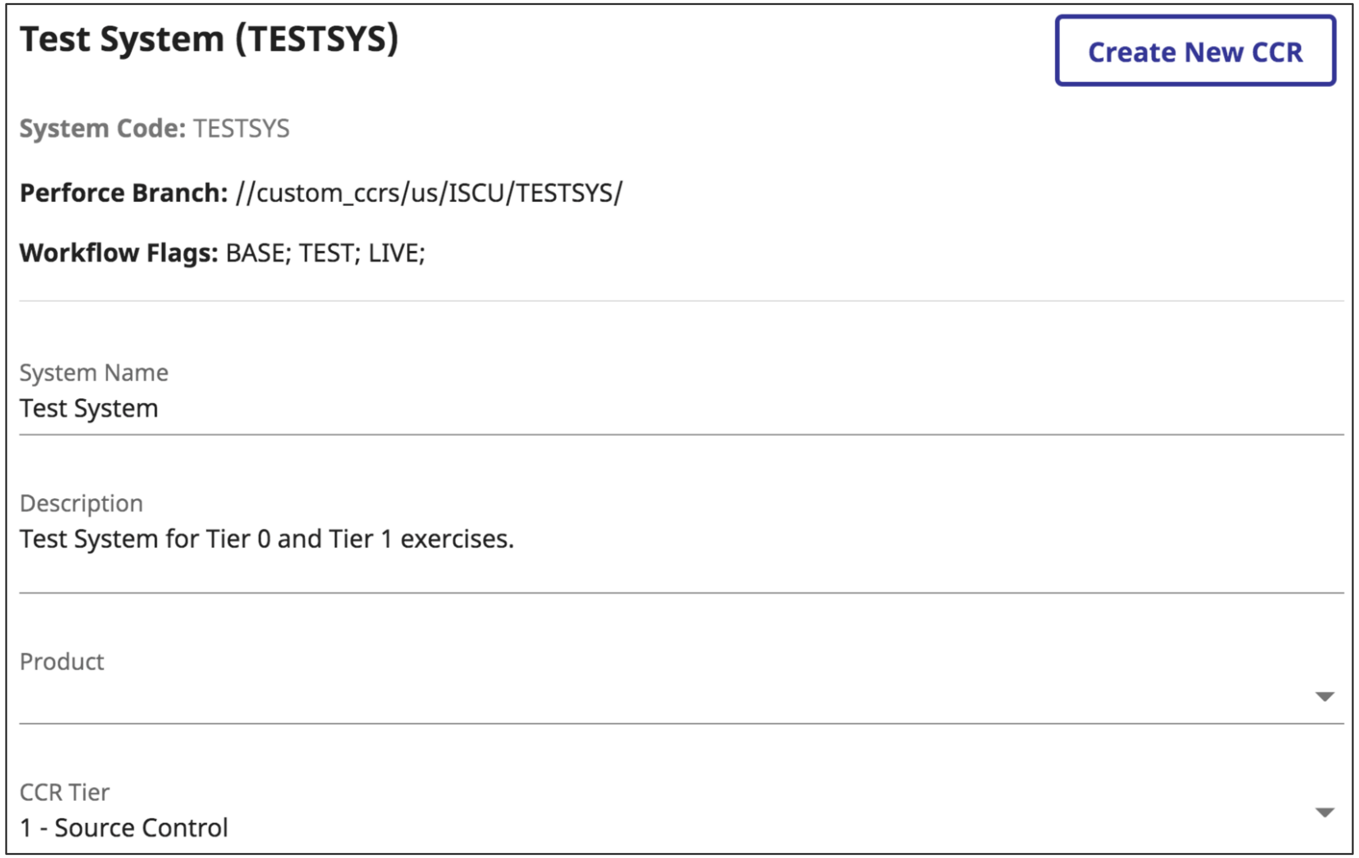
Product

CCR Tier

☑ System Includes BASE Environment
☑ System Includes TEST Environment
□ System Includes UAT Environment
☑ System Includes LIVE Environment

General System Settings

- General Details.
  - For reference only.
- System Tier Level.
  - Controls parts of workflow.



Test System (TESTSYS)

System Code: TESTSYS

Perforce Branch: //custom_ccrs/us/ISCU/TESTSYS/

Workflow Flags: BASE; TEST; LIVE;

System Name
Test System

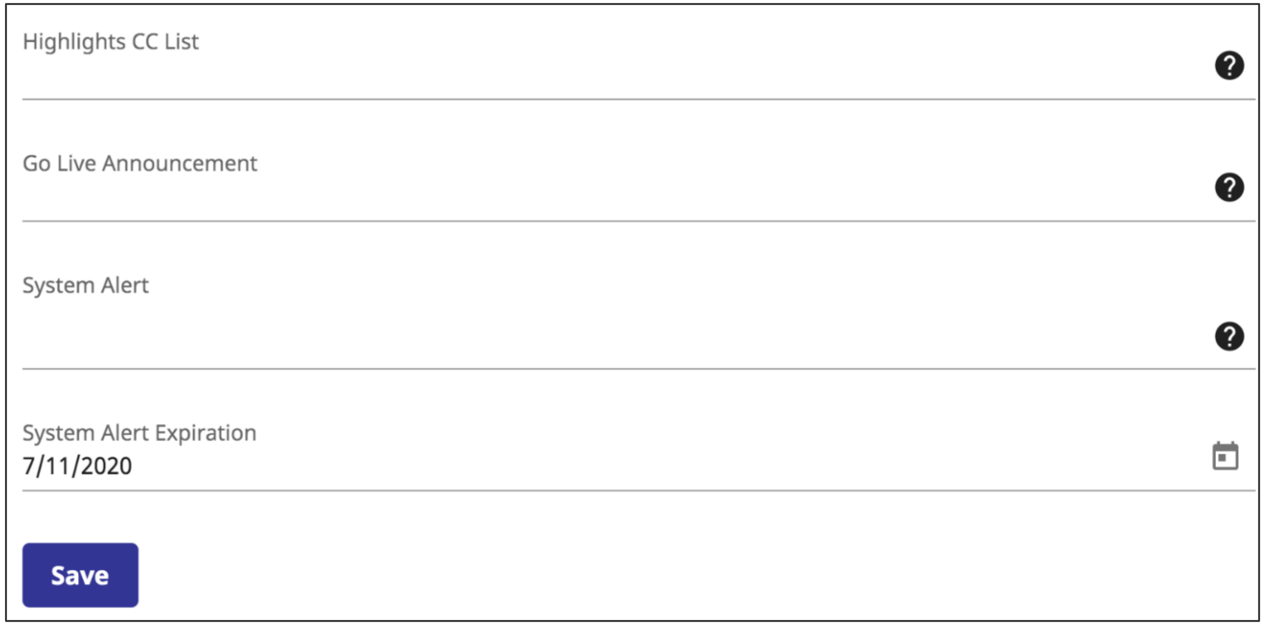
Description
Test System for Tier 0 and Tier 1 exercises.

Product

CCR Tier
1 - Source Control

General System Settings (cont.)

- Notification Recipients.
  - Highlights = Some transitions.
  - Go Live = markLIVEComplete
- System Alerts.
  - Shown on every CCR for System.



Highlights CC List

Go Live Announcement

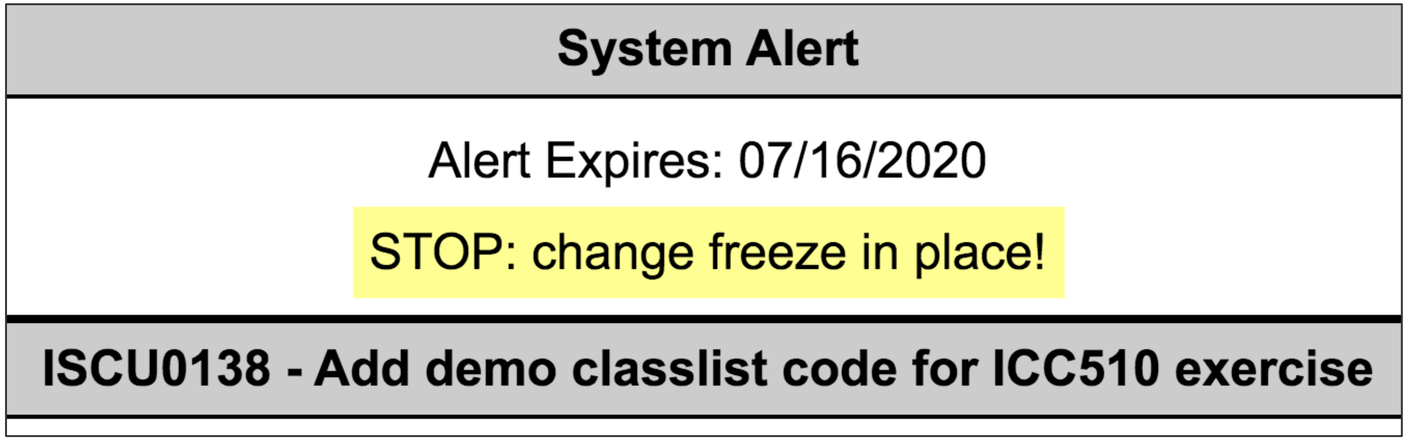
System Alert

System Alert Expiration
7/11/2020

Creating a System Alert

- Useful to indicate events such as change freeze.
- Shown on top of every CCR for System.

1. Edit System Alert field.
2. Add System Alert Expiration date beyond today.
3. Click Save.



System Alert

Alert Expires: 07/16/2020

STOP: change freeze in place!

ISCU0138 - Add demo classlist code for ICC510 exercise

Quiz: General System Settings

Question:

What does the CCR Tier on the System Details page indicate? Select all that apply.

A. The highest CCR Tier used in the System.

B. The minimum CCR Tier for all CCRs in the System.

C. The default CCR Tier for all CCRs created.

D. The CCR Tier of all CCRs in the System.

Quiz: General System Settings (cont.)

Answer:

A. The highest CCR Tier used in the system.

C. The default CCR Tier for all CCRs created.

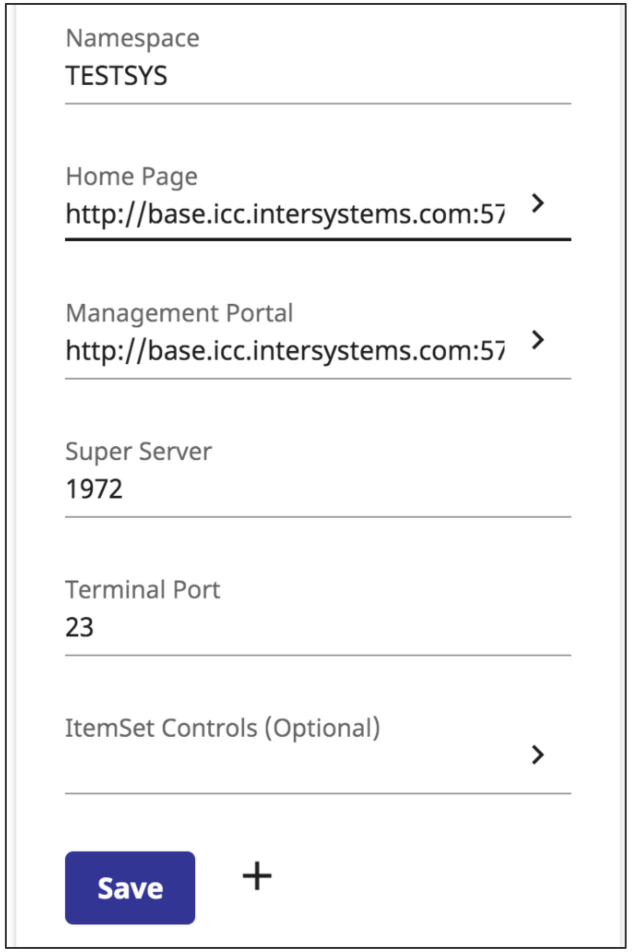
No CCR can be a higher Tier than CCR Tier on System Details.

CCRs can be modified to a lower Tier than CCR Tier on System Details.

Part 2: Environment Details

Environment Details

- Settings for environment available by expanding environment on right.
  - Namespace.
  - Homepage Link.
  - System Management Portal Link.
  - Super Server Port.
  - Terminal Port.
  - ItemSet Controls Customization.
  - Control to create secondary environment.



Namespace
TESTSYS

Home Page
http://base.icc.intersystems.com:57

Management Portal
http://base.icc.intersystems.com:57

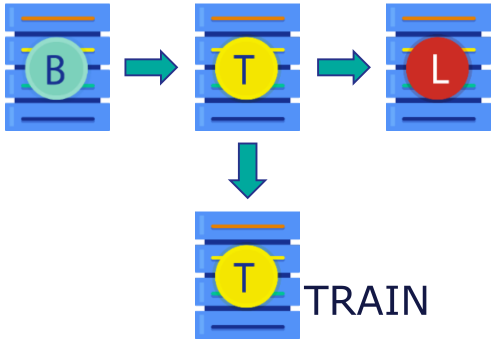
Super Server
1972

Terminal Port
23

ItemSet Controls (Optional)

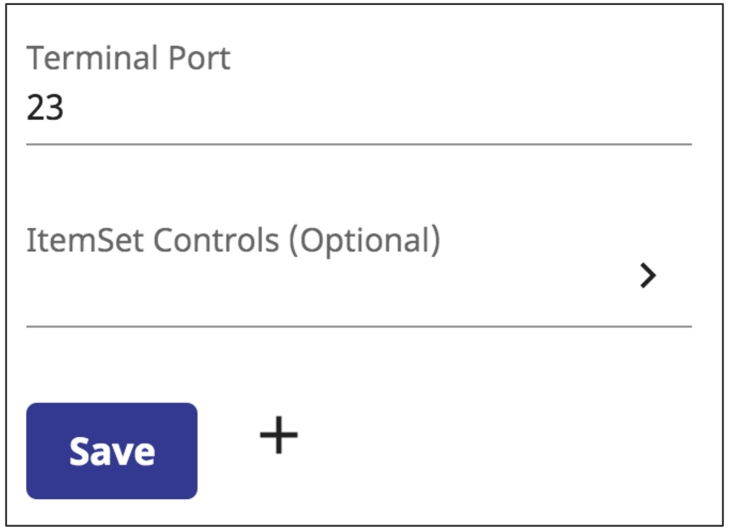
Secondary Environments (aka Peer Environments)

- Allows multiple namespaces to get identical source from same branch in Perforce.
  - Example: multiple Edge Gateways for HealthShare or training instance.
- Parallel environment to BASE, TEST, UAT or LIVE.
  - Deploy to TEST secondary environments when deploy to TEST.
- Require ItemSet flag ensures environments receives all updates to Primary.



Configure Secondary Environment

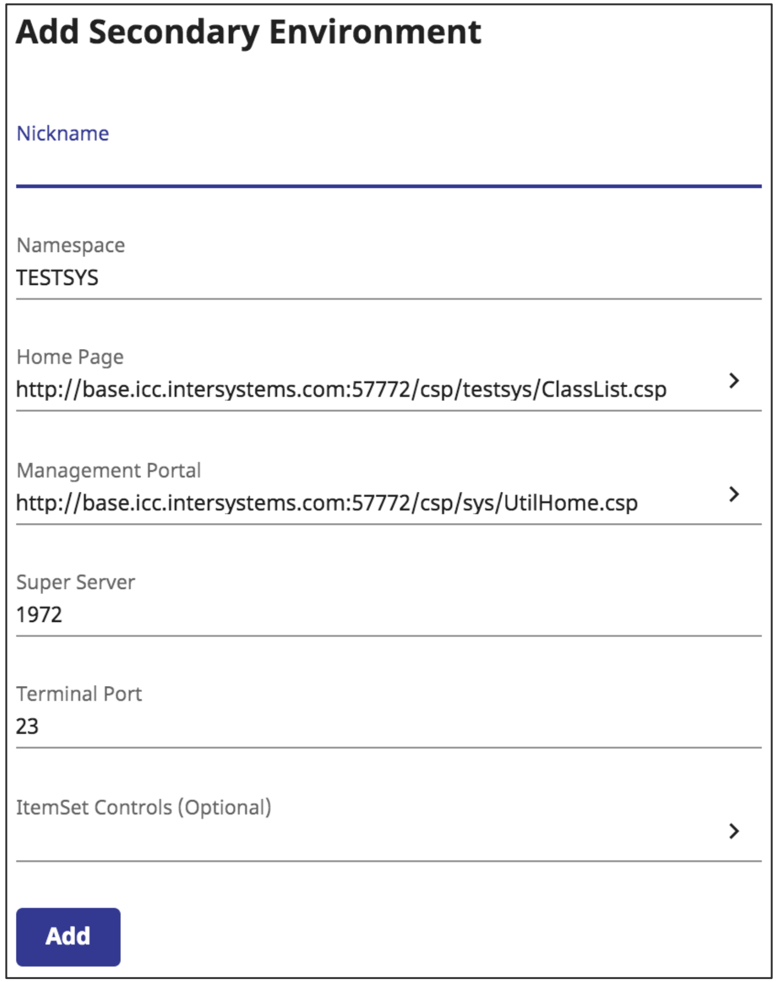
1. Expand appropriate primary environment.
2. Click + button next to Save.
3. Complete form.
   - Nickname shown for ItemSet as [deploy to nickname].



Terminal Port
23

ItemSet Controls (Optional)

Configure Secondary Environment (screenshot)



Add Secondary Environment

Nickname

Namespace
TESTSYS

Home Page
http://base.icc.intersystems.com:57772/csp/testsys/ClassList.csp

Management Portal
http://base.icc.intersystems.com:57772/csp/sys/UtilHome.csp

Super Server
1972

Terminal Port
23

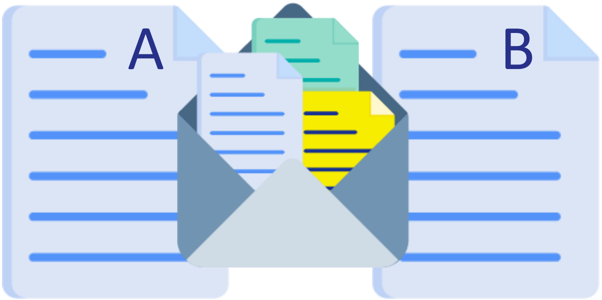
ItemSet Controls (Optional)

Secondary Environment Require ItemSet

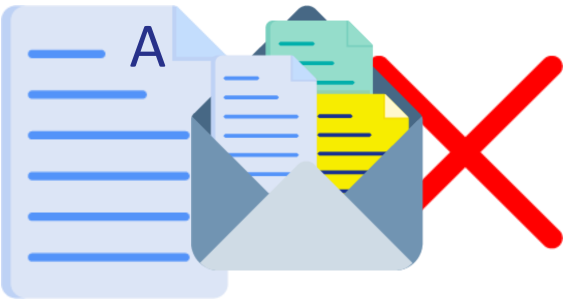
- Be very careful if Require ItemSet is cleared.
- Example of what could go wrong with Require ItemSet false (cleared):
    - Tier 1 CCR adds class definition B and edits preexisting class A to be dependent on B.
        - User accidentally does not deploy class to secondary environment because Require ItemSets is false.
    - Later, Tier 1 CCR edits class definition A (unrelated to previous change).
        - Can’t deploy this change to secondary environment because class A still dependent on class B.

Secondary Environment Require ItemSet (cont.)

Primary environment

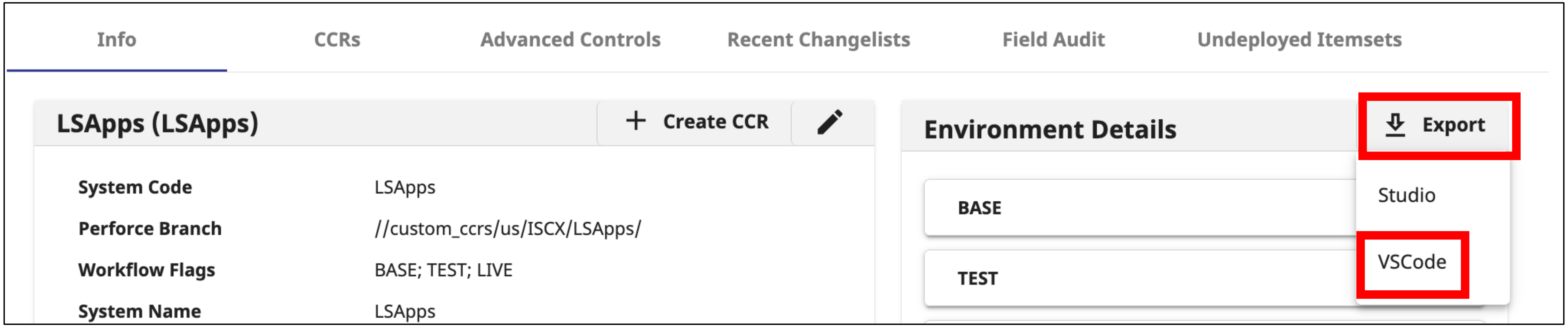


Secondary environment



Download VS Code Workspace

- Can download VS Code Workspace from Systems page.
  - Contains connection details for environments defined in System.
- Menu > Systems > Choose System > Export > VSCode.
  - Double click file to open in VS Code.



LSApps (LSApps)

System Code LSApps

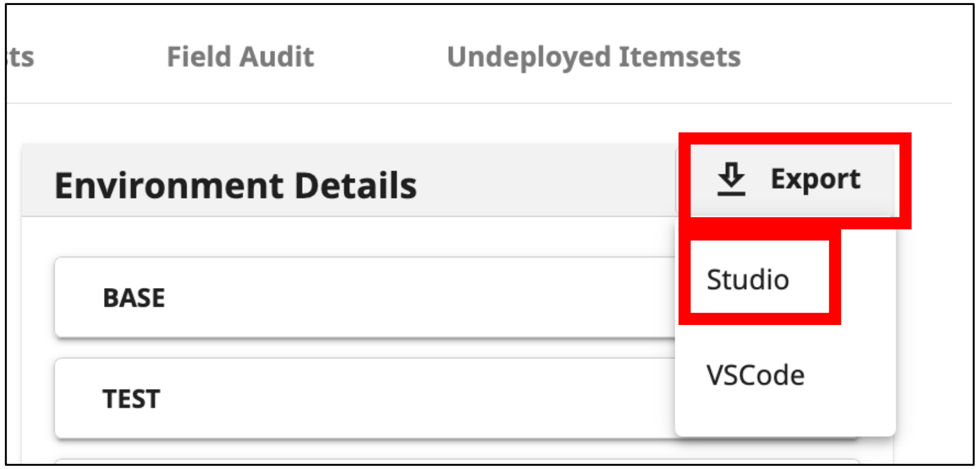
Perforce Branch //custom_ccrs/us/ISCX/LSApps/

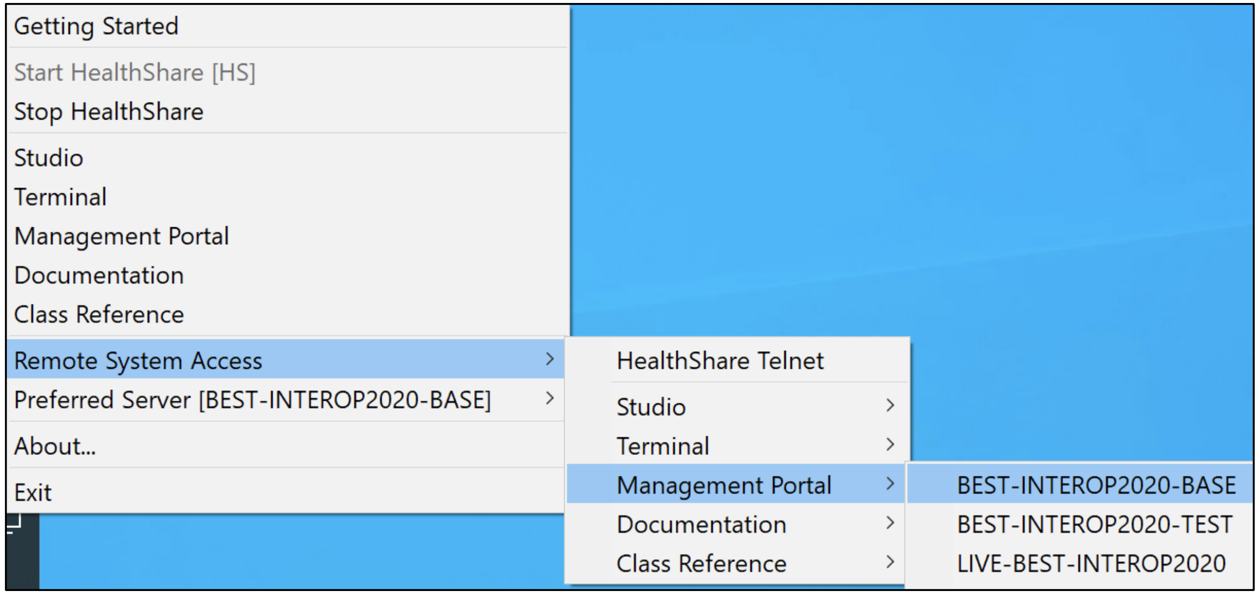
Workflow Flags BASE; TEST; LIVE

System Name LSApps

Windows: Configure Launcher

- Launcher provides remote system access to environments.
- Menu > Systems > Choose System > Export > Studio.
  - Run downloaded registry file.
    - Administrator credentials required.





Quiz: Environment Details

Question:

On a Windows machine, the best way to open a terminal session connected to the BASE environment is through the System Details page. True or False?

Answer:

False.

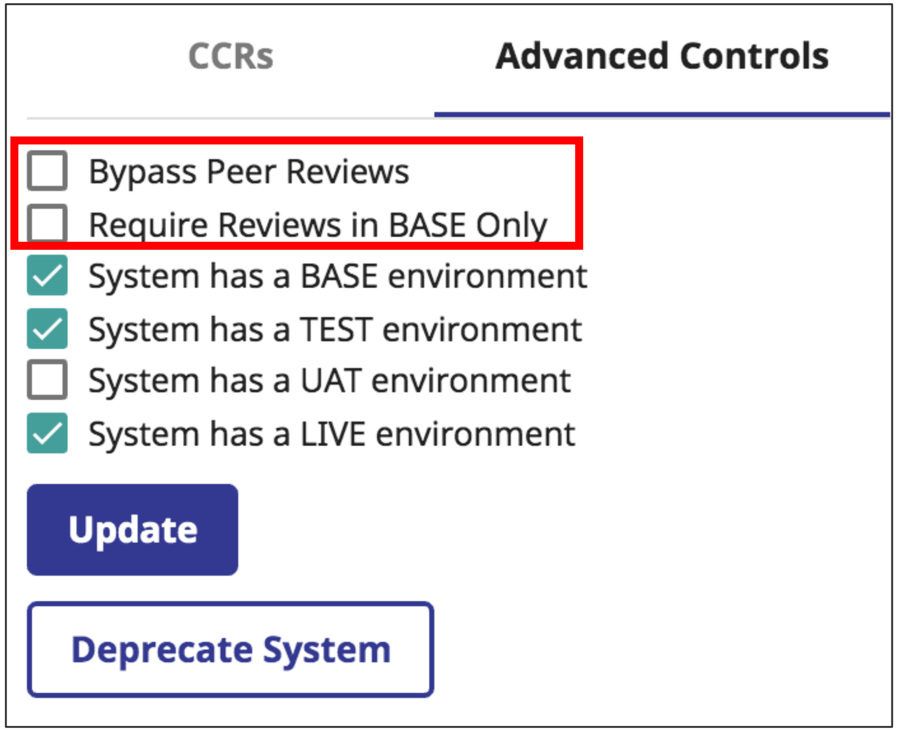
Depending on browser, it may not be possible to open a terminal session through the System Details.

Best to use Caché Server Manager.

Part 3: Advanced Controls

Peer Review System Workflow Options

- Modify the System's peer review workflow under Advanced Controls:
- Bypass peer reviews.
  - Skip all peer review states.
- Peer review BASE Only selected.
  - BASE peer reviews take place.
  - All others are skipped (TEST/UAT/LIVE).
- Neither option selected (default).
  - Each environment will include a peer review.

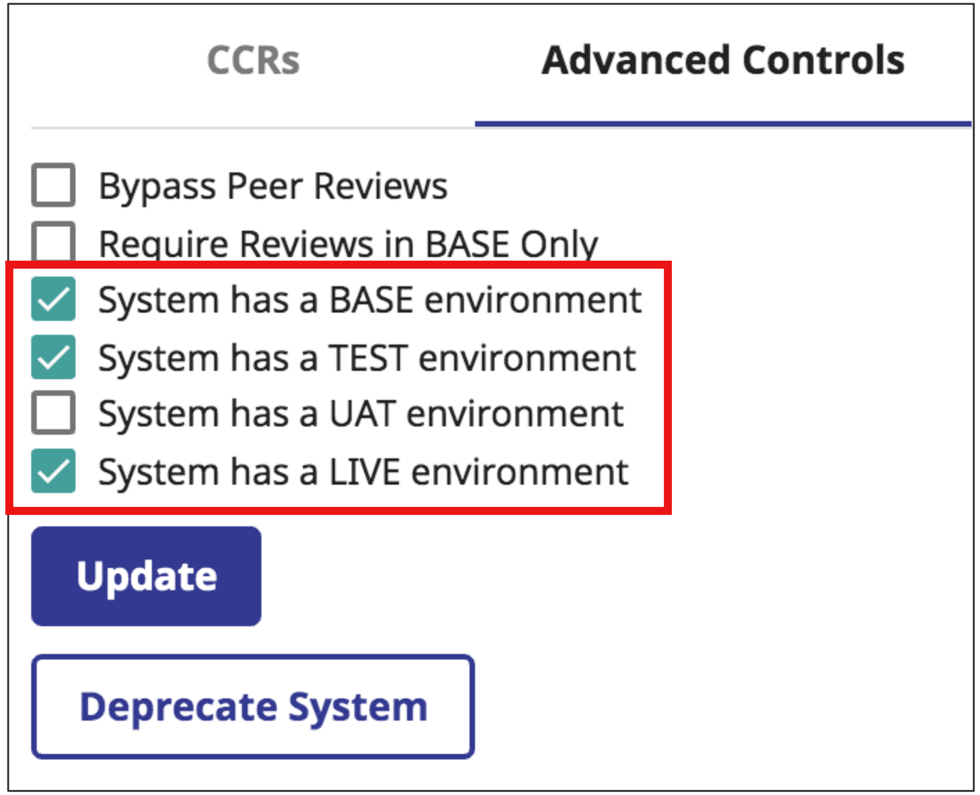


CCRs | Advanced Controls

- □ Bypass Peer Reviews
- □ Require Reviews in BASE Only
- ☑ System has a BASE environment
- ☑ System has a TEST environment
- □ System has a UAT environment
- ☑ System has a LIVE environment

Environment Workflow Options

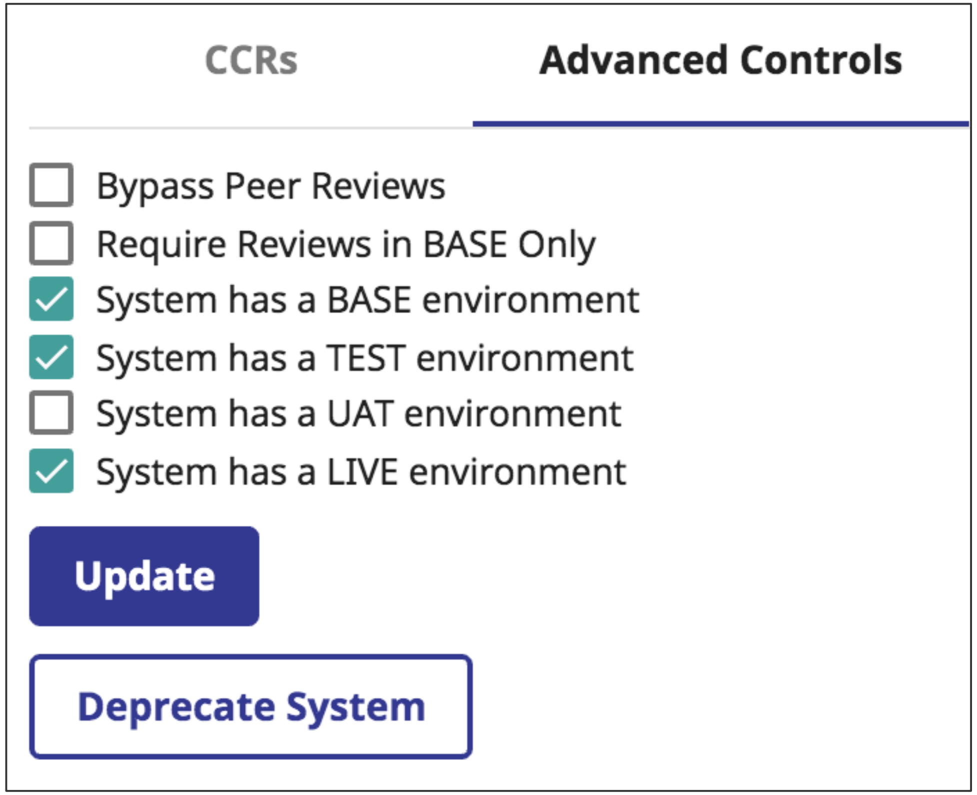
- Can tailor system workflow to include only relevant environments.
- BASE-only or LIVE-only workflows are supported.
- Workflows with 2 or more environments must include BASE.
- For Tier 1 and Tier 2 Systems, must baseline Perforce when adding environments.



□ Bypass Peer Reviews
□ Require Reviews in BASE Only
☑ System has a BASE environment
☑ System has a TEST environment
□ System has a UAT environment
☑ System has a LIVE environment

Deprecate System

- Deprecate unused systems.
- Removes system from dropdown lists.
- Cancels all active CCRs for system.
  - Warning: cannot be undone!

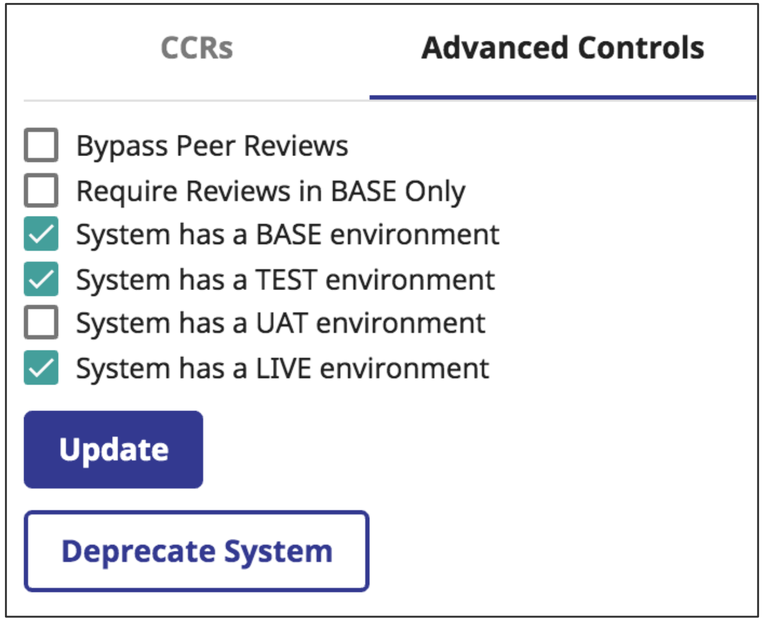


☐ Bypass Peer Reviews
☐ Require Reviews in BASE Only
☑ System has a BASE environment
☑ System has a TEST environment
☐ System has a UAT environment
☑ System has a LIVE environment

Quiz: Advanced Controls

Question:

The check boxes in this screenshot for BASE, TEST, UAT, and LIVE indicate which environments are used in the workflow for this System. True or False?



☐ Bypass Peer Reviews
☐ Require Reviews in BASE Only
☑ System has a BASE environment
☑ System has a TEST environment
☐ System has a UAT environment
☑ System has a LIVE environment

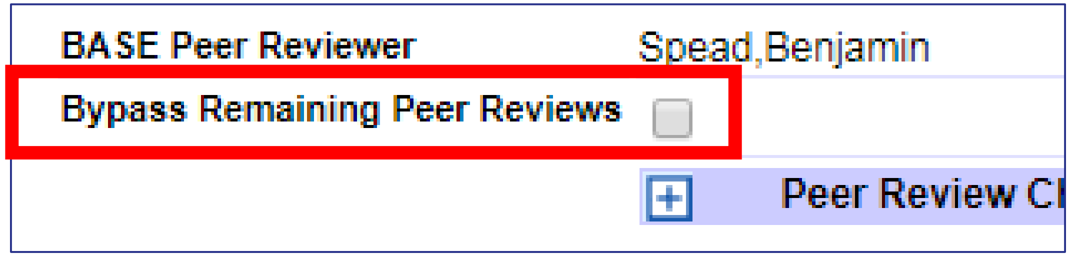
Answer:

True.

Part 4: System Architects

System Architects

- Users can be assigned to system as architects.
- Each organization with responsibility for a System can assign their own architects.
- Architects receive highlight email notifications for all CCRs.
  - For example when CCRs are opened, moved to a new phase, or closed.
- Architects receive all peer reviews by default.
  - Architects can "Bypass Remaining Peer Reviews" for any CCR by editing CCR Details pane.

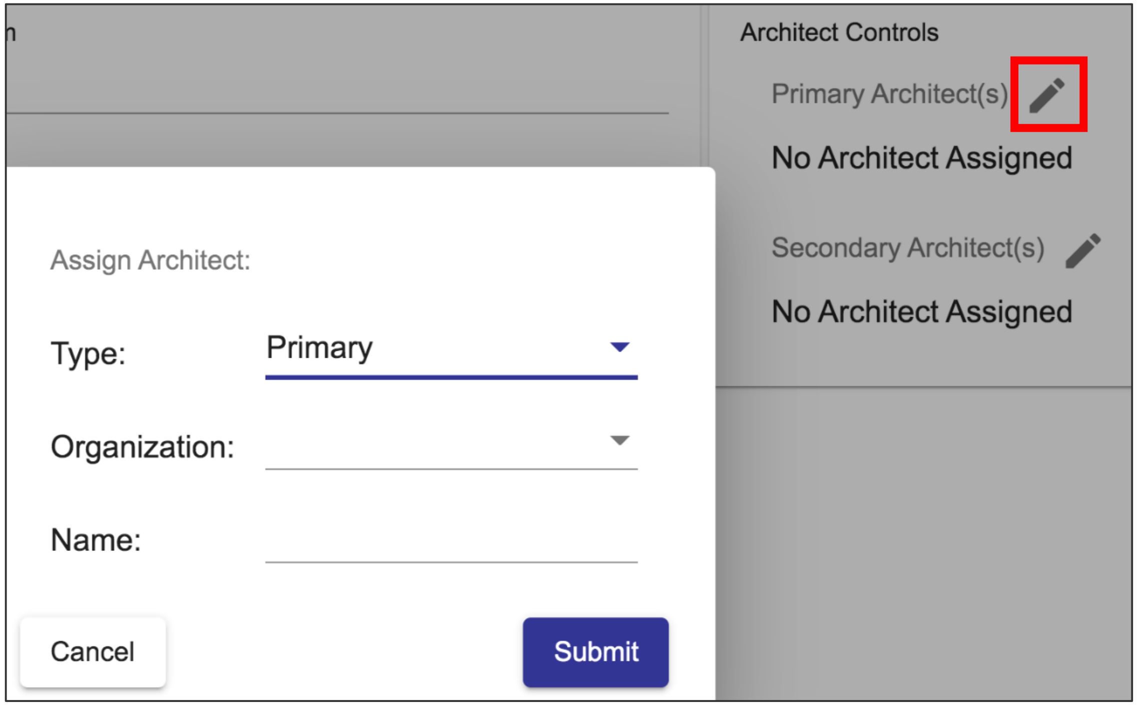


BASE Peer Reviewer Spead,Benjamin

Bypass Remaining Peer Reviews □

System Architects (cont.)

- Two kinds of architects:
  - Primary architect.
    - Default peer reviewer for all CCRs owned by their organization for that System.
  - Secondary architect.
    - Peer reviews all CCRs transitioned to XXXX_Pending_Peer_Review state by primary architect.
- Architects configured on System Details page.
  - Only editable by InterSystems employees.



Architect Controls

Primary Architect(s)

No Architect Assigned

Secondary Architect(s)

No Architect Assigned

Assign Architect:

Type: Primary

Organization:

Name:

Quiz: Primary Architects

Question:

Which of the following is true about any primary architect? Select all that apply.

A. Automatically receive Go Live announcements.

B. Can bypass remaining peer reviews for any CCR against their System.

C. Are the default peer reviewer for all CCRs for that System.

D. Can add a secondary architect.

Quiz: Primary Architects (cont.)

Answer:

B. Can bypass remaining peer reviews for any CCR against their System.

They receive highlight emails, not Go Live announcements automatically.

They are the default peer reviewer only for CCRs by their org for that System.

Only InterSystems employees can add architects.

Summary

- What are they key points for this module?

ICC 330:
CCR Tier 0 – Auxiliary Tools and Transitions

InterSystems
Creative data technology

Overview

- Configure peer review routing rules.
- Identify peer review models and workflow strategies.
- Configure peer review documents.

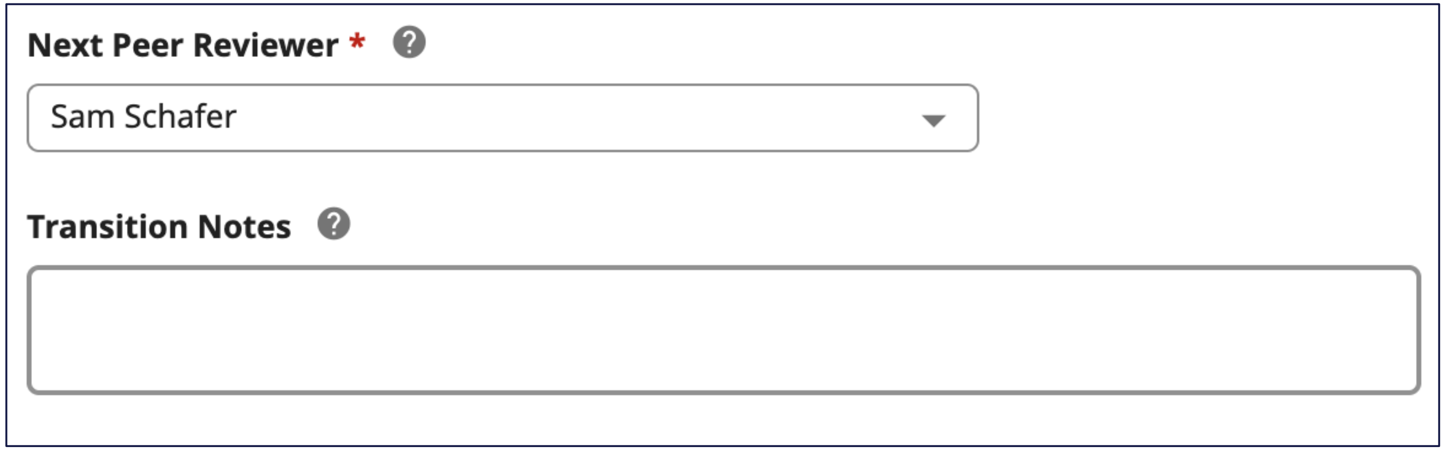
Overview (cont.)

- Additional transitions:
  - Reassign.
  - Merge.
  - Clone.
  - Cancel.
  - MarkIntegrationFailed.
  - MarkValidationFailed.
  - Revert.
  - ChangeSpec.
  - RequestOverride.

Part 1: Peer Reviews Options

Assigning a Peer Reviewer

- Several options for configuring default value of Next Peer Reviewer Name field.
  - Field shown during transitions into peer review states.
- Owner can change from default value to anyone else from Responsible Organization.



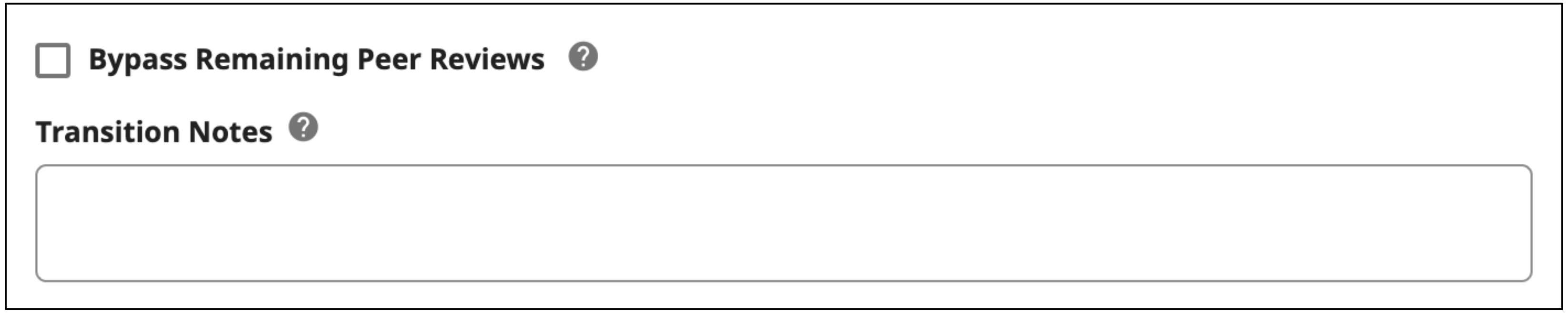
Next Peer Reviewer *

Sam Schafer

Transition Notes

System Architects

- Systems can have users designated as architects.
- Architects receive highlight email notifications for all CCRs.
  - For example when CCRs are opened, moved to a new phase, or closed.
- Architects receive all peer reviews by default.
- Architects can disable remaining peer reviews either:
  - By editing CCR Details Pane.
  - During pass peer review transition.

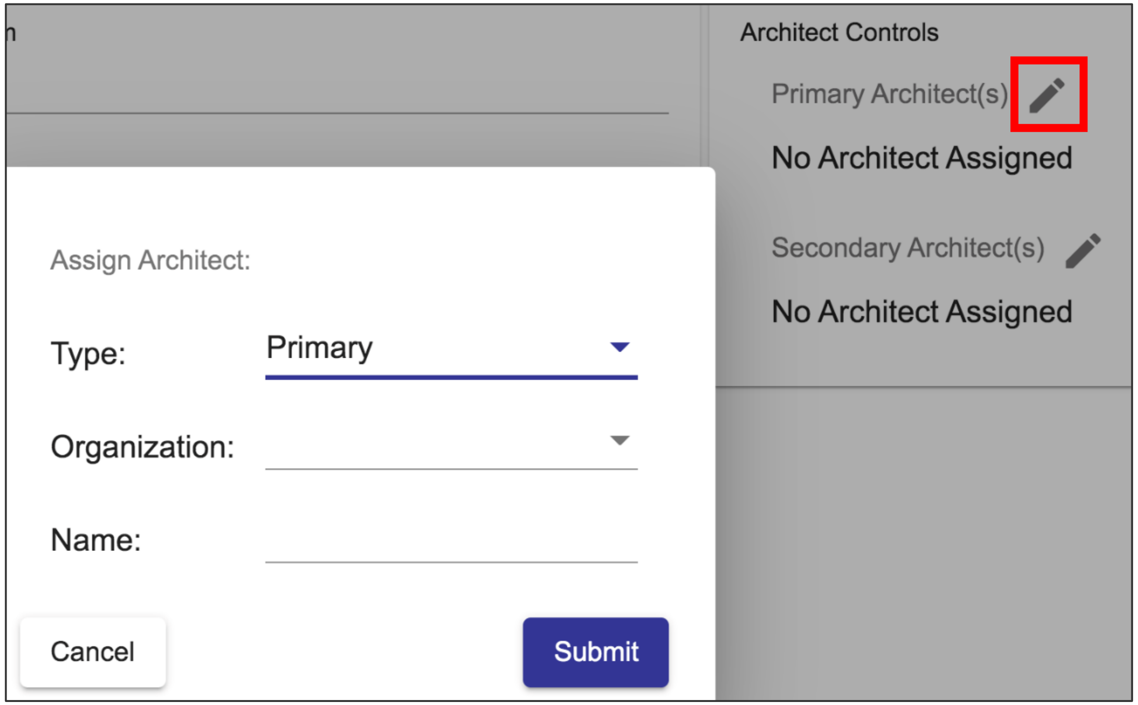


☐ Bypass Remaining Peer Reviews ?

Transition Notes ?

System Architects (cont.)

- Two kinds of architects:
  - Primary architect.
    - Default peer reviewer for all CCRs owned by their organization for that System.
  - Secondary architect.
    - Peer reviews all CCRs transitioned to XXXX_Pending_Peer_Review state by primary architect.
- Architects configured on System Details page.



Architect Controls

Primary Architect(s)

No Architect Assigned

Secondary Architect(s)

No Architect Assigned

Assign Architect:

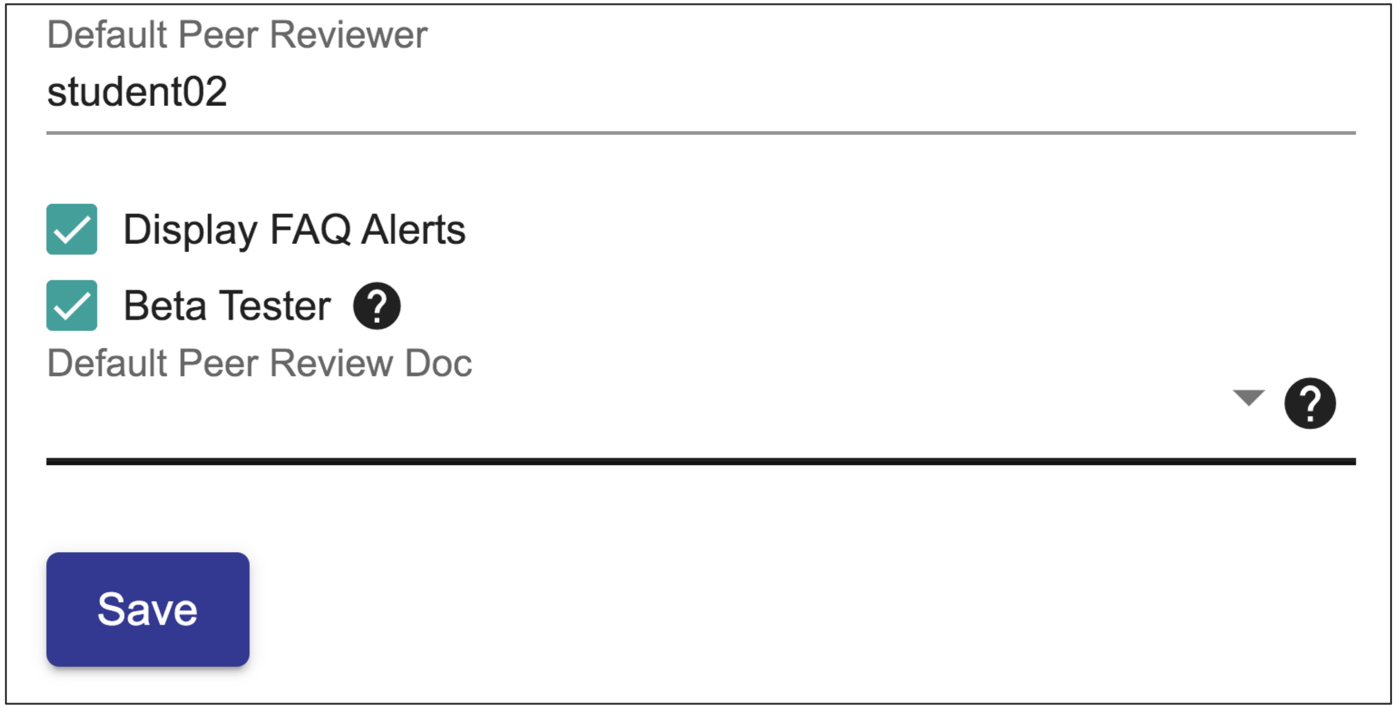
Type: Primary

Organization:

Name:

Default Peer Reviewer per User

1. Menu > Users.
2. Select the org.
3. Select the user.
4. Select the "Default Peer Reviewer."
5. Click "Save."



Default Peer Reviewer
student02

☑ Display FAQ Alerts

☑ Beta Tester ?

Default Peer Review Doc

Peer Review Routing

- Default value in drop down for next peer reviewer follows set of rules:
  1. Route to the primary architect for that (ResponsibleOrg + System) if one is assigned.
  2. Route to the secondary architect for that ResponsibleOrg + System if the CCR was authored by the primary architect AND there is a secondary architect assigned.
  3. Route to the default peer reviewer for that user if one is assigned on the User Details page.

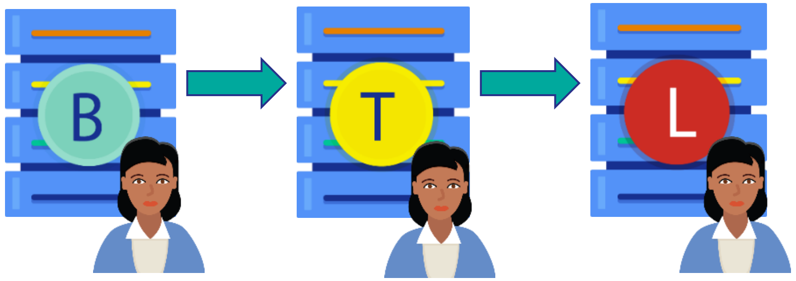
Peer Review Routing (cont.)

- Default value in drop down for next peer reviewer follows set of rules:

4. Route to the user’s manager.
   - Only for InterSystems employees.
5. Remains with user who transitioned into. XXXX_Pending_Peer_Review
   - This user still cannot passPeerReview.

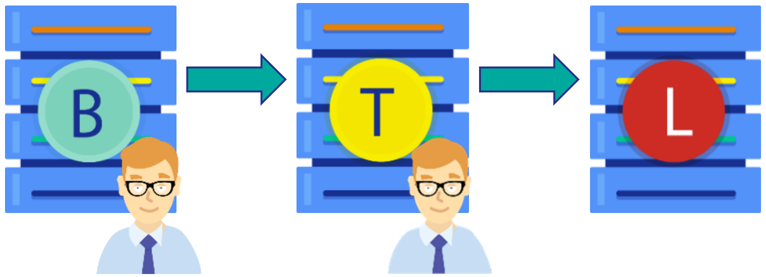
Peer Review Workflow Strategies

- Different peer review workflow strategies can be used for different scenarios.
- Standard peer review.
  - Every CCR gets peer reviewed in every environment.
  - Useful for providing most thorough peer review coverage.
  - Default setup (no configuration necessary).



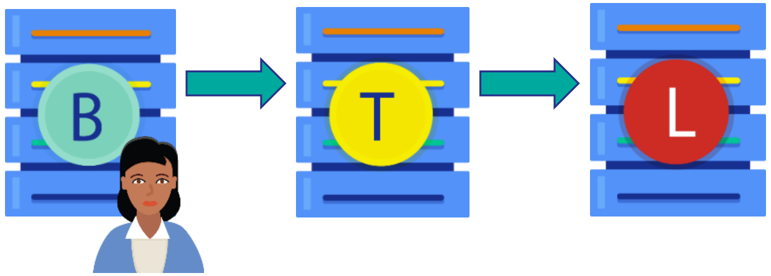
Peer Review Workflow Strategies (cont.)

- Architect with peer review bypass.
  - CCRs routed to architect for all reviews, who can "Bypass Remaining Peer Reviews" for a CCR at any time (even when not in a peer review state).
  - Useful for shortening workflow of low risk CCRs after initial review has passed.
  - Configured by assigning architects on System Details page.



Peer Review Workflow Strategies (cont.)

- BASE-Only peer review.
  - Useful for mid-phase of new projects.
    - Get a second set of eyes on all changes before the system goes into production.
  - Configured by selecting ‘Peer Review BASE Only’ under System Advanced Controls.



Peer Review Workflow Strategies (cont.)

- No peer review.
  - Should *only* be used during early BASE-only phase of a new implementation project.
    - Speed is of the essence.
    - Introduction of a broken change will not put anything at risk.
    - Once other environments are introduced to workflow, enable peer review.
  - Configured by selecting ‘Bypass Peer Reviews’ under System Advanced Controls.

Quiz: Peer Review Routing Configuration

Question:

Which peer review routing feature should be used when:

- Multiple teams are working on the System, and
- One person per team should handle peer reviews for that team.

A. Primary architects.

B. Default peer reviewer setting for each user.

C. Secondary architects.

D. Default group peer reviews.

Quiz: Peer Review Routing Configuration (cont.)

Answer:

B. Default peer reviewer setting for each user.

Since architects cannot be configured per team, only per responsible organization for a system, the only option is default peer reviewer setting.

Quiz: Primary/Secondary Architects

Question:

Which of the following is a feature only available to primary and secondary architects?

A. Receive peer reviews for CCRs by a different responsible organization.

B. Bypass remaining peer reviews on any CCR for that system where their organization is the responsible organization.

C. Perform peer reviews when system level Bypass Peer Reviews setting is true.

D. Pass their own peer reviews.

Quiz: Primary/Secondary Architects

Answer:

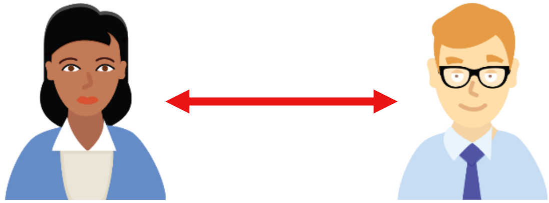
B. Bypass remaining peer reviews on any CCR for that system where their organization is the responsible organization.

A is wrong because it is never possible (only users from responsible organization can modify a CCR). C is wrong because no peer reviews are ever in workflow if bypass peer reviews system setting is true.

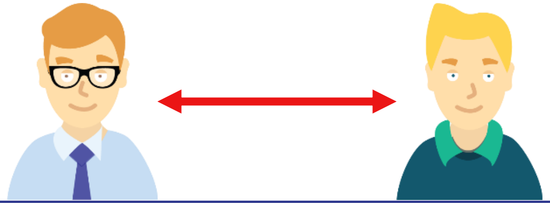
No one can pass their own peer review.

Peer Review Models

- Different peer review models provide different benefits.
- Peer programming.
  - Assign experienced colleagues as each other’s default peer reviewer.
  - Useful for cross-training on each others’ work.

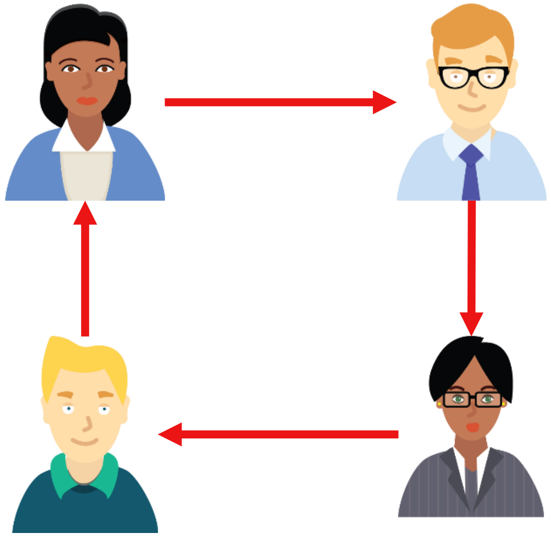


- Mentor/mentee.
  - Assign a senior person to be the default reviewer of a junior person and vice versa.
  - Allows the senior to instruct during reviews of the junior, and the junior to learn from work completed by the senior.



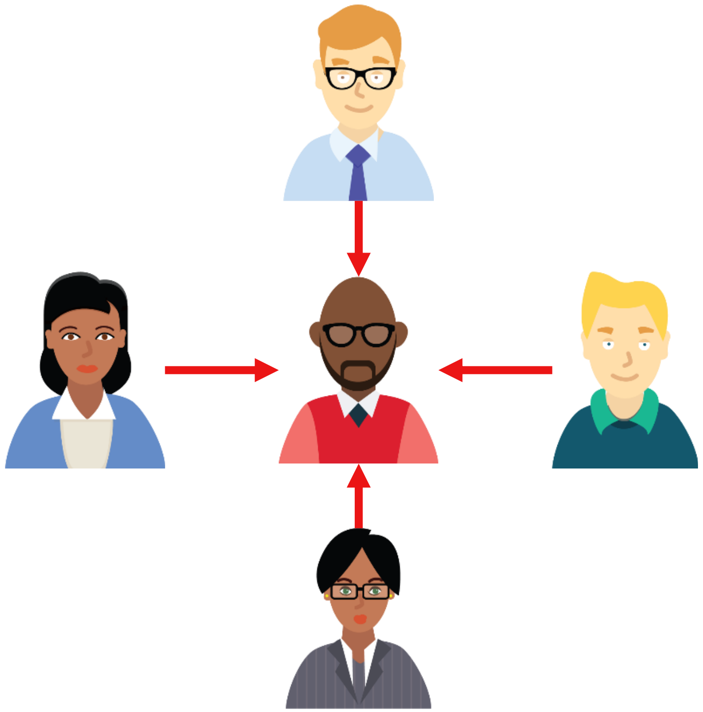
Peer Review Models (cont.)

- Round robin.
  - Assign colleagues in a cycle where each reviews and is reviewed by a different person.
  - Allows for broader collaboration within a team.
  - Useful for getting members to engage more broadly within the team.



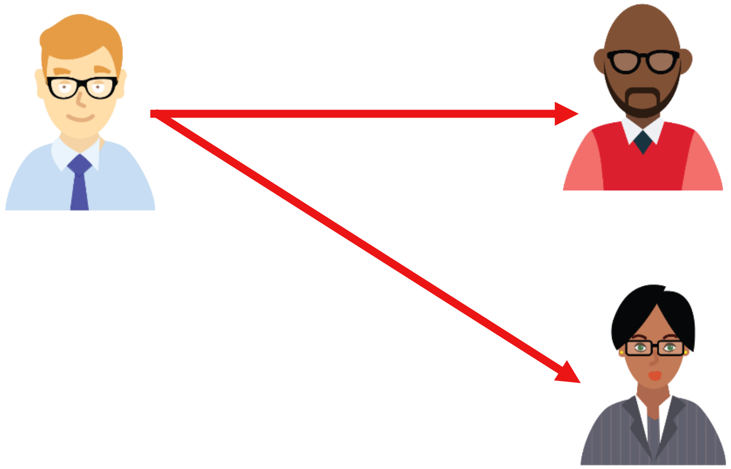
Peer Review Models (cont.)

- Hub and spoke.
  - Assign a central person as default peer reviewer for all other teammates.
  - Useful when there are multiple teams working on a System and a single primary architect will not suffice.



Peer Review Models (cont.)

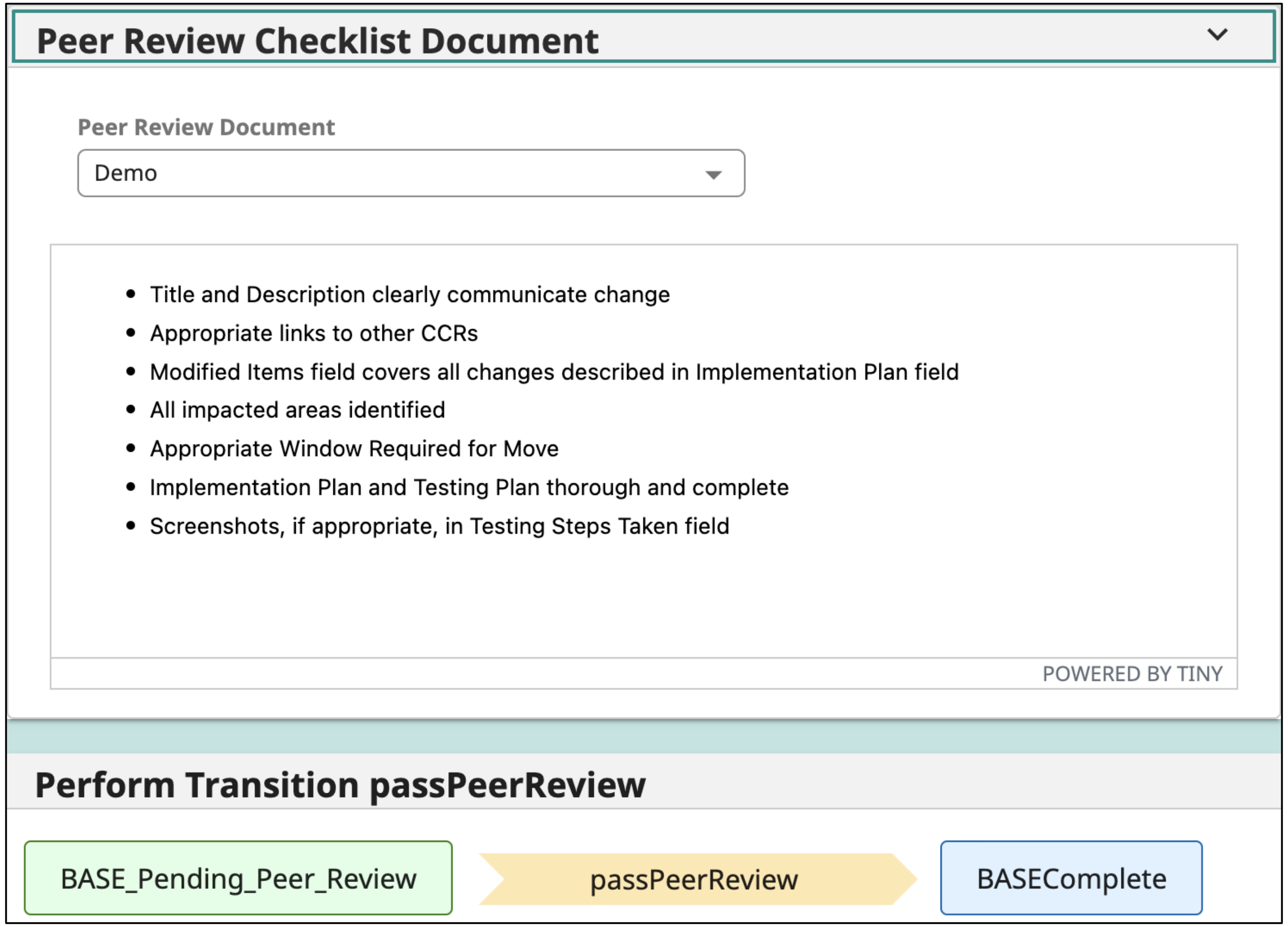
- Different peer reviewers in each phase.
  - Multiple people verify change and documentation.
    - Reduce risk of mistakes.
    - Increase cross-training on new changes.
  - No configuration tools in CCR to fully configure this option.



Peer Review Documents

- Can configure Peer Review Checklist Document to display on passPeerReview transition.
- Describes best practices for completing peer review.
  - For reference purpose only; no interactive check boxes.
- Collapsed by default.
- Default checklist for user autoselected.
  - Able to select other documents.

Peer Review Documents (cont.)



Peer Review Checklist Document

Peer Review Document: Demo

- Title and Description clearly communicate change
- Appropriate links to other CCRs
- Modified Items field covers all changes described in Implementation Plan field
- All impacted areas identified
- Appropriate Window Required for Move
- Implementation Plan and Testing Plan thorough and complete
- Screenshots, if appropriate, in Testing Steps Taken field

Perform Transition passPeerReview

Creating Peer Review Documents

- Menu > Peer Review Docs > Add New Document.
- One organization can have multiple peer review documents.
  - Different documents may make sense for technical vs application reviews.

Default Peer Review Documents

- Organization:
  - Menu > Organizations.
- System:
  - Menu > Systems.
- User:
  - Menu > Users.
- Order of precedence:
  - User.
  - System.
  - Organization.

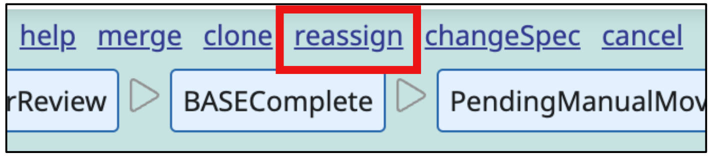
Do Exercise ICC330-1

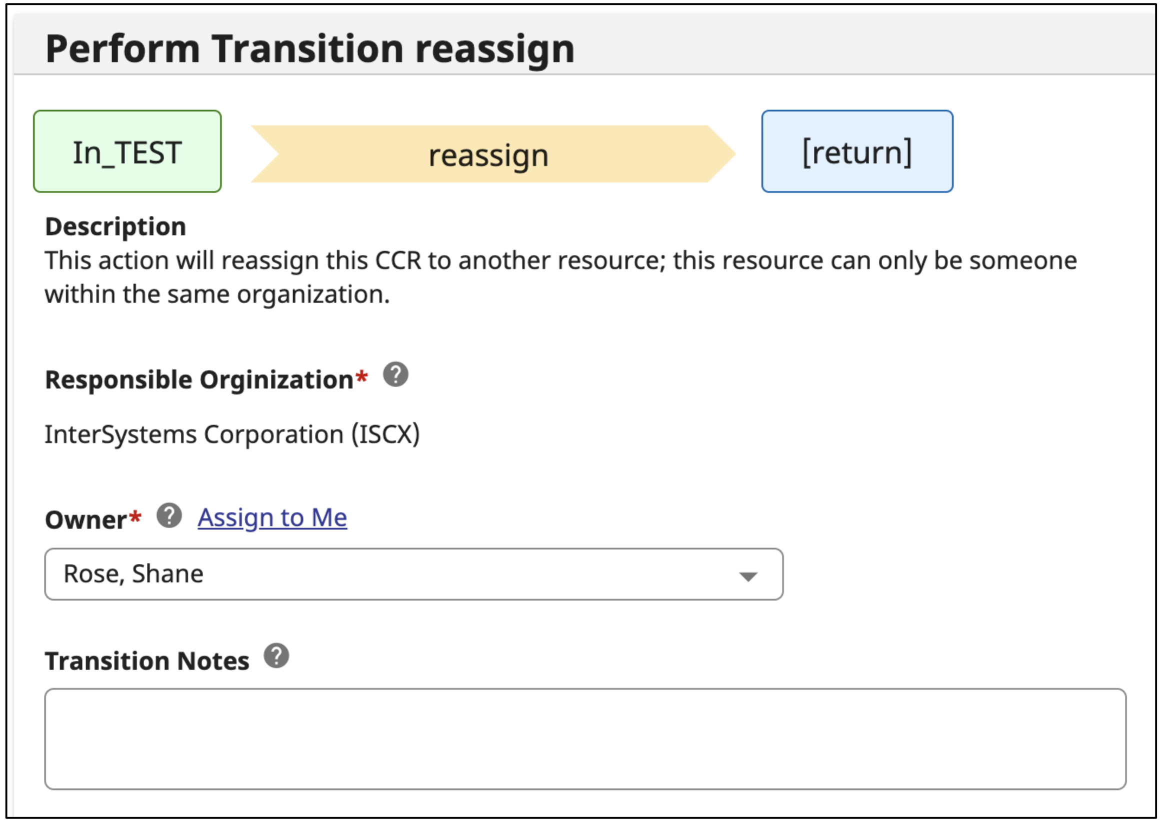
Part 2: Additional Transitions

- Reassign.
- Merge.
- Clone.
- Cancel.
- MarkIntegrationFailed.
- MarkValidationFailed.
- Revert.
- ChangeSpec.
- RequestOverride.

Reassign CCR Owner

- Users from responsible organization can change owner.
  - Not just current owner.
- Click reassign link at top of CCR.
- Select new Owner.
  - Assign to Me link specifies yourself.
- Optionally use transition notes to specify reason for reassign.
- Click the “reassign” button.





Description
This action will reassign this CCR to another resource; this resource can only be someone within the same organization.

Responsible Orginization*

InterSystems Corporation (ISCX)

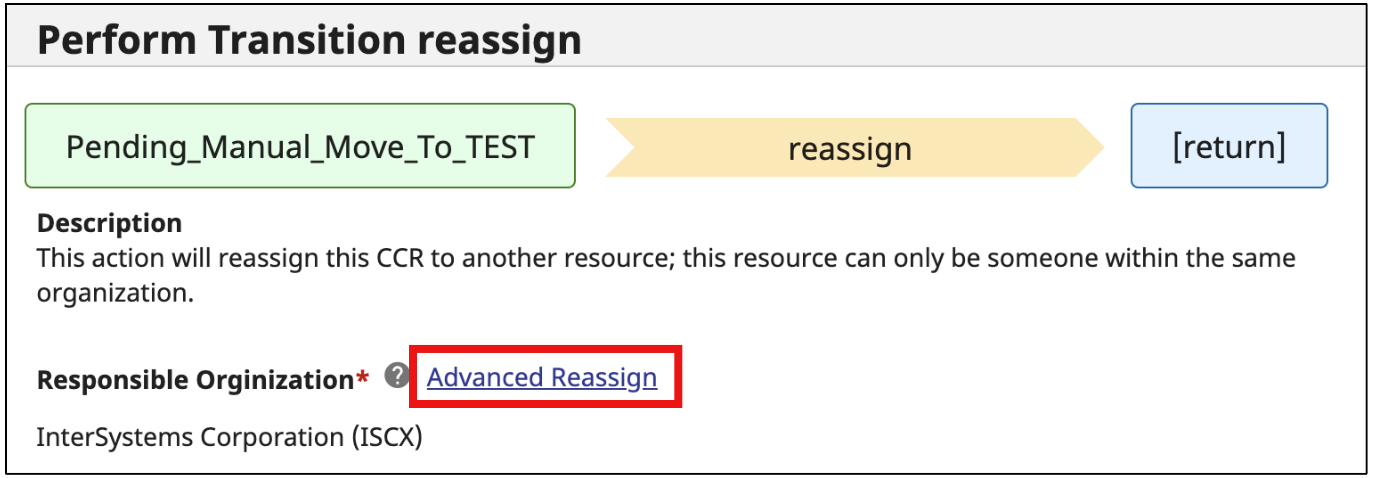
Owner* Assign to Me

Rose, Shane

Transition Notes

Reassign Responsible Organization

- InterSystems employees can change Responsible Organization.
- Click advanced reassign in reassign dialogue.
  - Changes Responsible Organization field from text field to drop-down menu.
- Select Responsible Organization.
- Select new owner.



Perform Transition reassign

Description
This action will reassign this CCR to another resource; this resource can only be someone within the same organization.

Responsible Orginization* ? Advanced Reassign

InterSystems Corporation (ISCX)

Quiz: Reassign a CCR

Question:

Only a CCR’s owner can reassign it to another person. True or False?

Answer:

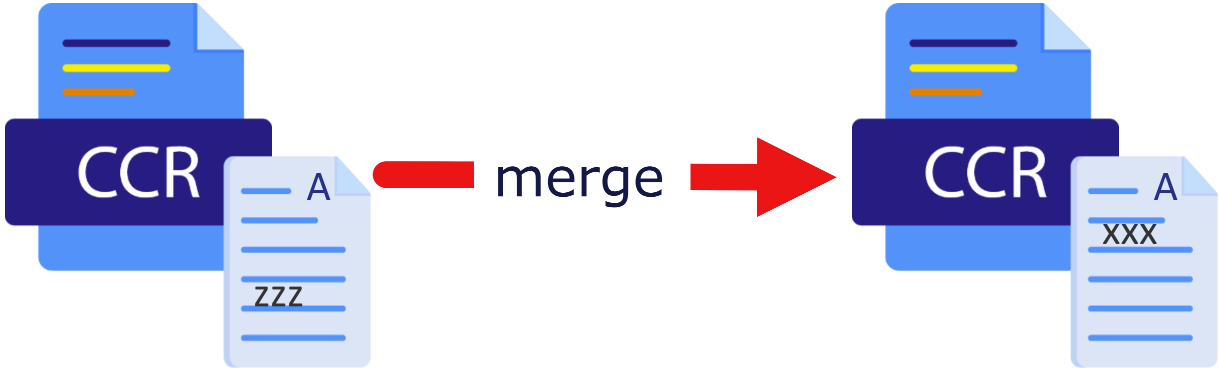
False.

Any user from the current responsible organization can reassign a CCR at any time.

An InterSystems employee can change the responsible organization and pick a new owner from that organization at any time.

Transition: merge

- Merging CCRs combines two or more changes into a single change.
- Moves CCR into MERGED phase and Merged state.
  - An endpoint phase and state.

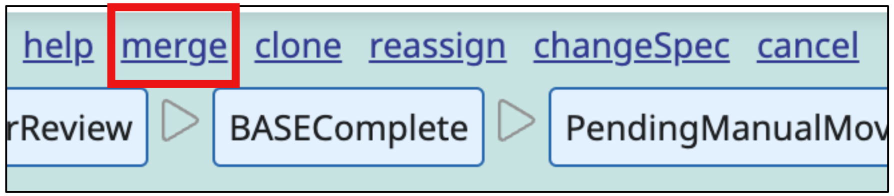


Transition: merge (cont.)

- Merging two CCRs will:
  - Append all fields from the source (aka “Merged From”) CCR to the target (aka “Merged To”) CCR.
  - Associate items in Perforce for the source CCR with the target CCR.
  - Create pointers between the source and target CCRs.
  - Transition the source CCR to a terminal “Merged” State.

Transition: merge (cont.)

- ‘merge’ link is at the top of every active CCR.



- A CCR can only be ‘merged’ into another CCR if
  - Both CCRs are in the same state.
  - Both CCRs are against the same System.
- Merges cannot be undone.

Quiz: Merge Restrictions

Question:

What requirements exist for two CCRs to be merged? (select all that apply)

A. Both CCRs have the same owner.

B. Both CCRs are for the same organization.

C. Both CCRs are against the same System.

D. Both CCRs are in the same CCR Tier (0/1/2).

E. Both CCRs are in the same State.

Quiz: Merge Restrictions (cont.)

Answer:

B. Both CCRs are for the same organization.

C. Both CCRs are against the same System.

E. Both CCRs are in the same State.

CCRs must be same State so neither skips a State; they must be same System to keep Perforce items in same Branch (same System implies same Org).

Quiz: Undoing a Merge

Question:

If a ‘merge’ was performed in error, InterSystems Support can assist in ‘unmerging’ the CCRs in question to return them to their original state. True or False?

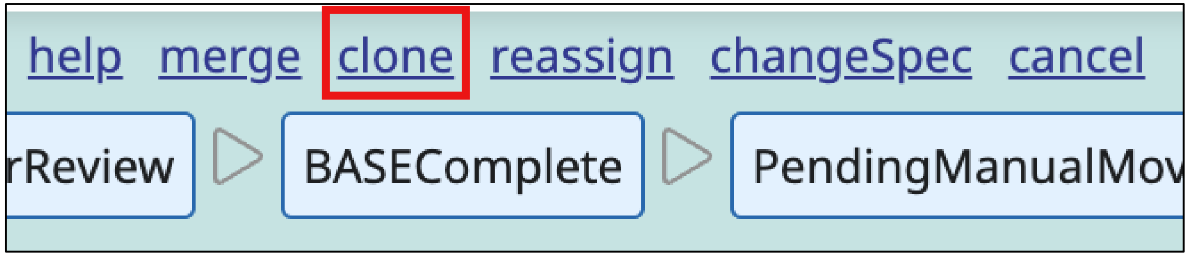
Answer:

False.

Merging CCRs is a non-reversible action.

Transition: clone

- Clone transition link at top of every CCR.
- Used instead of create transition.
- Cloning duplicates reusable content to create new CCR.
  - Automatic reuse of the Description, Testing Plan, Implementation Plan, etc.



Transition: clone (cont.)

- New CCR in In_Base state after clone transition.
- Helpful when a change needs to be repeated or reused.
  - Created for same System (‘repeated’ change) or against different System (‘reused’ change).
- User can integrate Perforce items from source CCR into BASE branch for cloned CCR.
  - Tier 1 or Tier 2 CCRs only.

Quiz: Cloning Timing

Question:

At what point during the workflow can a CCR be cloned?

A. Only during the BASE phase.

B. During any state in which the CCR is still considered ‘Active.’

C. Only after the CCR has been moved to closed.

D. At any time.

Quiz: Cloning Timing

Answer:

D. At any time.

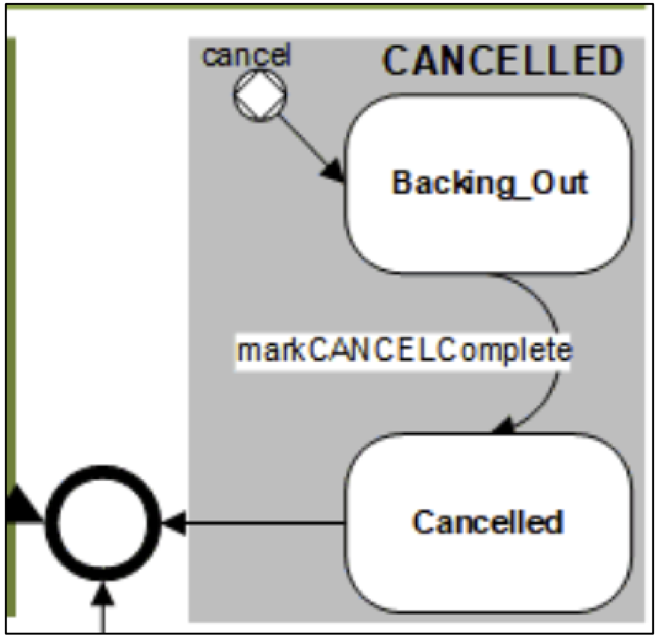
Of course, it may be more useful to clone a CCR which has been progressed further along its workflow. However, you can clone a CCR at any time in the creation workflow.

Transition: cancel

- Used when change no longer needed.
  - Never use to resolve CCR progression errors!
- Completely backout change from all environments according to backout plan.

Transition: cancel (cont.)

- Initiates workflow to cancel CCR.
  - Moves CCR to Backing_Out state in CANCELLED phase.
  - Once backout plan completed, perform markCANCELComplete.
    - Moves CCR to endpoint Cancelled state.
    - Makes clear changes successfully backed out.



Quiz: cancel

Question:

cancel is only necessary for CCRs past BASE phase. True or False?

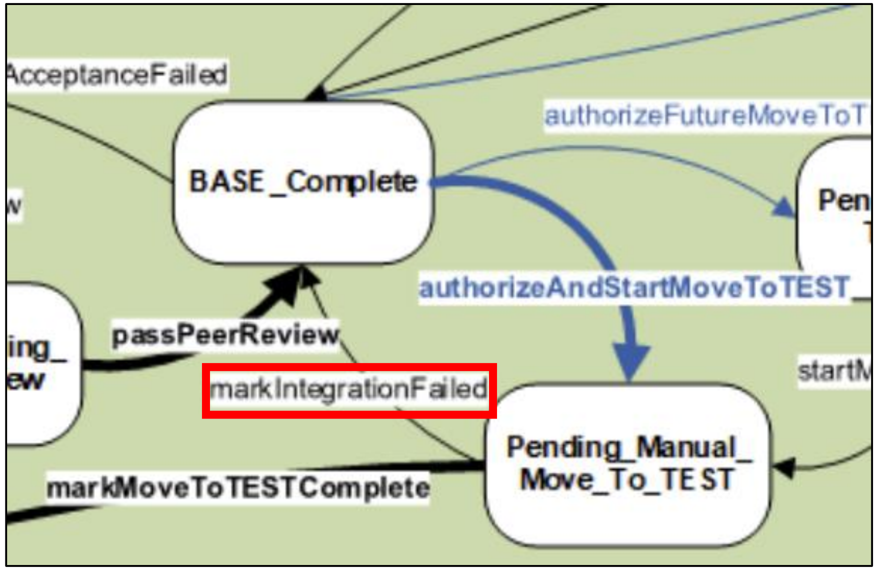
Answer:

False.

Always CCRs when necessary. Also, make sure to fully back out the CCR even if changes only made in BASE. Abandoned CCRs will cause crises.

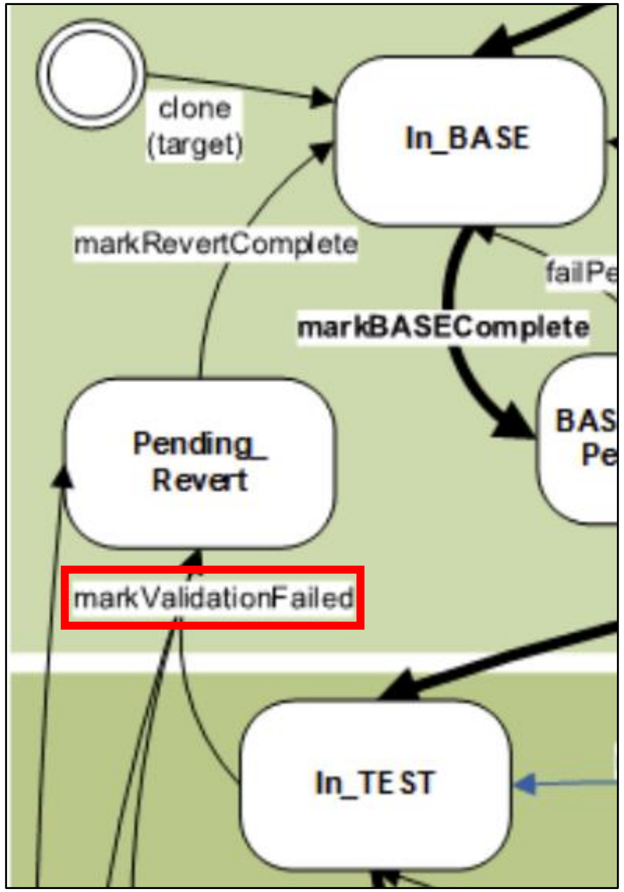
Transition: markIntegrationFailed

- Only for CCRs:
  - In Pending_Manual_Move_To_XXXX.
  - No submitted changes in next Phase.
  - No ongoing Perforce activity.
- Returns CCR to prior state.
- Use cases:
  - Enables best practice of short-lived Pending_Manual_Move_To_XXXX states.
  - Easier resolution of circular conflicts.
  - Enable retriggering automatic integration, when/if appropriate.



Transition: markValidationFailed

- Use for catastrophic failure of change in TEST / UAT / LIVE.
- markValidationFailed workflow:
  - Moves CCR to Pending_Revert state.
  - Follow backout plan to fully backout and then restore changes to BASE.
  - Perform markRevertComplete to transition to In_BASE.

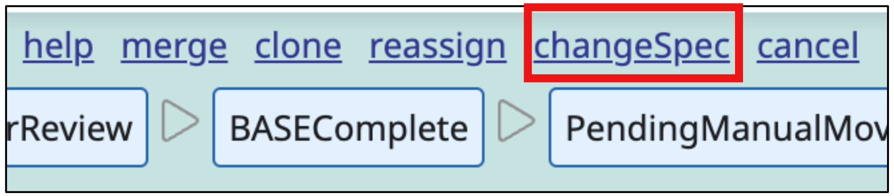


Revert

- Only for CCRs in Closed state.
- Used to reopen CCR and revert to In_BASE.
- Revert workflow:
  - Moves CCR to Pending_Revert state.
  - Fully backout changes, optionally restoring to BASE.
  - Perform markRevertComplete to transition CCR to In_BASE.
  - If appropriate, cancel CCR.

Transition: changeSpec

- ‘changeSpec’ transition link is at the top of every active CCR beyond In_PREP state.



- Used when need to spend time redefining the specification for a change.
  - Should not be In_BASE state because won’t be making changes for significant period of time.
  - Maintains history of change in specification by using same CCR, rather than cancelling.

Transition: changeSpec (cont.)

- Initiates workflow to return CCR to In_PREP state.
  - changeSpec moves CCR to Pending_Full_Revert state.
  - Revert all changes made in CCR progress so far.
  - After all changes reverted, perform markRevertComplete transition.
    - Moves to Pending_Spec_Rework.

Quiz: changeSpec

Question:

What is purpose of changeSpec?

A. Return active CCR to In_PREP to be able to modify backout plan.

B. Return active CCR to In_PREP to be able to modify testing plan.

C. Return active CCR to In_PREP while redefining the specification with customer.

D. All of the above.

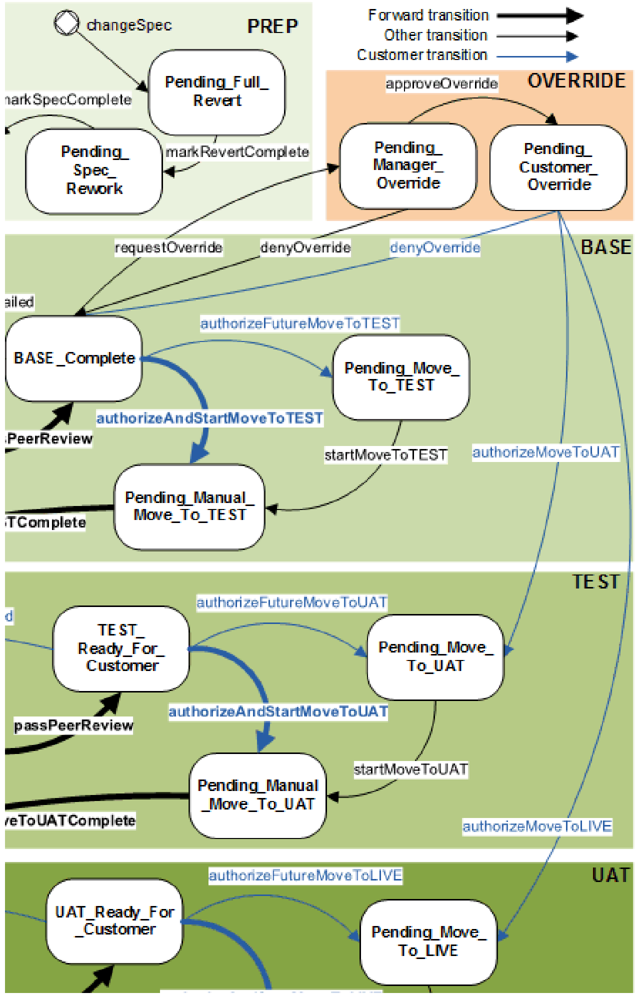
Quiz: changeSpec (cont.)

Answer:

C. Return active CCR to In_PREP while redefining the specification with customer.

You can modify the backout plan and testing plan at any point for an active CCR.

Skipping TEST and UAT Environments



Transition: requestOverride

- Optional transition for BASE_Complete state.
- Used to skip making changes to TEST and/or UAT.
- Only for Tier 0 CCRs.
  - Skipping branches for Tier 1 or Tier 2 will cause integration problems.

requestOverride Workflow

- Moves CCR to Pending_Manager_Override state in the OVERRIDE phase.
- CCR assigned to:
  - Primary architect, if defined.
  - Secondary architect, if primary architect made change.
  - User’s manager if no primary or secondary architect.
  - User that performed transition if no manager defined.
- Architect/manager has 2 options:
  - denyOverride, returning CCR to BASE_Complete state.
  - approveOverride, moving CCR to Pending_Customer_Override state.

requestOverride Workflow (cont.)

- If approved by architect or manager, customer has 2 to 3 options, depending on system architecture:
  - denyOverride, returning CCR to BASE_Complete state.
  - If UAT environment exists: authorizeMoveToUAT, moving CCR to Pending_Move_To_UAT state.
  - authorizeMoveToLIVE, moving CCR to Pending_Move_To_LIVE state.

Quiz: requestOverride

Question:

When is it okay to use requestOverride for a Tier 1 CCR?

A. For emergency changes.

B. For standard changes.

C. For normal changes.

D. Never.

Quiz: requestOverride (cont.)

Answer:

D. Never.

Tier 1 CCR’s involve code in source control. Having CCR controlled code in BASE and LIVE but not TEST will cause issues down the road.

Do Exercise ICC330-2

Summary

- What are the key points for this module?

ICC 340:
CCR Tier 0 –
Project Management

InterSystems
Creative data technology

Overview

- Groups.
- Email notifications.
- Organization dashboard.
- Reports.
- Related Records.

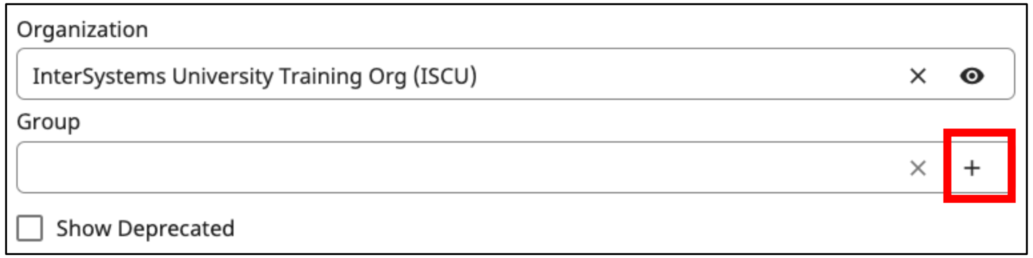
Part 1: Groups

Groups

- CCRs can be organized into a group for project management.
  - Example: CCRs related to a project.
  - 1 CCR can only belong to 1 group.
  - Group can contain CCRs from multiple Systems.
- Can view all CCRs in group.
- Can merge all CCRs in a group into 1 CCR.
  - Usually progress all CCRs to TEST_Ready_For_Customer then merge for go-LIVE.

Creating Groups

1. Menu > Groups.
2. Select organization.
3. Click + next to Group field.
4. Provide at least title and description.
5. Click Create Group.

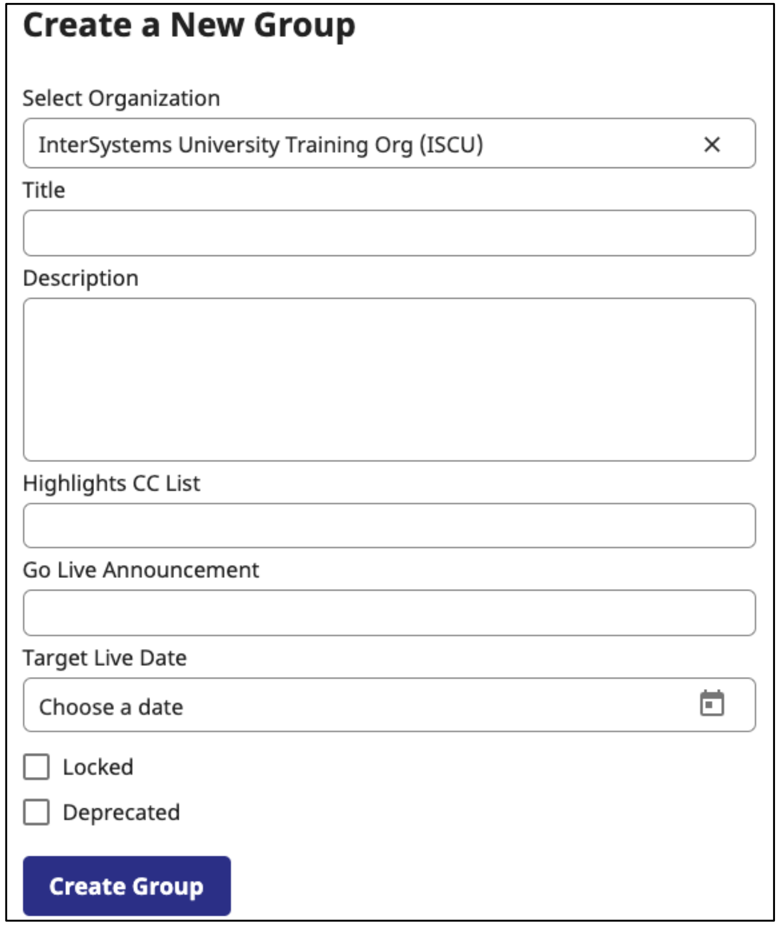


Organization

InterSystems University Training Org (ISCU)

Group

□ Show Deprecated



Create a New Group

Select Organization

InterSystems University Training Org (ISCU)

Title

Description

Highlights CC List

Go Live Announcement

Target Live Date

Choose a date

□ Locked

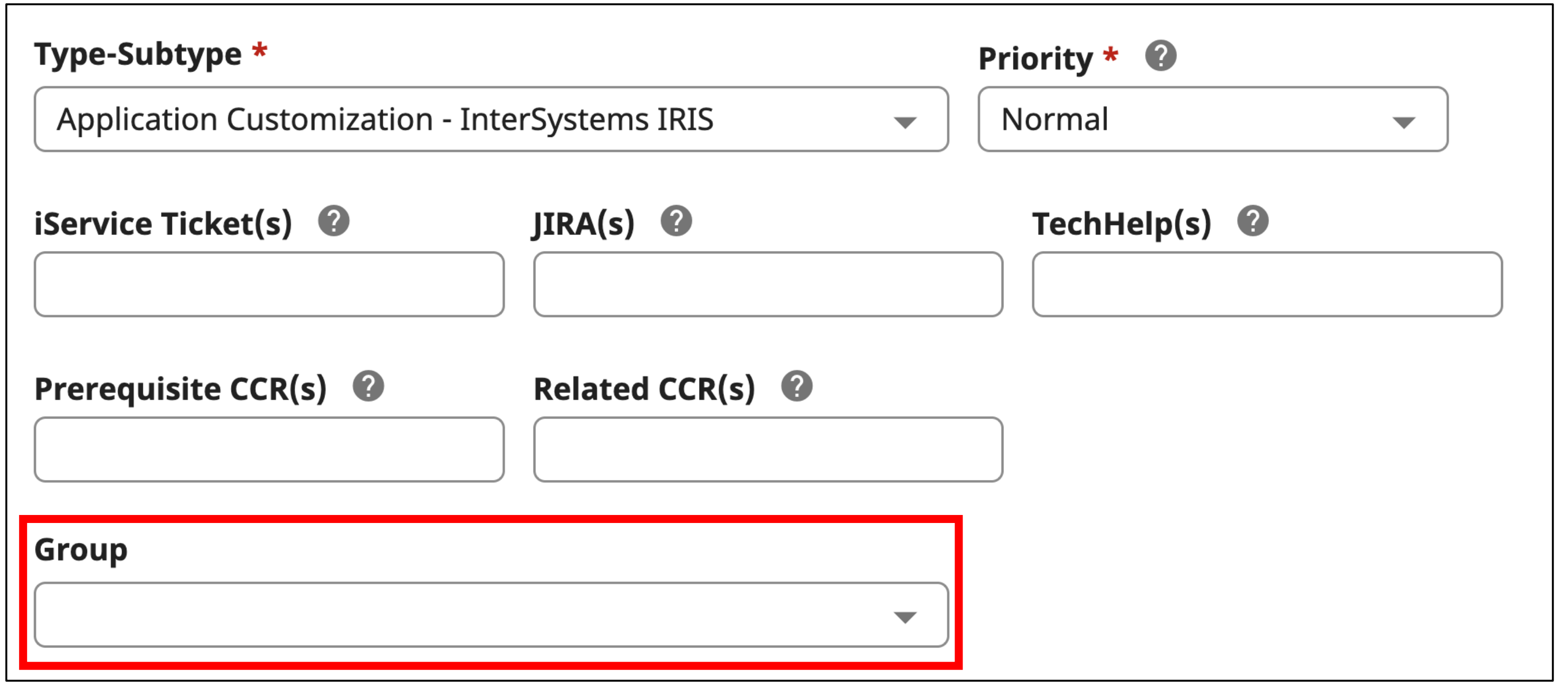
□ Deprecated

Create Group

Assigning CCRs to a Group

- A CCR can be added to a group:
  - At time of CCR creation.
  - By editing CCR details section of desired CCR.
  - Using Groups page.
    - Allows adding multiple CCRs at one time.

Assigning CCRs to a Group: Creation Time



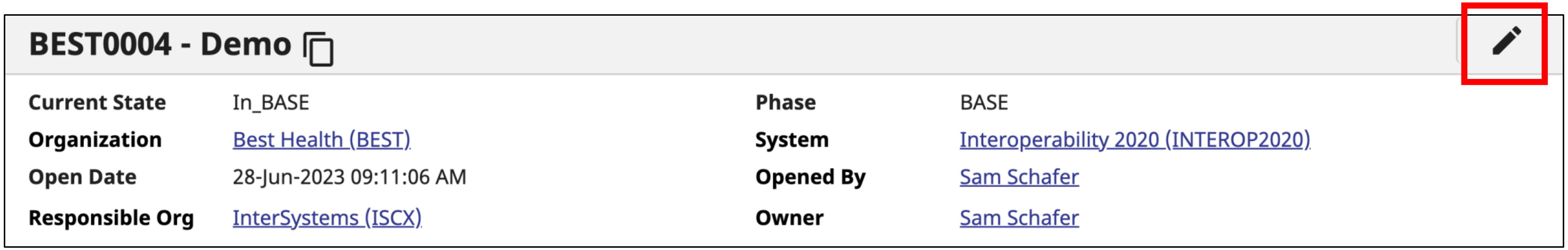
Type-Subtype *

Application Customization - InterSystems IRIS

Priority *

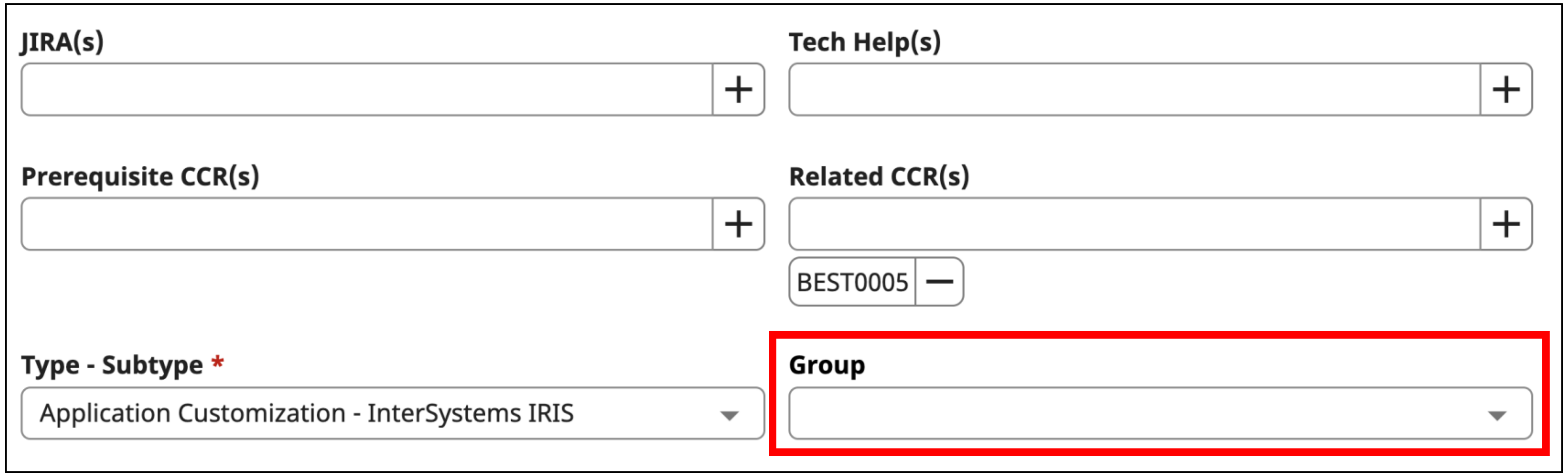
Normal

Assigning CCRs to a Group: Editing CCR Details



BEST0004 - Demo

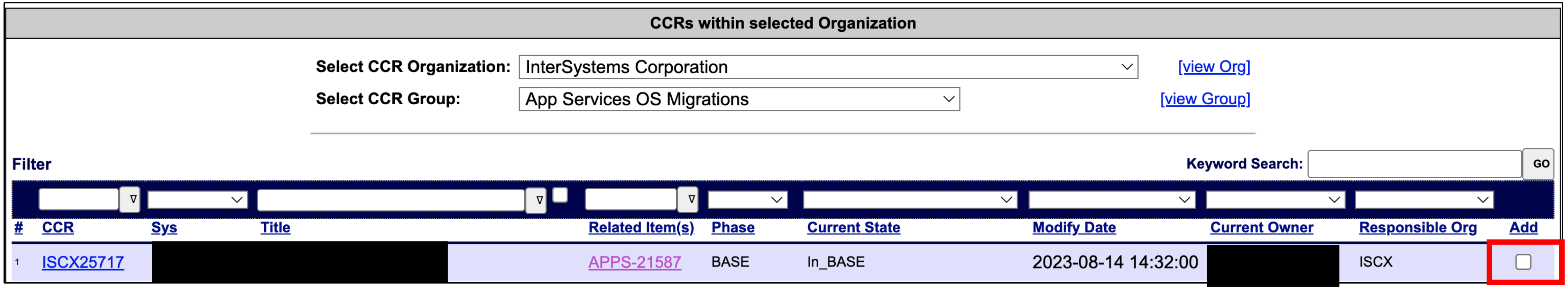
Current State	In_BASE	**Phase**	BASE
Organization	Best Health (BEST)	**System**	Interoperability 2020 (INTEROP2020)
Open Date	28-Jun-2023 09:11:06 AM	**Opened By**	Sam Schafer
Responsible Org	InterSystems (ISCX)	**Owner**	Sam Schafer



JIRA(s)	**Tech Help(s)**
Prerequisite CCR(s)	**Related CCR(s)** BEST0005
Type - Subtype * Application Customization - InterSystems IRIS	**Group**

Assigning CCRs to a Group: From Groups Page

1. Menu > Groups.
2. Select desired group from drop-down menu.
3. Click add Multiple CCRs button.
4. Select checkboxes in column named Add.
5. At bottom, click Add to Group button.



CCRs within selected Organization

Select CCR Organization: InterSystems Corporation [view Org]

Select CCR Group: App Services OS Migrations [view Group]

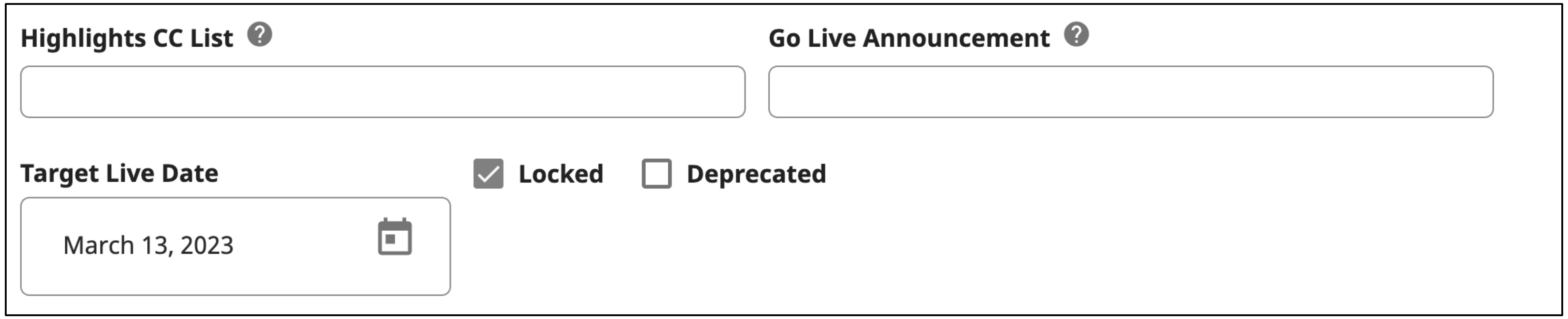
Filter

Keyword Search: GO

#	CCR	Sys	Title	Related Item(s)	Phase	Current State	Modify Date	Current Owner	Responsible Org	Add
1	ISCX25717	[redacted]	[redacted]	APPS-21587	BASE	In_BASE	2023-08-14 14:32:00	[redacted]	ISCX	□

Maintaining Groups

- Locked.
  - Prevents CCRs from being added to or removed from group.
- Deprecated.
  - Removes group from menus by default.
- Target Live Date.
  - For tracking purposes only.



Highlights CC List

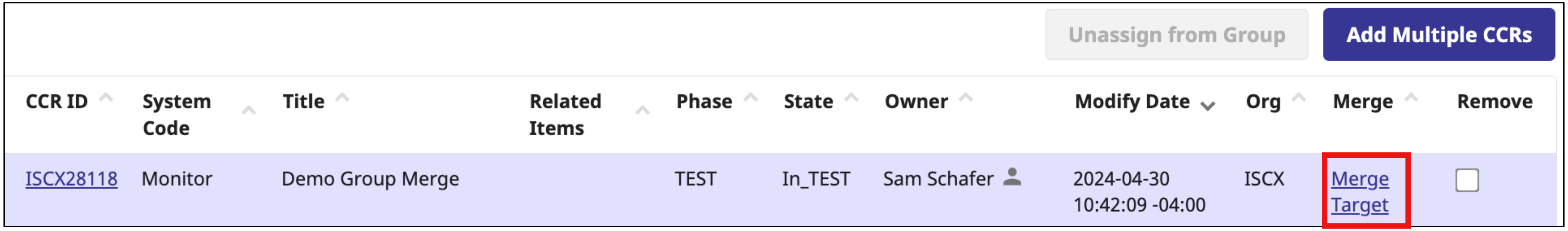
Go Live Announcement

Target Live Date: March 13, 2023

☑ Locked ☐ Deprecated

Group Merge

- 2+ CCRs in same state and for same system can be merged.
- Merges cannot be undone.
  - All merged changes must be progressed together; all or none.
- Merge Target = CCR that all other CCRs will be combined into.
  - Documentation appended, Perforce changelists moved over.
  - Merge Target then progressed through rest of workflow.
  - Merged CCRs enter inactive Merged state.



CCR ID	System Code	Title	Related Items	Phase	State	Owner	Modify Date	Org	Merge	Remove
ISCX28118	Monitor	Demo Group Merge		TEST	In_TEST	Sam Schafer	2024-04-30 10:42:09 -04:00	ISCX	Merge Target	□

Quiz: Finalizing Groups

Question:

Once a group contains all appropriate CCRs and a project manager wants to prevent that list from being changed, what should be done to the group?

A. Freeze the group.

B. Deprecate the group.

C. Lock the group.

D. Set the ‘Target Live Date.’

Quiz: Finalizing Groups (cont.)

Answer:

C. Lock the group.

Once a group has been locked on the Group Details page, then the group field is read-only for those CCRs already in the group, and the group name is not present in the group dropdown for CCRs not in that group.

Quiz: Legacy Groups

Question:

Once all CCRs in a group have gone LIVE and there is no longer a need to use that group or report on it, what final step should be taken?

A. N/A – Legacy groups clean up themselves.

B. Deprecate the group.

C. Lock the group.

D. Set the ‘Target Live Date.’

Quiz: Legacy Groups

Answer:

B. Deprecate the group.

Deprecating a group will remove it from the list on the Group Details page and in the group selection dropdown shown when editing a CCR.

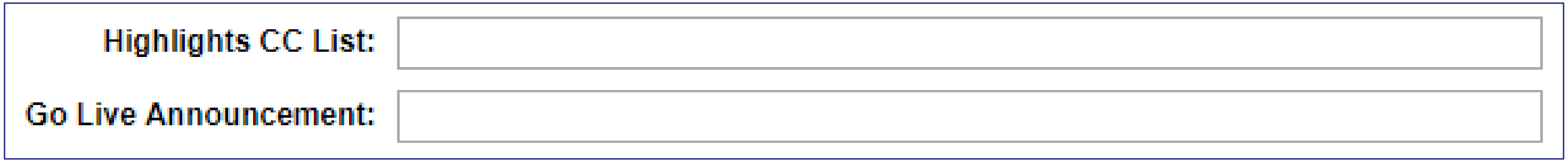
Part 2: CCR Email Notifications

Email Notifications Types

- CC List.
  - Email sent for every transition.
- Highlights CC List.
  - Emails sent upon entering: In_PREP, BASE_Complete, TEST_Ready_For_Customer, UAT_Ready_For_Customer, LIVE_Complete, Closed, Pending_Customer_Override, Cancelled, Merged.
- Go Live Announcement.
  - Email sent upon entering LIVE_Complete.
    - Contains CCR title, description and notes.

Email Notification Types Available

- Not all types available at all levels.
- Organization wide.
  - Highlights CC.
- System wide.
  - Highlights CC.
    - Architects automatically subscribed.
  - Go LIVE Announcement.



Highlights CC List:	
Go Live Announcement:	

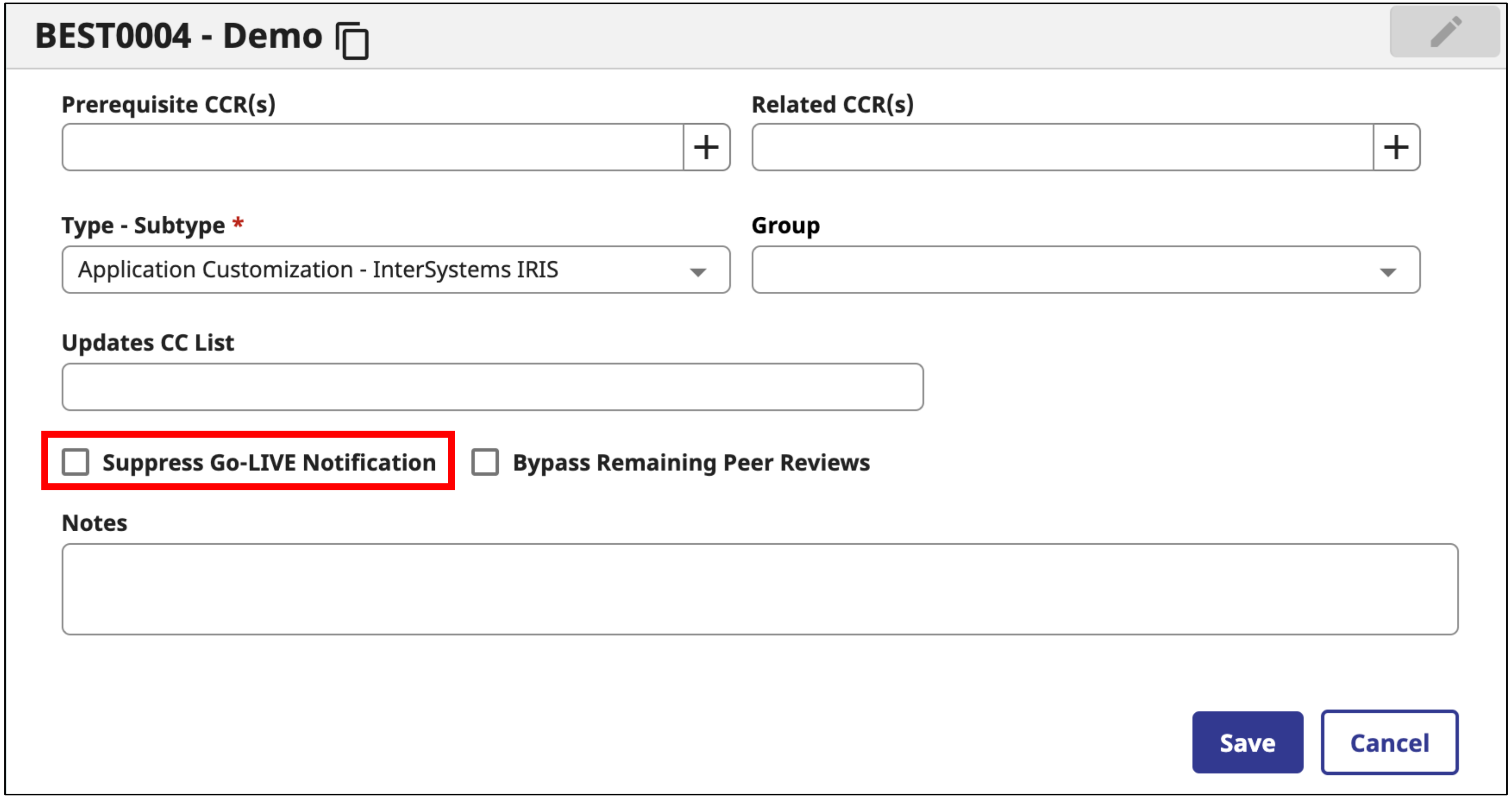
Email Notification Types Available (cont.)

- Group.
  - Highlights CC.
  - Go LIVE Announcement.
- Each CCR.
  - CC list.

Suppress Go-LIVE Notifications

- Some changes do not warrant go-LIVE announcements.
  - Renewing license key.
  - Updating unit tests.
  - Backend maintenance.
- Each CCR has flag to prevent sending go-LIVE announcement.
  - Edit CCR Details section > select Suppress Go-LIVE Notification > Save.

Suppress Go-LIVE Notifications (cont.)



BEST0004 - Demo

Prerequisite CCR(s)

Related CCR(s)

Type - Subtype *: Application Customization - InterSystems IRIS

Group:

Updates CC List:

□ Suppress Go-LIVE Notification

□ Bypass Remaining Peer Reviews

Notes:

Quiz: Email Notification Option

Question:
What should a user do if they are only interested in knowing when new changes affect all the LIVE Environments for a given organization?

A. Add themselves to the ‘Highlight CC’ email field for the org.
B. Add themselves to the ‘Highlight CC’ email field for each System in that org.
C. Add themselves to the ‘Go Live Announcement’ email field for each group in that org.
D. Add themselves to the ‘Go Live Announcement’ email field for each System in that org.

Quiz: Email Notification Option

Answer:

D. Add themselves to the ‘Go Live Announcement’ email field for each System in that org.

‘Highlight CC’ will send emails at earlier stages and will generate extra emails (so A. and B. won’t work); CCRs are not required to be in a group (so D. won’t work).

Quiz: Go-LIVE Notifications

Question:

On which transition of a CCR are go-LIVE notifications sent if suppress go-LIVE notifications has not been selected?

A. markLIVEComplete.

B. markMoveToLIVEComplete.

C. close.

D. authorizeAndStartMoveToLIVE.

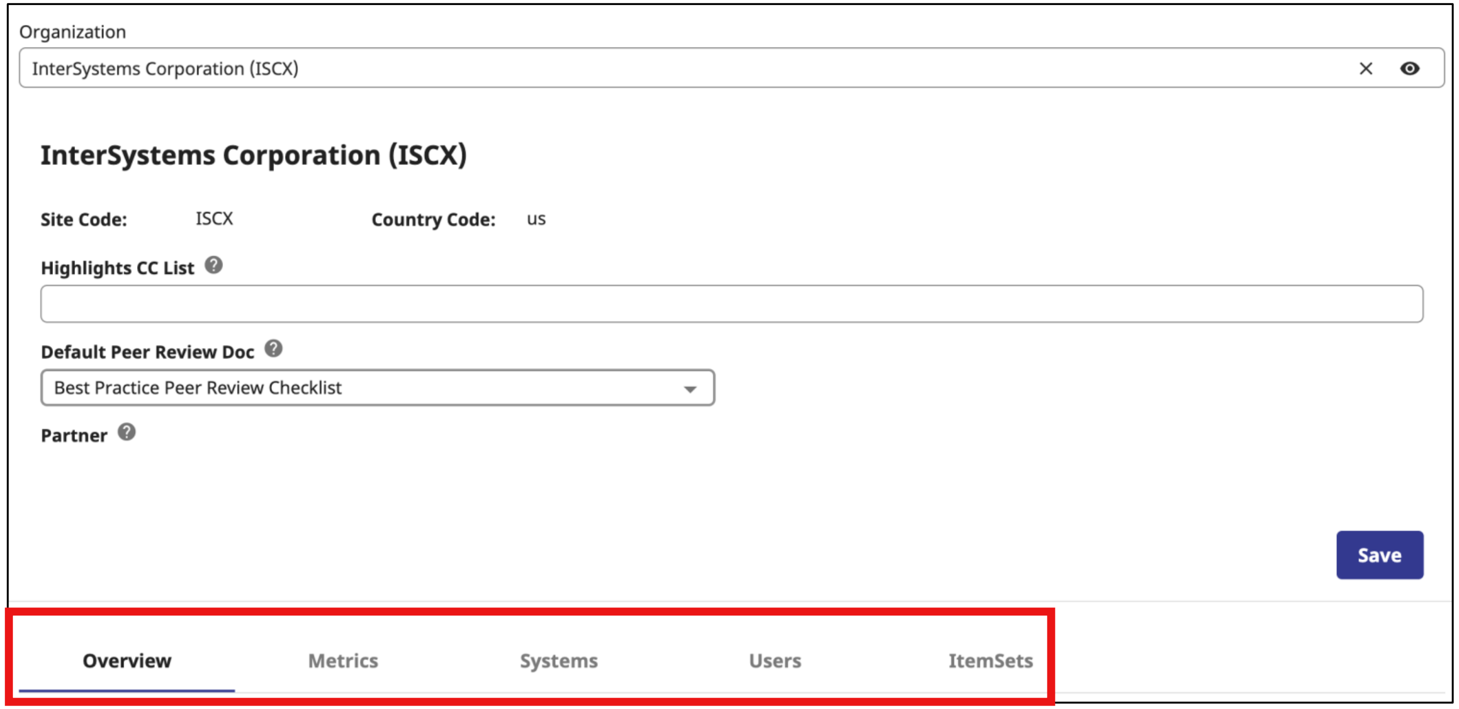
Answer:

A. markLIVEComplete.

Part 3: Organization Dashboard

Organization Dashboard

- Menu > Organizations.
- Shows key metrics and warnings about CCR usage.



Organization

InterSystems Corporation (ISCX)

InterSystems Corporation (ISCX)

Site Code: ISCX **Country Code:** us

Highlights CC List

Default Peer Review Doc

Best Practice Peer Review Checklist

Partner

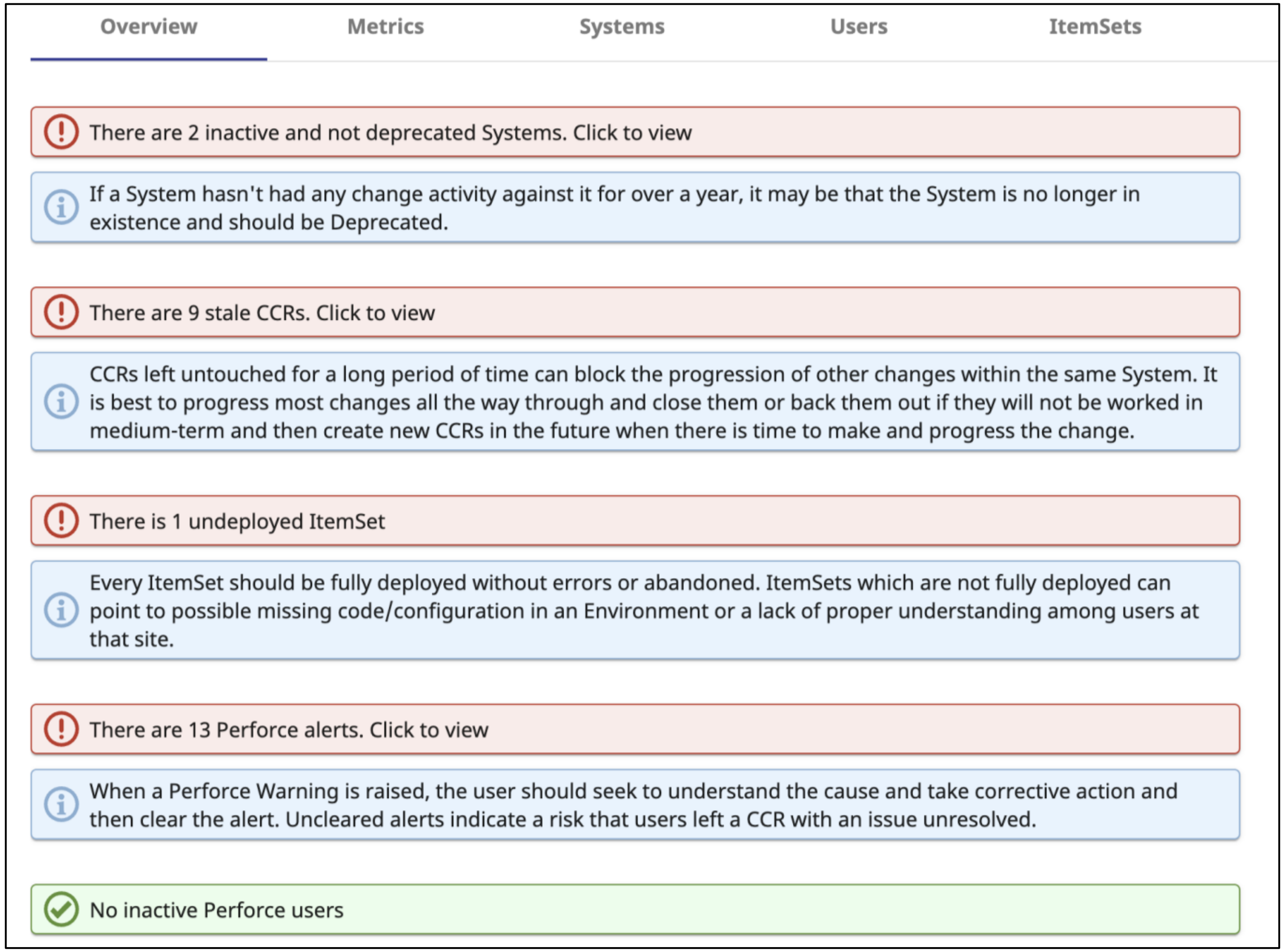
Organization Dashboard: Overview Tab

- Shows warnings for 5 key metrics.
  - Ideal value for each is zero.
- Inactive and not deprecated Systems.
  - Indicates candidate Systems for deprecation.
    - Only deprecate Systems for applications that no longer exist.
- Stale CCRs.
  - Highlights CCRs that are open without recent activity.
  - Could indicate CCRs abandoned and not properly cancelled, leading to merge-conflicts in the future.

Organization Dashboard: Overview Tab (cont.)

- Undeployed ItemSets.
  - ItemSets should always be deployed immediately.
  - See ICC400 for high-risk nature of undeployed ItemSets.
- Perforce alerts.
  - Alerts should be understood, corrected, and cleared.
- Inactive Perforce users.
  - Only users regularly using CCR should have Perforce licenses.

Organization Dashboard: Overview Tab (cont.)



There are 2 inactive and not deprecated Systems. Click to view

If a System hasn't had any change activity against it for over a year, it may be that the System is no longer in existence and should be Deprecated.

There are 9 stale CCRs. Click to view

CCRs left untouched for a long period of time can block the progression of other changes within the same System. It is best to progress most changes all the way through and close them or back them out if they will not be worked in medium-term and then create new CCRs in the future when there is time to make and progress the change.

There is 1 undeployed ItemSet

Every ItemSet should be fully deployed without errors or abandoned. ItemSets which are not fully deployed can point to possible missing code/configuration in an Environment or a lack of proper understanding among users at that site.

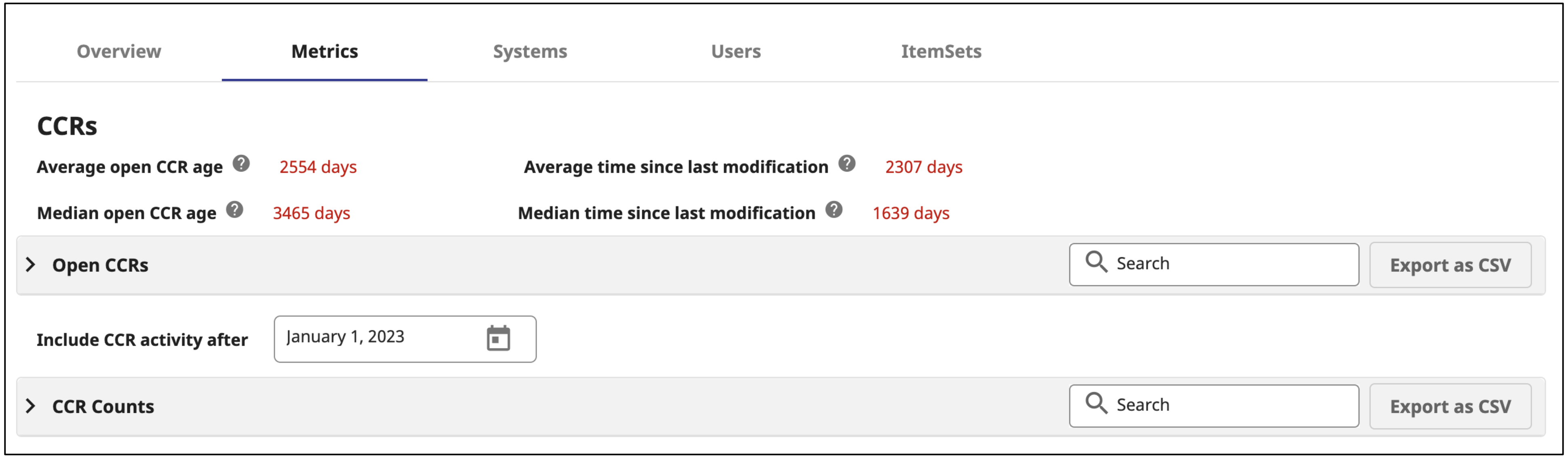
There are 13 Perforce alerts. Click to view

When a Perforce Warning is raised, the user should seek to understand the cause and take corrective action and then clear the alert. Uncleared alerts indicate a risk that users left a CCR with an issue unresolved.

✓ No inactive Perforce users

Organization Dashboard: Metrics Tab

- Shows more detailed metrics on CCR age and time since modification.
- Lists:
  - Open CCRs
  - CCR counts per System.



CCRs

Average open CCR age 2554 days

Average time since last modification 2307 days

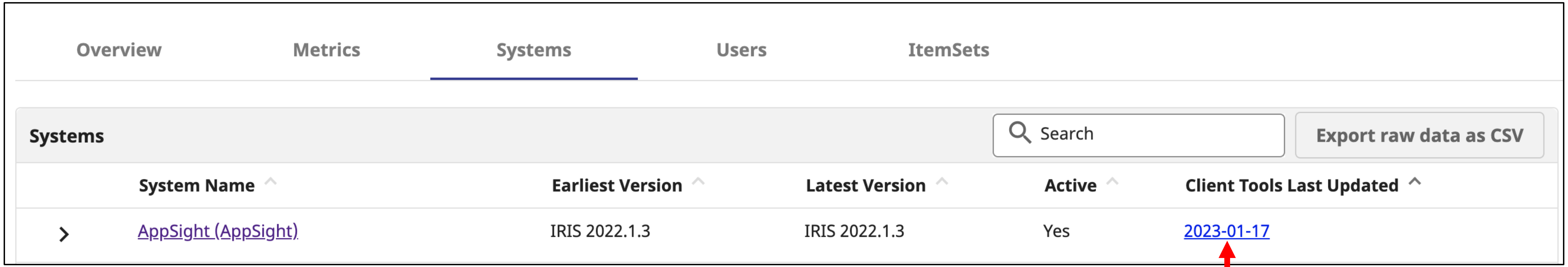
Median open CCR age 3465 days

Median time since last modification 1639 days

Include CCR activity after January 1, 2023

Organization Dashboard: Systems Tab

- Shows key information for each System.
  - Including last time Client Tools updated.



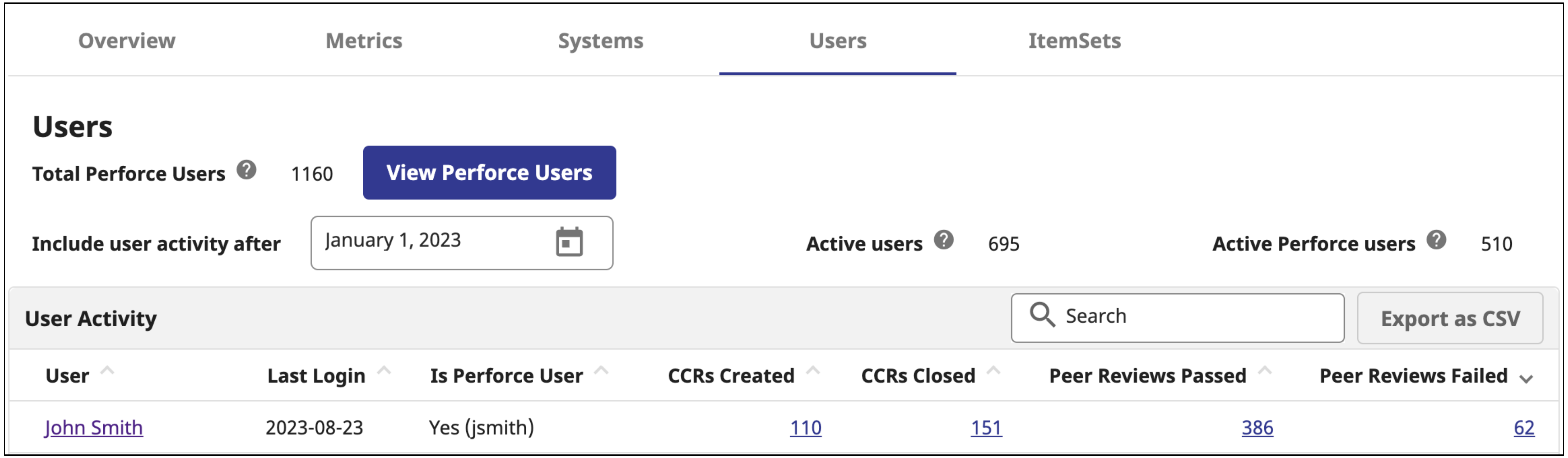
Systems

	System Name	Earliest Version	Latest Version	Active	Client Tools Last Updated
›	AppSight (AppSight)	IRIS 2022.1.3	IRIS 2022.1.3	Yes	2023-01-17

InterSystems employees can click to get list of Client Tool changes since last update.

Organization Dashboard: Users Tab

- Shows metrics per user.
  - Peer Reviews Passed / Failed = # of times that user performed passPeerReview or failPeerReview.
    - High number of passed with low number of failed indicates potentially poor peer reviews.



Users

Total Perforce Users ? 1160 View Perforce Users

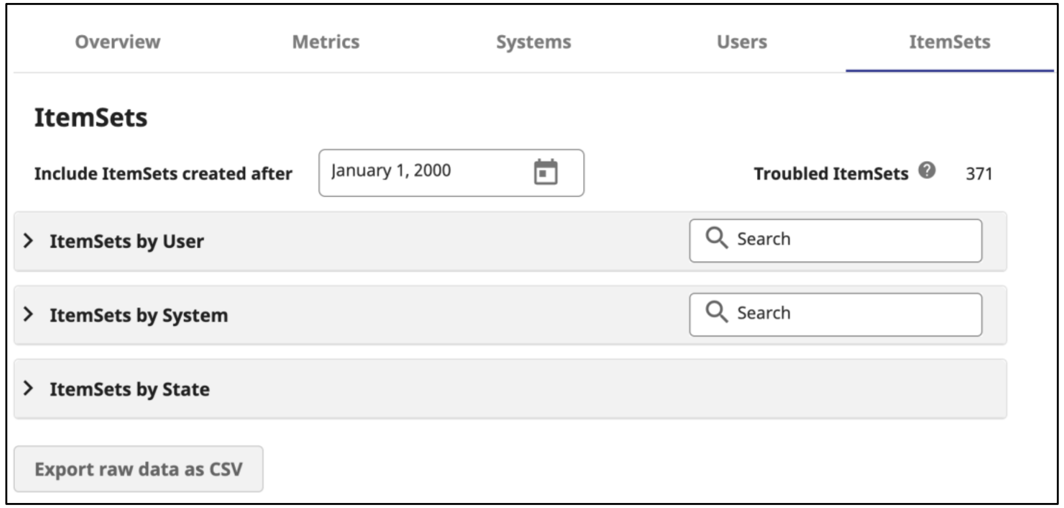
Include user activity after January 1, 2023 Active users ? 695 Active Perforce users ? 510

User Activity

User	Last Login	Is Perforce User	CCRs Created	CCRs Closed	Peer Reviews Passed	Peer Reviews Failed
John Smith	2023-08-23	Yes (jsmith)	110	151	386	62

Organization Dashboard: ItemSets Tab

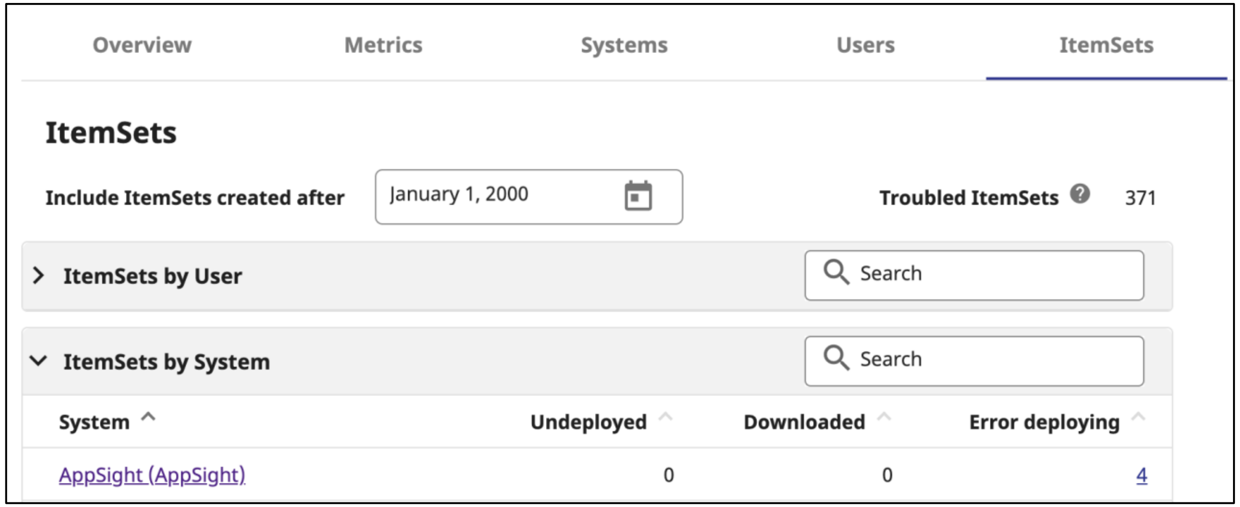
- Shows count of ItemSets in states:
  - Undeployed.
  - Downloaded.
  - Error deploying.
- Click number to view affected CCRs.



ItemSets

Include ItemSets created after January 1, 2000

Troubled ItemSets 371



ItemSets

Include ItemSets created after January 1, 2000

Troubled ItemSets 371

System	Undeployed	Downloaded	Error deploying
AppSight (AppSight)	0	0	4

Part 4: CCR Reports

CCR Reports

- Downloadable reports available on CCR reports page.
  - Menu > CCR Reports.
  - Formats include HTML, PDF, CSV.

CCR Reports (cont.)

- Reports available:
  - Active CCRs by Organization.
    - All currently active CCRs for organization.
  - Status Report By Organization.
    - CCRs that have gone LIVE or are scheduled to go LIVE during a specified date range, as well as all other active CCRs and cancelled CCRs, regardless of date range.
  - Documentation Review By User.
    - Report of documentation fields for CCRs modified in date range for each user in specified Responsible Organization.

CCRs in Status Report By Organization by Section

- Date From and To fields behave differently in each section of report.
- CCRs Moved to LIVE since Date From.
  - LIVEMoveTimeTarget field between Date To and Date From fields
  - Current phase is LIVE.
- CCRs schedule for move to LIVE between Date From and Date To.
  - LIVEMoveTimeTarget field between Date To and Date From fields.
  - Current phase is UAT.

CCRs in Status Report By Organization by Section (cont.)

- CCRs in UAT not yet scheduled for move to LIVE.
  - LIVEMoveTimeTarget not specified.
  - Current phase is UAT.
- All other sections do not use date fields.

Quiz: CCR Reports

Question:

Which report highlights CCRs scheduled for future go-live?

A. Active CCRs by organization.

B. Status Report by organization.

C. Documentation Review by user.

Answer:

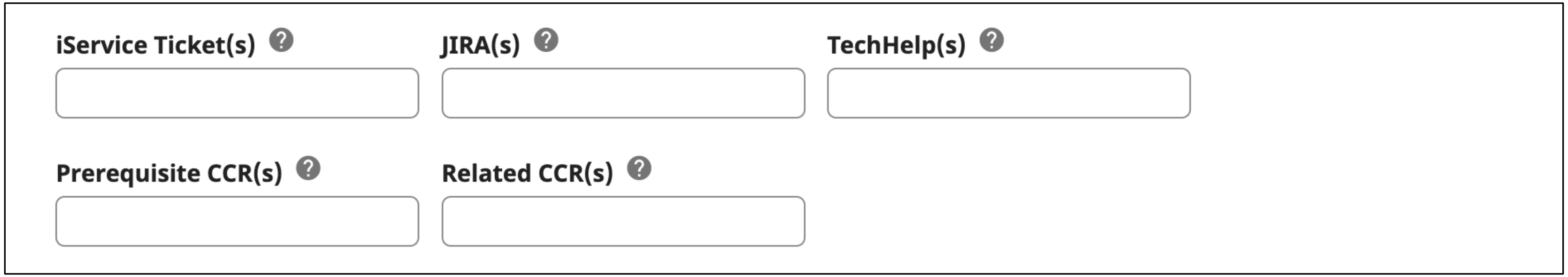
B. Status Report by organization.

This will list a section for CCRs scheduled to go LIVE in the future. You can specify the date range in which you would like to limit your search.

Part 5: Related Records

Linking CCRs to JIRA

- JIRA field on CCR defaults to InterSystems JIRA server.
  - Contact InterSystems to link to non-InterSystems JIRA server.
    - Can override at organization or per-System level.



iService Ticket(s) ?

JIRA(s) ?

TechHelp(s) ?

Prerequisite CCR(s) ?

Related CCR(s) ?

Summary

- What are the key points for this module?

ICC 350:
CCR Tier 0 –
Standard Changes

InterSystems

Creative data technology

Objectives

- Explain what standard changes are and how they differ from normal changes under ITIL.
- Demonstrate how to configure a standard change in CCR.
- Explain the properties of standard changes in CCR.

Part 1: Standard Changes

Standard Changes

- Under ITIL, a standard change is one that is:
  - Well-known, documented and proven.
  - Generally simple and recurrent.
  - Low risk.
  - Previously authorized.

Standard Changes (cont.)

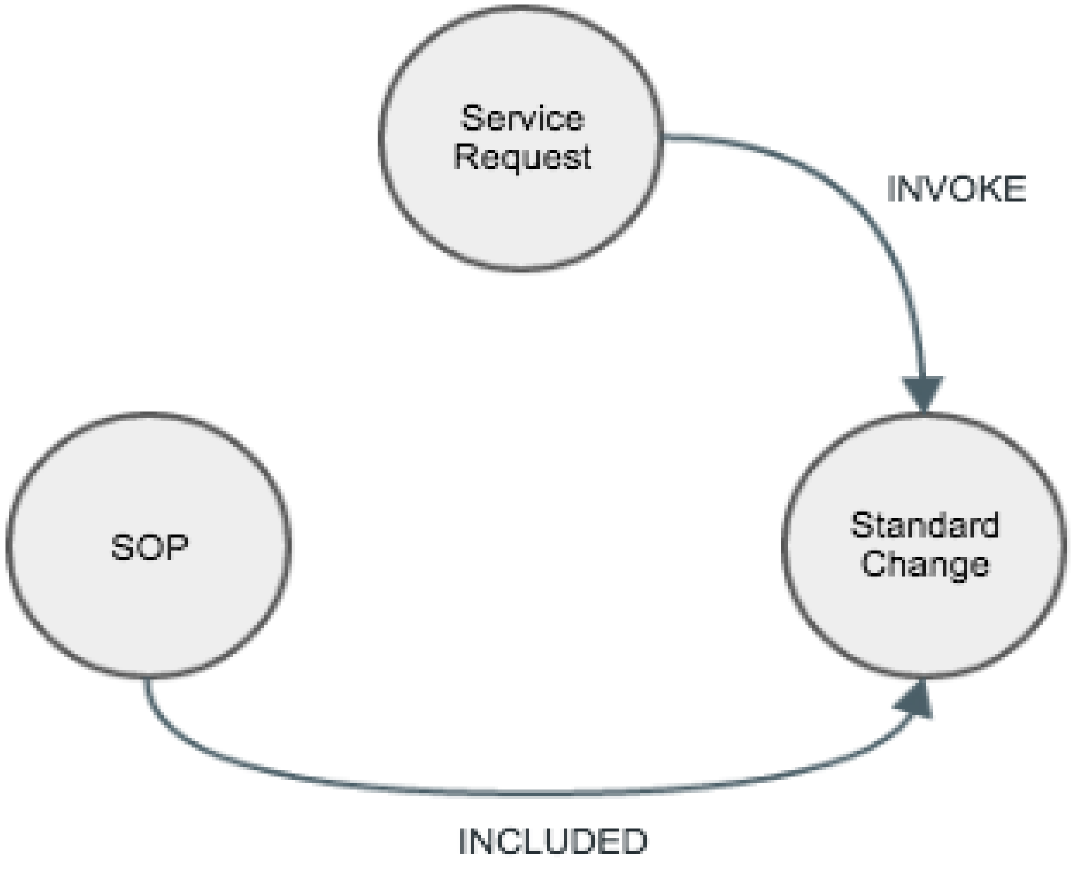
- Normal change control process unnecessary overhead for standard changes.
  - Causes organizations to not control standard changes.
- Use different control process for standard changes to ensure:
  - Documented / proven implementation, testing, and backout plans.
  - Documentation every time standard change performed.
    - Assists troubleshooting during crises.
  - Minimal change control overhead.

Management of Standard Changes

- Standard changes are often associated with service requests.
- There is typically a catalog of standard changes.
- Standard changes have to respect specific criteria.
- Standard changes have to be approved and regularly reviewed.
  - Particularly data fixes.
- Standard changes may be demoted to normal changes.
- Use of standard changes should be recorded.

Relationship between SR, SC and SOP

1. User issue a Service Request (SR).
2. Standard change is invoked (if present).
3. Standard Operating Procedure (SOP) is followed to implement the standard change.



Standard Change Approval

- Customers and InterSystems employees can approve standard changes.
  - This can vary based on the owner of the configuration item.
  - Approval must have expiration date.
- Example: for hosted sites InterSystems would create and approve most standard changes:
  - Restart specific services.
  - Purge log files.
  - Collect performance statistic information.

Quiz: Identifying Standard Changes

Question:

Which of the following can be considered a standard change? Select all that apply.

A. Purge log files.

B. Development change.

C. Synchronize mirror members.

D. New data fix.

E. Change to Site TrakCare Components.

F. Stop and Start HealthShare Production.

Quiz: Identifying Standard Changes (cont.)

Answer:

A. Purge log files.

C. Synchronize mirror members.

F. Stop and Start HealthShare Production.

low-risk, regularly occurring, well understood, repeatable

Part 2: Standard Changes within CCR

Standard Changes within CCR

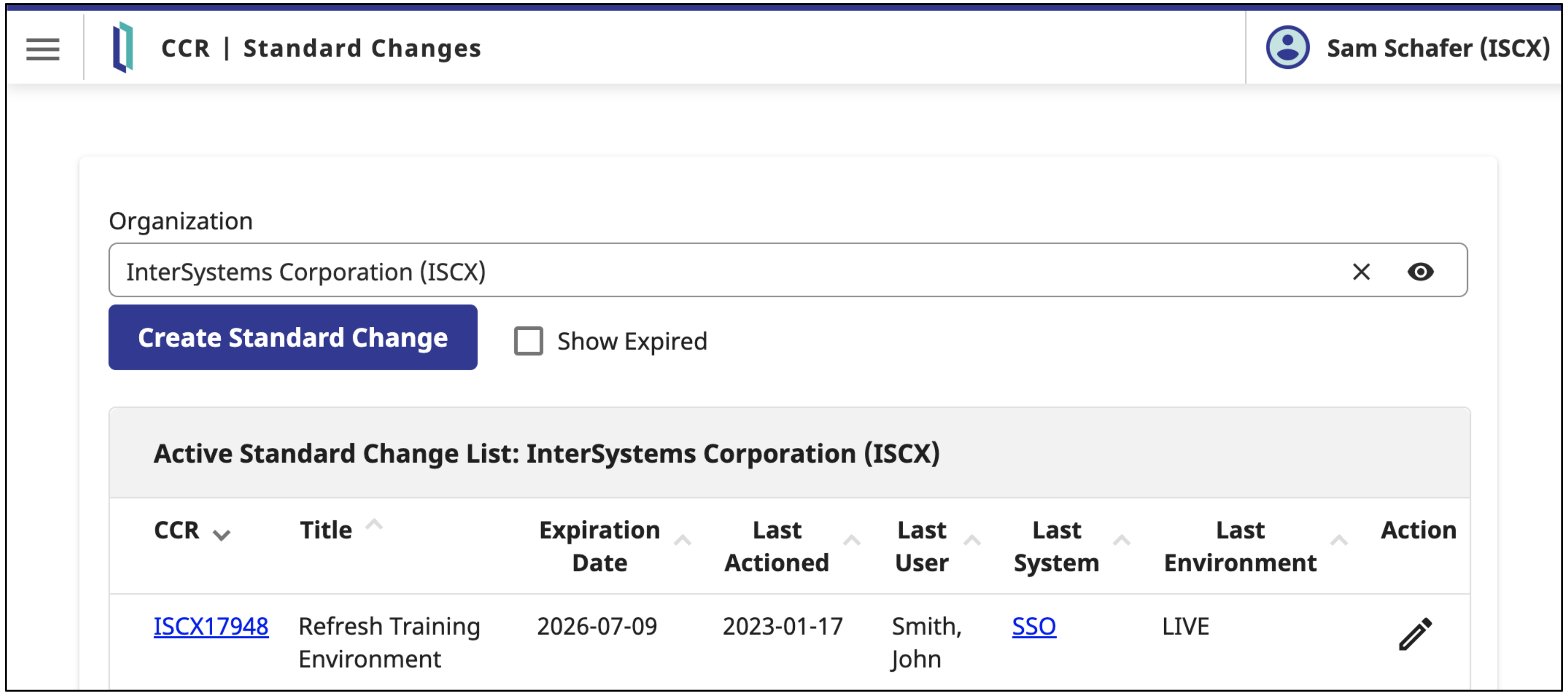
- CCR supports documenting standard changes.
- Can promote a CCR to be a standard change.
  - Documentation within that CCR then used as SOP for implementing change again.
  - CCR becomes read-only.
- To become a standard change, a CCR must be:
  - Tier 0.
  - Closed.

Standard Changes within CCR (cont.)

- Can action and view history of standard changes.
- Actioning standard change:
  - Does not require new CCR
    - It has already been completely specified.
  - Does not require additional approval
    - It has already been pre-authorized.

Access List of Standard Changes

- Menu > Standard Changes.
- Click CCR ID to view full CCR.



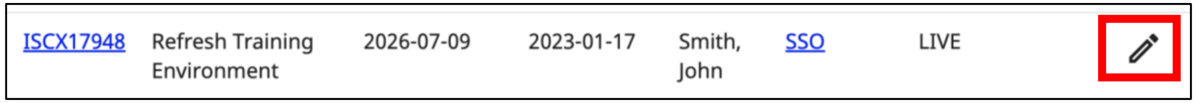
Organization: InterSystems Corporation (ISCX)

Active Standard Change List: InterSystems Corporation (ISCX)

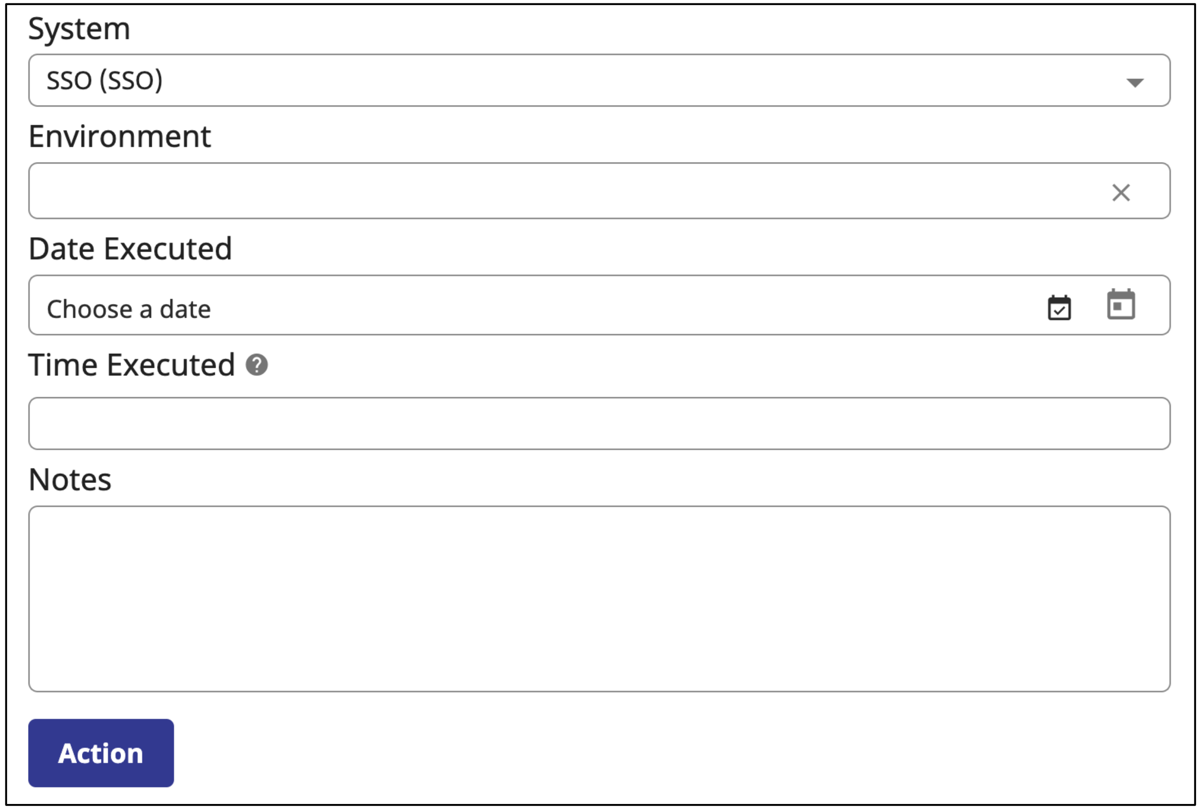
CCR	Title	Expiration Date	Last Actioned	Last User	Last System	Last Environment	Action
ISCX17948	Refresh Training Environment	2026-07-09	2023-01-17	Smith, John	SSO	LIVE	✎

Action Standard Change

1. Menu > Standard Changes.
2. Click pencil icon.
3. Specify Environment, Date, Time, and Notes (if applicable).
4. Click Action button.



ISCX17948	Refresh Training Environment	2026-07-09	2023-01-17	Smith, John	SSO	LIVE



System

SSO (SSO)

Environment

Date Executed

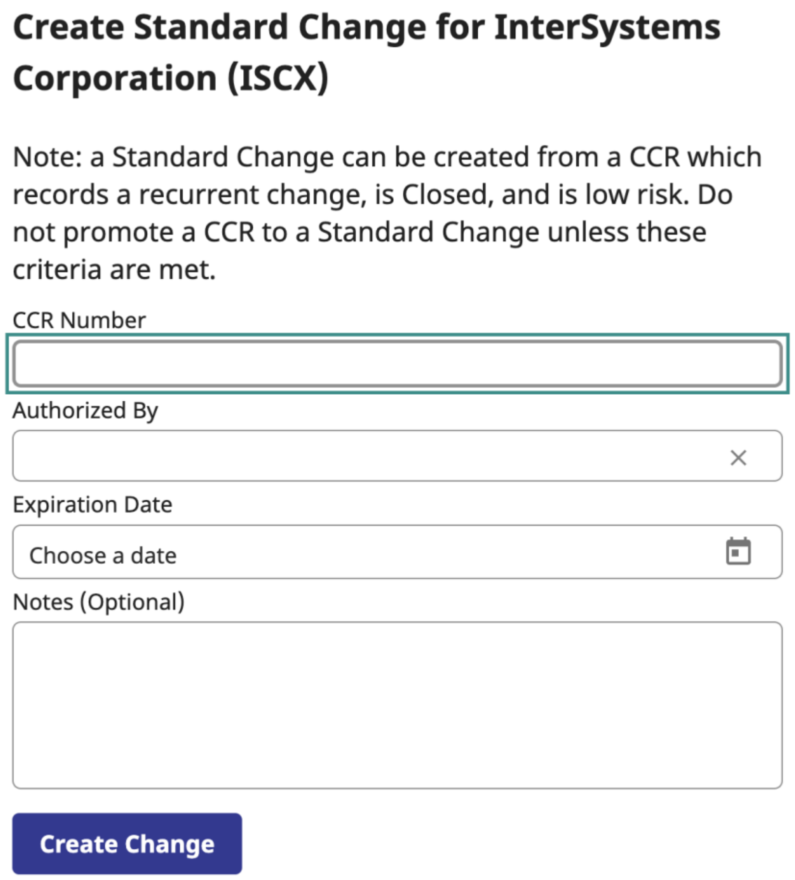
Choose a date

Time Executed

Notes

How To: Create New Standard Change

1. Menu > Standard Changes.
2. Click Create Standard Change button.
3. Complete form.
4. Click Create Change button.



Create Standard Change for InterSystems Corporation (ISCX)

Note: a Standard Change can be created from a CCR which records a recurrent change, is Closed, and is low risk. Do not promote a CCR to a Standard Change unless these criteria are met.

CCR Number

Authorized By

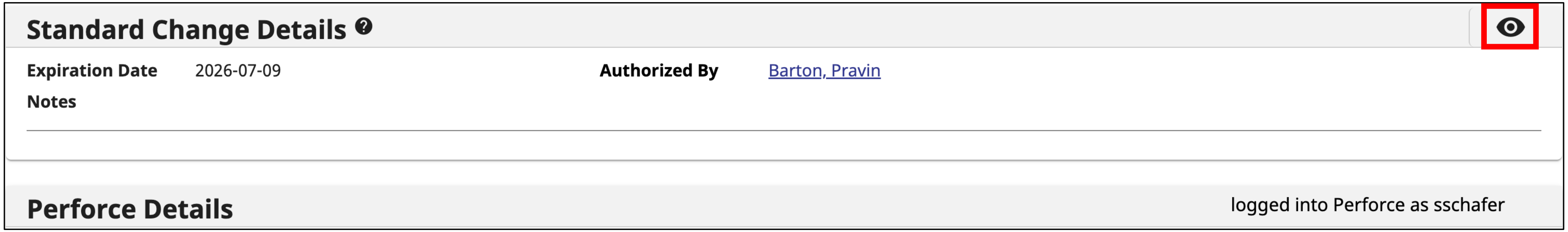
Expiration Date

Choose a date

Notes (Optional)

CCRs that are Standard Changes

- New Standard Changes Details above Perforce Details.
- Click eye to edit Expiration Date, Authorized By, or Notes.
- Rest of CCR read-only.



Standard Change Details

Expiration Date	2026-07-09	**Authorized By**	Barton, Pravin
Notes			

Perforce Details

logged into Perforce as sschafer

Standard Change Expiration Date

- Expired standard changes:
  - Cannot be actioned.
  - Not shown in list of standard changes by default.
- System alert displayed 1 week before expiration date.
- Edit Standard Change Details section to change expiration date.
  - Set date in future to re-authorize.
    - Usually review and re-authorize every 6-12 months.
  - Set yesterday to immediately revoke authorization.

Quiz: CCR Standard Changes

Question:

Which of the following is true about CCR standard changes? Select all that apply.

A. Standard changes are pre-authorized to be executed in a System.

B. Any closed Tier 0 CCR can be promoted to be a standard change.

C. The contents of a CCR become read-only when the CCR becomes a standard change.

Quiz: CCR Standard Changes

Answer:

All of the above.

Summary

- What are the key points for this module?

ICC 400: Introduction to CCR Transport

InterSystems

Creative data technology

Objectives

- Understand benefits of CCR Transport including when and how to use it properly.
- Define CCR Client Tools and describe their purpose.
- Identify automatic vs manual parts of CCR Transport.

Objectives (cont.)

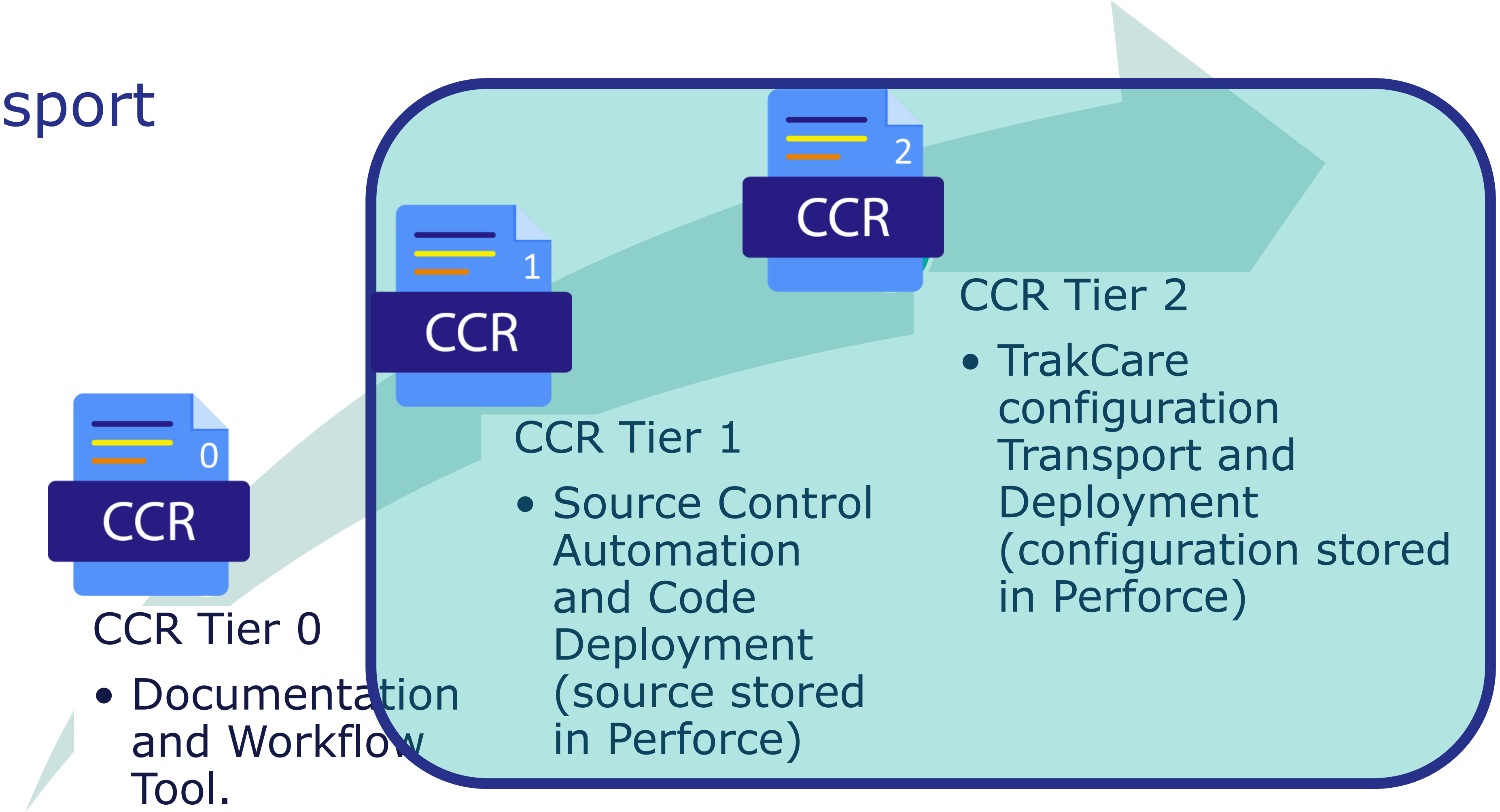
- Explain role and communication between:
  - Client environment.
  - CCR server.
  - Perforce Helix Server.
- Describe use of source workspace.

Part 1: CCR Transport:
Getting Items from Here to There

Review: CCR Usage Tiers

- CCR Transport

applies to:



CCR Tier 0

- Documentation and Workflow Tool.

CCR Tier 1

- Source Control Automation and Code Deployment (source stored in Perforce)

CCR Tier 2

- TrakCare configuration Transport and Deployment (configuration stored in Perforce)

CCR Transport: The Fundamentals

- Framework used to move items to or from environments.
- Only applies to Tier 1 and Tier 2 CCRs.
- Stores all items in central Perforce Helix Server.
- Depends on:
  - Namespace configured using Configure^%buildccr utility.
  - Outbound access to ccr.intersystems.com and login.intersystems.com on port 443.
    - CCR never requires direct inbound access.
  - Users having proper understanding of how to use CCR Transport.

CCR Transport: The Fundamentals (cont.)

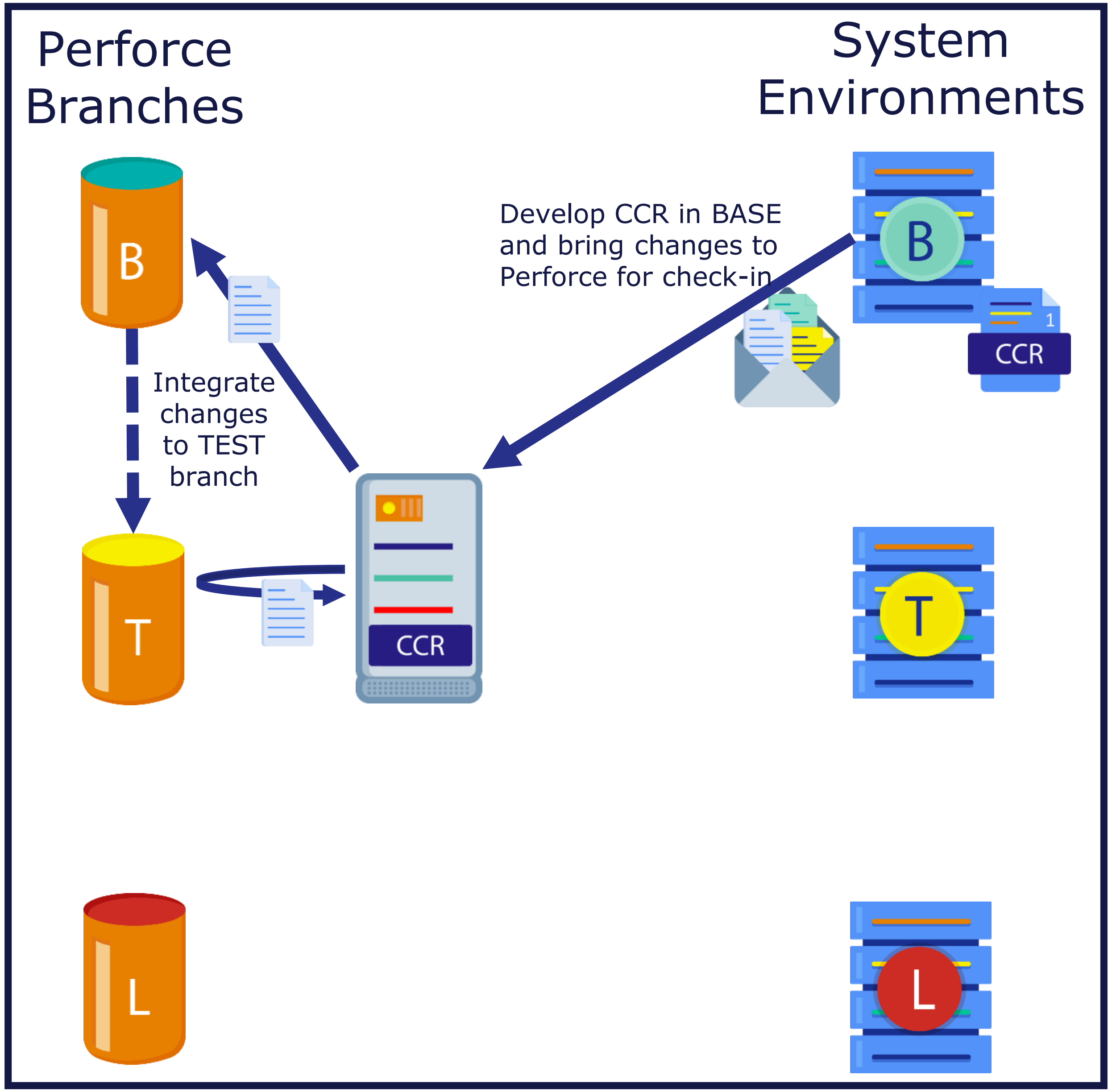
- Only InterSystems employees require direct access to Perforce for advanced debugging.

CCR Transport Terminology

- Item: Individual representation of part of an application.
  - Examples include individual:
    - Class definitions.
    - Routines.
    - Rows of a table.
- ItemSet: File that combines one or more Items along with metadata for transporting Items.
  - Think of it like an envelope for sending mail.
  - Only used for transport purposes, not persisting change.

CCR Transport Workflow

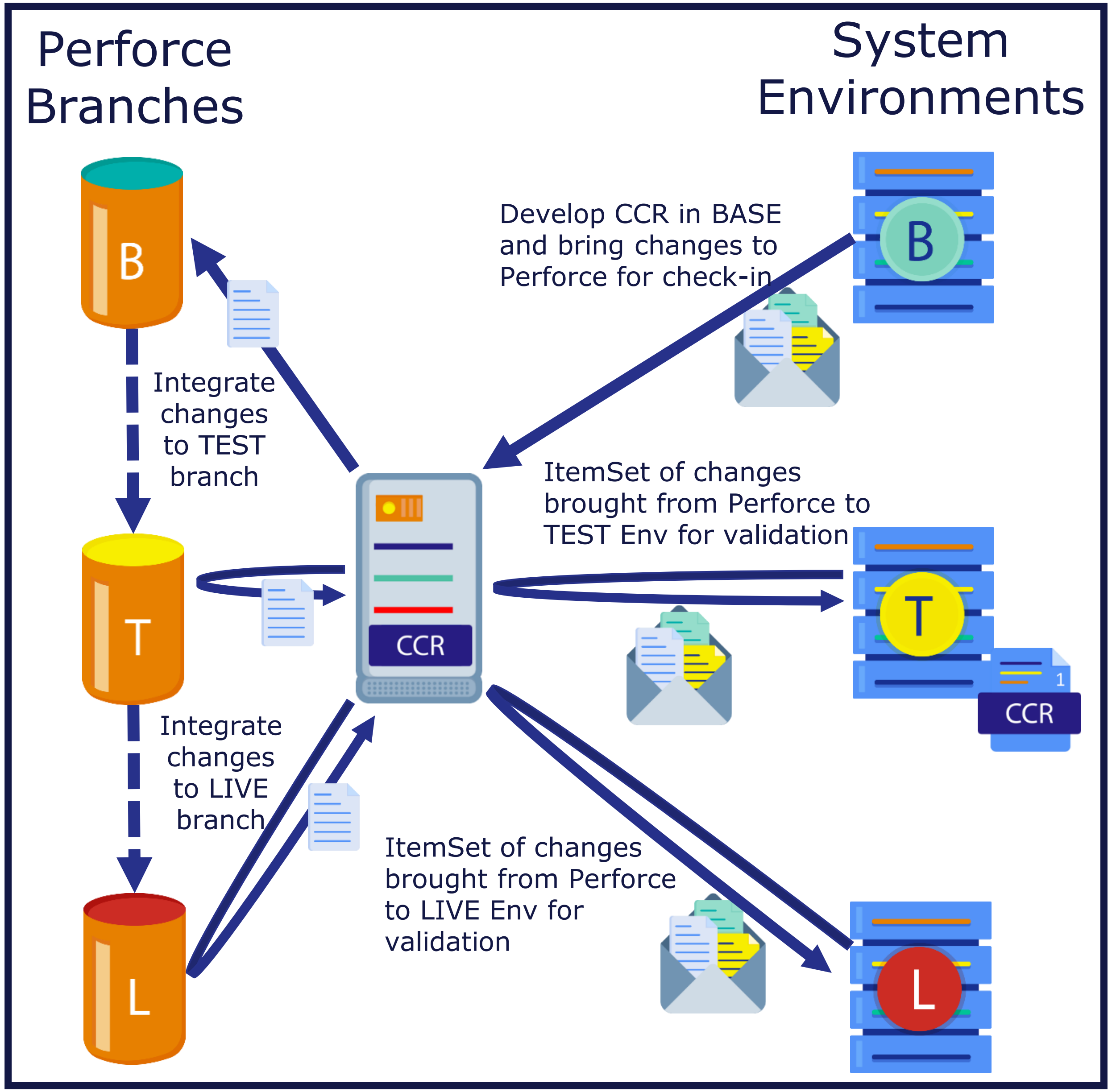
1. Create CCR to document change.
2. Make changes in BASE.
3. Send changes to Perforce via ItemSet.
4. CCR workflow triggers integration of changes from BASE to TEST branch in Perforce.



Perforce Branches
System Environments
B
T
L
Develop CCR in BASE and bring changes to Perforce for check-in
Integrate changes to TEST branch
CCR
B
CCR
T
L

CCR Transport Workflow

5. CCR delivers changes to TEST environment via ItemSet.
6. CCR workflow triggers integration of changes from TEST to LIVE branch in Perforce.
7. CCR delivers changes to LIVE environment via ItemSet.


Perforce Branches
System Environments
B
T
L
CCR
Develop CCR in BASE and bring changes to Perforce for check-in
Integrate changes to TEST branch
ItemSet of changes brought from Perforce to TEST Env for validation
Integrate changes to LIVE branch
ItemSet of changes brought from Perforce to LIVE Env for validation
B
T
L
1
CCR

Quiz: CCR Transport Delivery

Question:

What is the name of the container used to transport changes from the CCR server to an environment?

A. Changelist.

B. ItemSet.

C. DiffChunk.

D. Item.

E. SnailMail.

Quiz: CCR Transport Delivery (cont.)

Answer:

B. ItemSet.

An ItemSet is an XML file containing 1 or more Items along with metadata related to the CCR.

Part 2: CCR Client Tools

CCR Client Tools – An Introduction

- Library of code that comes with all InterSystems products.
  - 7 classes and 2 routines.
- Enables namespaces to act as clients to CCR.
- Stored in IRISLIB database.
  - Shared with all namespaces in instance.

CCR Client Tools – An Introduction (cont.)

- Can and should update without waiting for instance upgrade.
  - Low risk, easy, and mostly automated by progressing auto-generated Tier 1 CCR.
  - Updates include bug fixes and enhancements.
- Slides discussing new features in client tools refer to date added.
  - Environment has that feature if updated client tools since listed date.

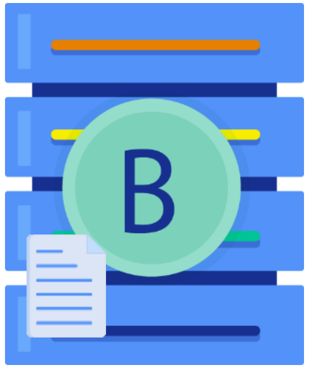
Part 3: CCR Transport Architecture

Source Workspace

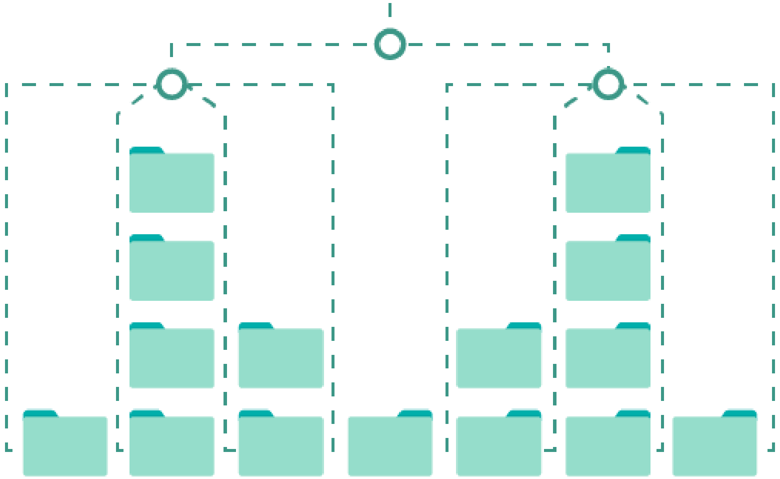
- Source workspace key component of CCR Transport functionality.
- Directory in filesystem of each environment.
- Folder structure under source workspace same as Perforce branch.
- Typically, files in source workspace updated by:
  - CCR Client Tools for Tier 1 changes.
  - TCC for Tier 2 changes.

Source Workspace

Database of
BASE instance



File system of
BASE machine

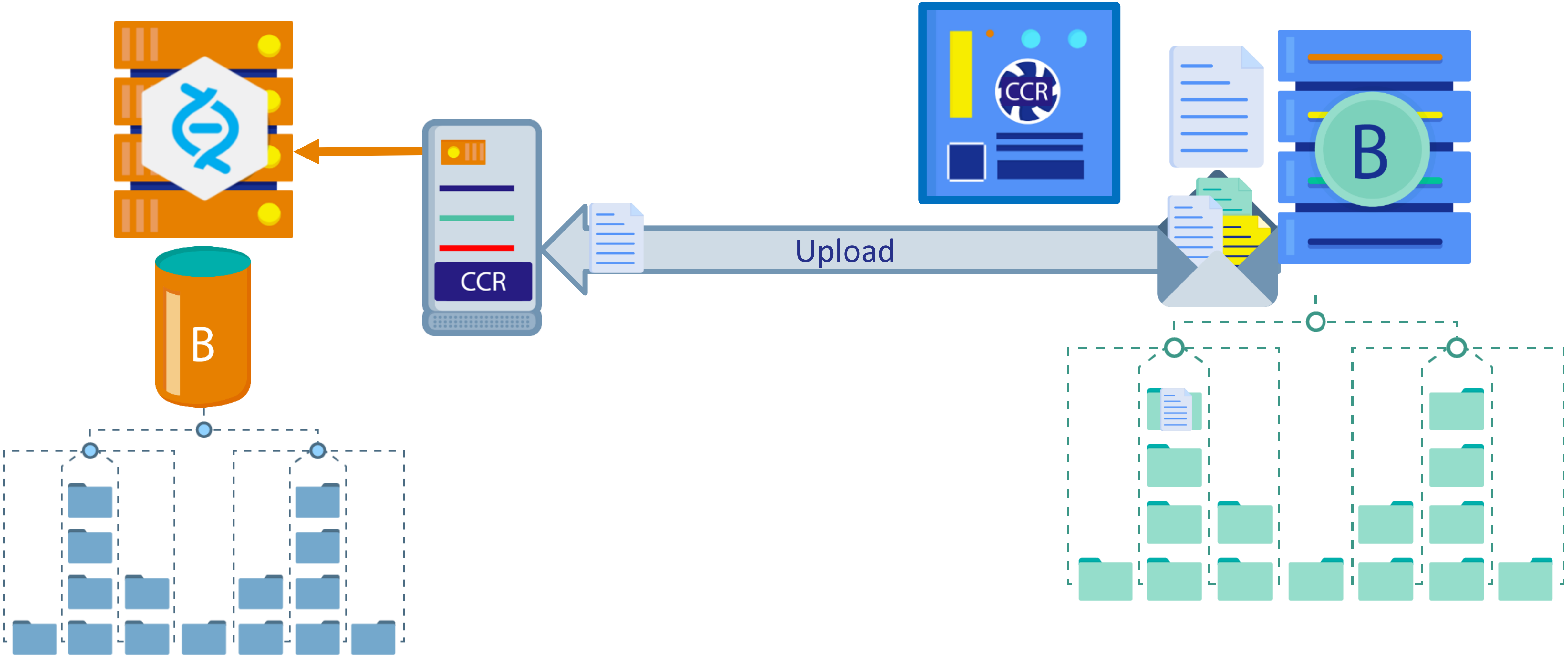


Connected vs Disconnected Mode

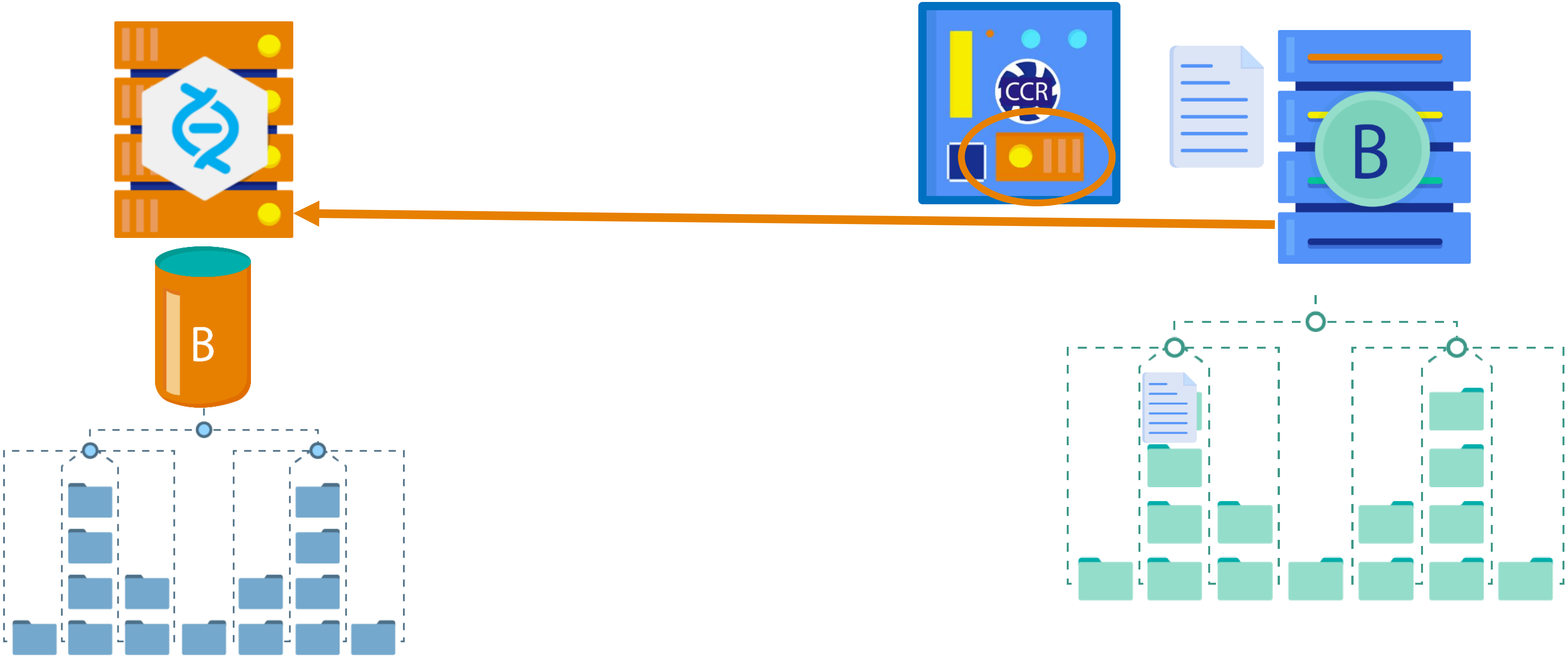
- Disconnected CCR Client.
  - BASE located outside of the InterSystems network.
  - CCR Client Tools use ItemSets to move source to/from Perforce.
  - TEST, UAT and LIVE are typically configured as Disconnected regardless of location.

- Connected CCR Client.
  - BASE located inside of the InterSystems network.
  - This allows the CCR Client Tools to communicate with the Perforce Helix Server Directly without using ItemSets.
    - Communication occurs via the p4 Client.
  - Typical configuration for InterSystems internal applications.

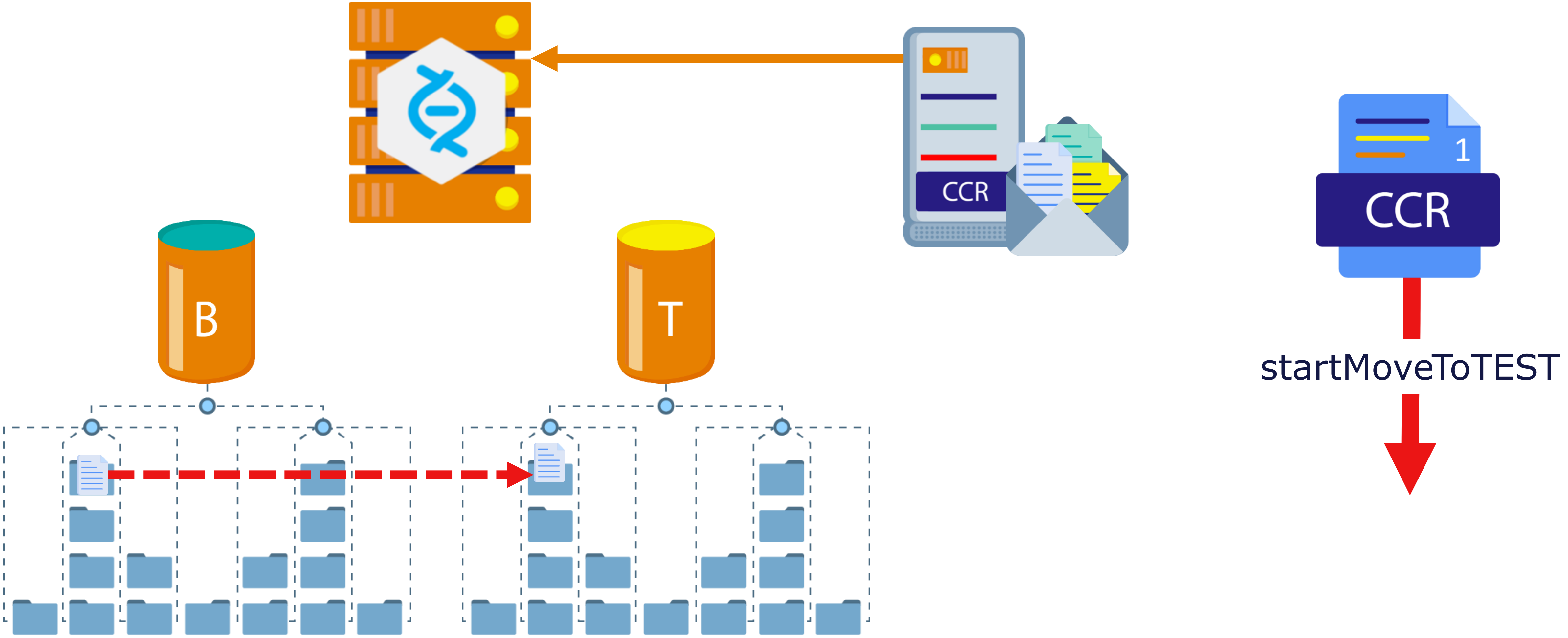
Transport Walkthrough: Sending Items from **Disconnected** BASE



Transport Walkthrough: Sending Items from **Connected** BASE

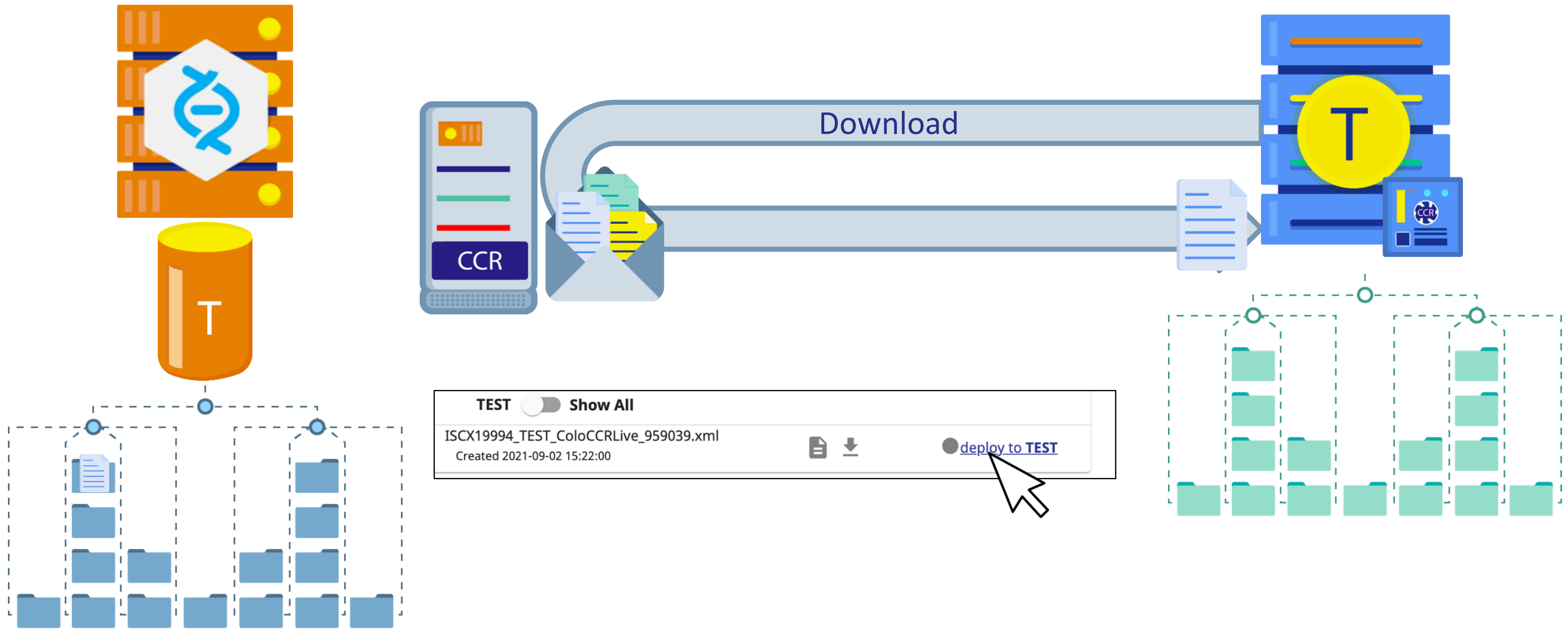


Transport Walkthrough: Integrating from BASE to TEST



startMoveToTEST

Transport Walkthrough: Sending Items To TEST

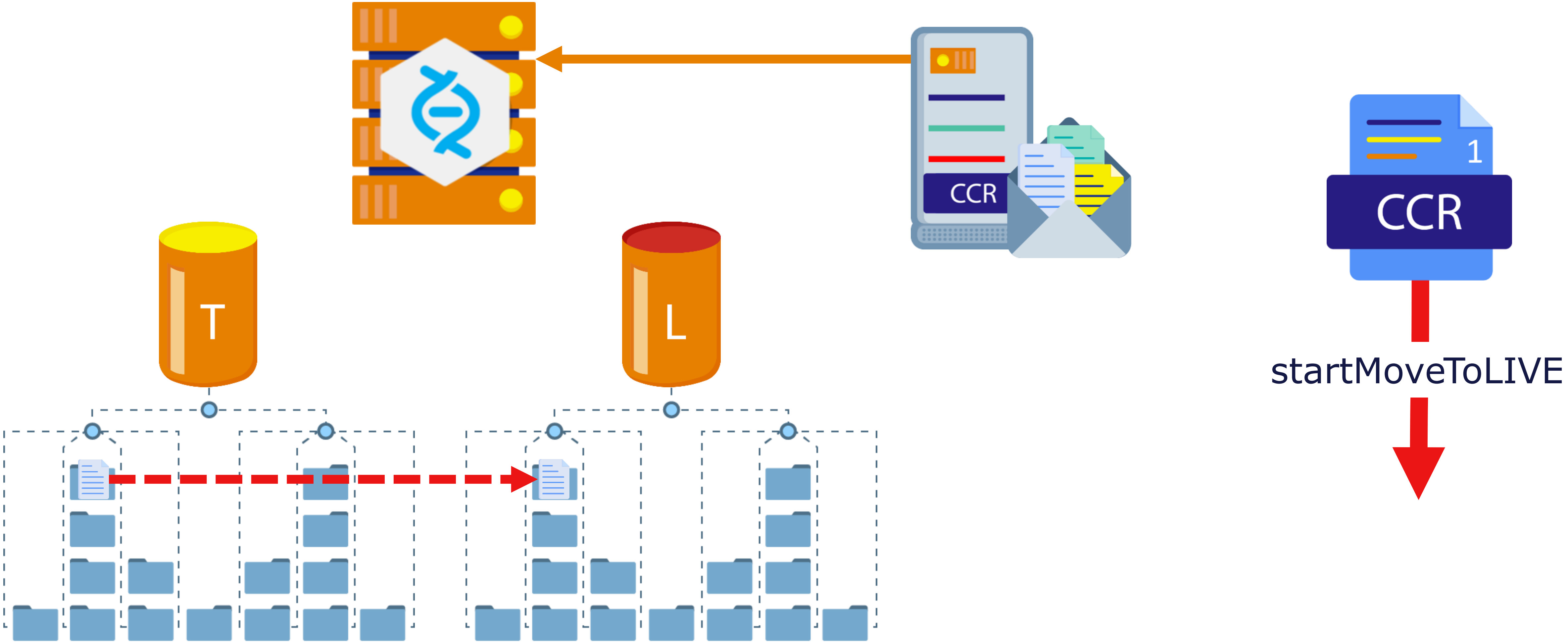


TEST ○ **Show All**

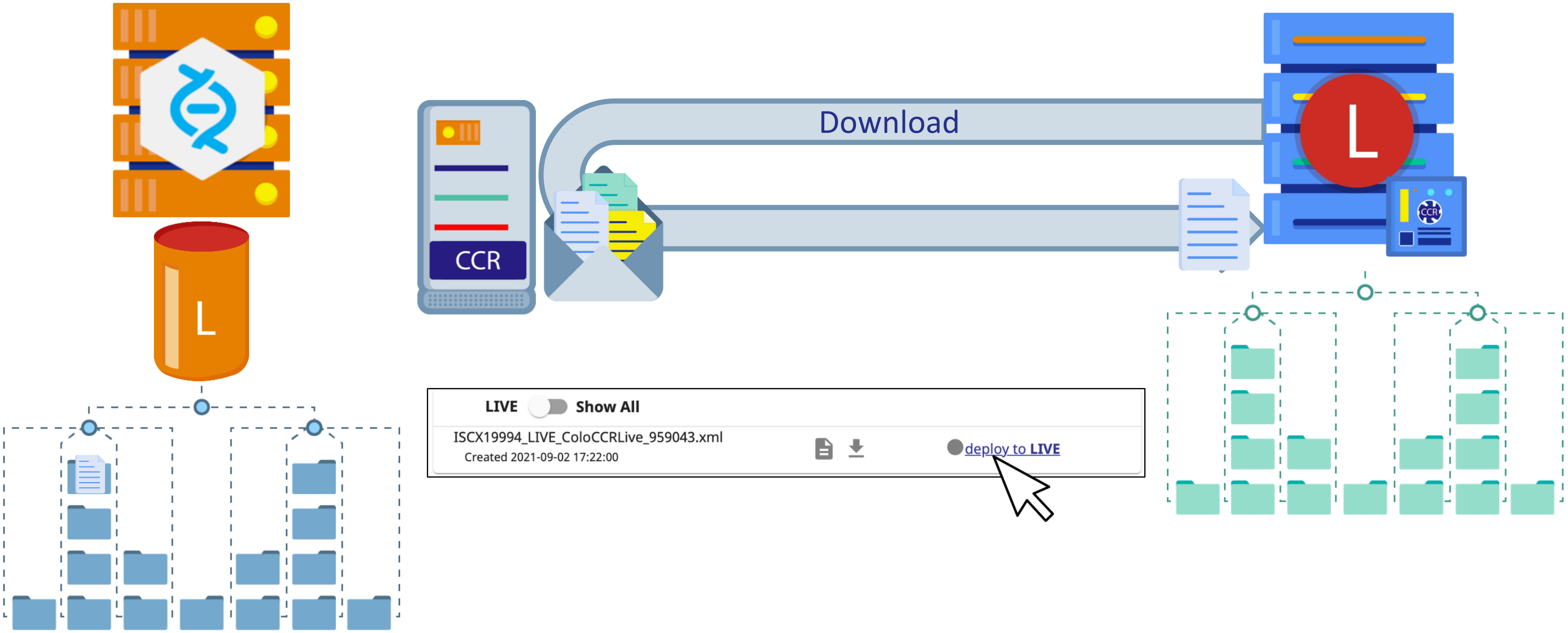
ISCX19994_TEST_ColoCCRLive_959039.xml
Created 2021-09-02 15:22:00

● deploy to **TEST**

Transport Walkthrough: Integrating from TEST to LIVE



Transport Walkthrough: Sending Items To LIVE



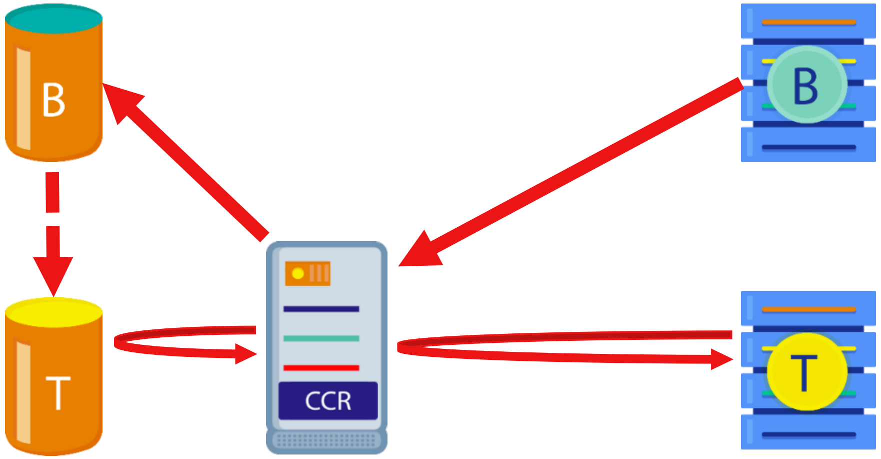
LIVE ○ Show All

ISCX19994_LIVE_ColoCCRLive_959043.xml
Created 2021-09-02 17:22:00

● deploy to **LIVE**

Danger: Changes to non-BASE Environments

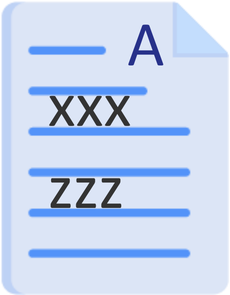
- All changes to controlled items must originate in BASE.
  - Exception: LIVE-only workflows.
- Never make changes to controlled items directly in TEST/UAT/LIVE environment.
  - Perforce must be the source of truth.
  - CCR Transport always uses complete copies of items in Perforce.
- Proper configuration locks TEST/UAT/LIVE from changes outside CCR workflow.



Always Deploy ItemSets

- Many high-risk problems can be caused by not deploying an ItemSet.
- Tier 1 CCR updates item A but does not deploy to LIVE.
- Later, another CCR also updates item A.
- Both changes now in LIVE.

Perforce LIVE branch



LIVE Environment

Quiz: Deployment

Question:

Performing the ‘startMoveToTEST’ transition on a CCR will automatically cause the item(s) to be deployed via an ItemSet to the TEST environment. True or False?

Answer:

False.

startMoveToTEST will automatically cause the changes to be *integrated* to the TEST branch, and an ItemSet to be *created*, but downloading the ItemSet is a user action.

Quiz: Deployment

Question:

When is it appropriate to NOT deploy an ItemSet?

A. Never.

B. If part of the testing plan fails.

C. When backing out a change.

D. If you are not a Perforce user.

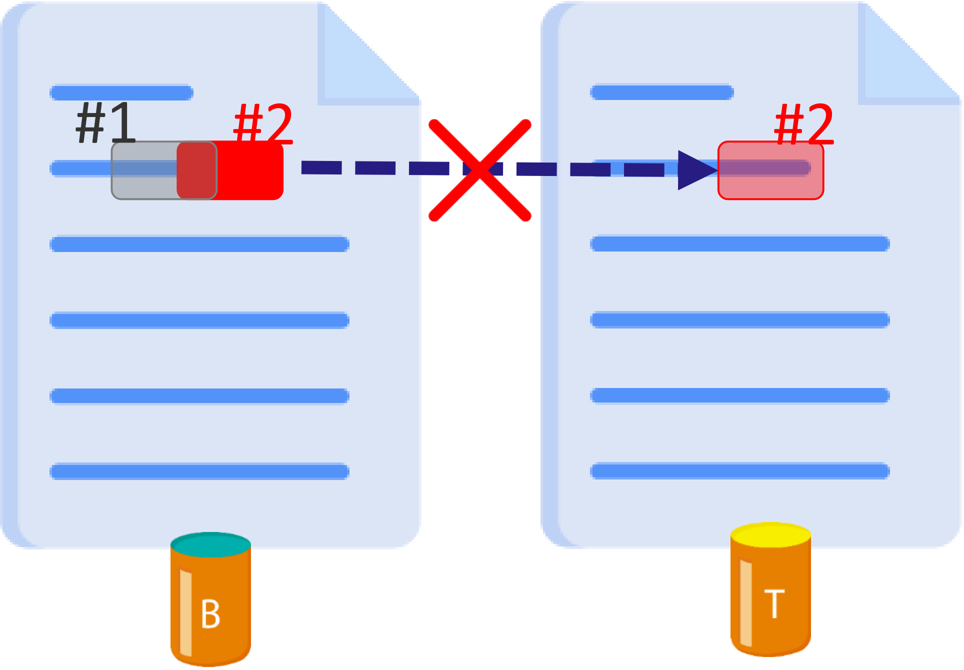
E. If there is a prerequisite CCR.

Quiz: Deployment (cont.)

Answer: A.
Always deploy an ItemSet to keep environments in sync with the state of Perforce.
B is wrong because deploy ItemSet happens before starting testing. C is wrong because backout happens in Perforce then need to deploy ItemSets to update environments. D is wrong because deploying ItemSets does not require interaction with Perforce (assuming ItemSet already generated). E is wrong because would not be able to perform authorizeAndStartMoveToXXXX transition and thus no ItemSet is created.

Merge Conflicts

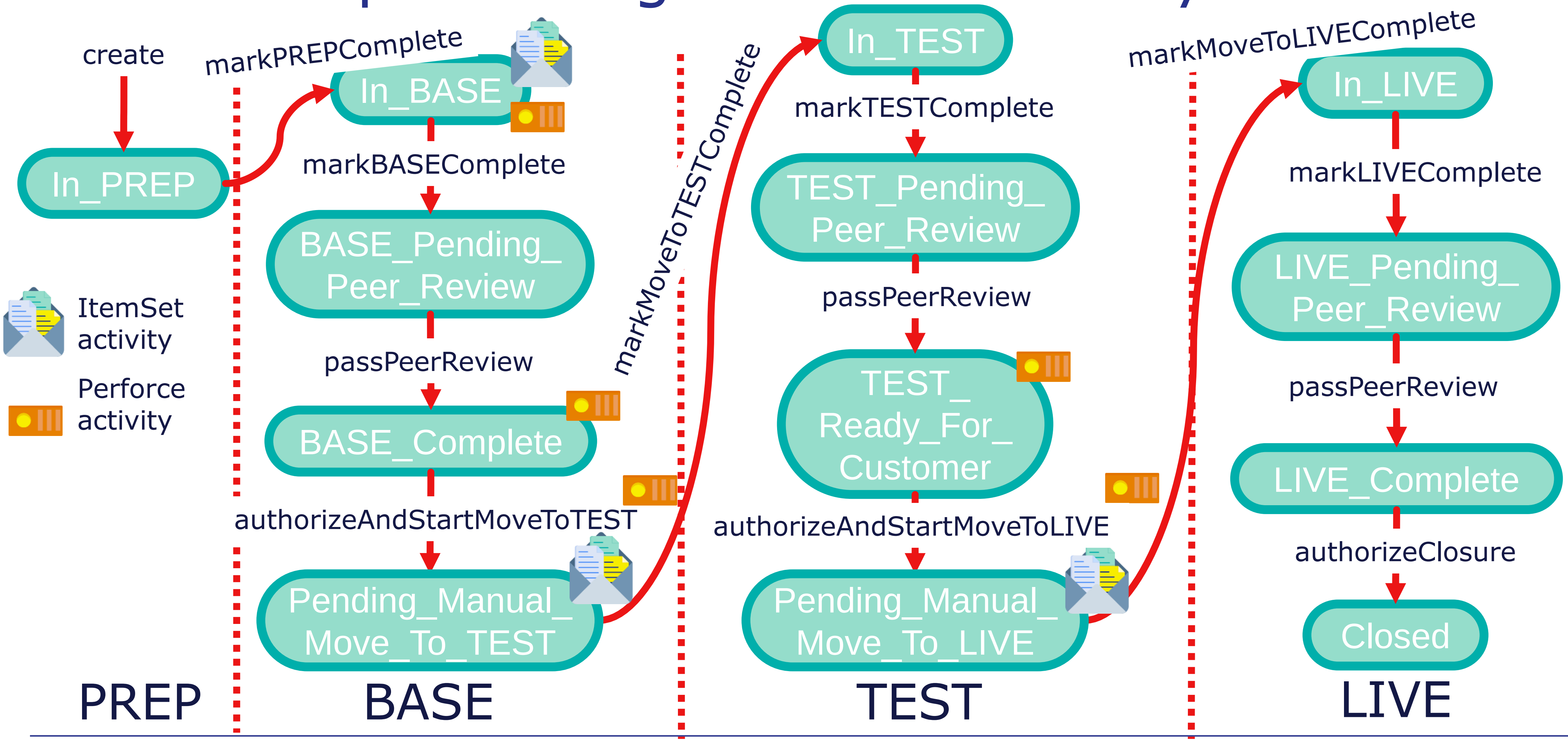
- Integration fails with merge conflict error when base for any diff chunk in source item does not exist in target environment.



Preview Integrations

- Preview integrations predict merge conflicts between Perforce branches.
- Does not commit changes to target branch.
- CCR does preview integrations between branch for current phase and branch for upcoming phase upon entering:
  - BASE_Complete.
  - TEST_Ready_For_Customer.
  - UAT_Ready_For_Customer.

CCR Transport along the CCR Journey…



CCR Transport Locations Explained

- In_BASE.
  - Initial ItemSet upload and CCR submit to Perforce.
  - No ItemSet activity in Connected BASE environments.
- BASE_Complete, TEST_Ready_For_Customer.
  - Preview integration in Perforce if user that performed passPeerReview transition logged into Perforce.
- authorizeAndStartMoveToTest, authorizeAndStartMoveToLive.
  - ItemSet created in Perforce, then manually deployed to environment.

CCR Transport Best Practices

- If you are unsure or run into an error: get assistance!
- Ensure baselining done properly when CCR first enabled.
  - Completed with help from InterSystems.
  - All existing items exported from an environment and submitted to Perforce.

CCR Transport Best Practices (cont.)

- Limit scope of individual CCRs.
- Progress CCR through workflow as quickly as safely as possible.
  - Stalled CCRs often block the work of others.
- Upload changes from BASE as soon as possible.
- Progress CCRs in order of configuration changes made.
- For cancel transition, ensure changes completely backed out and ItemSets deployed.

Improperly Cancelled CCR Causes Crises

- Only cancel CCR if change no longer needed.
  - Example: started creating configuration for new user but new user does not end up joining.
  - Cancel **never** solution for errors in change progression (such as merge conflicts).
- Always backout cancelled CCR.
  - Backed out of Perforce branch.
  - ItemSets deployed to all affected environments.

Improperly Cancelled CCR Causes Crises (cont.)

- Clean up merge conflicts.
  - Cancelling CCR with errors and recreating CCR is equivalent to leaving a crashed car on highway, getting into a new car and driving down same lane.
  - You have only made the original problem worse!

Quiz: Backing Out

Question:

Running the cancel transition is sufficient to reset everything if you ran into an issue and want to try again. True or False?

Answer:

False. Never cancel a CCR as a way of resolving an issue or to try to reset and try again. It will cause more problems and confusion.

If need to cancel, still need to complete the backout process. See ICC450 for details.

CCR Transport: The Audience

- Some of the other courses focused on CCR Transport (ICC4nn) are important prerequisites to CCR Tier 1 and CCR Tier 2 training.
- Courses dealing with direct Perforce interaction are only applicable to CCR users with access to the inside of the InterSystems network.

CCR Transport: The Audience (cont.)

- Additional CCR Transport material relates to:
  - Use of Perforce (applicable to InterSystems employees only).
  - The ins and outs of ItemSets.
  - Configuring CCR clients to send/receive changes from the CCR server.
  - Options for System Topology (number of environments, how they might share source, etc.).
  - Release management.
- While most would be helpful for anyone involved in Tier 1 or Tier 2 CCRs, some courses are for a more technical audience (e.g. Perforce and Debugging material).

Areas for Further Study for CCR Transport

- ICC410 – Perforce Basics.
- ICC420 – ItemSets.
- ICC430 – CCR Client Configuration.
- ICC440 – System Topology.
  - Environment Flow Options.
  - Secondary Environments.
  - Multi-Phase Implementation Strategy.

Areas for Further Study for CCR Transport (cont.)

- ICC450 – Backouts without using P4V and debugging.
- ICC460 – Debugging techniques for InterSystems employees and advanced Perforce concepts.
- ICC470 – Release Management.

Summary

- What are the key points for this module?

ICC 410: CCR Transport – Perforce Basics

InterSystems
Creative data technology

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.

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

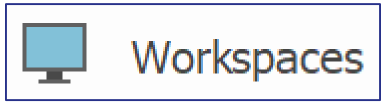
- P4.
  - Command line interface (CLI) client to Perforce.
  - Very useful for scripting and automation.
- P4V.
  - P4 CLI wrapped with visual client (desktop app).
  - Client provides advanced features.
- Free, but connecting to Perforce server requires a license.
  - Perforce does give away free 5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

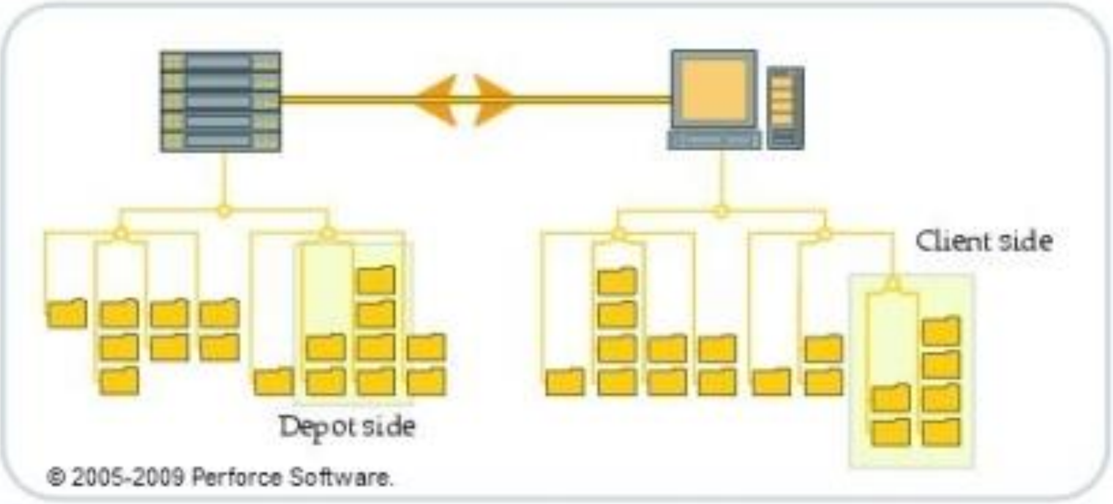
- Client workspace.



  - 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.



© 2005-2009 Perforce Software.

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.

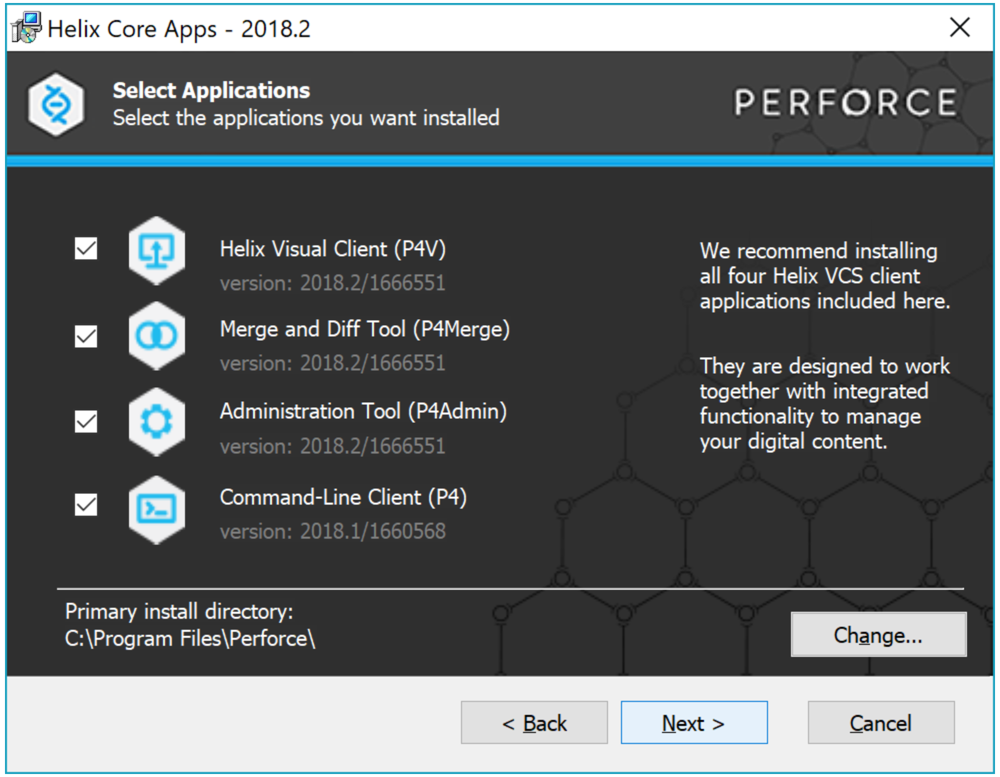
D. P4.

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.



Helix Core Apps - 2018.2

Select Applications

Select the applications you want installed

☑ Helix Visual Client (P4V)
version: 2018.2/1666551

☑ Merge and Diff Tool (P4Merge)
version: 2018.2/1666551

☑ Administration Tool (P4Admin)
version: 2018.2/1666551

☑ Command-Line Client (P4)
version: 2018.1/1660568

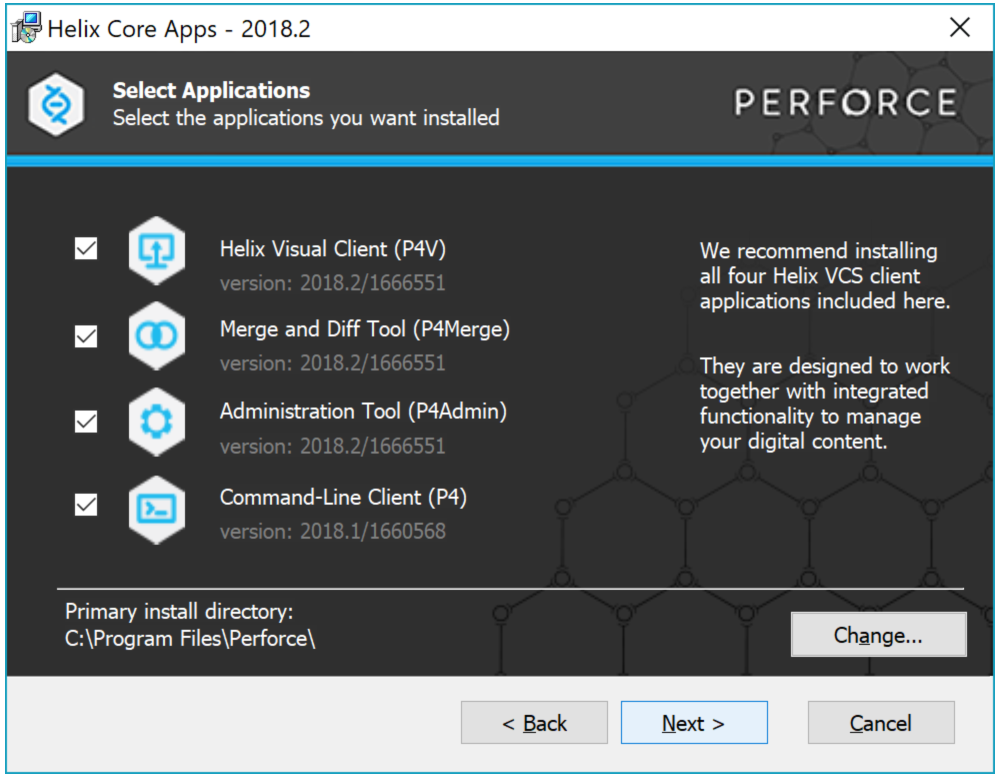
We recommend installing all four Helix VCS client applications included here.

They are designed to work together with integrated functionality to manage your digital content.

Primary install directory:
C:\Program Files\Perforce\

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

- Installation:
  1. Go to http://www.perforce.com/downloads/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."



Helix Core Apps - 2018.2

Select Applications
Select the applications you want installed

☑ Helix Visual Client (P4V)
version: 2018.2/1666551

☑ Merge and Diff Tool (P4Merge)
version: 2018.2/1666551

☑ Administration Tool (P4Admin)
version: 2018.2/1666551

☑ Command-Line Client (P4)
version: 2018.1/1660568

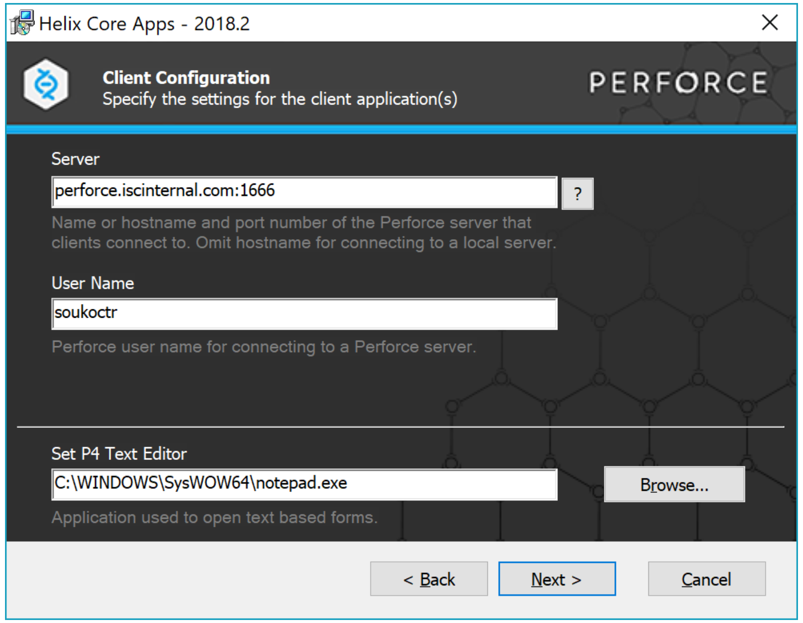
We recommend installing all four Helix VCS client applications included here.

They are designed to work together with integrated functionality to manage your digital content.

Primary install directory:
C:\Program Files\Perforce\

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)

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.

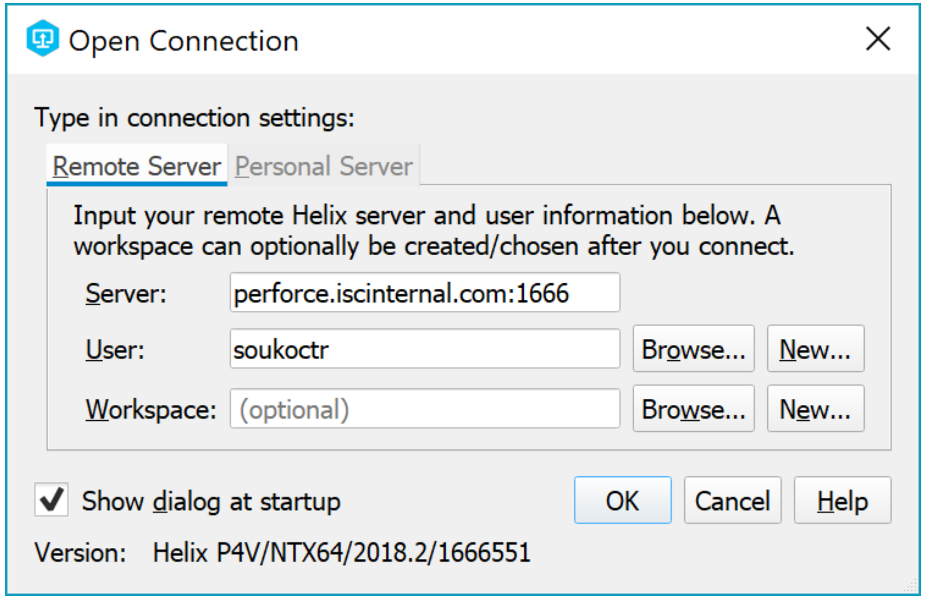
User Name
soukoctr
Perforce user name for connecting to a Perforce server.

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

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.



Open Connection

Type in connection settings:

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

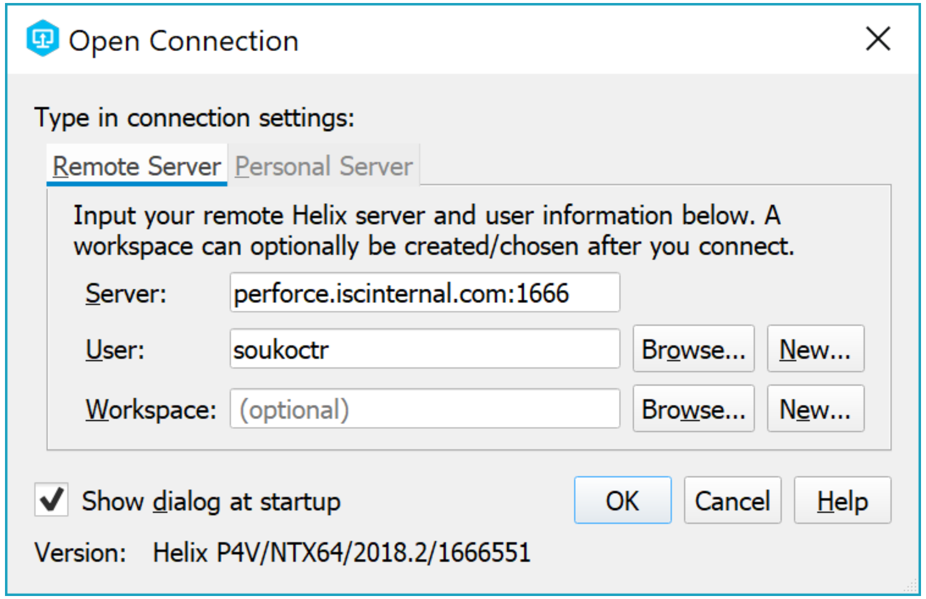
Workspace: (optional)

☑ Show dialog at startup

Version: Helix P4V/NTX64/2018.2/1666551

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:

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

Workspace: (optional)

☑ Show dialog at startup

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

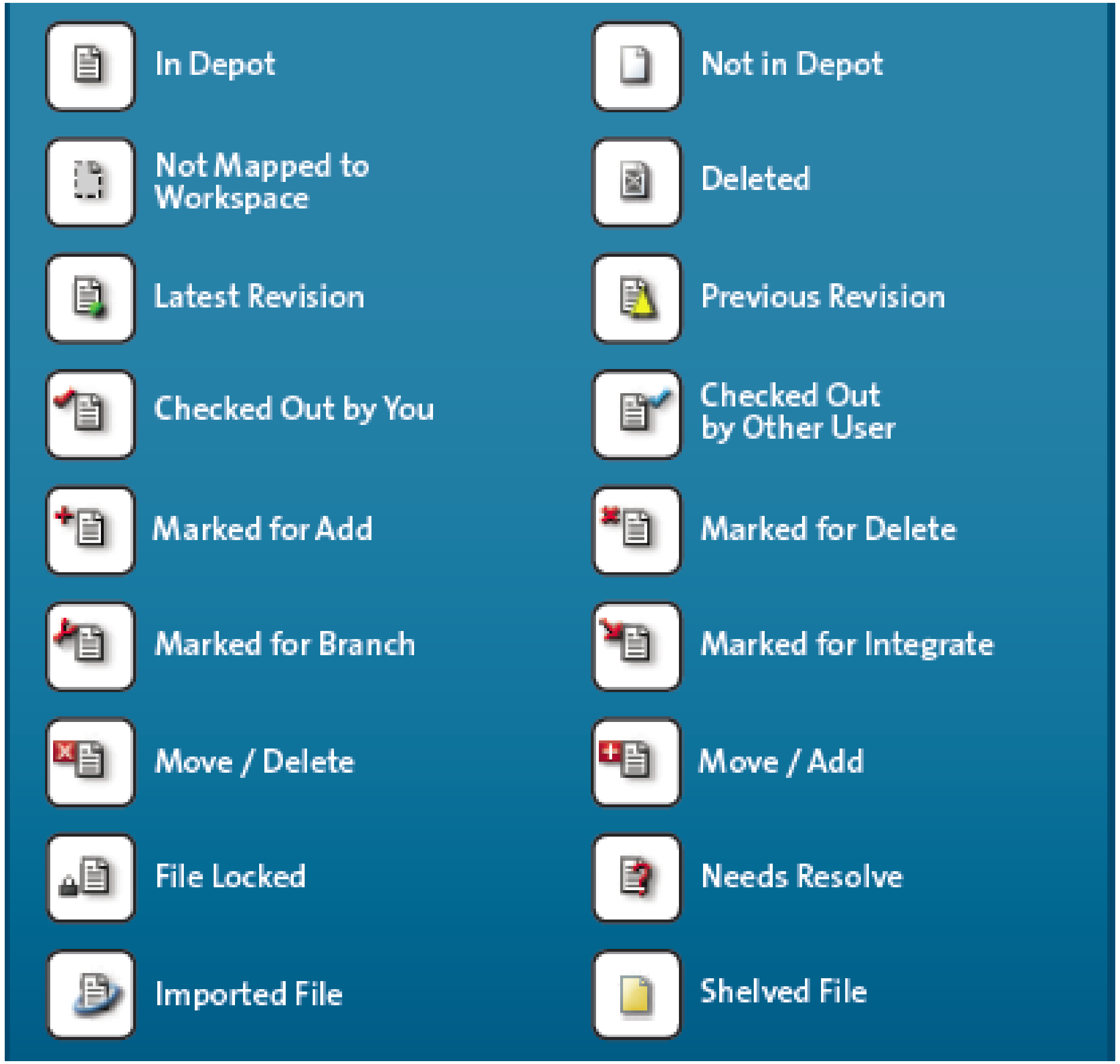
- Branch.
  - The folder structure of your Perforce depot.
    - Enables similar structures to represent each of the multiple environments in the System.

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.

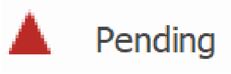
File status icons

- 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

P4V

Navigation	How To
Go To… • Changelist • User • Workspace • Job	1. [Ctrl]+g
Copy File Location	1. Click on a file 2. [Ctrl]+c
Highlight file within current tree (depot or workspace) whose path has been copied	1. Click on any file 2. [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
$Author$	User submitting the files.	$Author: bruno $
$Change$	Changelist number under which file was submitted.	$Change: 439 $
$Date$	Date of last submission (YYYY/MM/DD).	$Date: 2011/08/18 $
$DateTime$	Helix Server local date and time of last submission (YYYY/MM/DD hh:mm:ss).	$DateTime: 2011/08/18 23:17:02 $
$File$	Filename only, in depot syntax, without revision number.	$File: //depot/path/file.txt $
$Header$	Synonymous with Id.	$Header: //depot/path/file.txt#3 $
Id	Filename and revision number in depot syntax.	$Id: //depot/path/file.txt#3 $
$Revision$	Helix Server revision number.	$Revision: #3 $

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!

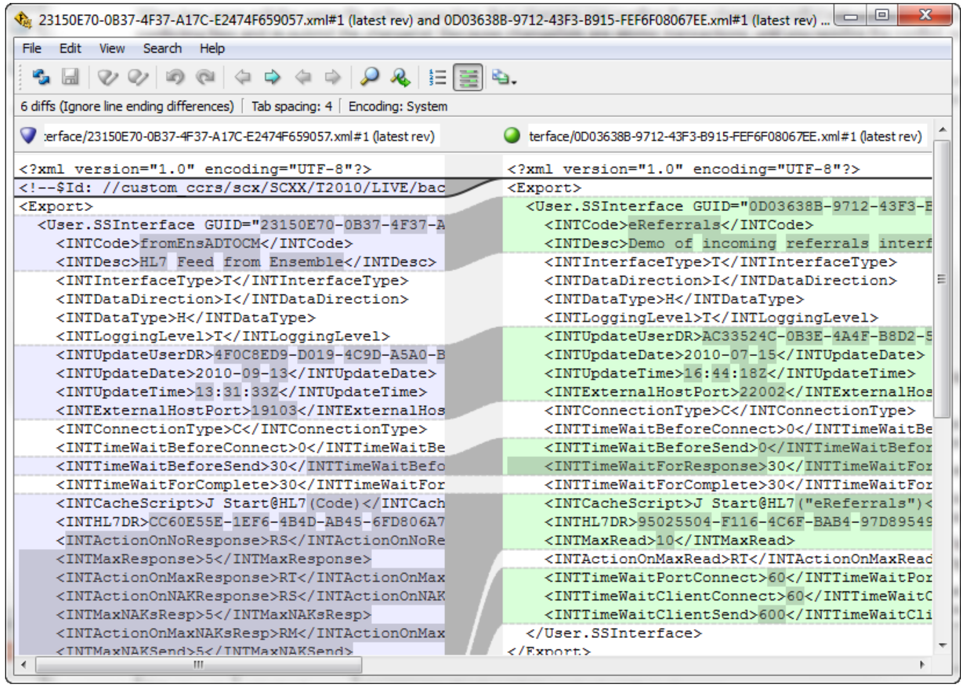
Shelving

- 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.


23150E70-0B37-4F37-A17C-E2474F659057.xml#1 (latest rev) and 0D03638B-9712-43F3-B915-FEF6F08067EE.xml#1 (latest rev) ...
File
Edit
View
Search
Help

6 diffs (Ignore line ending differences) | Tab spacing: 4 | Encoding: System

:erface/23150E70-0B37-4F37-A17C-E2474F659057.xml#1 (latest rev)

```
<?xml version="1.0" encoding="UTF-8"?>
<!--$Id: //custom ccrs/scx/SCXX/T2010/LIVE/bac
<Export>
  <User.SSInterface GUID="23150E70-0B37-4F37-A
    <INTCode>fromEnsADTOCM</INTCode>
    <INTDesc>HL7 Feed from Ensemble</INTDesc>
    <INTInterfaceType>T</INTInterfaceType>
    <INTDataDirection>I</INTDataDirection>
    <INTDataType>H</INTDataType>
    <INTLoggingLevel>T</INTLoggingLevel>
    <INTUpdateUserDR>4F0C8ED9-D019-4C9D-A5A0-B
    <INTUpdateDate>2010-09-13</INTUpdateDate>
    <INTUpdateTime>13:31:33Z</INTUpdateTime>
    <INTExternalHostPort>19103</INTExternalHos
    <INTConnectionType>C</INTConnectionType>
    <INTTimeWaitBeforeConnect>0</INTTimeWaitBe
    <INTTimeWaitBeforeSend>30</INTTimeWaitBefo
    <INTTimeWaitForComplete>30</INTTimeWaitFor
    <INTCacheScript>J Start@HL7(Code)</INTCach
    <INTHL7DR>CC60E55E-1EF6-4B4D-AB45-6FD806A7
    <INTActionOnNoResponse>RS</INTActionOnNoRe
    <INTMaxResponse>5</INTMaxResponse>
    <INTActionOnMaxResponse>RT</INTActionOnMax
    <INTActionOnNAKResponse>RS</INTActionOnNAK
    <INTMaxNAKsResp>5</INTMaxNAKsResp>
    <INTActionOnMaxNAKsResp>RM</INTActionOnMax
    <INTMaxNAKSend>5</INTMaxNAKSend>
```

terface/0D03638B-9712-43F3-B915-FEF6F08067EE.xml#1 (latest rev)

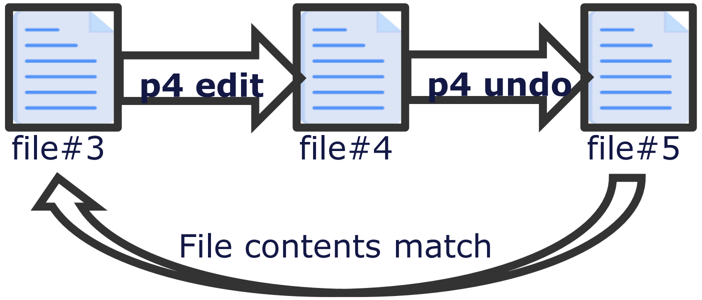
```
<?xml version="1.0" encoding="UTF-8"?>
<Export>
  <User.SSInterface GUID="0D03638B-9712-43F3-B
    <INTCode>eReferrals</INTCode>
    <INTDesc>Demo of incoming referrals interf
    <INTInterfaceType>T</INTInterfaceType>
    <INTDataDirection>I</INTDataDirection>
    <INTDataType>H</INTDataType>
    <INTLoggingLevel>T</INTLoggingLevel>
    <INTUpdateUserDR>AC33524C-0B3E-4A4F-B8D2-5
    <INTUpdateDate>2010-07-15</INTUpdateDate>
    <INTUpdateTime>16:44:18Z</INTUpdateTime>
    <INTExternalHostPort>22002</INTExternalHos
    <INTConnectionType>C</INTConnectionType>
    <INTTimeWaitBeforeConnect>0</INTTimeWaitBe
    <INTTimeWaitBeforeSend>0</INTTimeWaitBefor
    <INTTimeWaitForResponse>30</INTTimeWaitFor
    <INTTimeWaitForComplete>30</INTTimeWaitFor
    <INTCacheScript>J Start@HL7("eReferrals")<
    <INTHL7DR>95025504-F116-4C6F-BAB4-97D89549
    <INTMaxRead>10</INTMaxRead>
    <INTActionOnMaxRead>RT</INTActionOnMaxRead
    <INTTimeWaitPortConnect>60</INTTimeWaitPor
    <INTTimeWaitClientConnect>60</INTTimeWaitC
    <INTTimeWaitClientSend>600</INTTimeWaitCli
  </User.SSInterface>
</Export>
```

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.

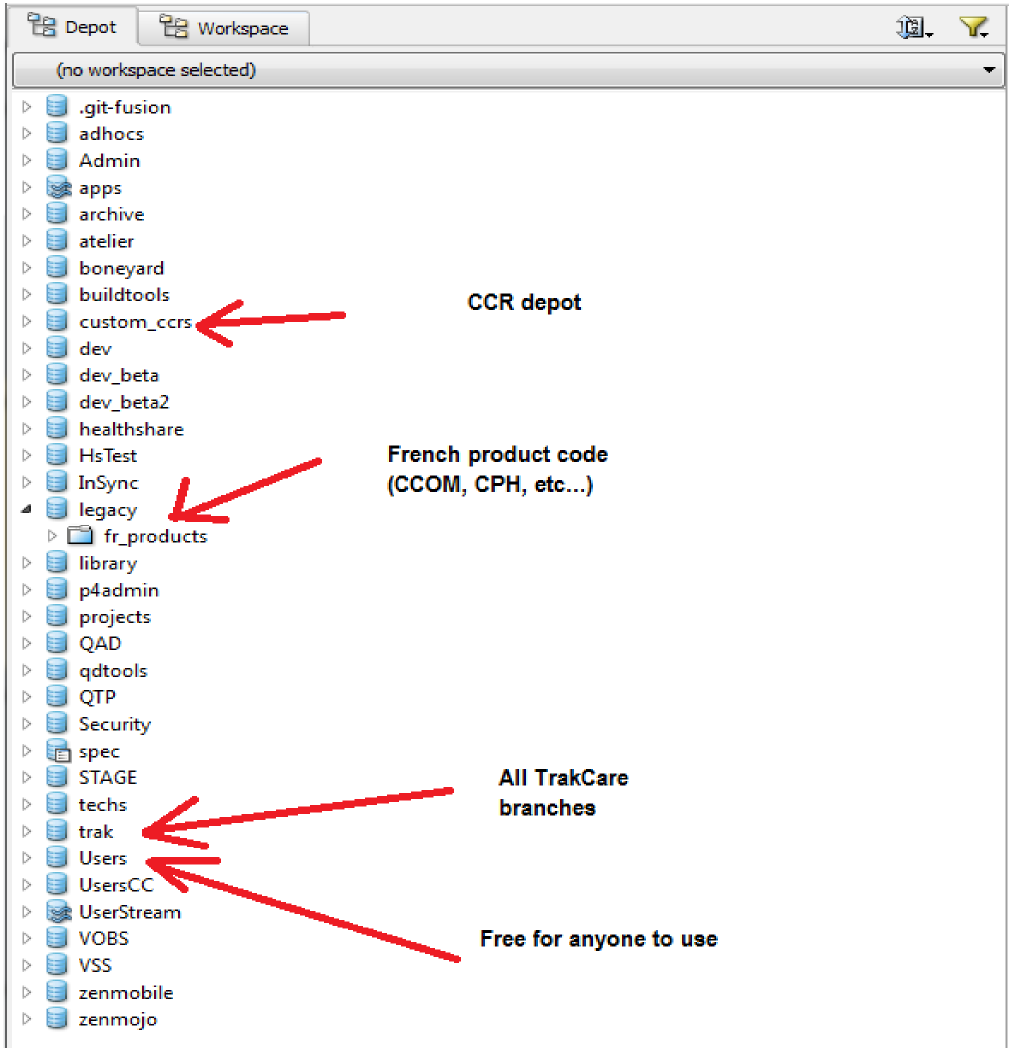
“Undo” will create a new revision of a file with content equal to an older revision… it does not remove subsequent revisions – they stay intact as part of the Revision History for future reference (or in case there is a need to Undo the Undo).

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.



.git-fusion
adhocs
Admin
apps
archive
atelier
boneyard
buildtools
custom_ccrs
dev
dev_beta
dev_beta2
healthshare
HsTest
InSync
legacy
fr_products
library
p4admin
projects
QAD
qdtools
QTP
Security
spec
STAGE
techs
trak
Users
UsersCC
UserStream
VOBS
VSS
zenmobile
zenmojo

CCR depot

French product code (CCOM, CPH, etc...)

All TrakCare branches

Free for anyone to use

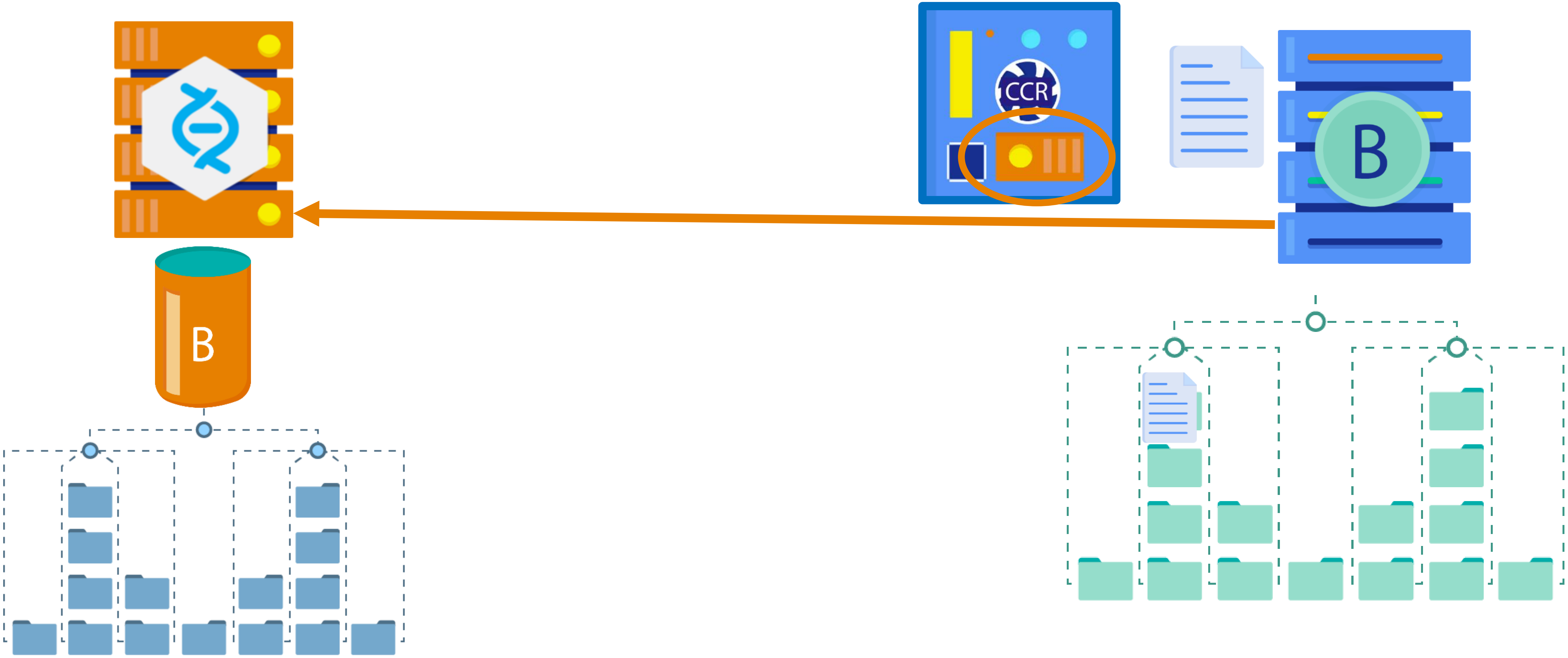
InterSystems IDE & Perforce Integration

- 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

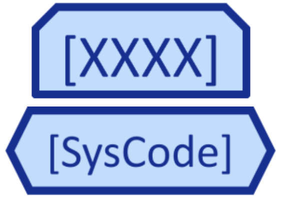
- There is a 1-to-1 relationship between a CCR and a P4 job.
  - But multiple changelists can be associated with a single job.
  - Possible to edit a job’s changelists.
    - Don’t forget to regenerate the ItemSet if you do!

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.

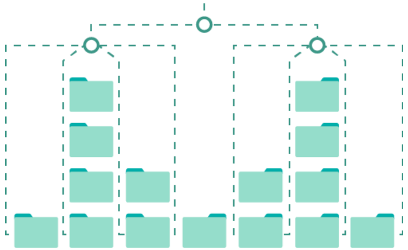
Review: CCR Branches

- 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.

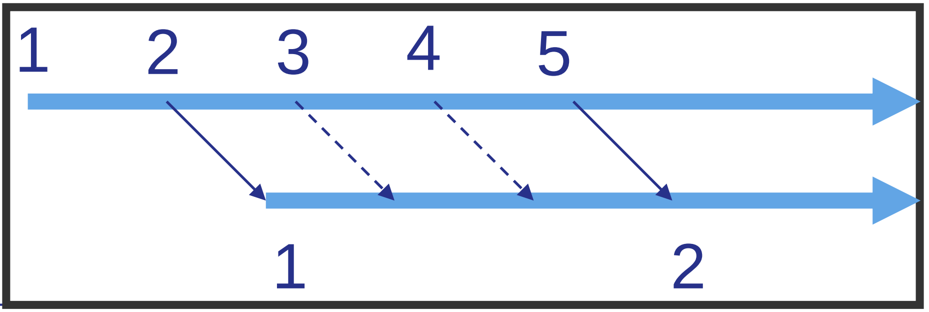
Part 6: Moving Changes between Branches

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.

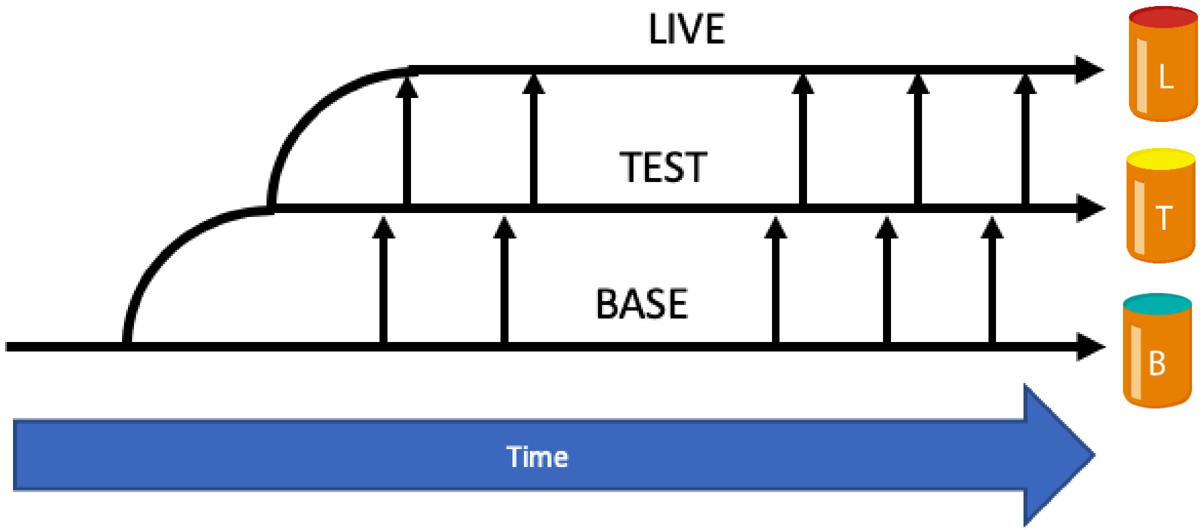
P4 Integrate

- 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.



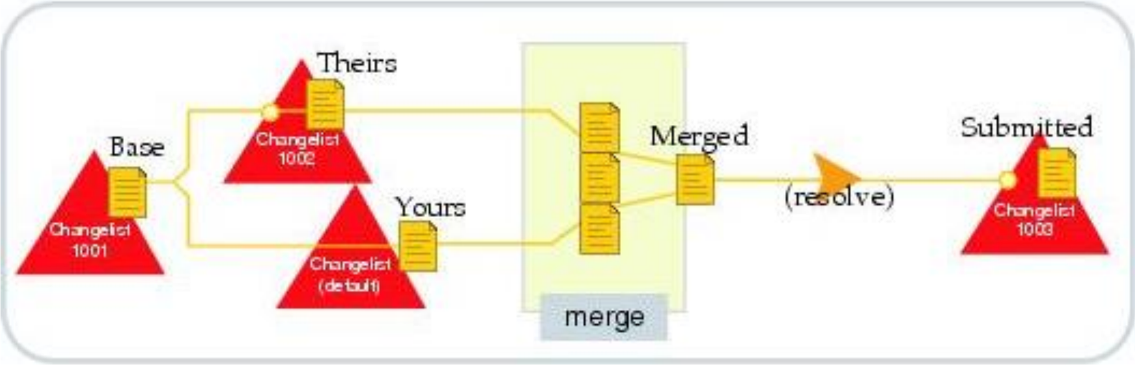
Codeline Management: Web Development Model

- 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.



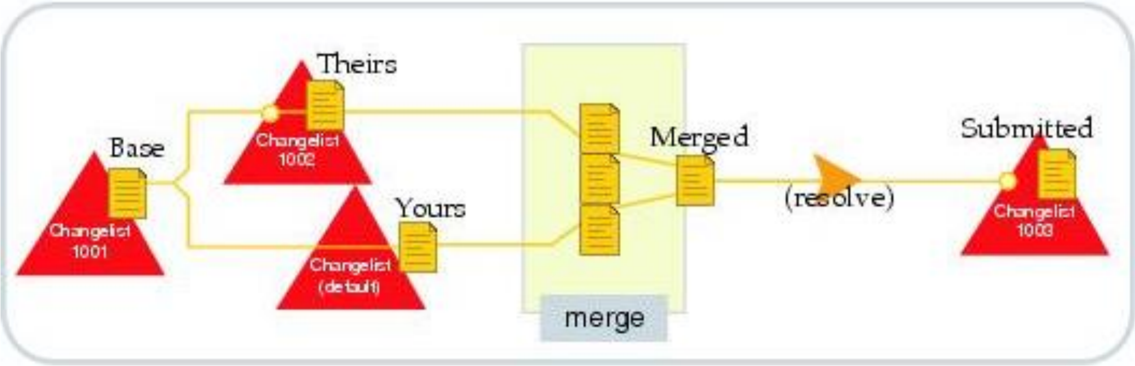
Resolve and Merge

- **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.

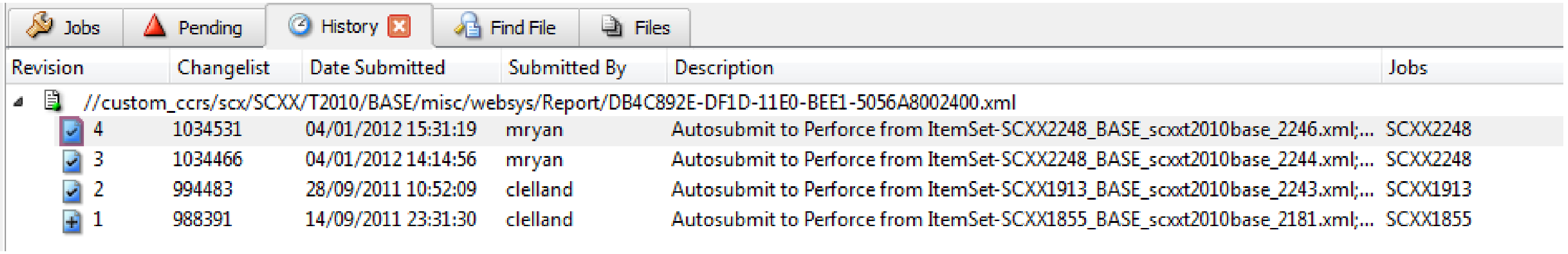
Part 7:
Tools for Developing and Maintaining Software

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

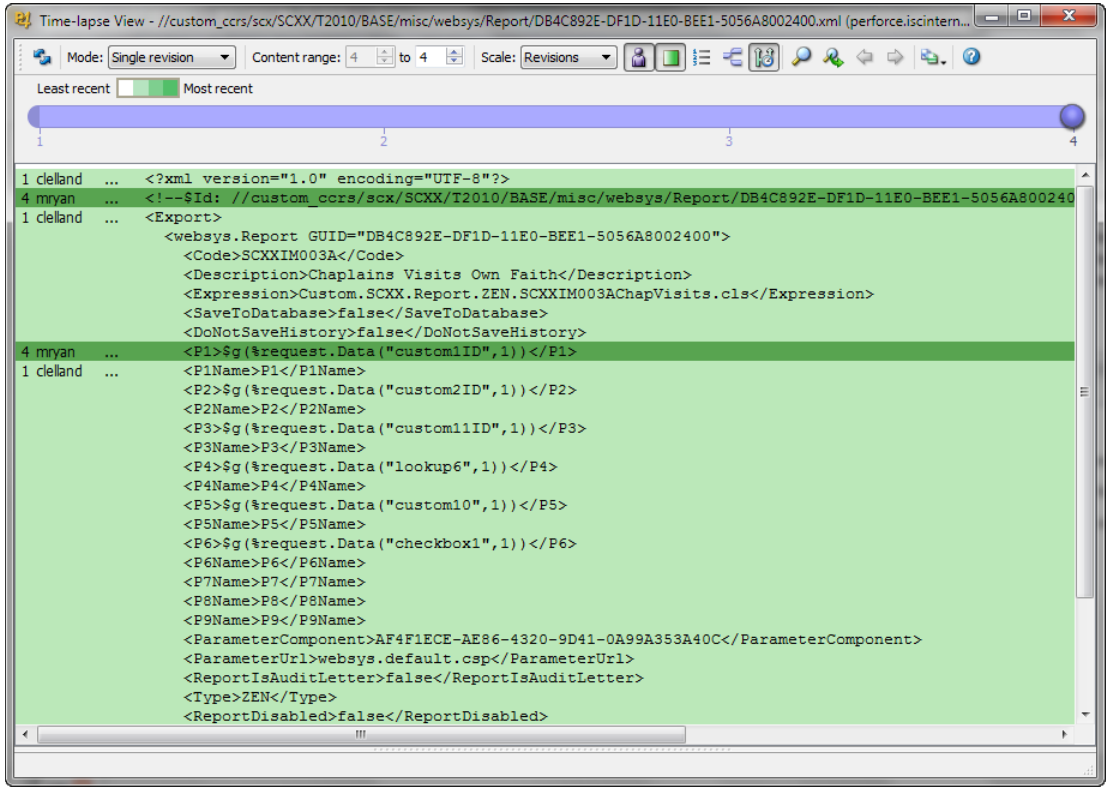
- File history.
  - How to: Click a file, then select History tab.
  - Submissions with timestamps.
  - Drag and drop comparison (Diff).
  - TIP: Jobs column is not shown by default but is very helpful!
    - Makes it clear what CCR/JIRA/Devlog was responsible for that revision.
    - Right-click on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



Revision	Changelist	Date Submitted	Submitted By	Description	Jobs
//custom_ccrs/scx/SCXX/T2010/BASE/misc/websys/Report/DB4C892E-DF1D-11E0-BEE1-5056A8002400.xml					
4	1034531	04/01/2012 15:31:19	mryan	Autosubmit to Perforce from ItemSet-SCXX2248_BASE_scxt2010base_2246.xml;...	SCXX2248
3	1034466	04/01/2012 14:14:56	mryan	Autosubmit to Perforce from ItemSet-SCXX2248_BASE_scxt2010base_2244.xml;...	SCXX2248
2	994483	28/09/2011 10:52:09	clelland	Autosubmit to Perforce from ItemSet-SCXX1913_BASE_scxt2010base_2243.xml;...	SCXX1913
1	988391	14/09/2011 23:31:30	clelland	Autosubmit to Perforce from ItemSet-SCXX1855_BASE_scxt2010base_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.



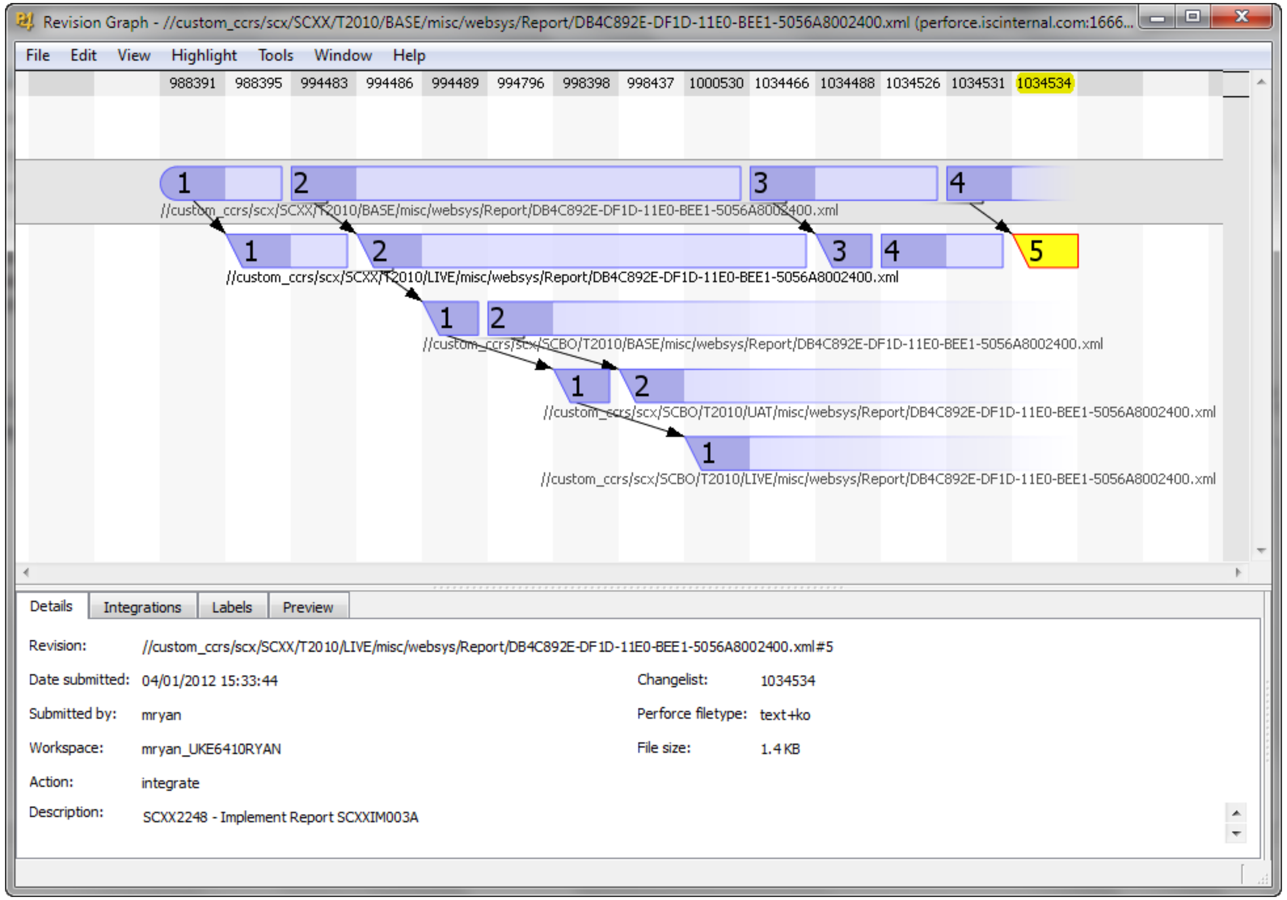
Time-lapse View - //custom_ccrs/scx/SCXX/T2010/BASE/misc/websys/Report/DB4C892E-DF1D-11E0-BEE1-5056A8002400.xml (perforce.iscintern...

Mode: Single revision | Content range: 4 to 4 | Scale: Revisions

```
1 clelland   ...   <?xml version="1.0" encoding="UTF-8"?>
4 mryan      ...   <!--$Id: //custom_ccrs/scx/SCXX/T2010/BASE/misc/websys/Report/DB4C892E-DF1D-11E0-BEE1-5056A800240
1 clelland   ...   <Export>
                     <websys.Report GUID="DB4C892E-DF1D-11E0-BEE1-5056A8002400">
                       <Code>SCXXIM003A</Code>
                       <Description>Chaplains Visits Own Faith</Description>
                       <Expression>Custom.SCXX.Report.ZEN.SCXXIM003AChapVisits.cls</Expression>
                       <SaveToDatabase>false</SaveToDatabase>
                       <DoNotSaveHistory>false</DoNotSaveHistory>
4 mryan      ...       <P1>$g(%request.Data("custom1ID",1))</P1>
1 clelland   ...       <P1Name>P1</P1Name>
                       <P2>$g(%request.Data("custom2ID",1))</P2>
                       <P2Name>P2</P2Name>
                       <P3>$g(%request.Data("custom11ID",1))</P3>
                       <P3Name>P3</P3Name>
                       <P4>$g(%request.Data("lookup6",1))</P4>
                       <P4Name>P4</P4Name>
                       <P5>$g(%request.Data("custom10",1))</P5>
                       <P5Name>P5</P5Name>
                       <P6>$g(%request.Data("checkbox1",1))</P6>
                       <P6Name>P6</P6Name>
                       <P7Name>P7</P7Name>
                       <P8Name>P8</P8Name>
                       <P9Name>P9</P9Name>
                       <ParameterComponent>AF4F1ECE-AE86-4320-9D41-0A99A353A40C</ParameterComponent>
                       <ParameterUrl>websys.default.csp</ParameterUrl>
                       <ReportIsAuditLetter>false</ReportIsAuditLetter>
                       <Type>ZEN</Type>
                       <ReportDisabled>false</ReportDisabled>
```

Revision Graph

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

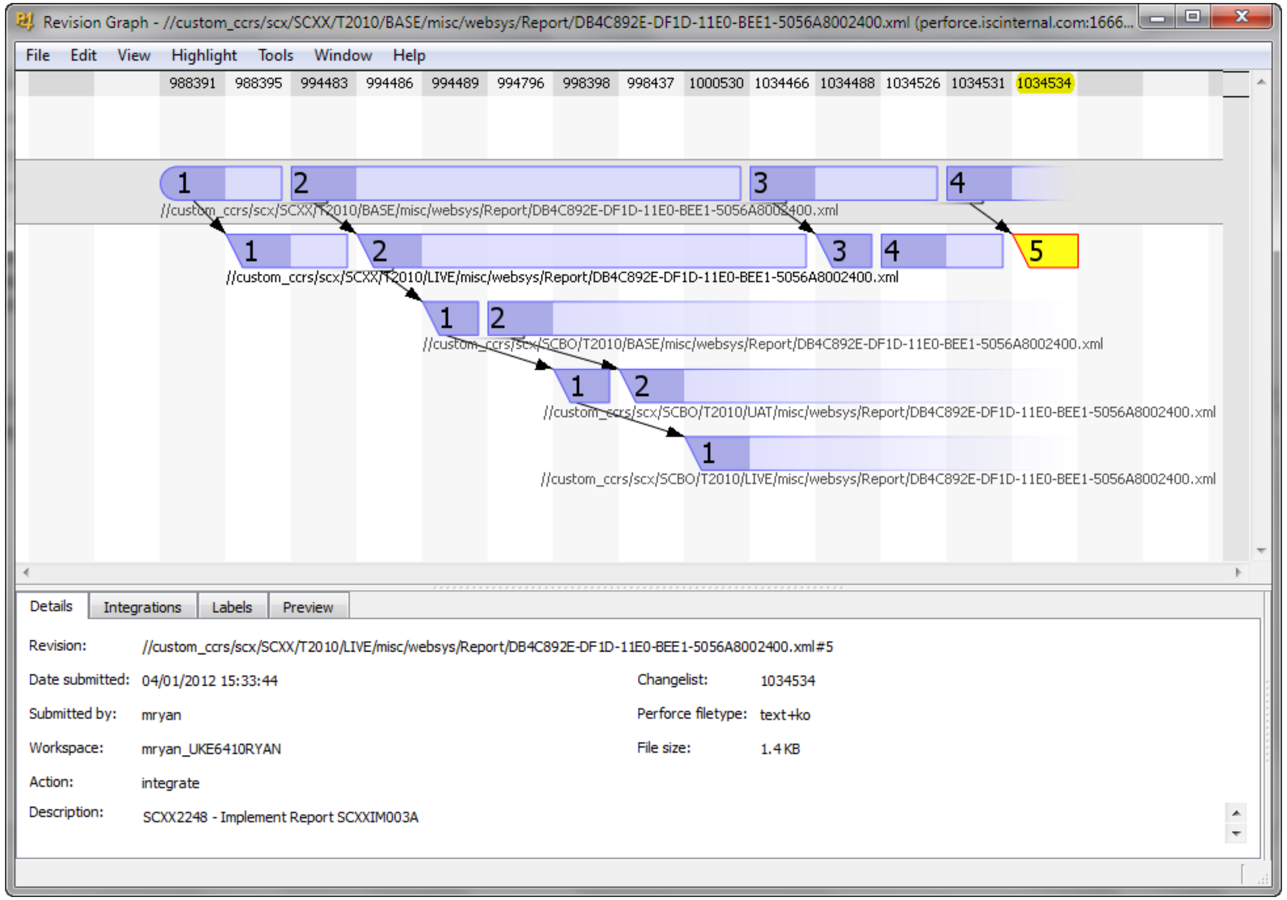


Revision Graph - //custom_ccrs/scx/SCXX/T2010/BASE/misc/websys/Report/DB4C892E-DF1D-11E0-BEE1-5056A8002400.xml (perforce.iscinternal.com:1666...
File
Edit
View
Highlight
Tools
Window
Help
988391
988395
994483
994486
994489
994796
998398
998437
1000530
1034466
1034488
1034526
1034531
1034534
1
2
3
4
//custom_ccrs/scx/SCXX/T2010/BASE/misc/websys/Report/DB4C892E-DF1D-11E0-BEE1-5056A8002400.xml
1
2
3
4
5
//custom_ccrs/scx/SCXX/T2010/LIVE/misc/websys/Report/DB4C892E-DF1D-11E0-BEE1-5056A8002400.xml
1
2
//custom_ccrs/scx/SCBO/T2010/BASE/misc/websys/Report/DB4C892E-DF1D-11E0-BEE1-5056A8002400.xml
1
2
//custom_ccrs/scx/SCBO/T2010/UAT/misc/websys/Report/DB4C892E-DF1D-11E0-BEE1-5056A8002400.xml
1
//custom_ccrs/scx/SCBO/T2010/LIVE/misc/websys/Report/DB4C892E-DF1D-11E0-BEE1-5056A8002400.xml
Details
Integrations
Labels
Preview

Revision:	//custom_ccrs/scx/SCXX/T2010/LIVE/misc/websys/Report/DB4C892E-DF1D-11E0-BEE1-5056A8002400.xml#5		
Date submitted:	04/01/2012 15:33:44	Changelist:	1034534
Submitted by:	mryan	Perforce filetype:	text+ko
Workspace:	mryan_UKE6410RYAN	File size:	1.4 KB
Action:	integrate		
Description:	SCXX2248 - Implement Report SCXXIM003A		

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.



Revision Graph - //custom_ccrs/scx/SCXX/T2010/BASE/misc/websys/Report/DB4C892E-DF1D-11E0-BEE1-5056A8002400.xml (perforce.iscinternal.com:1666...
File Edit View Highlight Tools Window Help
988391 988395 994483 994486 994489 994796 998398 998437 1000530 1034466 1034488 1034526 1034531 1034534
1 2 3 4
//custom_ccrs/scx/SCXX/T2010/BASE/misc/websys/Report/DB4C892E-DF1D-11E0-BEE1-5056A8002400.xml
1 2 3 4 5
//custom_ccrs/scx/SCXX/T2010/LIVE/misc/websys/Report/DB4C892E-DF1D-11E0-BEE1-5056A8002400.xml
1 2
//custom_ccrs/scx/SCBO/T2010/BASE/misc/websys/Report/DB4C892E-DF1D-11E0-BEE1-5056A8002400.xml
1 2
//custom_ccrs/scx/SCBO/T2010/UAT/misc/websys/Report/DB4C892E-DF1D-11E0-BEE1-5056A8002400.xml
1
//custom_ccrs/scx/SCBO/T2010/LIVE/misc/websys/Report/DB4C892E-DF1D-11E0-BEE1-5056A8002400.xml
Details Integrations Labels Preview

Revision: //custom_ccrs/scx/SCXX/T2010/LIVE/misc/websys/Report/DB4C892E-DF1D-11E0-BEE1-5056A8002400.xml#5

Date submitted: 04/01/2012 15:33:44

Changelist: 1034534

Submitted by: mryan

Perforce filetype: text+ko

Workspace: mryan_UKE6410RYAN

File size: 1.4 KB

Action: integrate

Description: SCXX2248 - Implement Report SCXXIM003A

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?

ICC 420: CCR Transport – Items and ItemSets

InterSystems

Creative data technology

Objectives

- Identify information and actions available in Perforce details section of CCR.
- Describe how changes are moved between Perforce branches.
- Identify contents of ItemSets.
- Explain the use of the source workspace.
- Identify purpose of baselining.
- Identify actions that require a Perforce account.

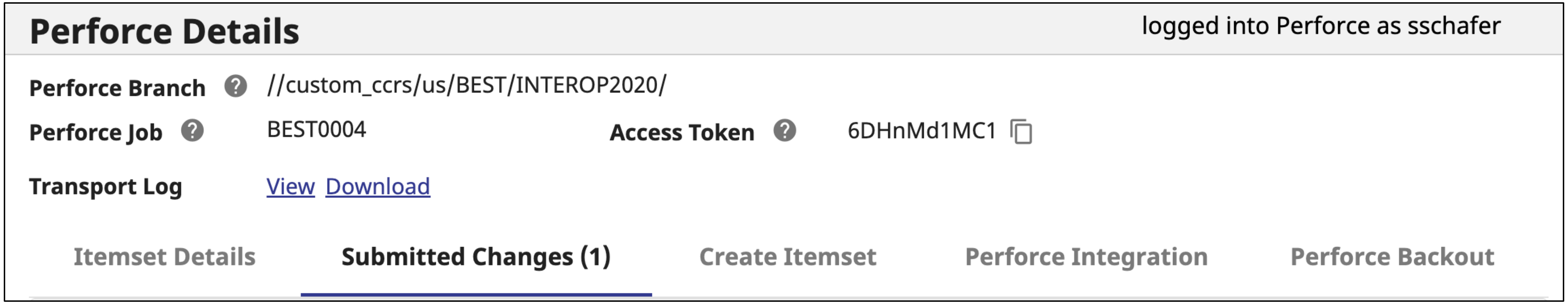
Part 1: Perforce and the CCR Record

Review: Perforce Users

- All CCR users who need CCR to take actions in Perforce on their behalf must have Perforce username and password.
  - Including non-InterSystems users.

Perforce Details Section

- Contains CCR Transport information and actions.
  - Perforce login status.
  - Access Token.
  - Transport Log.
  - ItemSets.
  - Submitted Changes.
  - Perforce actions and ItemSet creation.



Perforce Details

logged into Perforce as sschafer

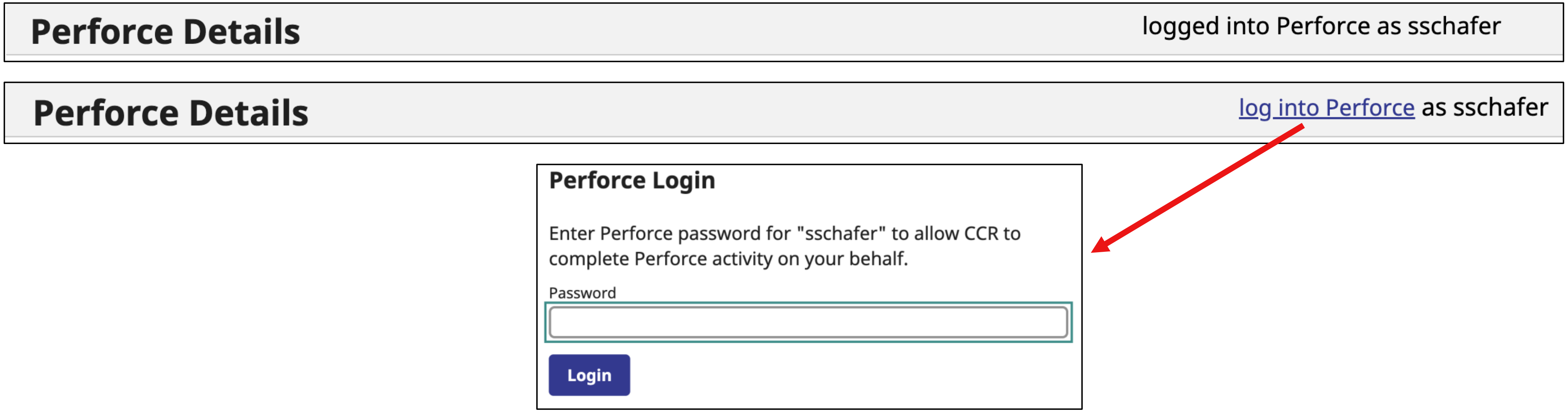
Perforce Branch ? //custom_ccrs/us/BEST/INTEROP2020/

Perforce Job ? BEST0004 **Access Token** ? 6DHnMd1MC1

Transport Log View Download

Logging into Perforce via CCR

- CCR assumes same username in Perforce.
- CCR attempts to use same password for Perforce on logging into CCR.
  - If Perforce and CCR password different, must supply Perforce password to CCR.
- Perforce Details section of every CCR shows login status.



Perforce Details logged into Perforce as sschafer

Perforce Details log into Perforce as sschafer

Perforce Login

Enter Perforce password for "sschafer" to allow CCR to complete Perforce activity on your behalf.

Password

Username Not a Perforce User Error

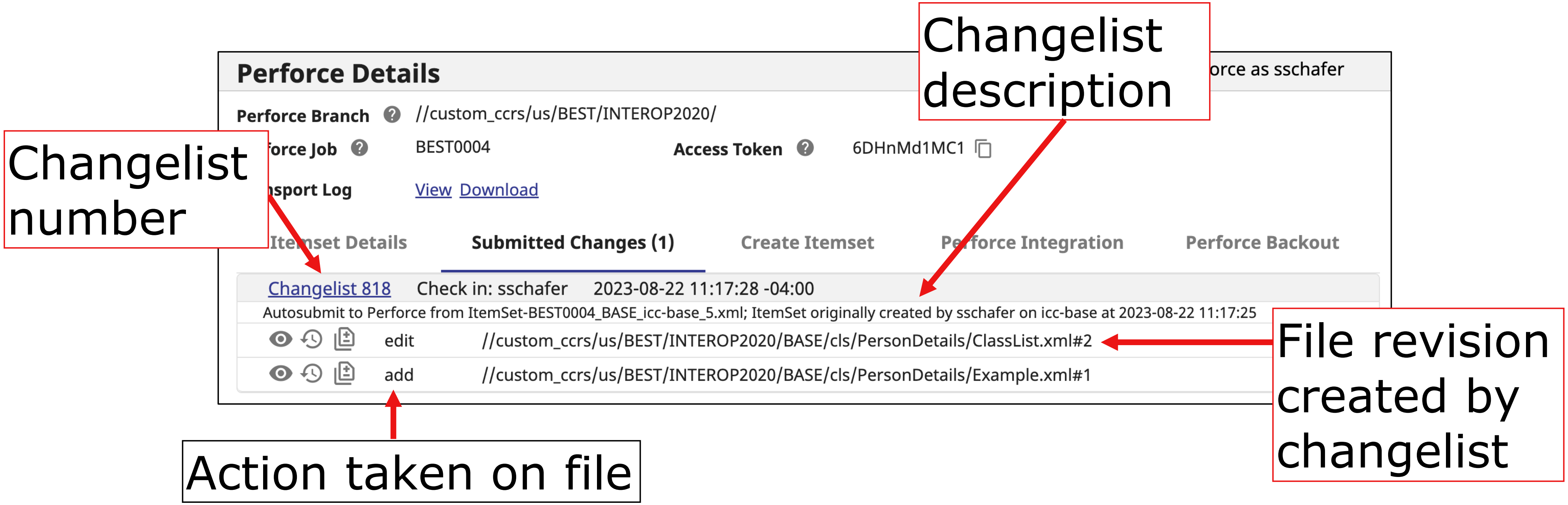
- New CCR users might see message “user is not a Perforce user”.
  - CCR regularly queries Perforce to determine which users are also Perforce users.
  - InterSystems employees can force resynchronization of Perforce users.
    - Menu > Refresh Perforce Users.

Terminology: Changelist

- Changelist = Perforce term for an atomic set of changes in 1 transaction.
  - All changelists have a numerical ID.
- CCR submits all items in an ItemSet as one changelist tagged with CCR ID.
  - Does not submit ItemSet to Perforce; submits items in ItemSet.
- authorizeAndStartMoveToTEST integrates all changelists tagged with that CCR ID from BASE to TEST branch.
  - Submits integration as one changelist tagged with CCR ID.

Viewing Submitted Changes

- Click Submitted Changes tab to view all changelists in Perforce associated with this CCR.



Perforce Details

orce as sschafer

Perforce Branch ? //custom_ccrs/us/BEST/INTEROP2020/

force Job ? BEST0004

Access Token ? 6DHnMd1MC1

nsport 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

Changelist description

Changelist number

File revision created by changelist

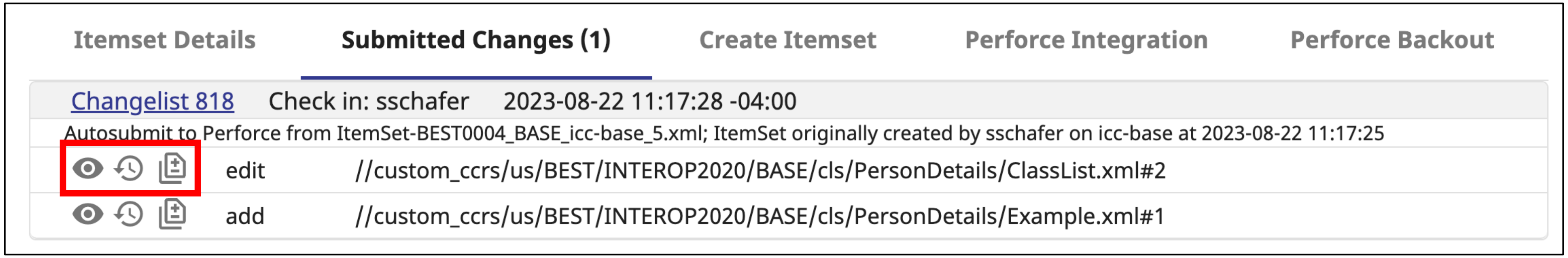
Action taken on file

Viewing Submitted Changes (cont.)

- Changelist number.
  - Links to Swarm (InterSystems employees only).
- Action taken on file options:
  - Add, edit, delete, branch, integrate, undo.
- Notice no information about ItemSets.
  - ItemSets effectively disposable once used for transport.

Actions Under View Submitted Changes

- View.
  - Show entire contents of the item at that revision.
  - Links to page where able to diff against any revision.
- History.
  - Shows color coded list of revisions to item.
- Diff.
  - Show diff chunks changed at that revision.



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

CCR Transport: Errors

- Errors with Perforce or ItemSets appear in Perforce Details.
- Never ignore orange or red messages in Perforce Details.
  - CCR may allow progression of change despite error messages.
    - Does not mean should progress CCR.
    - Flexibility to progress despite error necessary in some **rare** situations.
- Ignoring errors without fully understanding and resolving has caused crises.

Quiz: Perforce Details Section

Question:

Which one of the following is NEVER shown in the Perforce Details section of a CCR?

A. View Transport Log link.

B. Log into Perforce link.

C. List of submitted changes.

D. Access Token.

E. Link to directly edit item.

Quiz: Perforce Details Section

Answer:

E. Link to directly edit item.

There is no way to directly edit an item in the Perforce Details section.

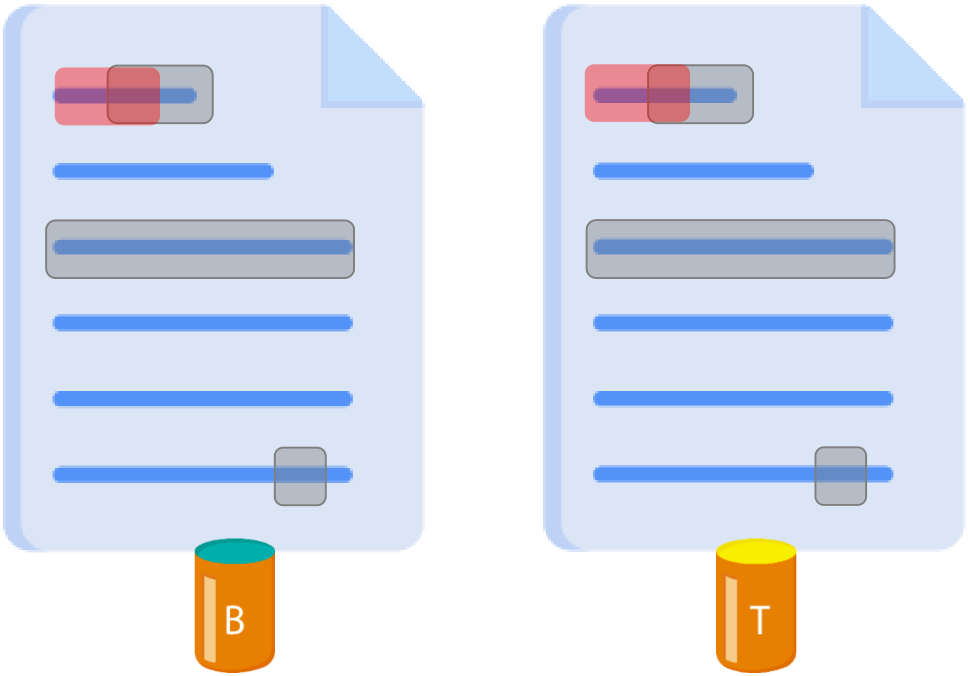
Part 2: Moving Changes Between Perforce Branches

Items and Diff Chunks

- Diff chunk = sections of file that differs from previous revision.
  - Insert, update, or delete.
- Perforce identifies and records diff chunks.
- Enables independently moving different CCRs editing different sections of item.

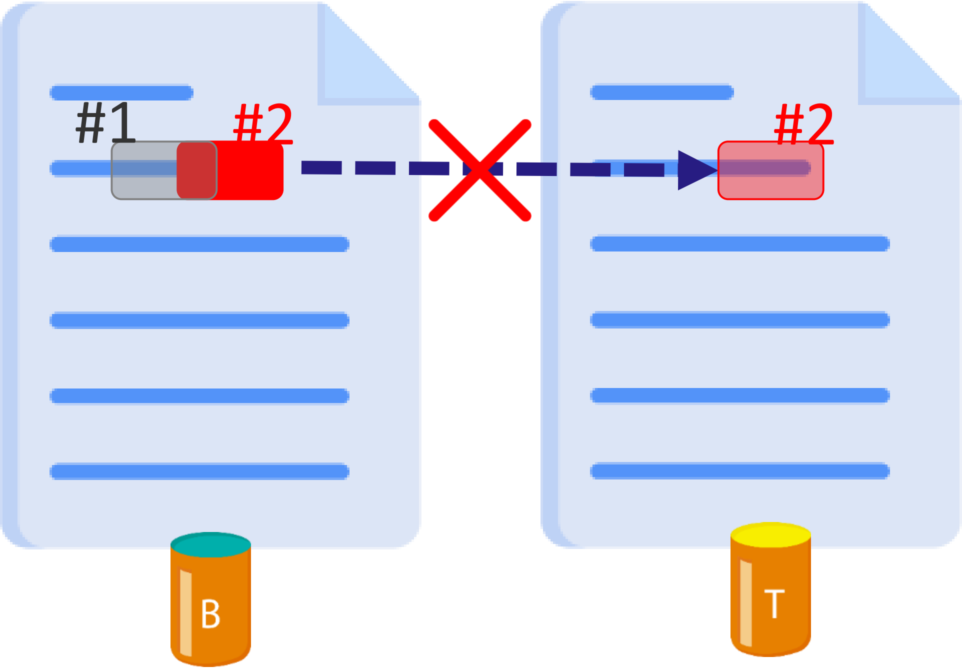
Moving Diff Chunks Between Branches

- Integrations only move diff chunks associated with that CCR record.
  - Exception: entire file copied if does not exist in target branch.
- Integration successful if base for each diff chunk in source item exists in target item.



Merge Conflicts

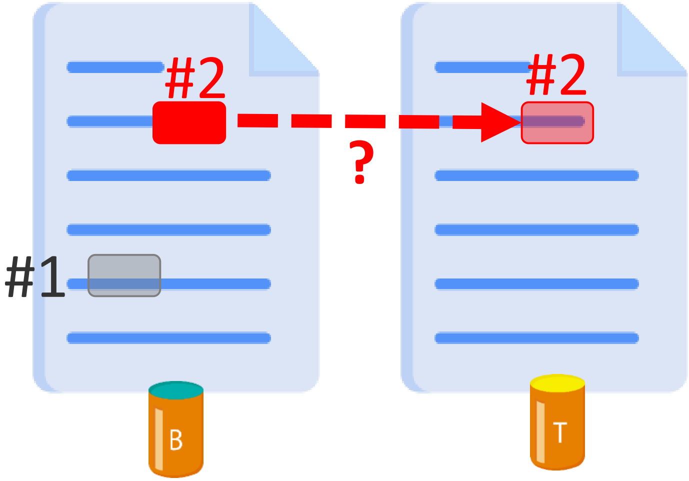
- Integration fails with merge conflict error when base for any diff chunk in source item does not exist in target environment.



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: False.

Since change #2 does not overlap with #1, change #2 can successfully progress to TEST without #1.

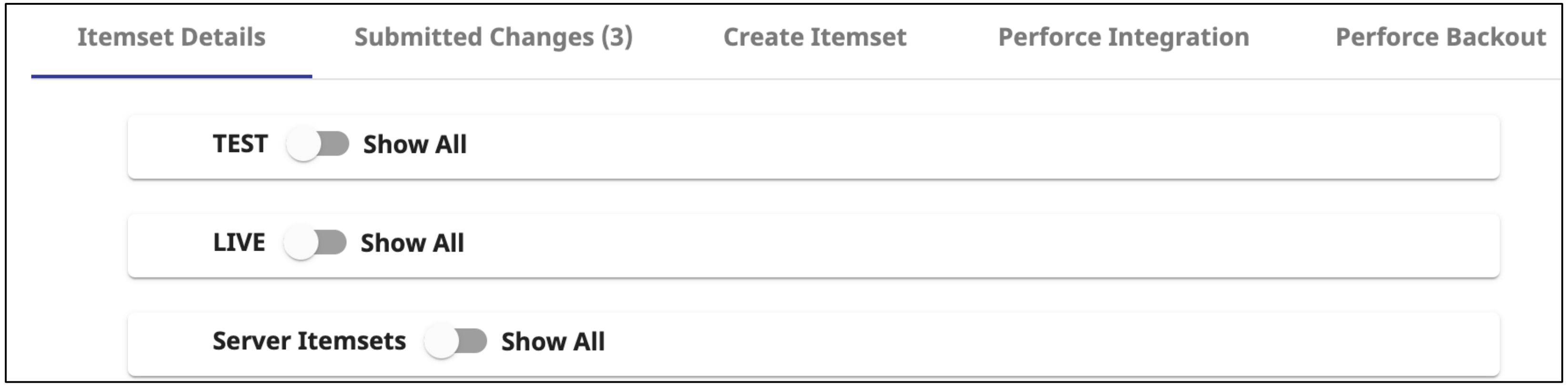
Part 3: Understanding ItemSets

ItemSet Contents

- ItemSets contain:
  - Metadata.
  - Complete copies of each item.
    - Not just diff chunks!

ItemSet for Server vs ItemSet for Client

- CCR record lists ItemSets for server and ItemSets for client.
- ItemSets for server = submitted to CCR from BASE Environment.
  - Hidden if successfully committed to Perforce.
- ItemSets for client = ItemSets generated by CCR to deploy to Environments.
  - Hidden if successfully deployed to all required environments.
- Toggle to view hidden ItemSets.



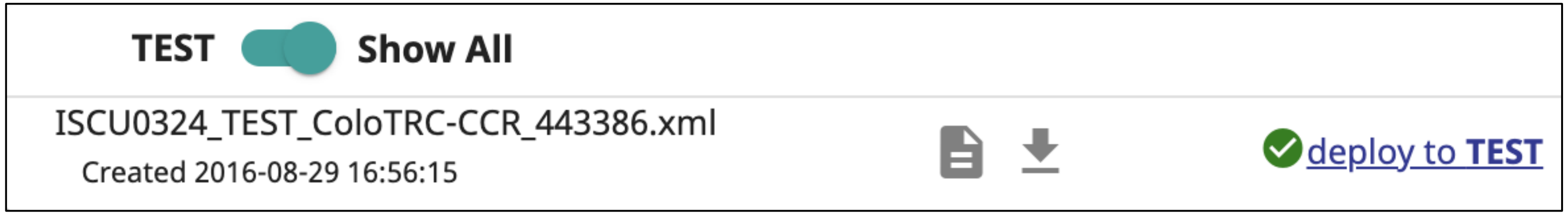
TEST Show All

LIVE Show All

Server Itemsets Show All

ItemSets for Client Status

- Undeployed. ●
  - Have not attempted to deploy ItemSet.
- Deployed. ✓
  - No further action required.
- Error. !
  - Must fix an issue before progressing CCR.
- Downloaded. ✓
  - Sent to environment filesystem but not loaded into namespace.



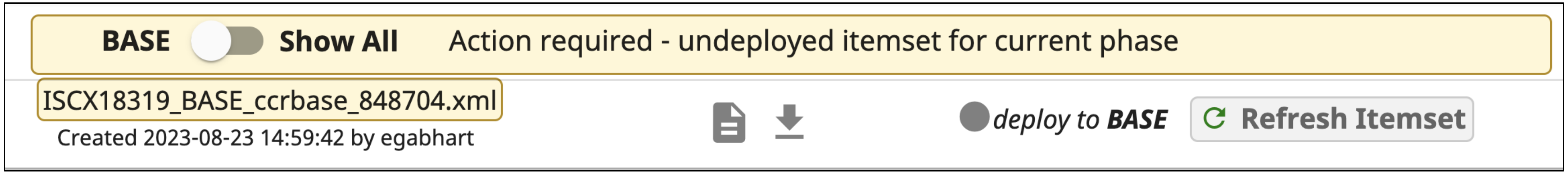
TEST Show All

ISCU0324_TEST_ColoTRC-CCR_443386.xml
Created 2016-08-29 16:56:15

✓ deploy to **TEST**

ItemSets for Client Status: Stale ItemSet

- ItemSets expire after 24 hours.
  - Helps prevent deploying ItemSets with old versions of items.
- Click Refresh Itemset button to generate new ItemSet based on latest version of items.



BASE Show All Action required - undeployed itemset for current phase

ISCX18319_BASE_ccrbase_848704.xml

Created 2023-08-23 14:59:42 by egabhart

Quiz: ItemSet Contents

Question:

ItemSets contain metadata and diff chunks of items (rather than the full item) associated with a specific CCR.

True or False?

Answer: False

Diff chunks associated with a CCR are integrated to a target branch and merged in to create a new revision of an item and associate those revisions with the CCR (via a new changelist). ItemSets contain all of the new revisions (complete files) of the items for that CCR.

Part 4: CCR Source Workspace

Source Workspace Read/Write File Settings

- Read/Write setting of files part of concurrency control framework.
  - Do not change read/write setting of any files.
  - Read only = not checked out.
  - Read/Write = checked out.

CCR Branches

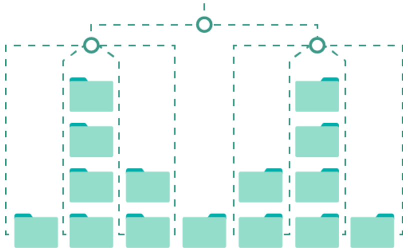
- Branch = folder location for items stored within Perforce:
  - //custom_ccrs/<Cc>/<Site>/<Sys>/<Env>/.
    - custom_ccrs – top level for CCR apps.
    - <Cc> – Country code.
    - <Site> – Site code.
    - <Sys> – System code.
    - <Env> – BASE/TEST/UAT/LIVE.

[XXXX]

[SysCode]

CCR Branches (cont.)

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



Tier 1 Branch Subfolders

- Items automatically exported to appropriate subfolders of source workspace and imported on ItemSet delivery by CCR Client Tools based on item type:
  - cls/ (class files).
  - cspapp/ (web files e.g. JavaScript, CSP, CSS).
  - inc/ (include files).
  - prj/ (project files).
  - rtn/ (routines).
  - rul/ (Interoperability Message Rules).

Tier 1 Branch Subfolders (cont.)

- Items automatically exported to appropriate subfolders of source workspace and imported on ItemSet delivery by CCR Client Tools based on item type:
  - lut/ (Interoperability Lookup Tables).
  - schema/hl7/ (HLv2 Custom Schemas).
  - ds/ (Business Intelligence dashboard definitions).
- Other subfolders can be created at will (e.g. “/java”).
  - Write custom ImplementCCR routine or CCR Event Handler for automation.

Integration Rules for Special Subdirectories

- All items checked into source branch, except directories noted below, are integrated to target.
- /internal subdirectory never integrated to LIVE.
  - Use case: unit test files, sample data.
- /backup subdirectory never integrated.
  - Use case: System Default Settings.

Quiz: Source Workspace

Question:

Which of the following best describes the structure of the source workspace?

A. A directory structure containing numbered copies of each item, corresponding to different version of that item.

B. A collection of XML snippet files containing the ‘diff chunk’ of each edit submitted by users.

C. A file-based replicate of the Perforce branch for that environment, containing the same structures and the content of the ‘head’ revisions of the items.

Quiz: Source Workspace

Answer:

C. A file-based replicate of the Perforce branch for that environment, containing the same structures and the content of the ‘head’ revisions of the items.

For TEST and LIVE, the local workspace should always exactly match those branches; for BASE it should match the BASE branch except for items actively being edited.

Part 5: The Importance of Baselining

What is Baselining?

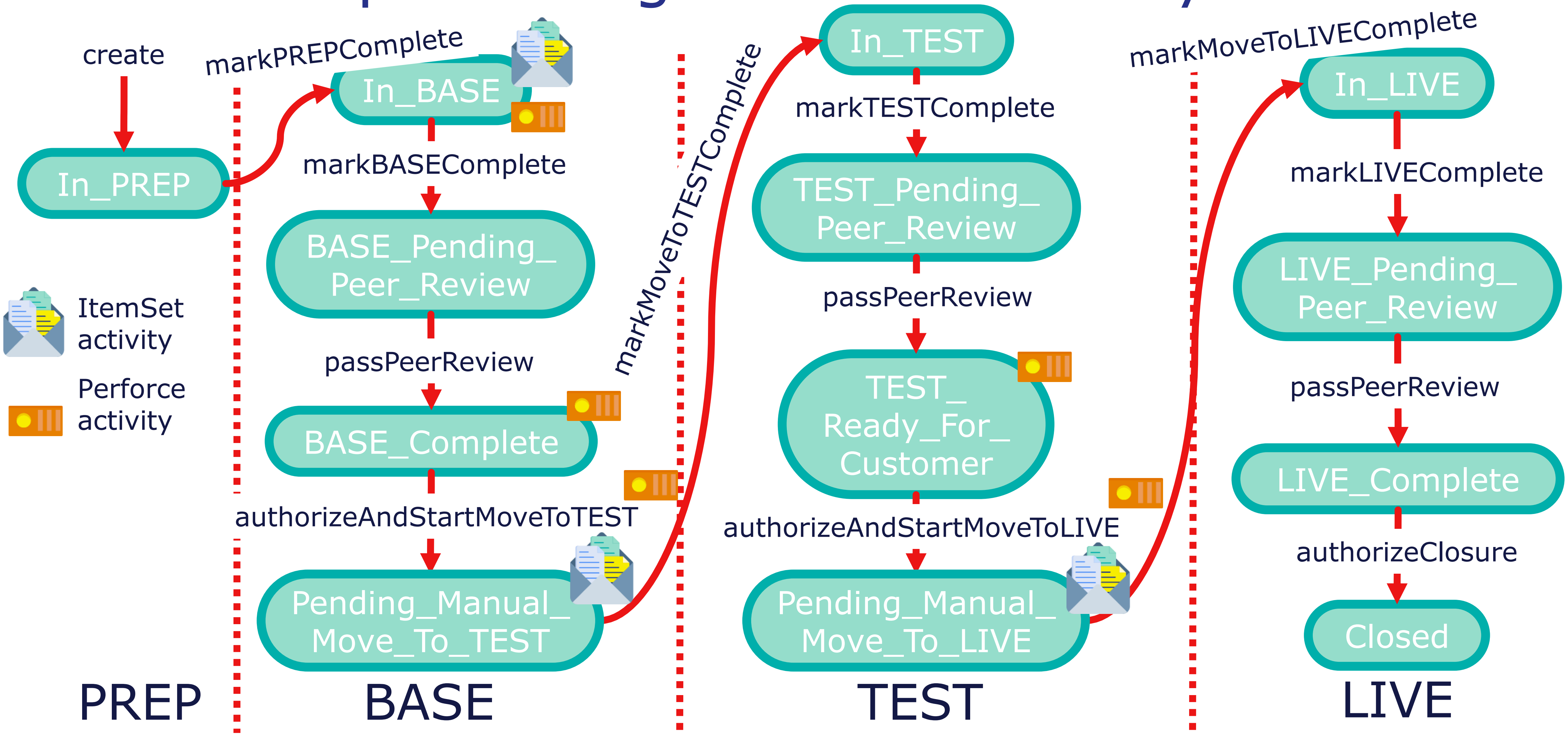
- Synchronizes Perforce and environments.
- InterSystems employee usually involved in process.
  - Tier 1 baselines sometimes require direct Perforce access.
  - Tier 2 baselines require direct Perforce access.

Why is Baselining Important?

- Establishes starting point in Perforce.
- Editing item that was never added to Perforce means:
  - Cannot diff change.
    - Perforce does not have previous revision.
  - Cannot backout.
    - Would delete entire item.
- Establishes clean integration history.
  - Bad practices lead to diff chunks inconsistent between environments.

Part 6: Actions Requiring Perforce Account

CCR Transport along the CCR Journey…



Actions in CCR that Require Perforce User

- Creating and deploying ItemSets to environments does not require Perforce account.
- Actions that require being logged into Perforce.
  - Clicking startMoveToXXXX transition on Tier 1 or Tier 2 CCRs.
  - Manually triggering integration or backout through Perforce Details section of CCR.
  - Integration previews.

Actions in CCR that Require Perforce User (cont.)

- Behavior when attempting action but not logged into Perforce via CCR.
  - Users with no Perforce account blocked from performing action.
  - Users with Perforce account but not logged in via CCR:
    - [Log into Perforce] field flashes and turns yellow.
  - Exception:
    - Previews in CCR workflow will be bypassed if user not logged into Perforce.

Uploading ItemSet from environment to CCR

- If user does not have Perforce account:
  - ItemSet queued on CCR server.
  - Entire queue for an organization committed to Perforce when either:
    - Any user logged into Perforce uploads an ItemSet.
    - Any user logged into Perforce clicks [commit queue] link next to uncommitted ItemSet in CCR record.
  - When queue committed, all changelists reference user who caused queue to clear.

Quiz: Perforce Account Required

Question:

Only a Perforce user can perform a ‘startMoveTo’ transition for Tier 0, 1, and 2 CCRs. True or False?

Answer: False.

Tier 0 CCRs do not trigger any changes in Perforce therefore no Perforce usernames are required.

Quiz: Perforce Account Required 2

Question:

Which of the following actions require being logged into Perforce?

A. Viewing submitted changes.

B. Deploying an ItemSet.

C. Performing a passPeerReview transition.

D. Performing the authorizeAndStartMoveToTEST transition.

Quiz: Perforce Account Required 2

Answer:

D. Performing the authorizeAndStartMoveToTEST transition.

The authorizeAndStartMoveToTEST transition must perform the integration from BASE to TEST in Perforce in order to successfully complete. The passPeerReview transition will bypass the preview integration if you are not logged into Perforce.

Part 7: Advanced Information

CCR Client Tools – What is included?

- %Studio.SourceControl.* classes.
  - ISC.cls - Source control hooks class; controls Atelier/Studio behavior.
  - UI.cls - User interface for CCR actions (ItemSets bundling, loading, etc.).
  - Change.cls - Persistent record of “who did what?” (aka uncommitted queue).
  - ItemSet.cls - Persistent class for ItemSets.
  - ItemSetWS.cls - Web service client for moving ItemSets to/from CCR server.
  - CCREventHandler.cls - Abstract super-class for creating custom CCR event handlers for your application (aka “build hooks”).

CCR Client Tools – What is included? (cont.)

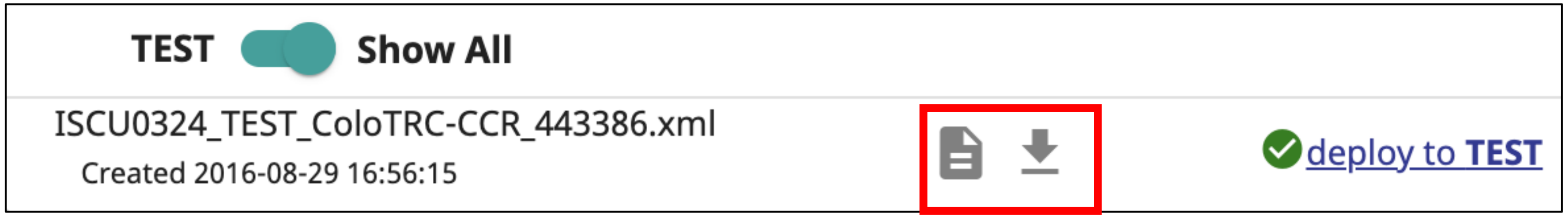
- Routines.
  - %buildccr.mac - Command Line interface for CCR configuration, etc.
  - %buildccrRefresh.mac - Logic for refreshing namespace from source workspace.

ItemSet Logs

- All ItemSet activity in environment logged.
- Logs accessible via Perforce Details on CCR record.
- If log fails to upload to CCR, access in InterSystems command prompt of environment.
  - Do ItemSetLog^%buildccr.

View ItemSet information

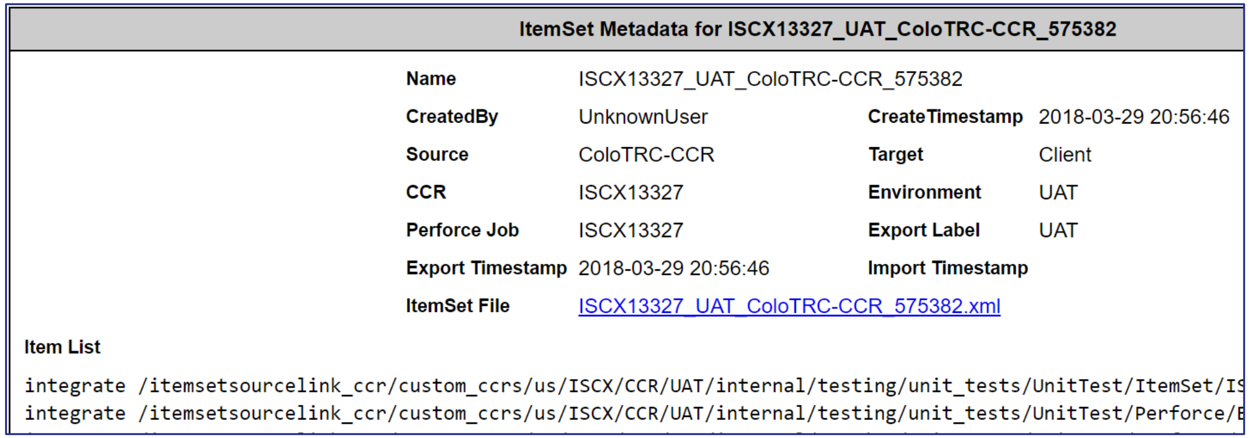
- 📄 = View ItemSet activity log.
  - Logged activitity specific to this ItemSet.
- ⬇ = Download file locally.



ISCU0324_TEST_ColoTRC-CCR_443386.xml
Created 2016-08-29 16:56:15

ItemSet Structure

- Click the [details] link by an ItemSet to see the following:
  - Name (a concatenation of CCR ID, environment, source server name, ItemSet ID).
  - Target (server or client).
  - Source server name.
  - CCR / Perforce job.
  - Environment.
  - Item List (includes action).
  - Payload.
    - UTF8 encoded XML block of all content for items in the ItemSet.



ItemSet Metadata for ISCX13327_UAT_ColoTRC-CCR_575382

Name	ISCX13327_UAT_ColoTRC-CCR_575382		
CreatedBy	UnknownUser	**CreateTimestamp**	2018-03-29 20:56:46
Source	ColoTRC-CCR	**Target**	Client
CCR	ISCX13327	**Environment**	UAT
Perforce Job	ISCX13327	**Export Label**	UAT
Export Timestamp	2018-03-29 20:56:46	**Import Timestamp**	
ItemSet File	ISCX13327_UAT_ColoTRC-CCR_575382.xml		

Item List

```
integrate /itemsetsourcelink_ccr/custom_ccrs/us/ISCX/CCR/UAT/internal/testing/unit_tests/UnitTest/ItemSet/IS
integrate /itemsetsourcelink_ccr/custom_ccrs/us/ISCX/CCR/UAT/internal/testing/unit_tests/UnitTest/Perforce/E
```

Source Workspace and ItemSets

- Creation of an ItemSet on client.
  - Items copied from workspace into ItemSet Payload XML file created in /<Workspace>/ItemSets/Export/<SiteCode>/ directory.
    - Payload file has the same filename as the ItemSet but with a “p” at the end.
  - ItemSet file containing Payload plus metadata created in same location.
  - Items marked as read-only in workspace following ItemSet creation.
- Loading of an ItemSet from server.
  - ItemSet downloaded from CCR server and deposited into /<Workspace>/ItemSets/Import/<SiteCode>/.
  - Payload file extracted from ItemSet.
  - Items extracted from Payload file and deposited in workspace branch locations.
    - Existing files marked Read/Write and then overwritten with contents of ItemSet Payload.

Summary

- What are the key points for this module?

ICC 430: CCR Transport – Client Configuration

InterSystems
Creative data technology

Objectives

- Understand how to respond to prompts from Configure^%buildccr.
- Maintain CCR Client Tools.
- Map Configuration globals.
- User permission requirements.
- Secondary BASE environments.
- ECP configuration.

Part 1: Introduction to CCR Client Configuration

Updates to CCR

- 2 different code bases.
  - Centralized CCR application.
  - CCR client tool code in each environment.
- Updates to centralized CCR application instantly available.
  - Anything in CCR UI.
- Client tool updates implemented with Tier 1 CCR.
  - Updates may include:
    - Bug fixes.
    - New features in client-side UI such as bundle and upload process.

CCR Client Tools

- Stored in IRISLIB or CACHELIB database of all products.
  - Shared among all namespaces (environments) in installation.
- Set of classes and routines.
  - %Studio.SourceControl package.
- ^%buildccr.
  - Primary configuration routine.
- Provides GUI and APIs for communicating with CCR.
- Can update without upgrading product version.

Maintaining CCR Client Tools

- Always update CCR Client Tools through same System.
  - Maintains accurate version history in Perforce.
    - After configuring, progress Client Tool update CCR to baseline tools and receive any recent updates.
  - Configure system for %SYS namespace or HSCUSTOM namespace.
    - Also use for instance wide changes such as memory allocations.
- Check version using do Version^%buildccr.
- Reload tools after in-place upgrade.
  - Upgrade introduced outdated client tools because IRISLIB/CACHELIB replaced.
  - `Do ReloadTools^%buildccr.`

How To: Update CCR Client Tools

1. Go to appropriate System Details page.
2. Click Update Client Tools button.
   - Can only be actioned by Perforce users.
   - Integrates current client tools to BASE branch of System.
   - Creates a CCR in In_BASE state with:
     - Prefilled documentation for process.
     - Changelist from integration attached.
     - ItemSet ready for deployment to BASE.

How To: Update CCR Client Tools (cont.)

3. Progress generated CCR through normal workflow.
   - Execute prefilled implementation plan and testing plan in all environments at appropriate times in workflow.
   - Document and perform peer reviews as with any other Tier 1 CCR.

Quiz: Updating CCR Client Tools

Question:

Which of the following is true about updating CCR Client Tools via the System Details page? Select all that apply.

A. Automatically creates a CCR which documents how to update the client tools.

B. Will integrate the head revision of these files into the LIVE branch for that System.

C. Can only be actioned by a Perforce user.

D. Requires a Perforce user to be logged in.

Quiz: Updating CCR Client Tools (cont.)

Answer:

A. Automatically creates a CCR which documents how to update the client tools.

C. Can only be actioned by a Perforce user.

D. Requires a Perforce user to be logged in.

It will integrate the head revision of these files into the BASE branch for that System, therefore B is incorrect.

CCR Client Configuration

- Each environment in Tier 1 or 2 system requires configuration.
- Configure as connected or disconnected.
- Accurate configuration critical to ensure:
  - Correct procedure enforcement.
  - Appropriate use of Perforce structures.
  - Proper behaviour of CCR Transport.

CCR Client Configuration (cont.)

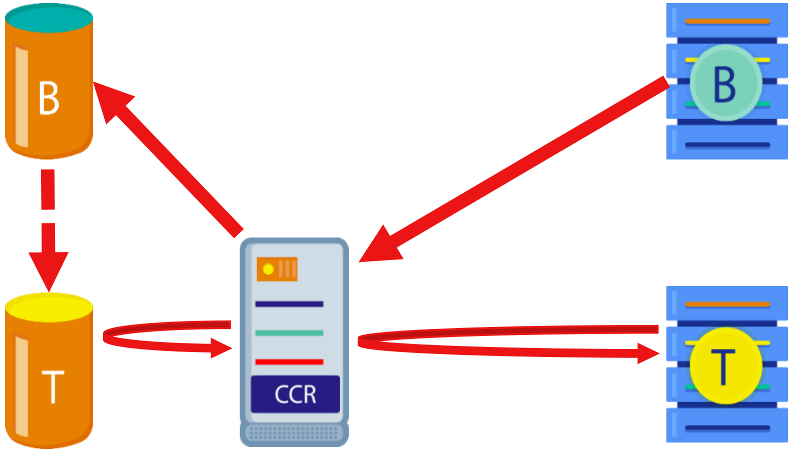
- CCR client configuration steps:
  - Create a CCR to document responses to configuration prompts.
  - Run Configure^%buildccr routine in BASE.
  - Progress CCR record and run Configure^%buildccr in for each environment in each phase.

Review: Connected vs. Disconnected

- Clients can be configured as connected or disconnected.
- Connected environment.
  - Communicates directly with the InterSystems Perforce server.
  - Must be within the InterSystems firewall.
  - Uses the p4 client to issue commands to Perforce.
- A Disconnected environment.
  - No direct access to Perforce.
  - Uses CCR server as a proxy to submit changes to Perforce.
  - Uses ItemSets for moving items to and from CCR server.
  - Can be within or outside the InterSystems network.
  - All customer environments.

Review: Changes in non-BASE

- All changes to controlled items must originate in BASE.
  - Exception: LIVE-only workflows.
- Never make changes to controlled items directly in TEST/UAT/LIVE environment.
  - Perforce must be the source of truth.
  - CCR Transport always uses complete copies of items in Perforce.
- Proper configuration locks TEST/UAT/LIVE from changes outside CCR workflow.



Connected vs Disconnected

- Connected BASE when possible.
  - Better diffing tools.
  - Possible for each developer to have their own BASE.
    - See ICC440 for BASE development options.
- TEST, UAT, LIVE disconnected.
  - No good reason to be connected and easy to deploy ItemSets.

Quiz: CCR Client Configuration

Question:

It is recommended that LIVE environments are configured as ‘Connected’. True or False?

Answer:

False.

All changes will be moved to LIVE environments via ItemSets even if they are on the InterSystems network.

Quiz: CCR Client Configuration

Question:

Locked LIVE environments must be unlocked to deploy ItemSets. True or False?

Answer:

False.

Part 2: Creating a CCR To Configure the Client

Creating the CCR Record

- Create organization and System in CCR.
  - See ICC320.
- Create Tier 0 CCR.
- Suggested documentation:
  - Title: Configuration of CCR Client Tools.
  - Description: Initial configuration of CCR Client Tools for environments in this System.
  - Type-SubType: Application Customization - <Product>.
  - CCR Tier: 0 - Documentation Only.

Creating the CCR Record (cont.)

- Suggested documentation:
  - Modified Items: source control and CCR Client Settings for <namespace>.
  - Impacted Areas: behavior of Studio, Atelier, and Management Portal source control hooks.

CCR Documentation: In_BASE

- Implementation plan.
  - < Specify responses to Configure^%buildccr >.
- Backout plan.
  - If part way through running routine, interrupt routine.
- Testing plan.
  1. Capture output of Summary^%buildccr.
  2. Test download/upload of ItemSets by progressing a Client Tool Update.

Part 3: Configuring CCR Client Tools

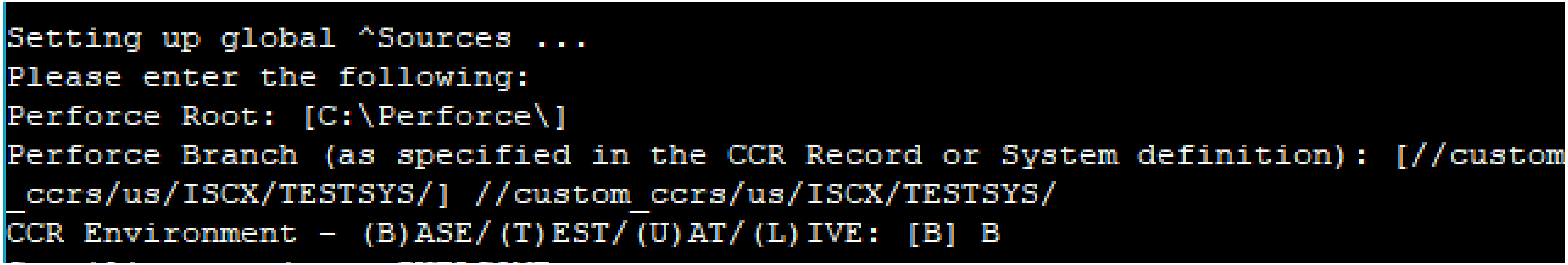
Configuring CCR Client Tools

1. Open Terminal session on target instance.
2. Switch to namespace to be configured.
3. Enter do Configure^%buildccr.

Configure^%buildccr prompts

- Perforce Root.
  - Location on file system where source workspace structures will be built.
    - Example: C:\Source.
- Perforce Branch.
  - Copy paste from Perforce Details pane of CCR documenting configuration process.
    - Example: //custom_ccrs/us/ISCU/TESTSYS/.
- CCR environment.
  - Type character corresponding to environment type:
    - **B** for BASE / **T** for TEST / **U** for UAT / **L** for LIVE.

Configure^%buildccr prompts (cont.)



```
Setting up global ^Sources ...
Please enter the following:
Perforce Root: [C:\Perforce\]
Perforce Branch (as specified in the CCR Record or System definition): [//custom
_ccrs/us/ISCX/TESTSYS/] //custom_ccrs/us/ISCX/TESTSYS/
CCR Environment - (B)ASE/(T)EST/(U)AT/(L)IVE: [B] B
```

Prompts for Non-BASE Environments

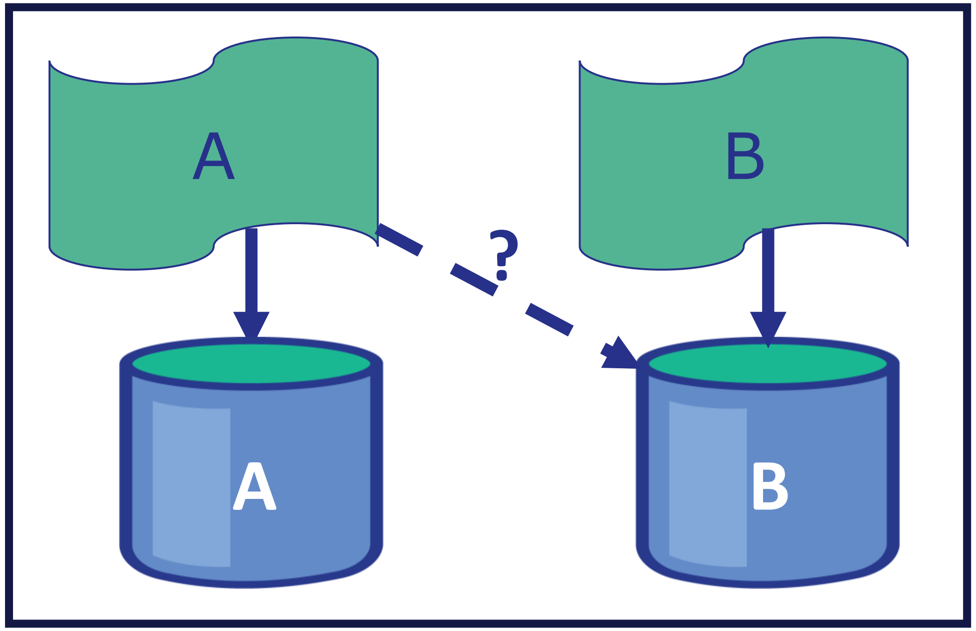
- Lock environment.
  - Disables check out and add to source control functionality to prevent editing any item in source control.
  - **Always respond yes.**
- If respond yes to lock, prompts whether to Admin Lock.
  - Admin lock removes information in UI on how to unlock environment.
    - Most users should not even know how to unlock environment.
    - Unlocking environment causes missing changes because Perforce no longer source of truth.
  - Best practice: yes.

Configure^%buildccr Prompts (cont.)

- Add or Change any CSP mappings.
  - Usually respond no.
- Automatically tag with Perforce version.
  - Adds SrcVer parameter when saving class definition if it does not exist.
  - Indicates branch and revision of class retrieved from Perforce.
  - Best Practice: yes.

Configure^%buildccr Prompts (cont.)

- Ensure mapped items cannot be changed in other namespaces.
  - Concurrency control risks.
- Treat items mapped from other databases as read-only in Studio.
  - Also applies to Management Portal pages with hooks.
  - Yes = can only edit items in default database(s) of namespace.
  - Best practice:
    - Tier 1 = yes.
    - Tier 2 = probably no.
- Use separate Systems for namespace A and B.



Configure^%buildccr Prompts (cont.)

- Keep history of changes.
  - Determines whether to maintain or delete metadata for changes.
    - Username, time stamp, filename, and more.
    - Metadata stored in %Studio_SourceControl.Change table.
    - Does not maintain copies of changes items or what was changed; only metadata.
  - Yes = set committed field for that change to true, with timestamp.
  - No = committed changes deleted from change table.
  - Best practice: yes because can be useful in debugging.
    - Consider setting up task to purge %Studio_SourceControl.Change table.

Configure^%buildccr prompts (cont.)

- Will this namespace ever communicate directly with the ISC Perforce server?
  - (Y)es if want connected mode.
  - (N)o if want disconnected mode.
  - Remember:
    - All customers work with disconnected.
    - Almost all non-BASE environments are disconnected.

Prompts for Connected Mode: Initialize Perforce Credentials

- Would you like to do this now?
  - Yes.
- Will more than one person be developing in this instance?
- Please enter the following:
  - Perforce Username?
  - Perforce Password?
- Perforce workspace?
  - Must first create workspace using p4v.

Prompts for Connected Mode: Initialize Perforce Credentials (cont.)

- Do you want to use a Perforce Alt Directory?
- Always sync the head revision of a file prior to check-out?
  - In other words: always edit latest revision (as opposed to revision currently in environment).
  - Best practice: yes.

Configure^%buildccr prompts (cont.)

- Will you be importing/exporting ItemSets from this namespace?
  - Always respond Yes.
  - Creates /itemsetsourcelink CSP Application for use in CCR Transport if it does not exist.
  - Creates (if necessary) and initializes ItemSetClient SSL configuration.
- Configuration summary displayed.
  - Copy-paste to Testing Steps Taken in XXX field.

Result of Configure^%buildccr

- Environment configured to use CCR.
- Within Studio, when create new item, will be prompted to add to source control.
- After baselining, existing items must be checked out in BASE for editing.
- In non-BASE environments, only able to edit items in source control through CCR process.
- Studio will automatically import and export items added to source control in the proper format and into the proper directory structure.

Quiz: Configure^%buildccr

Question:

Which of the following is a prompt when configuring client tools for BASE? Select all that apply.

A. Whether the instance will communicate directly with Perforce.

B. The location on the file system where the workspace structures will be built.

C. Whether classes will be automatically tagged with Perforce version.

D. Whether to keep a history of changes.

E. Whether to treat items mapped from other databases as read-only in Studio.

Quiz: Configure^%buildccr

Answer:

All of the above.

Part 4: Progressing the Client Configuration CCR

Progressing the CCR Record

- Progress CCR to next phase and run Configure^%buildccr for all environments in each phase.
- At any time, the following command will show the current CCR configuration information for a particular namespace:
  - `do Summary^%buildccr.`

Summary^%buildccr

- Environment Details: “Org”, “System”, “Environment.”
- Studio Source Control Class: Version #, SrcVer Auto-insert, Mapped Item Lock.
- Studio Source Control UI: Version # and $Id.
- Change Class: Version#, $Id, Maintain History.
- ItemSet Class: Version#, $Id.
- ItemSetSourceLink (for Export/Import of ItemSets): CSP Application Name and Path.
- ItemSet WebService Class: Version #, $Id.
- ItemSet SSL Configuration.

Summary^%buildccr (cont.)

- Perforce Details:
  - Perforce Root: e.g. C:\Perforce\.
  - Current Perforce Branch: e.g. C:\Perforce\custom_ccrs\us\ISCX\TESTSYS\BASE\.
  - CSP Application Mappings for Perforce.
  - Perforce Connectivity State.
  - Development mode: e.g. Single-User.

Summary^%buildccr (cont.)

- Perforce Details:
  - Credentials:
    - Perforce Username.
    - Perforce Password.
  - Perforce Workspace.
  - Perforce Directory.
  - Perforce AutoSync: e.g. Enabled.

Part 5: Other ^%buildccr options

Other ^%buildccr options

3. `do Download^%buildccr` - Download an ItemSet from CCR Server.
4. `do Load^%buildccr` - Load ItemSet contents into environment.
5. `do Bundle^%buildccr` - Bundle uncommitted changes into an ItemSet.
6. `do Upload^%buildccr` - Upload a created ItemSet to CCR Server.

Other ^%buildccr options

7. `do DisplayUncommitted^%buildccr` - Display list of local uncommitted changes.
8. `do ItemSetLog^%buildccr` - find and display the log for an ItemSet.
9. `do Version^%buildccr` - report version of CCR client classes.
10. `do Refresh^%buildccr` - refreshes namespace from sources on disk.
11. `do ReloadTools^%buildccr` - reloads CCR Client Tools from local disk.

Part 6: Additional Client Configuration Considerations

Unix® and Linux Source Workspace Ownership

- Files and folders must be owned by user running Management Portal and Studio processes.
  - Usually irisusr or cacheusr.

Minimum User Permissions for CCR Transport

- SQL Select on %Studio_SourceControl.Change table.
  - BASE only.
- Read on %DB_IRISLIB or %DB_CACHELIB.
- Read on %DB_IRISSYS or %DB_CACHESYS.
- Read/Write on resource protecting default database(s) for namespace.

Minimum User Permissions for CCR Transport (cont.)

- Appropriate privileges for making the change.
  - For example, use on %Development resource required to make changes in Studio.
- Create role for privileges specific to using CCR Transport to ensure anyone using CCR has appropriate privileges.

Secondary BASE Environments

- Cannot have multiple disconnected BASE environments able to submit changes to Perforce.
  - User A checks out item X from BASE1.
  - User B checks out item X from BASE2.
    - No concurrency control across environments, therefore checkout successful.
  - User A bundles and uploads their version of item X.
  - User B bundles and uploads their version of item X.
    - Item does not contain user A’s changes.
    - Current revision of item no longer contains user A’s changes.

Secondary BASE Environments (cont.)

- To lock secondary BASE environments:

  ```
  do ##class(%Studio.SourceControl.ISC).Lock(1).
  ```

  - Argument of 1 admin locks.

Secondary BASE Environments (cont.)

- If want to be able to make changes in secondary BASE environments for testing,

  ```
  set
  ^SYS("SourceControl","ItemSetConfig","NoUpload")=1.
  ```

  - In this case, do not lock environment.
  - NoUpload flag disables uploading changes to CCR.
  - Does not mean want secondary environments always different.
    - Only for testing before developing change in primary BASE environment.
    - Always use undo checkout to revert changes.
  - When an item is checked out, cannot deploy ItemSets containing that item to that environment.

Mapping out Configuration Globals

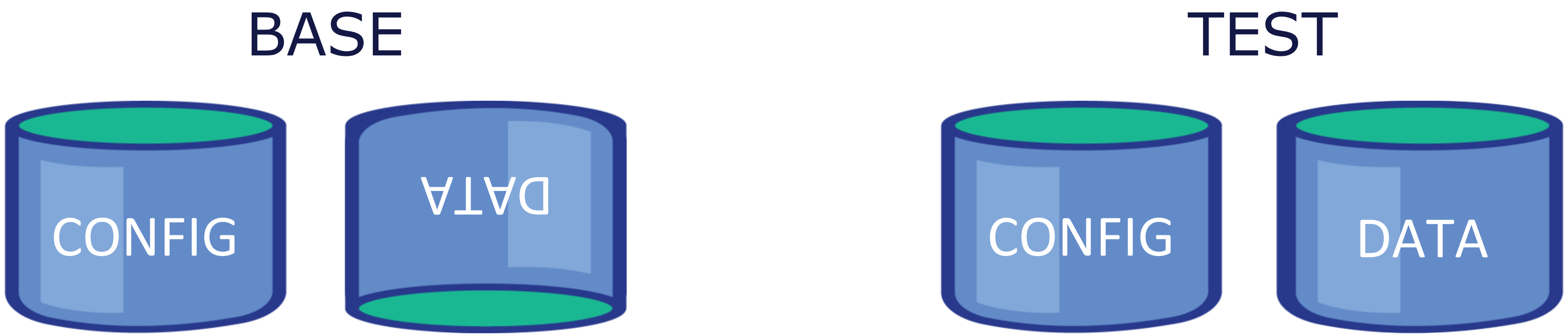
- CCR client configuration globals control behaviour for namespace.
- Some sites prefer to have configuration globals in separate database.
  - Facilitates data refresh on testing environments.

Mapping out Configuration Globals (cont.)

- Globals to map:
  - ^Sources.
  - ^SYS("SourceControl").
  - ^SYS("SourceControlClass").
  - ^SYS("ChangeControlClass").
    - TrakCare environments only.
- ^%SYS("SourceControl") also used but do not map.

Mapping out Logging and History Globals

- Logs and change history should not be mapped.
  - Do not want to maintain during refresh.
  - Should be carried over from source of refresh.
- Logging and history globals include:
  - ^Studio.SourceControl.Change*.
  - ^Studio.SourceControl.ItemSet*.



Part 7: Configuring CCR for ECP Deployments

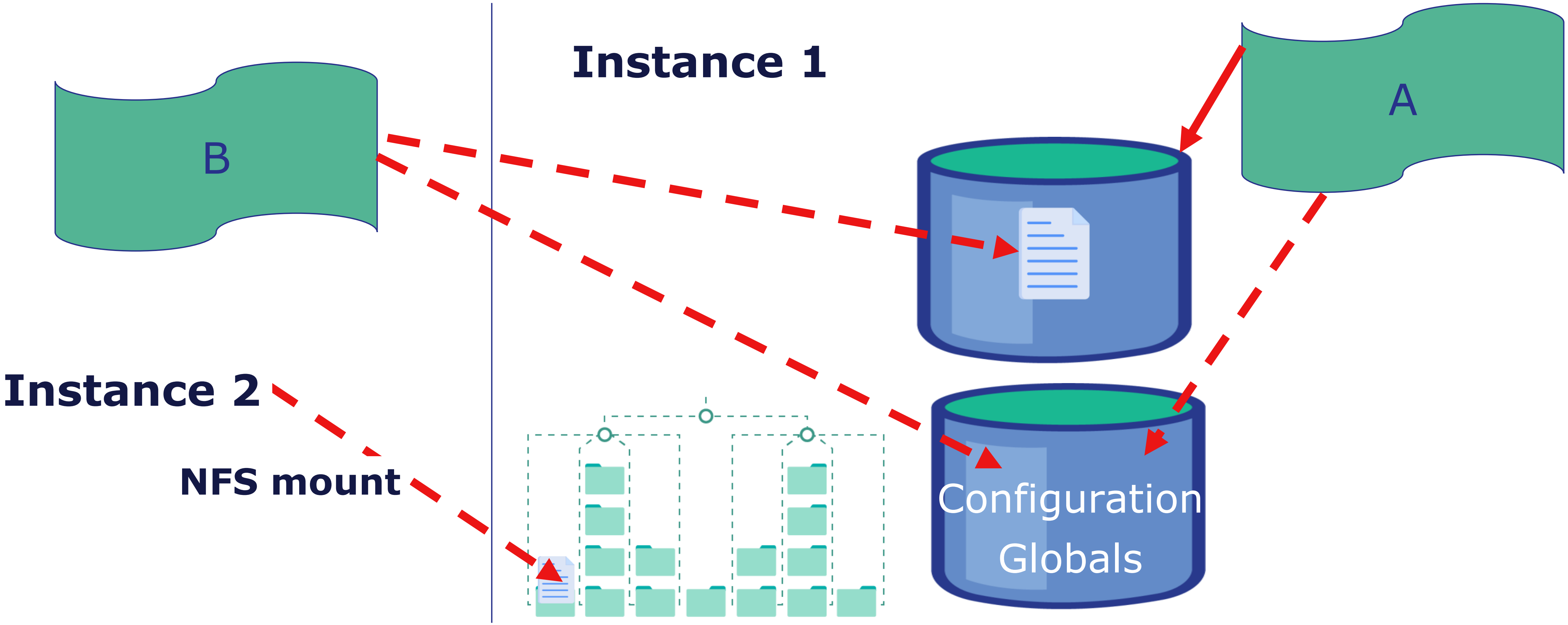
Complexity of CCR for ECP Deployments

- 2 ways each node must be able to communicate:
  - CCR client configuration, Logging and Storage globals.
    - Concurrency control.
  - Source workspace in file system of 1 node.
- When items are checked out:
  - Checkout information is stored in the %Studio.SourceControl.Change table.
  - Exported files (stored in Perforce workspace) are changed between read-only and read/write.

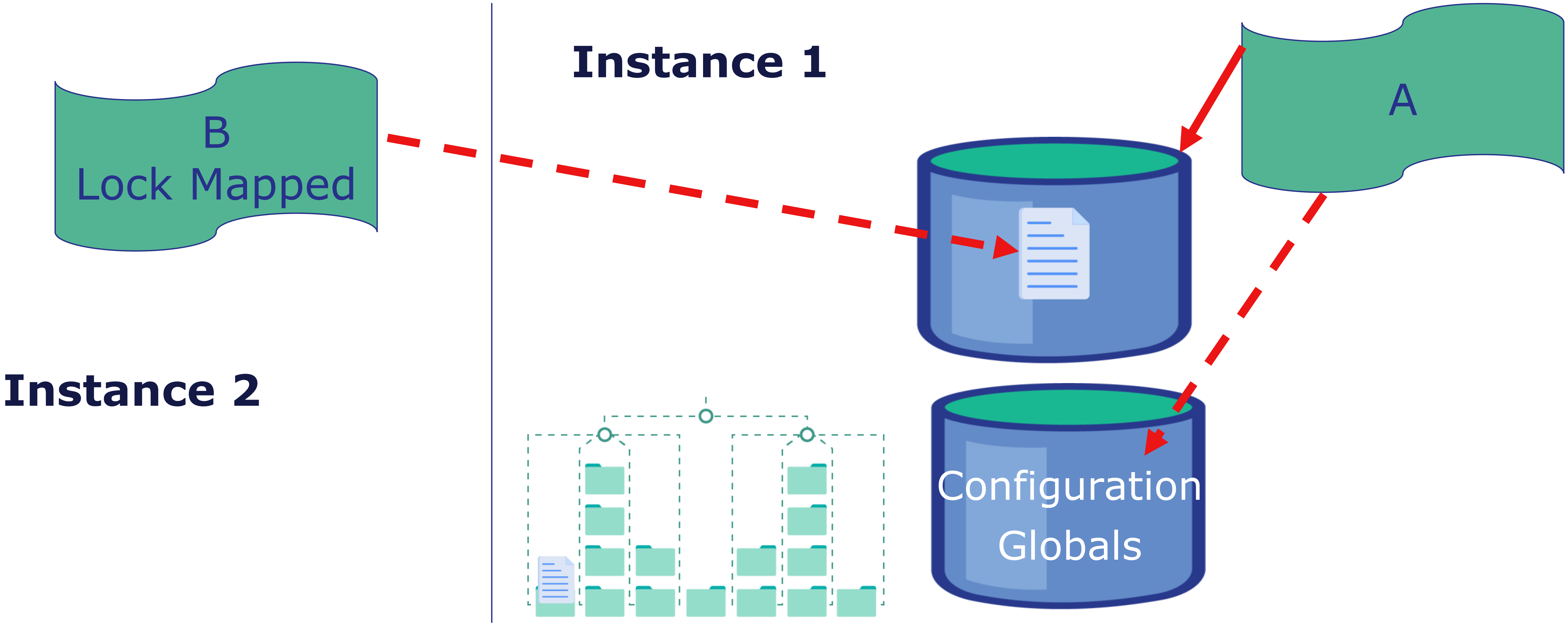
Complexity of CCR for ECP Deployments (cont.)

- Any instance running CCR client logic must have access to both of these in order to function properly.
  - Alternatively, nodes that cannot access this information must be locked.

Configuring CCR for ECP Deployments (choice 1)



Configuring CCR for ECP Deployments (choice 2)



Configuring CCR for ECP Deployments (cont.)

- Architecturally there are two options for using CCR in an ECP configuration:
  - Create a single DB on the DB Server which contains all CCR globals and create an NFS mount point which all nodes can use as the shared Perforce workspace.
    - NOTE: Even with a mapped set of configuration globals, Configure^%buildccr must be run in each instance at least once to initialize the SSL configuration required to use the CCR Web Services.
  - Only allow changes to be made on a single instance in the ECP cluster.
    - Usually the data server.

Configuring CCR for ECP Deployments (cont.)

- NFS mount considered weak point, detracting from robust ECP architecture.

Configuring CCR for ECP Deployments (cont.)

- Therefore, preferred architecture for LIVE is to have single point of entry for manual changes. Typically consists of:
  - Standalone VM running Apache.
    - Directs requests directly to DB server as front end for changes made via Ensemble, InterSystems IRIS™ Integration, HealthShare, or TrakCare.
  - Policy mandate that all Studio connections must be made directly to data server.
  - Application servers also require separate database of configuration globals.
    - If BASE, configure as locked using `do ##class(%Studio.SourceControl.ISC).Lock(1)`.

Quiz: Additional Client Configuration Considerations

Question:

CCR client configuration globals should be mapped to SYSCONFIG. True or False?

Answer:

True.

They should be mapped to SYSCONFIG in order to be retained during a refresh.

Quiz: Additional Client Configuration Considerations

Question:

CCR client logging and storage globals should be retained after a refresh. True or False?

Answer:

False.

Logs and change history from the refresh source should be reflected on the target after the refresh.

Quiz: Additional Client Configuration Considerations

Question: Which of the following is true about configuring CCR for ECP deployments? Select all that apply.

A. Requires additional configuration to ensure various pieces of the architecture stay in sync.

B. CCR client configuration and logging and storage globals need to communicate between nodes.

C. Different nodes require access to Perforce workspace file system (stored in the root node of ^Sources).

D. Any instance running CCR client logic must have access to information about checked out items.

Quiz: Additional Client Configuration Considerations (cont.)

Answer:

All of the above.

Summary

- What are the key points for this module?

ICC 440: CCR Transport – System Topology

InterSystems
Creative data technology

Objectives

- Deploy changes to secondary Environments.
- Identify Environment workflow options.
- Discuss private, connected BASE Environments.

Part 1: Changing System Workflow

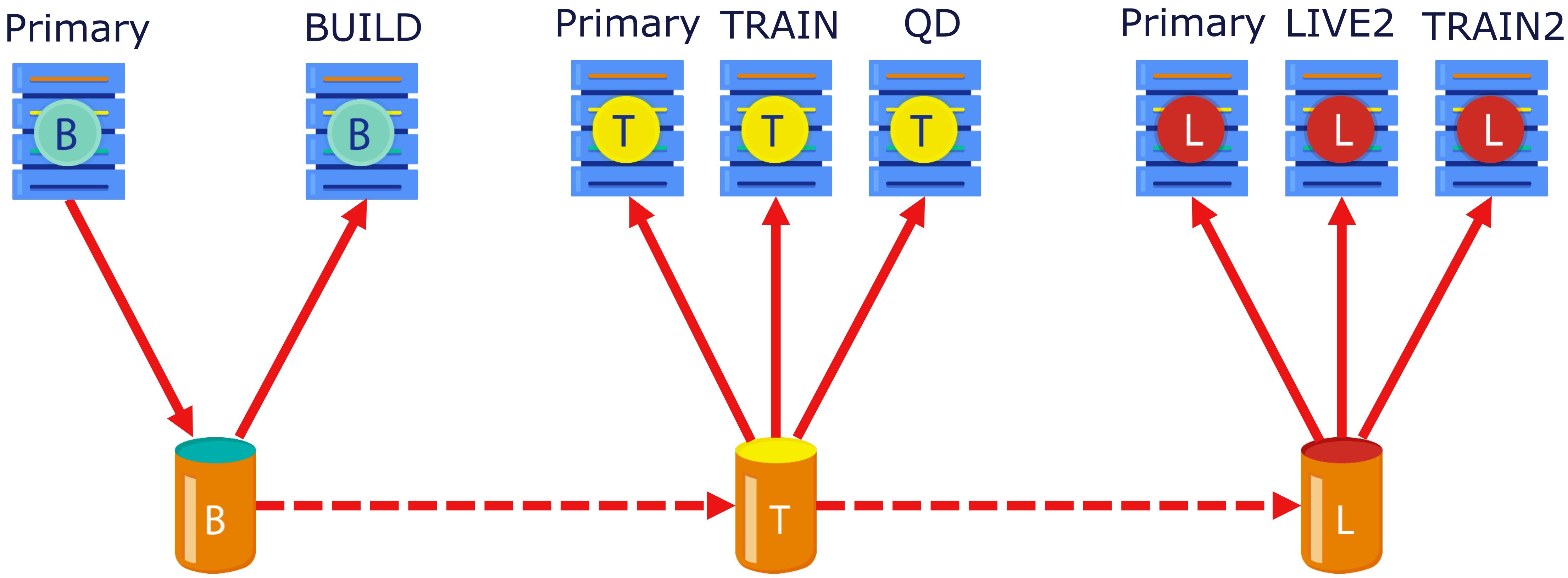
Review: Environment Workflow Options

- System workflow can be tailored to include only the Relevant environments.
- BASE-only or LIVE-only workflows are supported.
- Workflows with 2 or more environments must include BASE.
- Must update Perforce branches when changing included environments for Tier 1 and Tier 2 Systems.
  - Create new Perforce branch for new environments via integration.
    - This must be handled by an InterSystems employee.
  - Clone new environments from an existing environment.

Secondary Environments

- Additional environments to receive changes at same time as corresponding primary environment.
  - Any primary environment can have 1+ secondary environments.
- Secondary environments for BASE require deployment of ItemSet during In_BASE state.
  - Automatically created when submit changes if require ItemSet setting true.

Secondary Environments (cont.)



Secondary BASE Options

- Secondary BASE environments cannot identify whether items checked out on primary.
- Secondary environments should be either:
  - Read-only.
  - Disabled for upload.
    - Can checkout items but cannot upload changes.
  - See ICC430 for how to configure.

Secondary BASE Upload Disabled Workflow

1. Checkout item in secondary BASE to test possible change.
2. Use Undo Checkout action in secondary BASE.
   - Cannot deploy ItemSets containing items checked out in target environment.
3. Make and upload change in primary BASE.
4. Deploy ItemSet to all BASE environments.
5. Test in other BASE environments.
6. Progress CCR normally.

Deploying to Secondary Environments

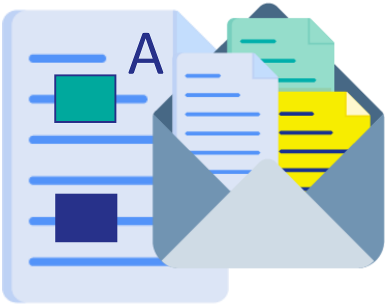
1. Make sure popups allowed by browser.
2. Log into Management Portal of all environments.
3. Click [Deploy to All Rqd] ItemSet deployment link.
    - New browser tab for primary environment and each secondary.
    - All tabs automatically close if deployment successful.
    - If not logged into environment, tab stays open with login page.
    - Error in deployment causes tab to remain open and turn red.

Secondary Environment Require ItemSet

- Be very careful if Require ItemSet system configuration setting cleared.
- Example of what could go wrong with Require ItemSet false (cleared):
  - Tier 1 CCR changes 1 line of code.
    - User accidentally does not deploy class to secondary environment because Require ItemSets is false.
  - Later, Tier 1 CCR changes a different line of code, unrelated.
    - Unknowingly deploys both changed lines to secondary environment.

Secondary Environment Require ItemSet

Primary environment



Secondary environment

Secondary Environment Require ItemSet (cont.)

- Example of what could go wrong with Require ItemSet false (cleared):
  - Tier 1 CCR adds class definition B and edits preexisting class A to be dependent on B.
    - User accidentally does not deploy class to secondary environment because Require ItemSets is false.

Primary environment

Secondary Environment Require ItemSet (cont.)

- Example of what could go wrong with Require ItemSet false (cleared):
  - Later, Tier 1 CCR edits class definition A (unrelated to previous change).
    - Can’t deploy this change to secondary environment because class A still dependent on class B.

Secondary environment

TRAIN Environments

- Often a project will require a TRAIN environment dedicated for End User Training.
  - This is separate from TEST to avoid collisions between testers and trainers.
  - Normally it is desirable for the configuration on TRAIN to mirror that of TEST.
- Option 1: Periodically refresh TRAIN from TEST (or LIVE).
  - Pros: Precise control over TRAIN contents and timing of updates.
  - Cons: Time intensive and long delay before new configuration available for training; requires Tech resource for refresh.

TRAIN Environments (cont.)

- Option 2: Configure TRAIN as a secondary environment for TEST.
  - Pros: TRAIN always up to date (ItemSets pushed to TRAIN and TEST simultaneously); does not require Tech resource.
  - Cons: Manual changes must be done in TEST and TRAIN; Configuration not ready for training use will be in TRAIN.

Quiz: Changing System Workflow

Question:

A system can be configured to only use TEST and LIVE environments. True or False?

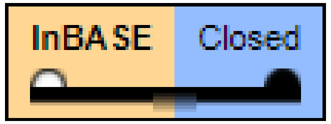
Answer: False.

Any system with 2 or more environments must include BASE.

Part 2: Implementation-Specific Workflows

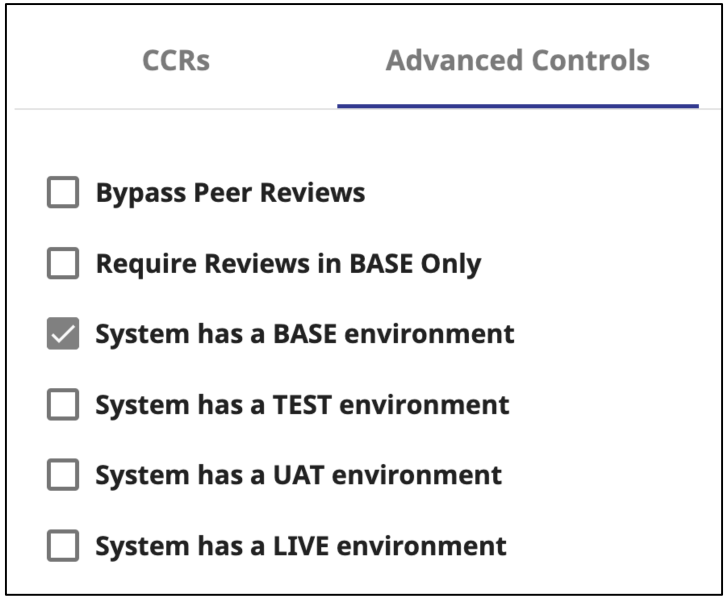
BASE-Only Workflows

- Early in implementation projects, only use BASE environment.
  - No need to promote changes to a LIVE environment.
- Allows use of:
  - CCR to document changes.
  - Perforce for roll-back or recovery.
  - An abbreviated workflow to speed up CCR lifecycle without significant risk.
    - Can be as short as only In_BASE state.
    - With peer reviews enabled, CCR progresses through BASE_Pending_Peer_Review prior to Closed.



Configure BASE-Only Workflows

- Only select BASE under Advanced Controls on System Details page.
  - Controls which Phases included in CCR progression.
  - Does not automatically configure BASE environment.



☐ Bypass Peer Reviews

☐ Require Reviews in BASE Only

☑ System has a BASE environment

☐ System has a TEST environment

☐ System has a UAT environment

☐ System has a LIVE environment

BASE/TEST Workflows

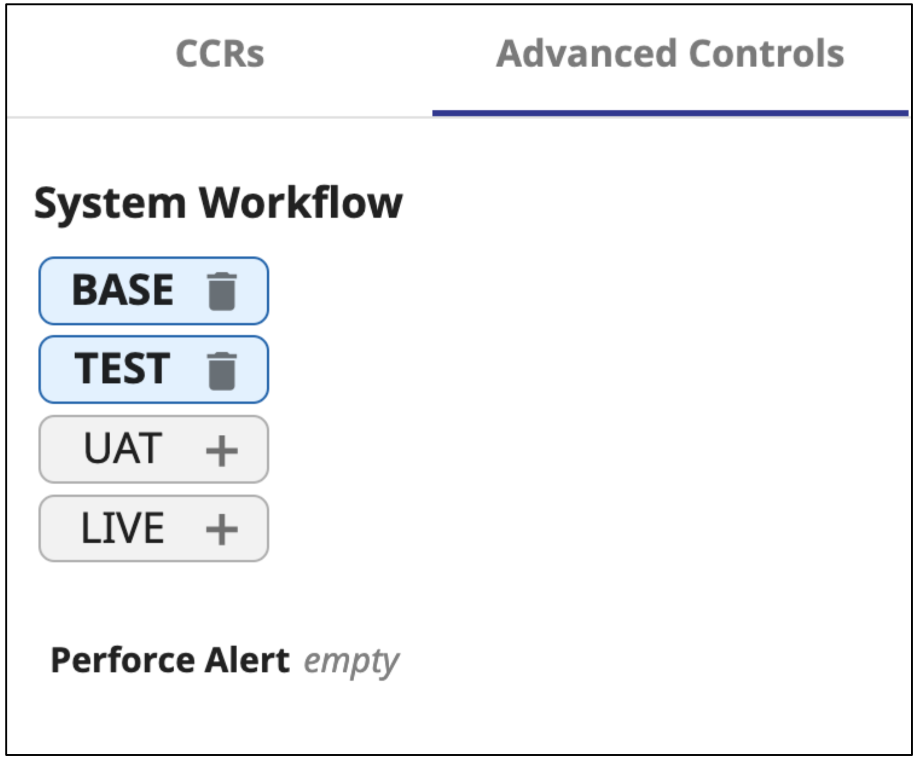
- Add TEST environment prior to project entering validation.
  - Clone BASE environment and Perforce branch.
- Promotion of changes via CCR Transport during validation steps prepares for smooth go-LIVE.

Adding Environment to Workflow

1. Progress active CCRs out of Environment prior to Environment to be added.
   - Example: When adding UAT to existing BASE > TEST > LIVE workflow, can have active CCRs in BASE and LIVE but not TEST.
2. Clone latest Environment in workflow.
   - Example: Clone TEST when adding LIVE to existing BASE > TEST workflow.
3. Add new Environment to CCR System.
   - Tier 1 & 2: have CCR automatically populate Perforce branch.
     - Must populate Perforce branch to prevent merge conflicts in future!

Adding Environment to CCR System Details

1. Go to Advanced Controls tab of System.
2. Click add icon for Environment add.
3. For Tier 1 & 2 Systems: Choose branch to copy from.
   - Default = latest in workflow.
     - Usually appropriate.
   - Note: Deletes existing items in branch.
   - Can choose Do Not Initialize Branch.
     - You must manually initialize branch in P4V.
   - Status of intialization indicated in Perforce Alert field.



System Workflow

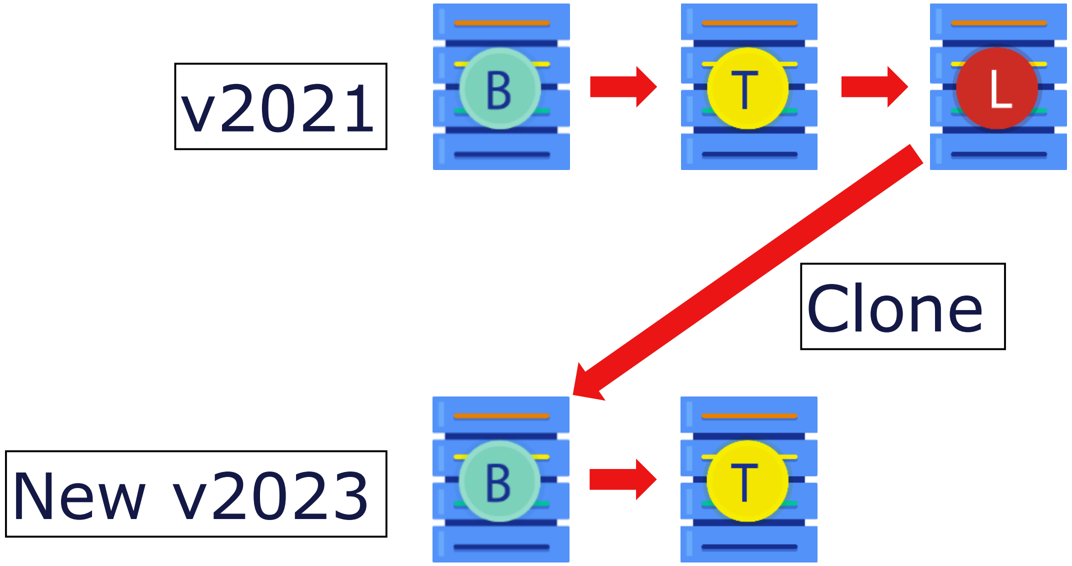
Perforce Alert *empty*

CCR for Multi-Phase Implementations

- Multi-phase implementations need to allow next phase configuration changes separately from required Business As Usual (BAU) changes.
  - Example: Upgrades.
- Next phase work should be done in its own system to reduce collisions.

CCR for Multi-Phase Implementations (cont.)

- Example change flow for an upgrade.



v2021

New v2023

Clone

Quiz: Implementation-Specific Workflows

Use secondary environments for multi-phase projects.

True or False?

Answer: False.

Secondary environments contain the same code as their primary environments and only receive code pushed to the primary.

Multi-phased projects can clone a 2nd System from the BAU System in order to keep the new phase isolated.

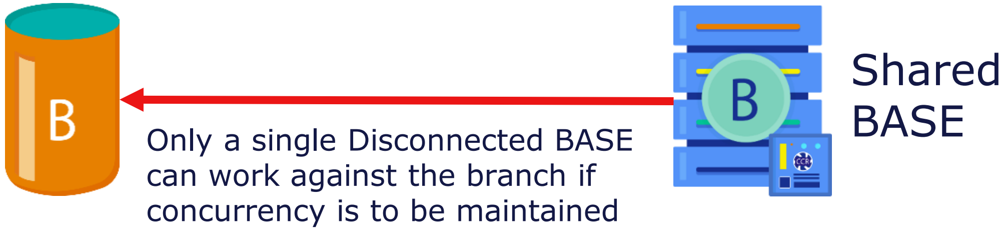
Part 3: Selecting a BASE Topology

Supported Strategies for BASE

- Concurrency of source must always be considered.
- Concurrency differs whether working in Disconnected or Connected mode.

Supported Strategies for BASE (cont.)

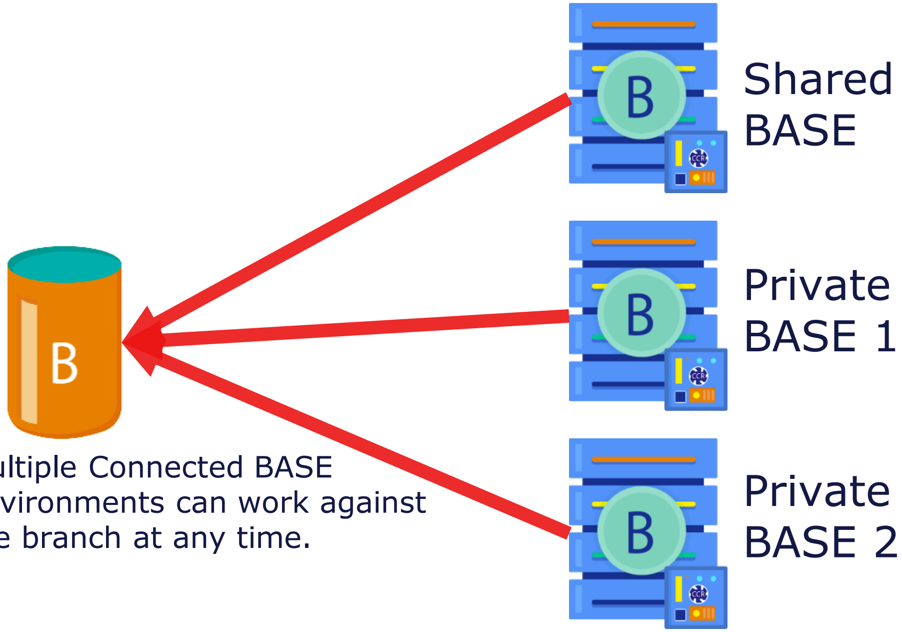
- **Disconnected** clients manage concurrency locally via CCR Client Tools.
  - Check-out is recorded as item being Read/Write, and owner recorded in %Studio_SourceControl.Change.
  - Only **one** Disconnected BASE can be used to submit changes against the branch.
  - No way to control concurrency between multiple instances.
  - If require multiple BASE environments for functional testing (e.g., several HS Edge servers):
    - Designate one as Master for all editing.
    - Lock others to only accept changes via BASE ItemSets.



Only a single Disconnected BASE can work against the branch if concurrency is to be maintained

Connected Clients for BASE

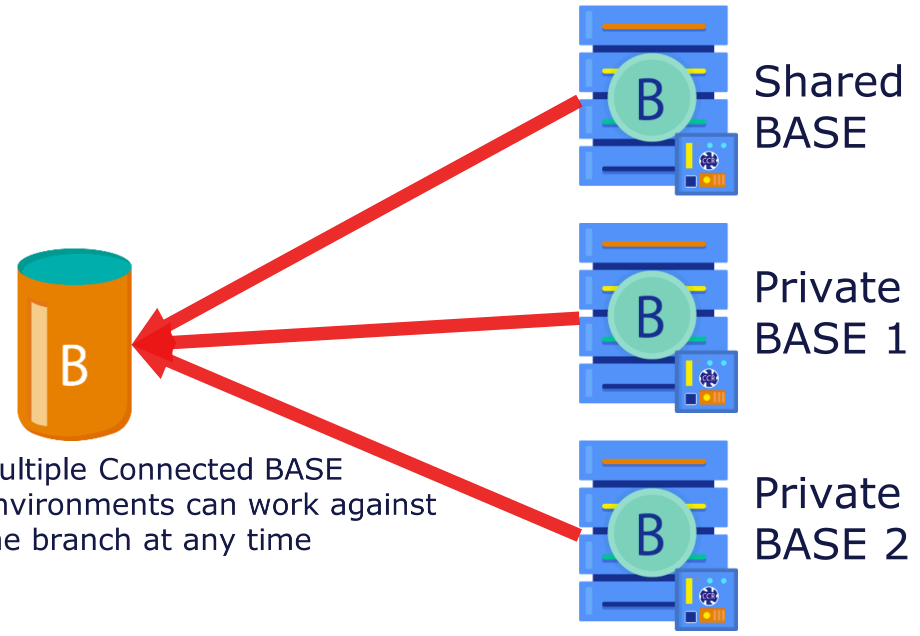
- Perforce server manages concurrency.
  - Perforce tracks what version has been checked out by whom.
  - Requires merging before check-in if a newer version submitted by another user.
- Any combination of the following is supported:
  - Shared BASE used by multiple developers.
    - Only one user can check out a file on shared BASE at a time.
    - That file can be checked out by other Connected BASEs.
  - Private BASE used by a single developer.



Multiple Connected BASE environments can work against the branch at any time.

Connected Clients for BASE (cont.)

- If multiple Connected BASE environments, ensure all BASEs receive all changes.
    - Scripted CCR Refresh can be used, or…
    - ItemSet deployment to all BASE environments.



Multiple Connected BASE environments can work against the branch at any time

Quiz: Selecting a BASE Topology

Question:

To protect data integrity in Connected mode, you should only have 1 shared BASE environment. True or False?

Answer: False.

In Connected mode you can safely have more than 1 BASE environment because the Perforce server can maintain concurrency.

In Disconnected mode, you should only have 1 shared BASE environment.

Summary

- What are they key points for this module?

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.

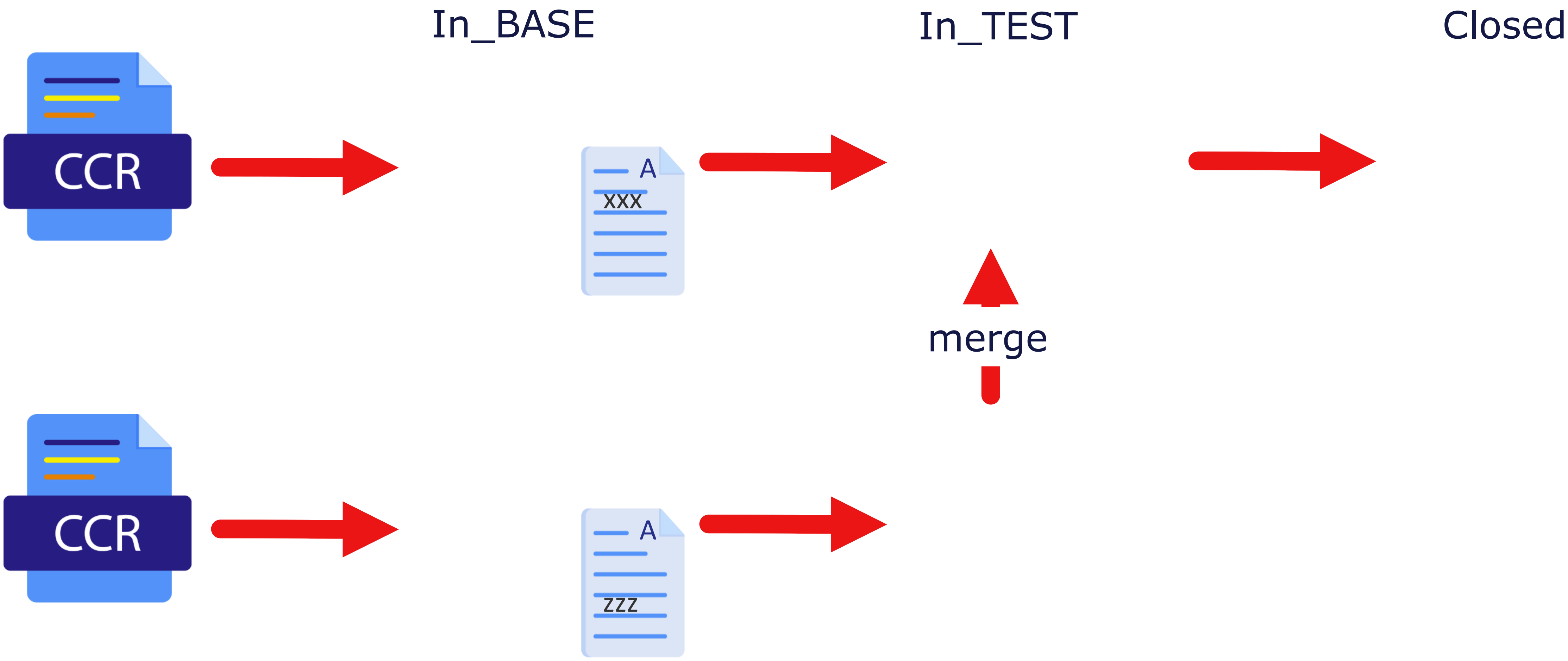
How To: Use Catch-up CCR

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?

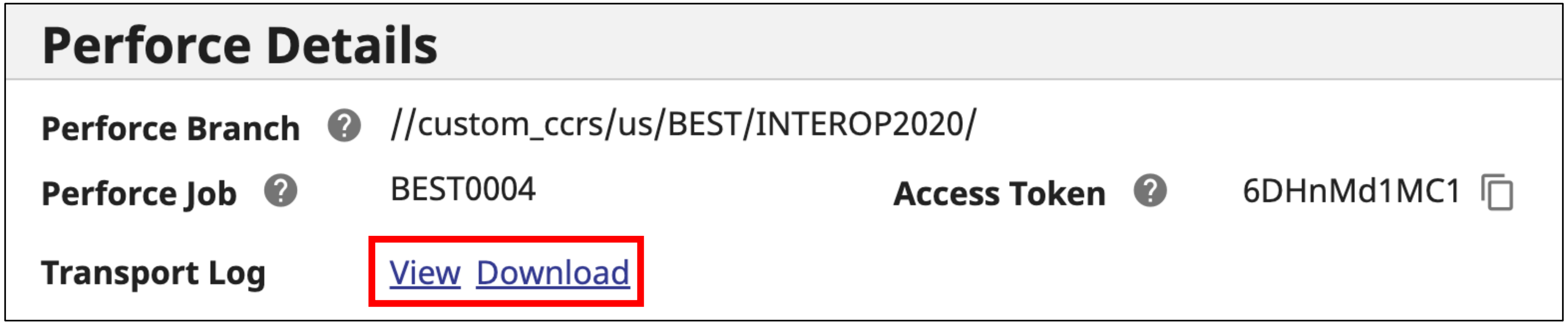
Answer:

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.

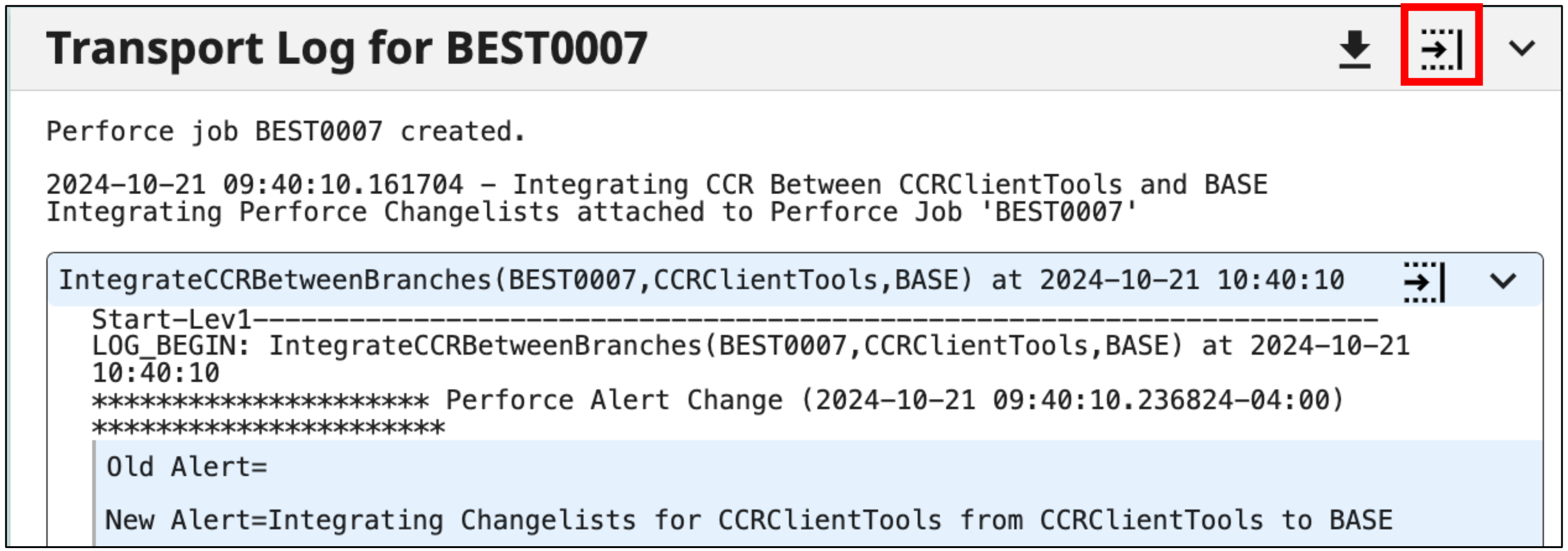


Perforce Details

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.

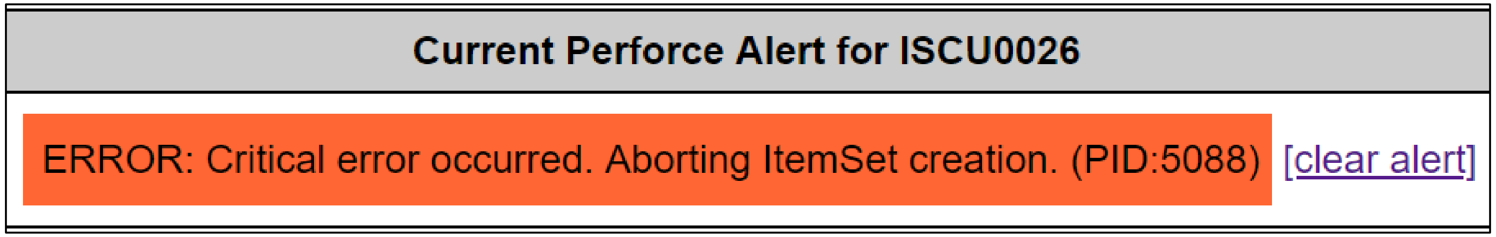


Transport Log for BEST0007

```
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:

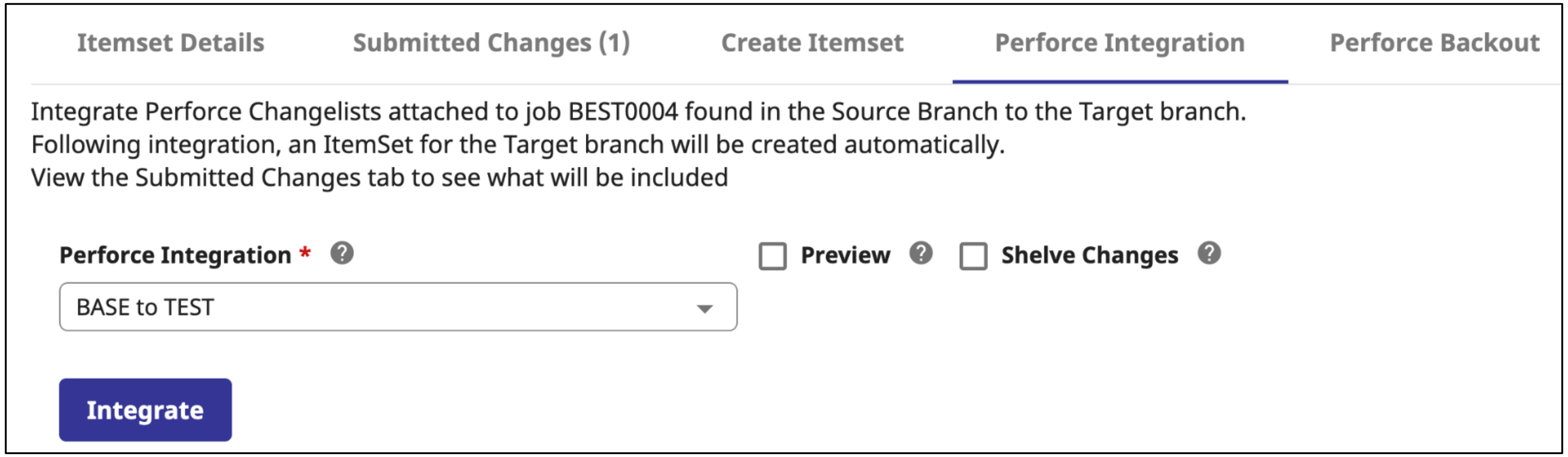
Transport Logs includes all logging which occurred by Perforce-related actions which occurred against that CCR or its ItemSets. True or False?

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

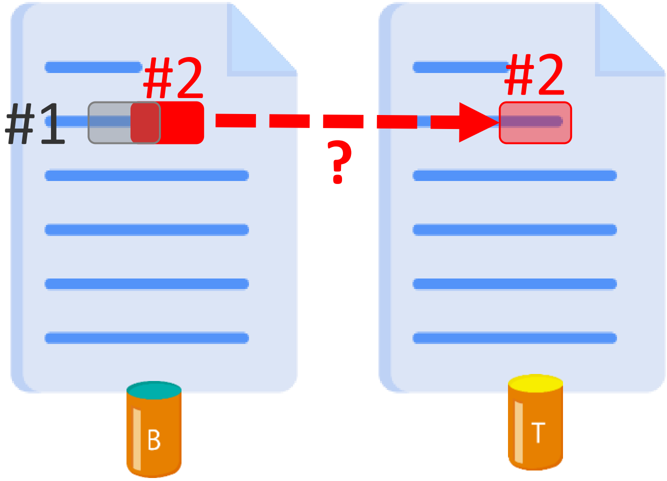
□ Preview □ Shelve Changes

Part 5: Perforce Integration Conflicts

Quiz: Merge Conflicts

Question:

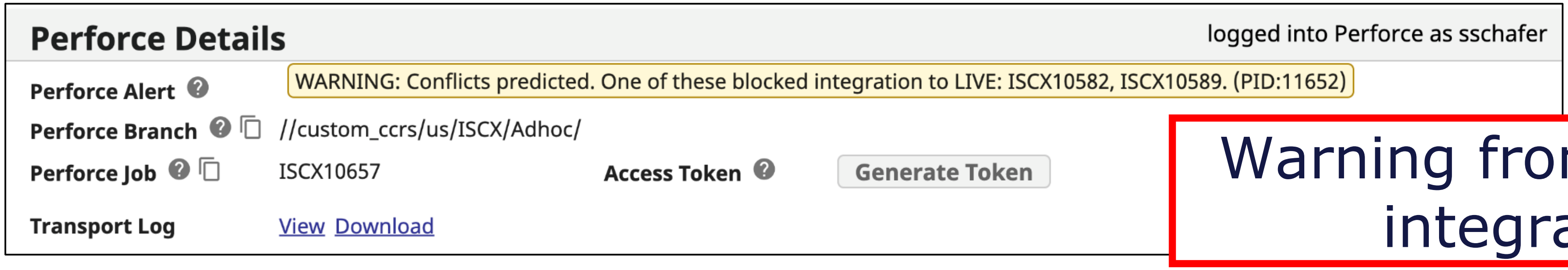
The following will cause a merge conflict. Change #1 is made but not progressed before #2 is made and progressed to TEST. True or False?



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

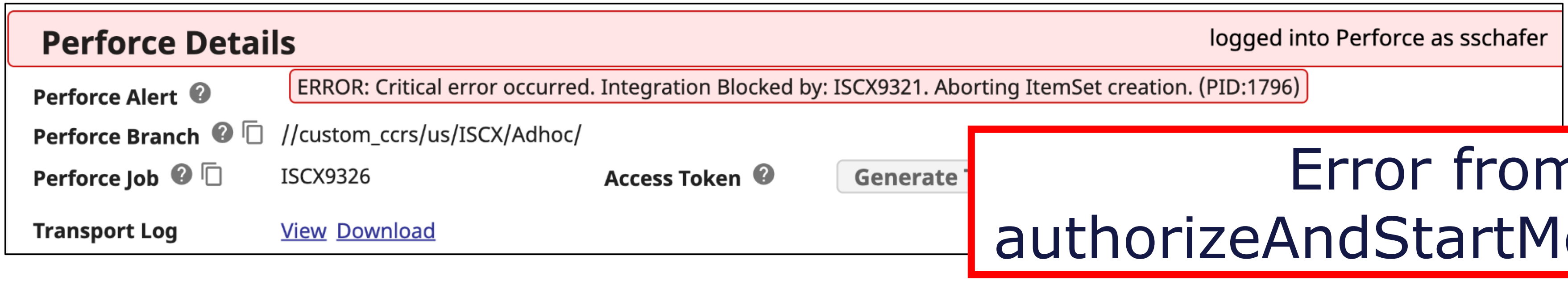
Perforce Alert ? WARNING: Conflicts predicted. One of these blocked integration to LIVE: ISCX10582, ISCX10589. (PID:11652)

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/

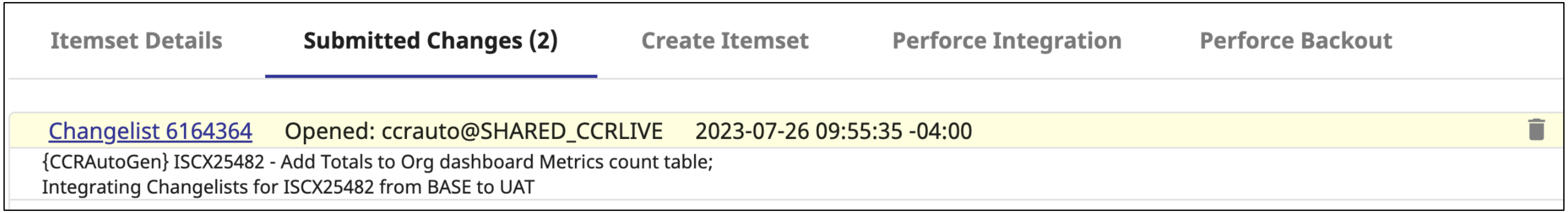
Perforce Job ? ISCX9326 **Access Token** ? Generate

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.

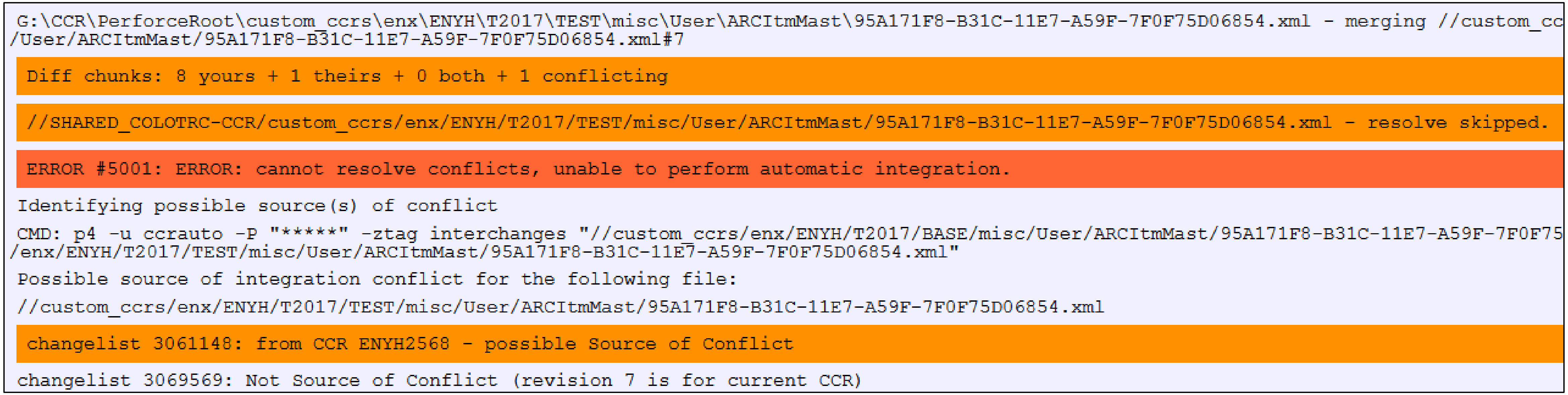


Changelist 6164364 Opened: ccrauto@SHARED_CCRLIVE 2023-07-26 09:55:35 -04:00

{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.

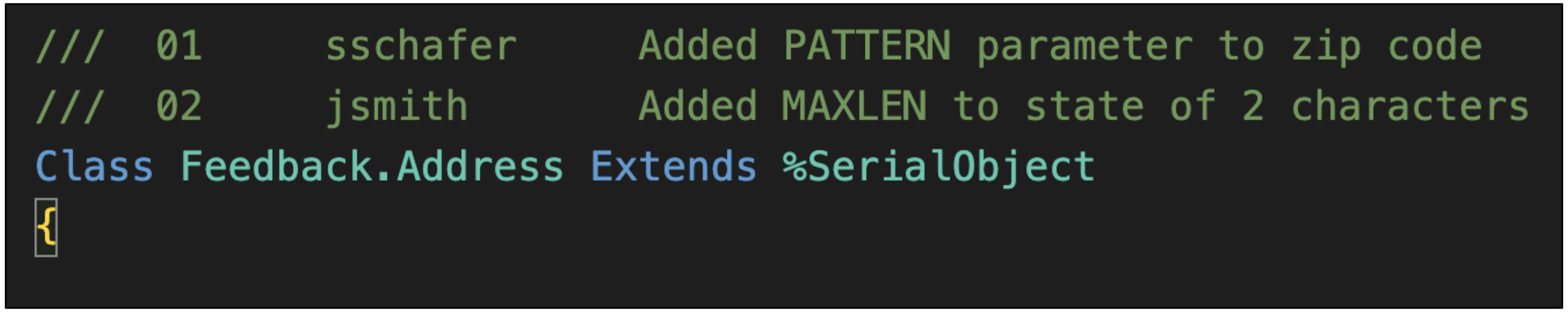
1. Perform markIntegrationFailed transition on CCR A.
2. Edit CCR A to specify CCR B as prerequisite.
3. Progress CCR B.
4. Optional: Perform preview integration on CCR A to confirm all merge conflicts resolved.
4. Progress CCR A.
   - CCR B must progress ahead of CCR A in terms of phase.

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.

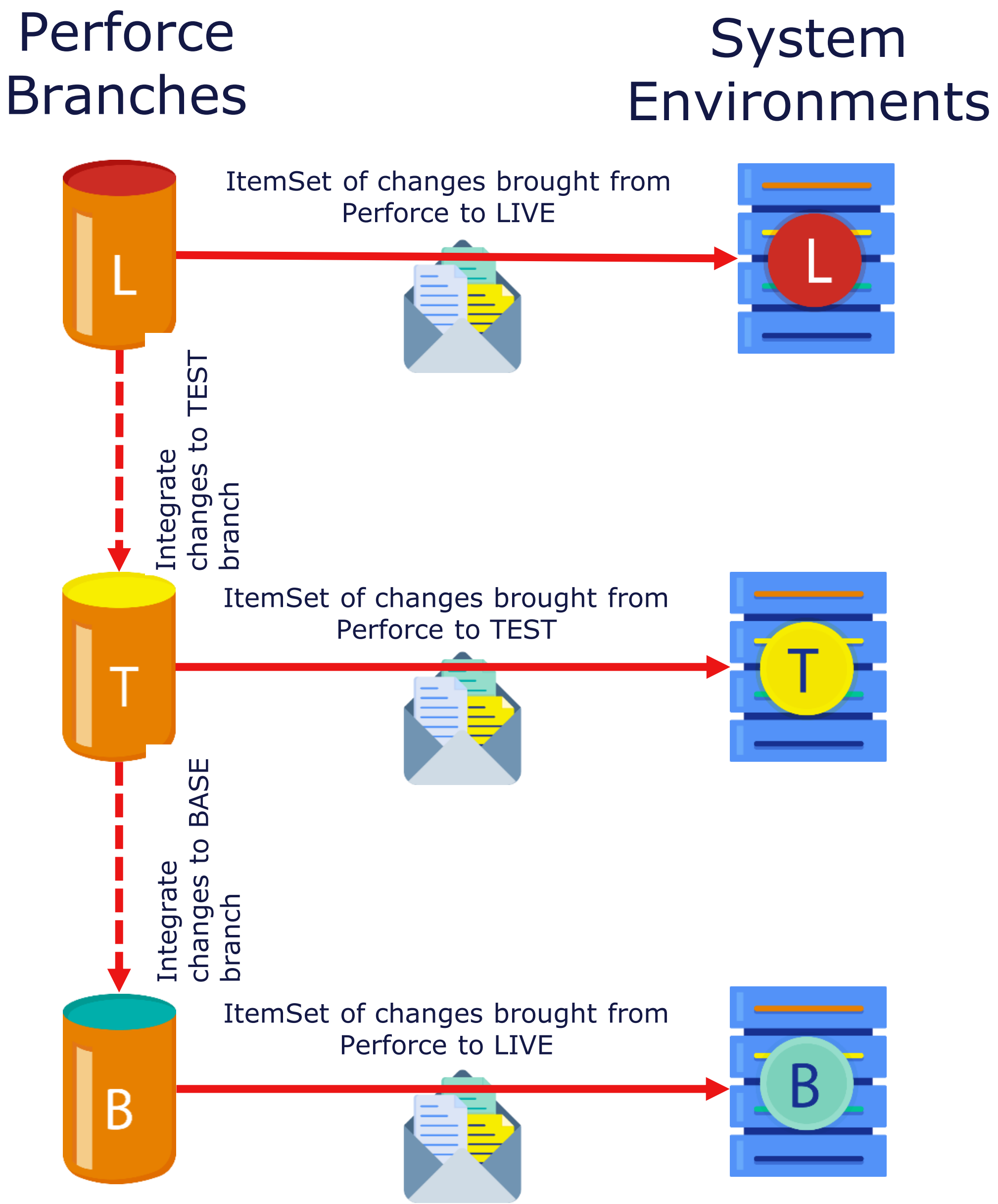
Progressing Edit before Add: Why It Cannot Be Allowed

- 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

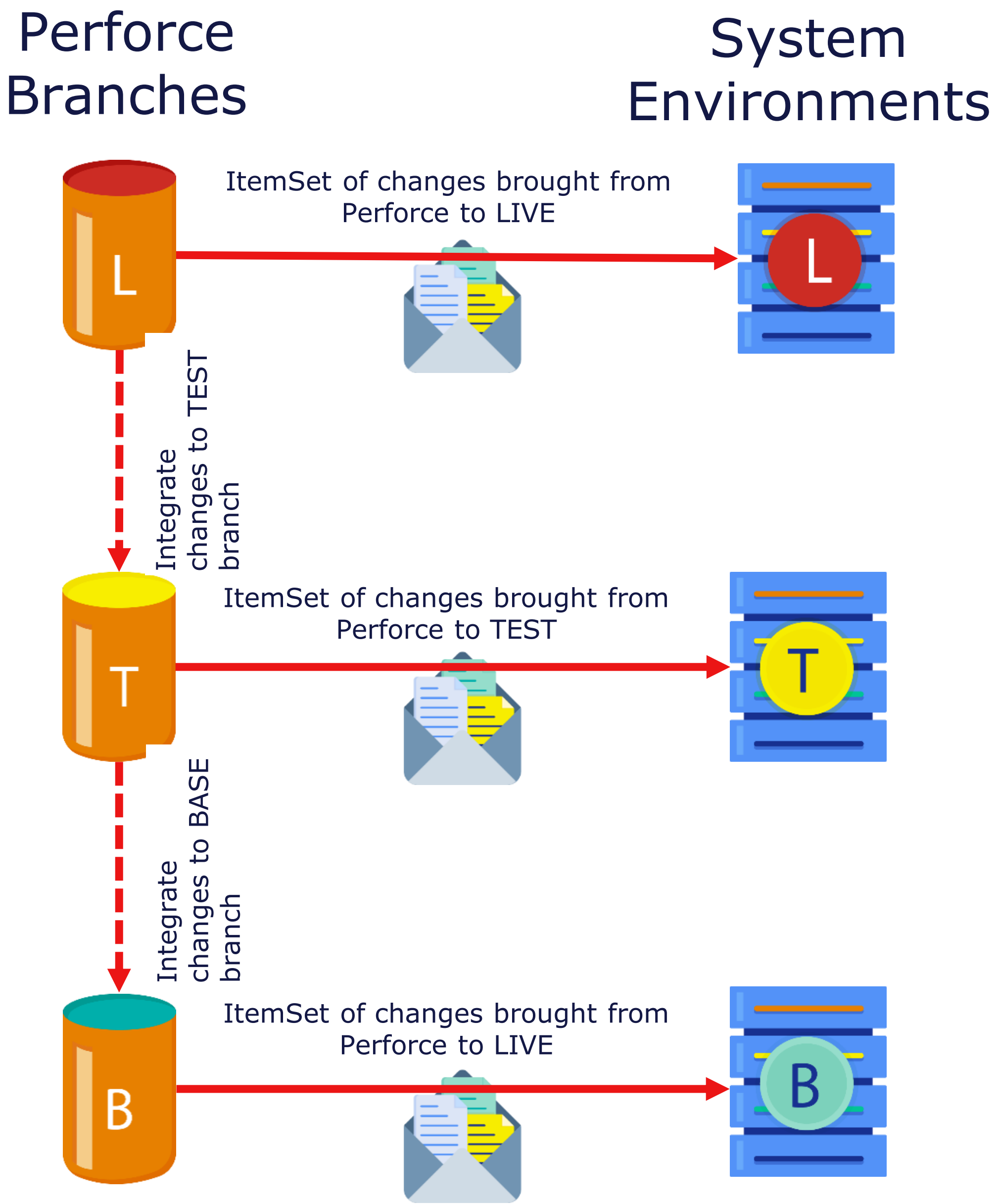
Backout Overview

- 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

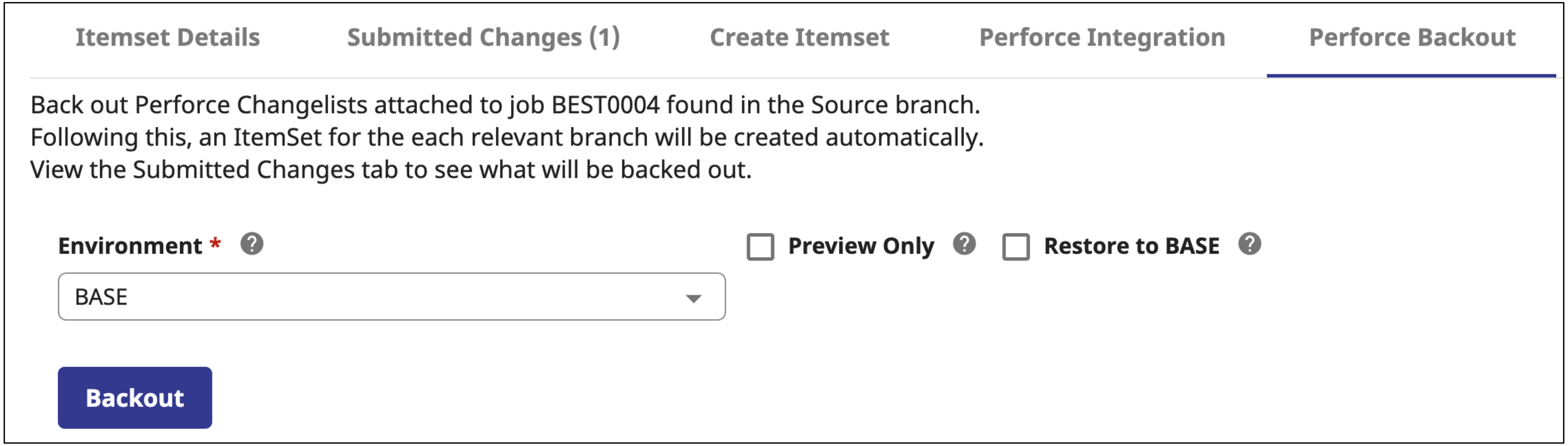
- CCR automatically backs out all changes from all environments for:
  - Tier 1 CCRs only.
  - cancel and changeSpec transitions.
- ItemSets for each affected environment generated if Perforce backout successful.
  - Deploy all generated ItemSets like any other ItemSet.
- Progress CCR through next transition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

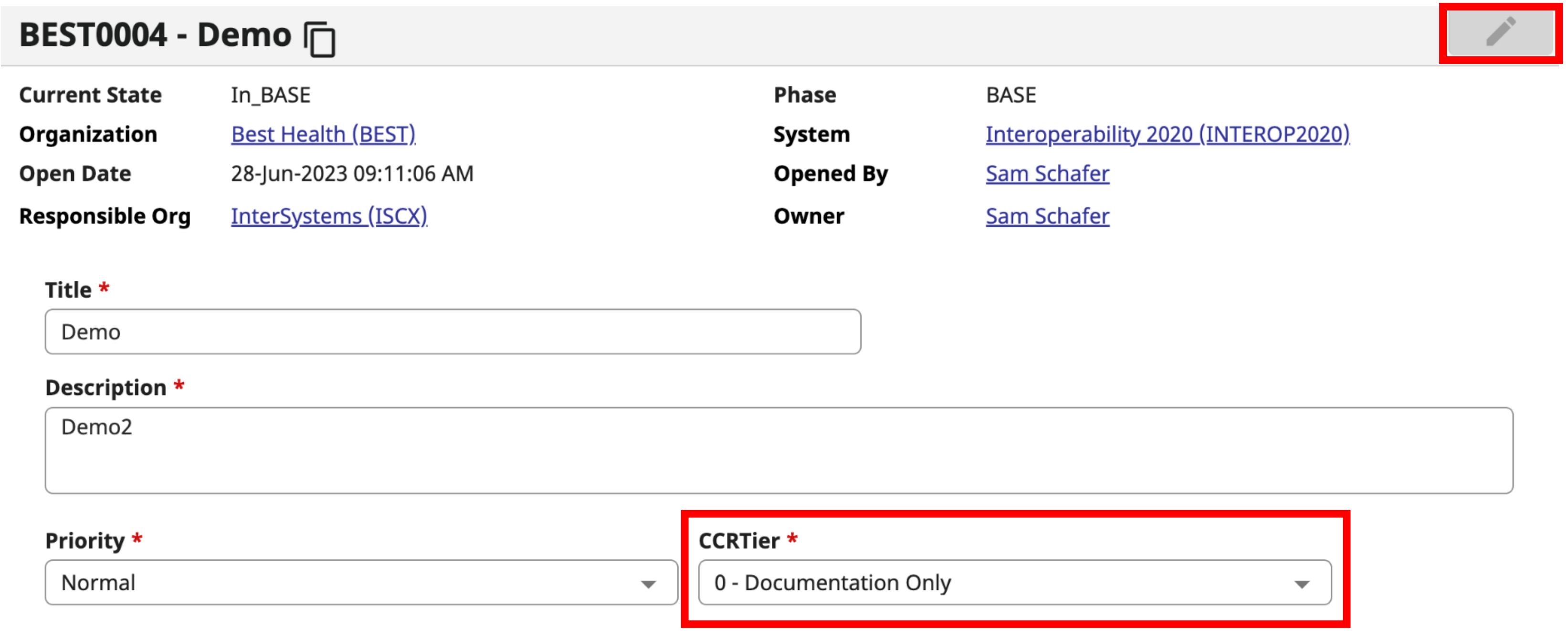
C. It is not necessary when cancelling a CCR that did not enter LIVE phase.

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

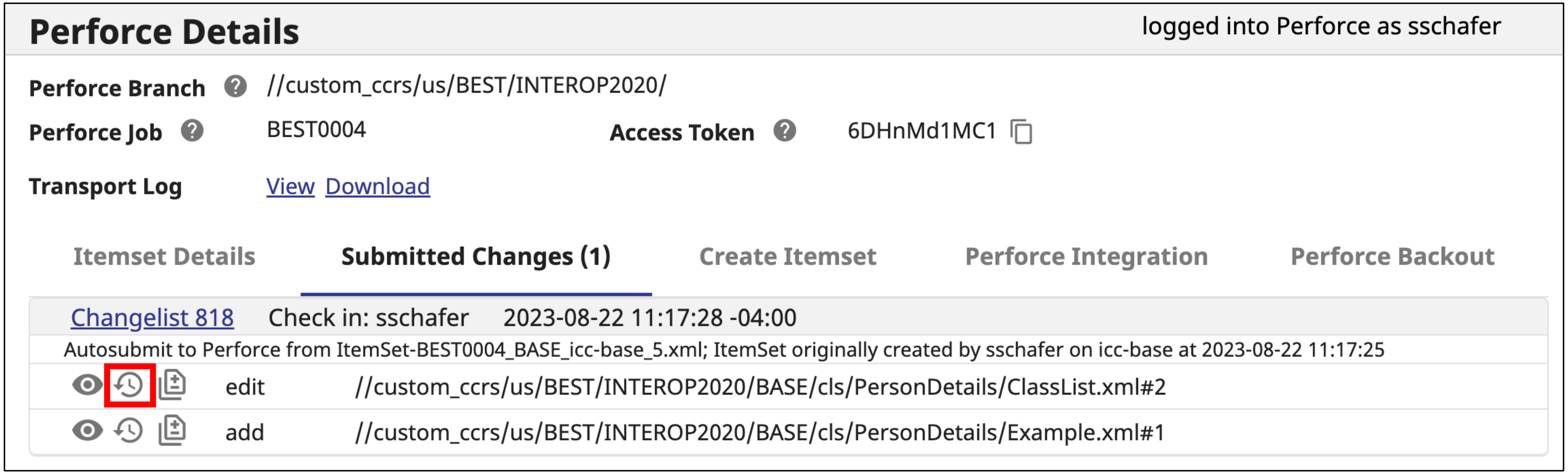
- Go to appropriate System Details page.
- Click Update Client Tools button.
  - Can only be actioned by Perforce users.
  - Creates Tier 1 CCR and integrates current client tools to BASE branch of System.
- Progress generated CCR through normal workflow.
  - Make sure to deploy generated ItemSet to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

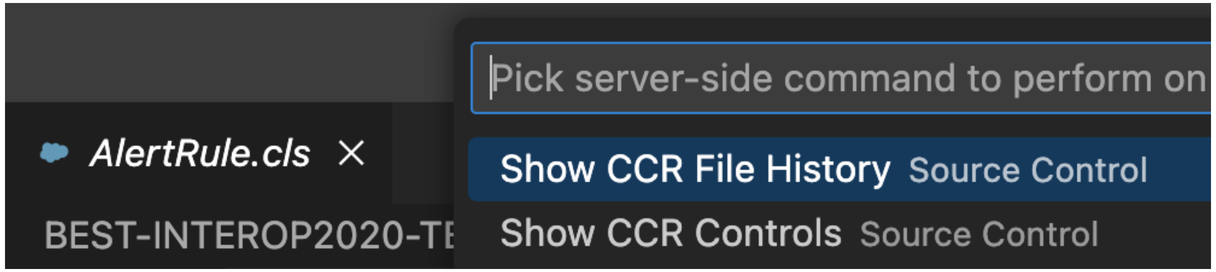
Changelist 818 Check in: sschafer 2023-08-22 11:17:28 -04:00

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.

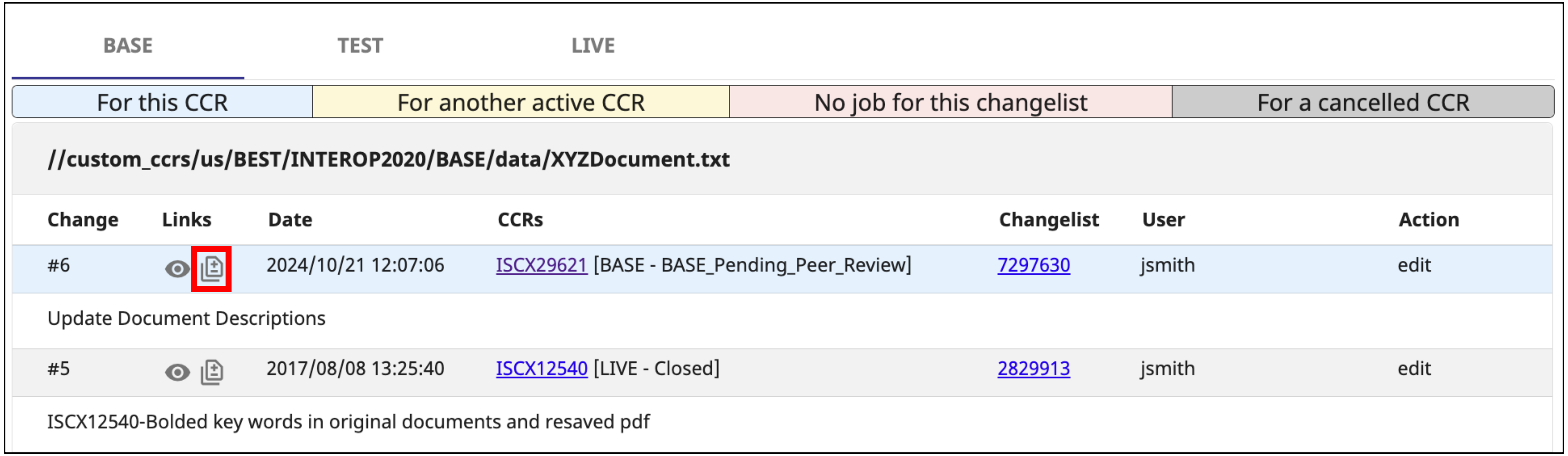


Pick server-side command to perform on

AlertRule.cls
BEST-INTEROP2020-TE
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.



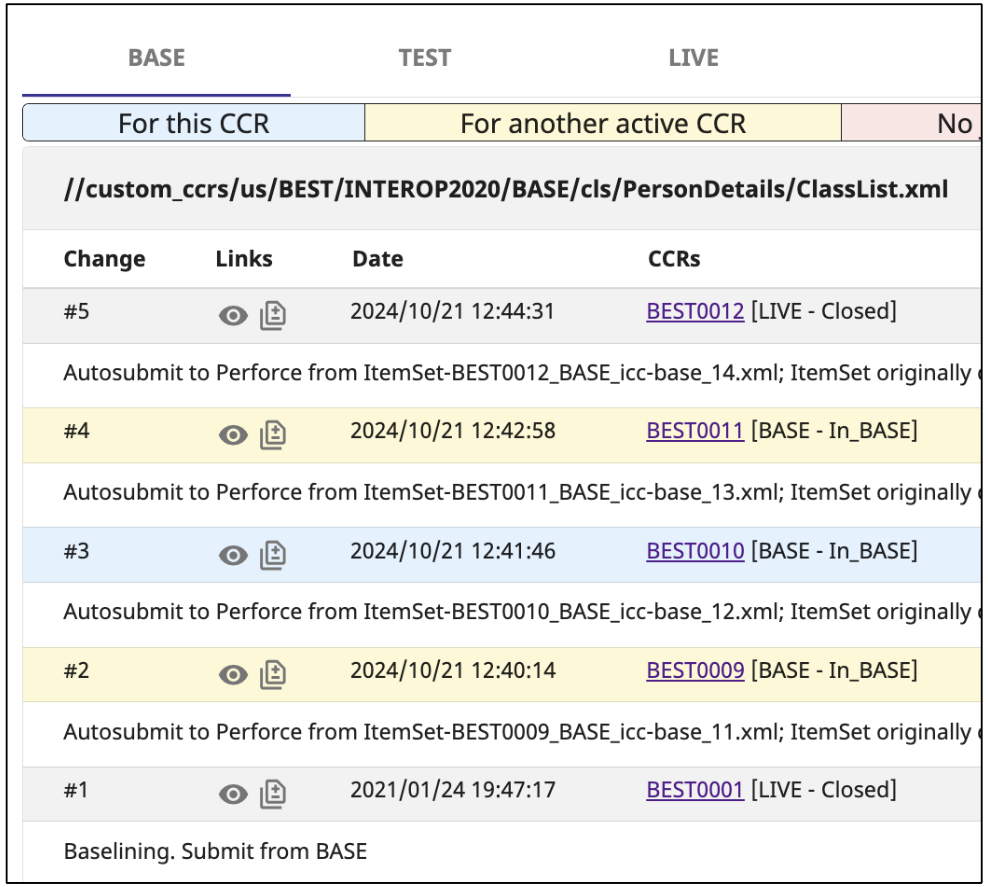
For this CCR	For another active CCR	No job for this changelist	For a cancelled CCR

//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.



BASE TEST LIVE

For this CCR | For another active CCR | No

//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:

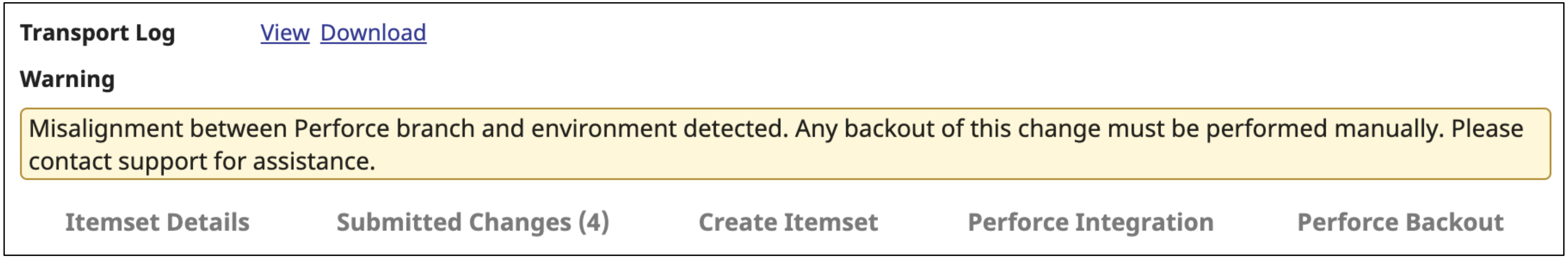
The ‘history’ link under show submitted changes has links to show diffs of the current version against every other version. True or False?

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?

ICC 460: CCR Transport – Perforce Debugging & Advanced Topics

InterSystems

Creative data technology

Objectives

- Be able to use p4V and Swarm to make changes directly to items on the Perforce server.
- Define view paths and how they are used within Perforce.
- Describe Perforce command line interface variables, how they are used and setup.
- Explain how Perforce references files during synchronisation.
- Illustrate how Perforce prevents conflicts using locks.
- Leverage branch hygiene tool.

Objectives (cont.)

- Setup Perforce notifications to monitor specific branches.
- Demonstrate how to move a changelist to a new CCR using Swarm and P4V.
- Explain how to refresh Perforce branches.
- Demonstrate how to backout changes using Perforce.
- Fix integration history.

Part 1: Advanced Perforce Concepts

View Paths

- View path: //depotMapping/... //targetMapping/...
  - Examples:
    - A Client Workspace View Path between a depot and a client: //MYDEPOT/MyApp/MAIN/... //MYDEPOT_MyApp_MAIN_Dev/...
    - A Branch View Path between two depot codelines: //MYDEPOT/MyApp/MAIN/....//MYDEPOT/MyApp/BRANCHES/1.0.0/...
- Overlay & Exclusions – used to remove or add specific locations to a View Path.
  - +//mapping3/... //mapping4/...
  - -//mapping3/... //mapping5/...

View Paths (cont.)

- Special characters for View Paths: space, *, ..., %, @, #.
  - Perforce wildcards – avoid in filenames.
  - Use ASCII code or quote.

CLI: P4 Session Variables

- To use the CLI (Command Line Interface) it is a good idea to set up your CLI variables:
  - P4CONFIG = p4config.txt (Windows) ; .p4 (UNIX).
    - P4PORT=server:port.
    - P4CLIENT=clientName.
    - P4USER=lanID.
    - P4CHARSET= utf8 (Optional).
- If not logged, must issue “p4 login” command first.
- Always check where you are connected with “p4 info” before issuing commands by the CLI.

CLI: Perforce Session

- Hierarchy of elements defining your P4 environment and command settings:
  - On the command line, using flags.
  - In a configuration file, if P4CONFIG is set.
  - User environment variables (on UNIX or Windows).
  - System environment variables (on Windows, system-wide environment variables are not necessarily the same thing as user environment variables).

CLI: Perforce Session (cont.)

- Hierarchy of elements defining your P4 environment and command settings:
  - On Windows, in the Perforce User Registry:
    - p4 set setting=value.
  - On Windows, in the Perforce System Registry:
    - p4 set -s setting=value.
- You can check your environment variables with “p4 set” before issuing commands with the CLI.

Working on Files: Synchronizing

- p4 sync.
- @, # are the wildcards to use to specify versions:
  - @ = relative reference.
    - changelist.
    - date.
    - Label.
    - etc…
  - # = hard reference (specific revision).
    - Useful for fetching specific revision of one file.
    - Not very useful for group of files.
    - #head specifies the ‘latest’ revision.

Preventing Conflicts: Lock

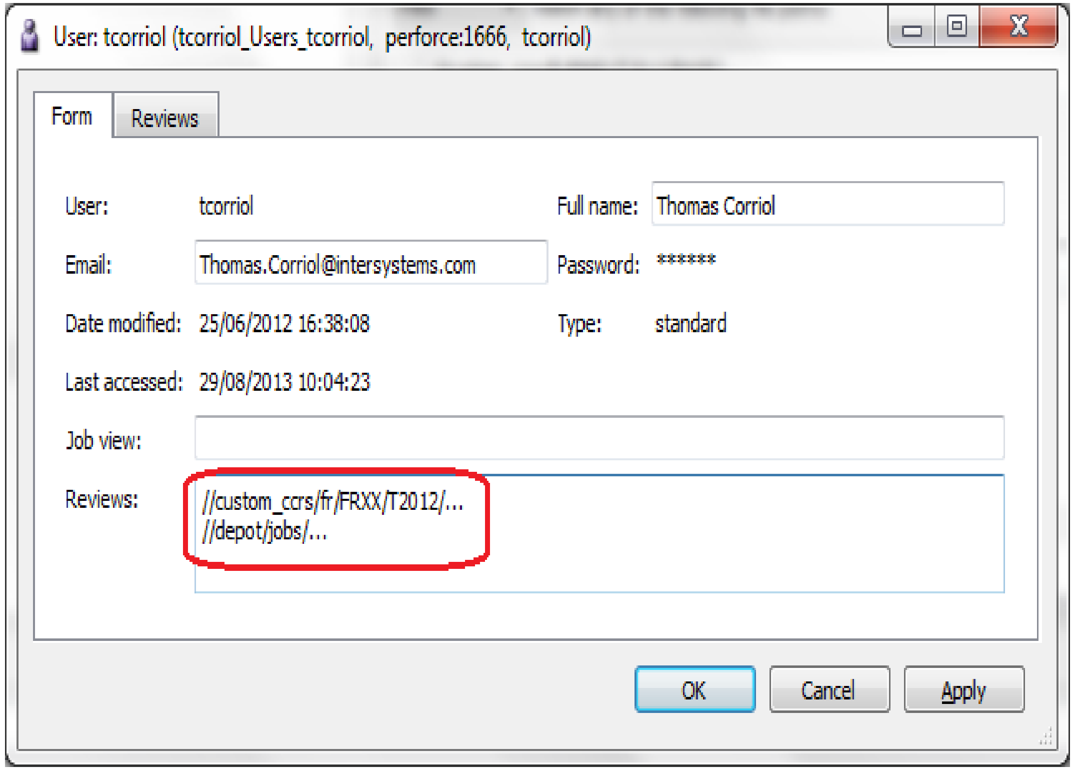
- Optimistic locking.
  - Use Lock command in P4V.
  - Prevents other user checking in the file before you check in the file.
- Pessimistic locking.
  - Handled on the server via specific filetypes.
  - Prevents other user from checking out the file before you check in the file.
    - It is helpful to use pessimistic for certain files types, e.g. PPT and XLS files which can’t be merged.
  - Contact Tech Services if there is a specific file type for which you want pessimistic locking in your depot.

Performing Very Large Branching Virtually

- There are some cases where a large number of files need to be branched to a new location.
  - E.g., populating environment branches after a CCR Baseline, or making a new Project Branch for a product.
- By default, all files get synced to the local workspace as part of an integration, which can be very time consuming.
  - Especially on remote connections.
- To branch large volumes of files from depot to depot without syncing them locally:
  - Use the -v switch for p4 integrate, which will perform the branching virtually and will be much faster.

Perforce Notifications via “Reviews”

- Changelists notifications via email can be set up to ‘watch’ a specific branch.
  - Allows specific branches to be ‘monitored.’
  - Configurable by user.
  - Menu “Connection > Edit Current User...”
    - Put a branch under “Reviews.”



User: tcorriol (tcorriol_Users_tcorriol, perforce:1666, tcorriol)

User:	tcorriol	Full name:	Thomas Corriol
Email:	Thomas.Corriol@intersystems.com	Password:	******
Date modified:	25/06/2012 16:38:08	Type:	standard
Last accessed:	29/08/2013 10:04:23		
Job view:			
Reviews:	//custom_ccrs/fr/FRXX/T2012/... //depot/jobs/...		

Perforce Notifications via “Reviews” (cont.)

- Changelists notifications via email can be set up to ‘watch’ a specific branch.
  - When a changelist is submitted against a branch, an email gets sent immediately with:
    - User.
    - Changelist description.
    - Files impacted.
    - Job.
  - “//depot/jobs” to get jobs notification (you will get ALL of them).
  - Metadata is not protected, and shared on a Server.

Quiz: Advanced Perforce Concepts

Question:

If the Perforce system default for check-outs is optimistic locking, both John and Sandy can checkout the same file.

True or False?

Answer: True.

If Sandy wants to prevent John or anyone else from checking out the file, she should use the lock command after checking out the file.

Part 2:
Moving a Cancelled Changelist to a new CCR

Moving a Changelist from One CCR to Another

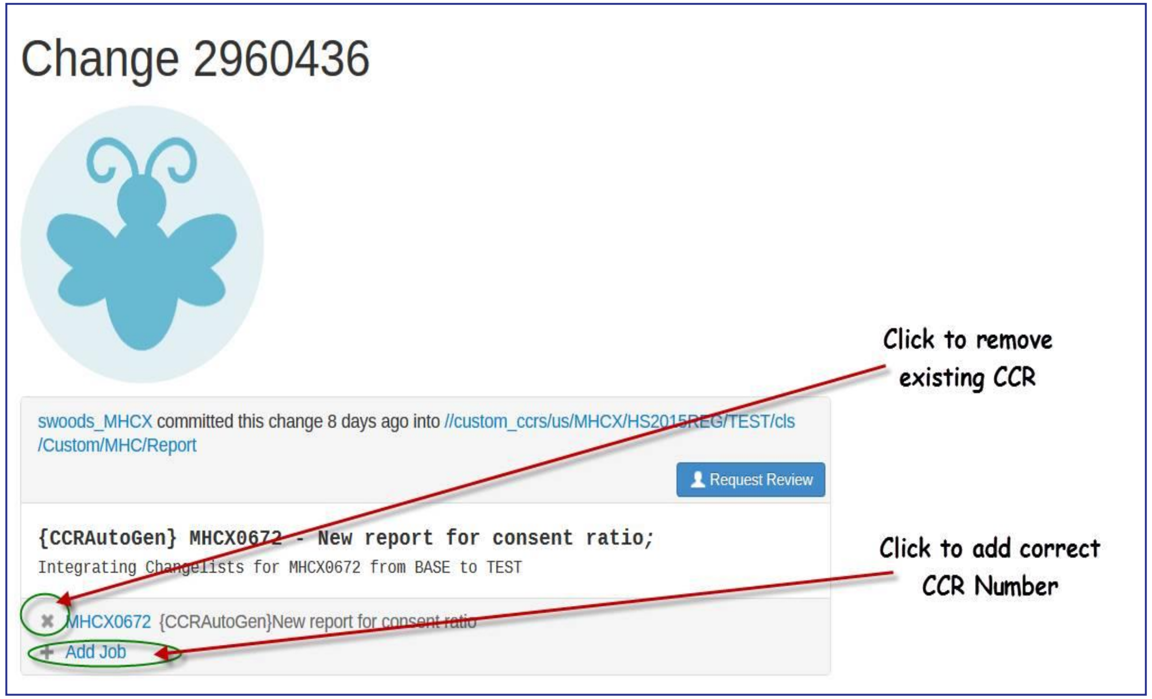
- There are some cases where a changelist needs to be moved from one CCR to another:
  - An ItemSet was uploaded against the wrong CCR by accident.
  - A CCR was cancelled by mistake (and the changelists were not backed out) and the user wishes to attach the changelist to a new CCR to progress the change.
- Review: changelists are associated with CCRs via Perforce jobs.

Moving a Changelist from One CCR to Another (cont.)

- Jobs can be edited on submitted changelists via:
  - Swarm.
  - p4V.
- Adding a new job (which references a CCR) can be done only by InterSystems employees as it requires direct access to Perforce.
  - Customers should contact InterSystems Support for assistance in moving a changelist.

Using Swarm to Change a Job on a changelist

- Click the changelist number on the CCR that you need to move, then it will open it up in Swarm (using your Perforce credentials).
- Add the correct CCR number by clicking the “Add Job” hyperlink.
- Remove the link to the incorrect one by clicking the X beside it.



Change 2960436

swoods_MHCX committed this change 8 days ago into //custom_ccrs/us/MHCX/HS2015REG/TEST/cls/Custom/MHC/Report

```
{CCRAutoGen} MHCX0672 - New report for consent ratio;
Integrating Changelists for MHCX0672 from BASE to TEST
```

✖ MHCX0672 {CCRAutoGen}New report for consent ratio

Click to remove existing CCR

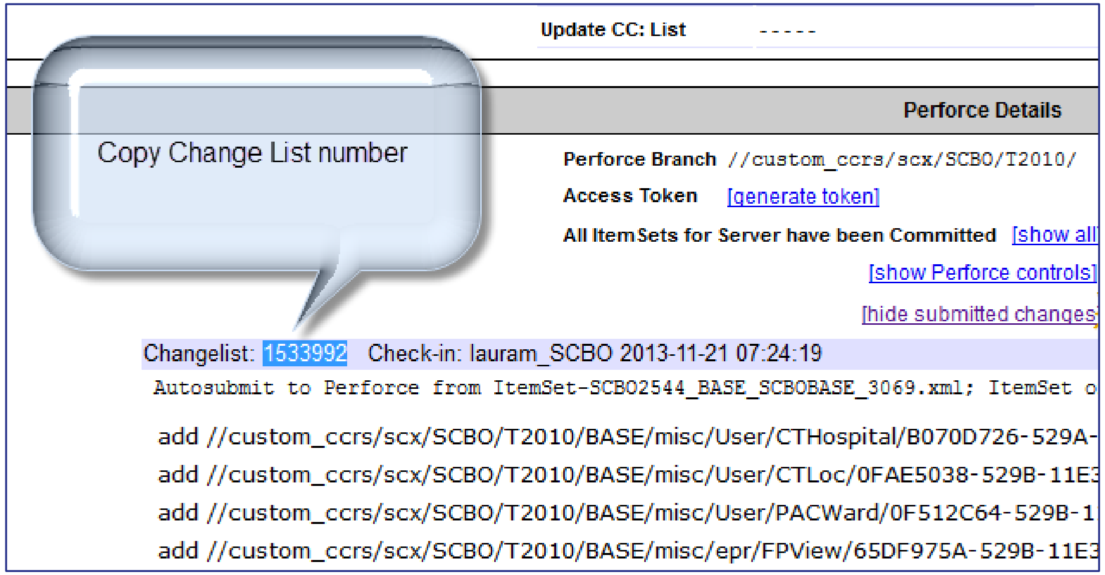
Click to add correct CCR Number

Using p4V to Change a Job on a Changelist

- Steps:
  1. Log into CCR online application.
  2. Ensure that the new CCR is in a phase of In_BASE.
  3. Open the CCR with which the changelist is currently associated.
  4. Select Show Submitted Changes hyperlink within the Perforce Details Section.

Using p4V to Change a Job on a Changelist (cont.)

- Steps:
  5. Copy the changelist Number.
  6. Log into p4V.
  7. Open the changelist (CTRL+G).
  8. Paste the changelist number.
  9. Click on OK.



Copy Change List number

Update CC: List -----

Perforce Details

Perforce Branch //custom_ccrs/scx/SCBO/T2010/

Access Token [generate token]

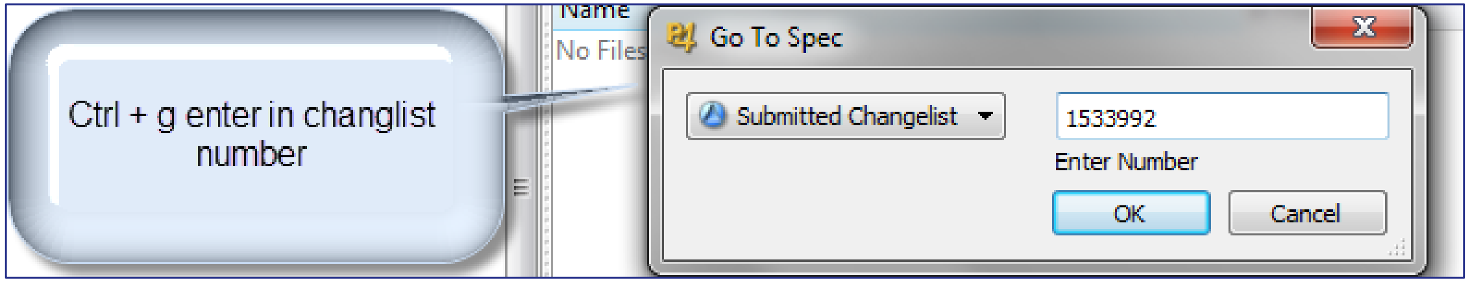
All ItemSets for Server have been Committed [show all

[show Perforce controls]

[hide submitted changes]

Changelist: 1533992 Check-in: lauram_SCBO 2013-11-21 07:24:19

```
Autosubmit to Perforce from ItemSet-SCBO2544_BASE_SCBOBASE_3069.xml; ItemSet o
add //custom_ccrs/scx/SCBO/T2010/BASE/misc/User/CTHospital/B070D726-529A-
add //custom_ccrs/scx/SCBO/T2010/BASE/misc/User/CTLoc/0FAE5038-529B-11E3
add //custom_ccrs/scx/SCBO/T2010/BASE/misc/User/PACWard/0F512C64-529B-1
add //custom_ccrs/scx/SCBO/T2010/BASE/misc/epr/FPView/65DF975A-529B-11E3
```



Ctrl + g enter in changlist number

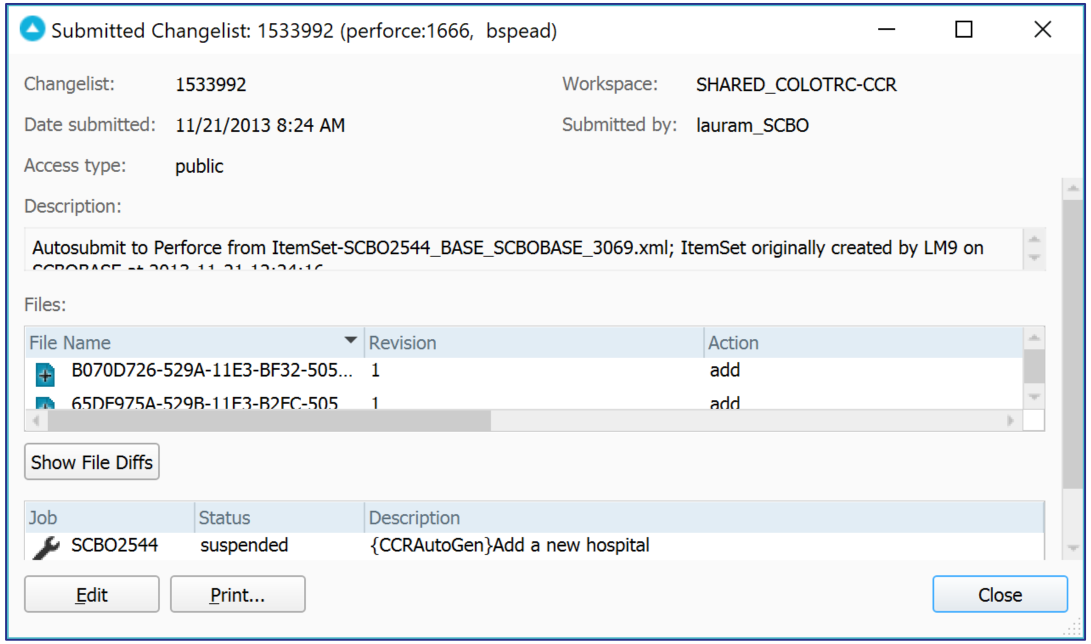
Go To Spec

Submitted Changelist: 1533992

Enter Number

Using p4V to Change a Job on a Changelist (cont.)

- Steps:
  10. Click Edit.
  11. Enter new CCR number in Job field.
  12. Click Add.
  13. Confirm the job is now listed.
  14. Right-click the old job > Remove from changelist.
  15. Click OK.



Submitted Changelist: 1533992 (perforce:1666, bspead)

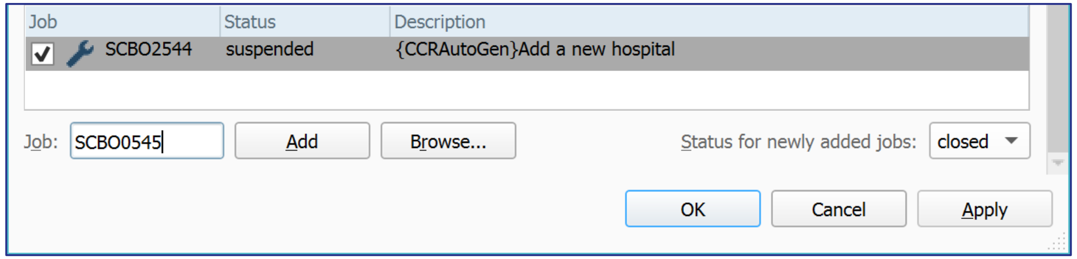
Changelist: 1533992
Workspace: SHARED_COLOTRC-CCR
Date submitted: 11/21/2013 8:24 AM
Submitted by: lauram_SCBO
Access type: public

Description:
Autosubmit to Perforce from ItemSet-SCBO2544_BASE_SCBOBASE_3069.xml; ItemSet originally created by LM9 on SCBOBASE at 2013-11-21 12:24:16

Files:

File Name	Revision	Action
B070D726-529A-11E3-BF32-505...	1	add
65DF975A-529B-11E3-B2FC-505...	1	add

Job	Status	Description
SCBO2544	suspended	{CCRAutoGen}Add a new hospital



Job	Status	Description
☑ SCBO2544	suspended	{CCRAutoGen}Add a new hospital

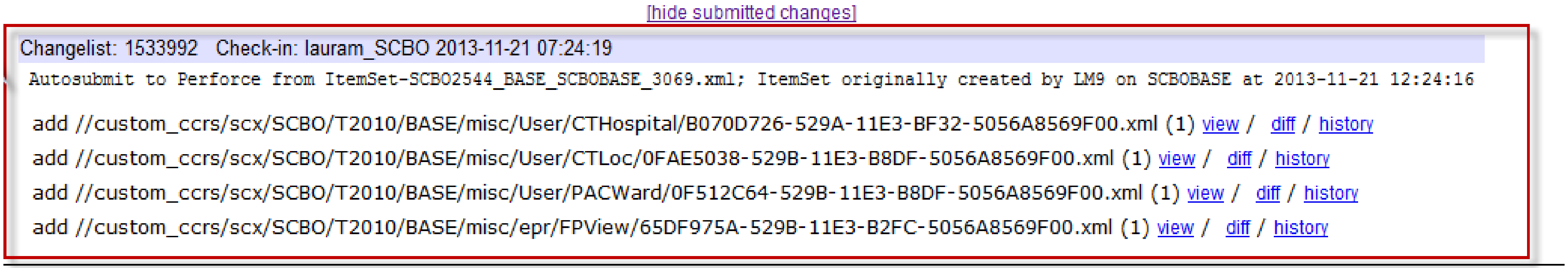
Job: SCBO0545

Status for newly added jobs: closed

Using p4V to Change a Job on a Changelist (cont.)

- Steps:

16. Return to new CCR and refresh page.
17. Verify changelist now appears under [show Submitted Changes].
18. Progress CCR according to normal workflow.



[hide submitted changes]

Changelist: 1533992   Check-in: lauram_SCBO 2013-11-21 07:24:19

Autosubmit to Perforce from ItemSet-SCBO2544_BASE_SCBOBASE_3069.xml; ItemSet originally created by LM9 on SCBOBASE at 2013-11-21 12:24:16

add //custom_ccrs/scx/SCBO/T2010/BASE/misc/User/CTHospital/B070D726-529A-11E3-BF32-5056A8569F00.xml (1) view / diff / history
add //custom_ccrs/scx/SCBO/T2010/BASE/misc/User/CTLoc/0FAE5038-529B-11E3-B8DF-5056A8569F00.xml (1) view / diff / history
add //custom_ccrs/scx/SCBO/T2010/BASE/misc/User/PACWard/0F512C64-529B-11E3-B8DF-5056A8569F00.xml (1) view / diff / history
add //custom_ccrs/scx/SCBO/T2010/BASE/misc/epr/FPView/65DF975A-529B-11E3-B2FC-5056A8569F00.xml (1) view / diff / history

Quiz: Moving Cancelled Changelist to a New CCR

Question:

Moving a cancelled changelist to a new CCR requires re-submitting content in a new changelist with the new CCR number. True or False?

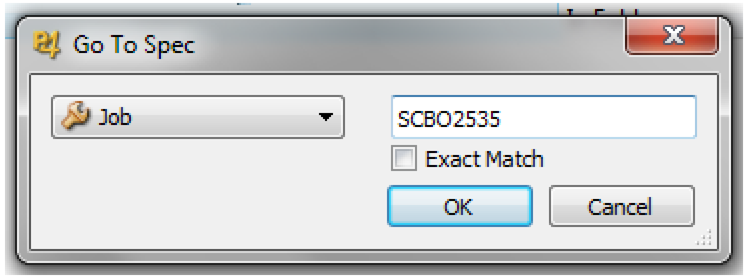
Answer: False.

You are merely updating the same changelist with the new job number. You are not re-submitting the changes.

Part 3: Back out Changes Using Perforce

Backout Changes using Perforce

- Only way backout some (rather than all) of the changelists submitted while In_BASE:
  1. Log into Perforce.
  2. Search > Go To…
     - Keyboard shortcut: Ctrl + G.
  3. Choose job.
  4. Enter your CCR number.
  5. Click OK.



Go To Spec

Job: SCBO2535

□ Exact Match

Backout changes using Perforce (cont.)

- Only way backout some (rather than all) of the changelists submitted while In_BASE:

6. Right-click > Undo Changes in Changelist #####.
7. Pending changelist: New.
8. Click Save to Changelist.
9. Repeat for each changelist that should be backed out.
    - Note: May need to submit changelist per instructions on next slide before backing out another changelist. Necessary when backing out multiple changelists on same file.

Backout changes using Perforce (cont.)

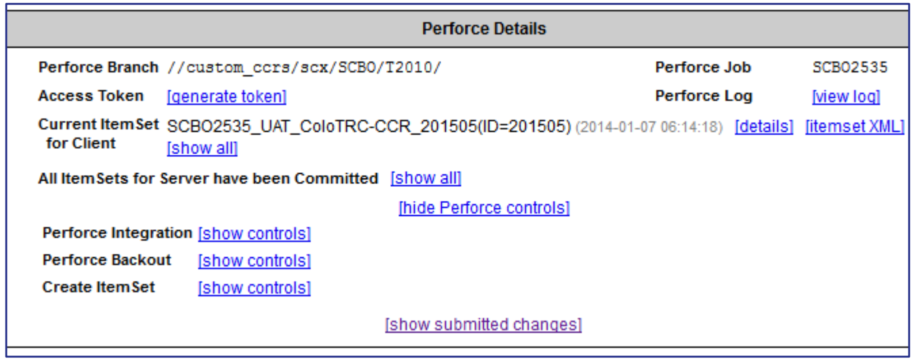
- Only way backout some (rather than all) of the changelists submitted while In_BASE:

10. Select the Pending Change List Tab.
11. Right-click changelist with the backout > Submit.
12. Add in the changelist description.
13. Enter job number (CCR number).
14. Click Add.
15. Click Submit.

Backout changes using Perforce (cont.)

- Only way backout some (rather than all of the changelists submitted while In_BASE:

16. Open the CCR.
17. Show Perforce controls.
18. Click [show controls] for Create ItemSet.
19. Select Target environment: BASE.
20. Deploy ItemSet.



Perforce Details

Perforce Branch //custom_ccrs/scx/SCBO/T2010/ Perforce Job SCBO2535

Access Token [generate token] Perforce Log [view log]

Current ItemSet for Client SCBO2535_UAT_ColoTRC-CCR_201505(ID=201505) (2014-01-07 06:14:18) [details] [itemset XML] [show all]

All ItemSets for Server have been Committed [show all]

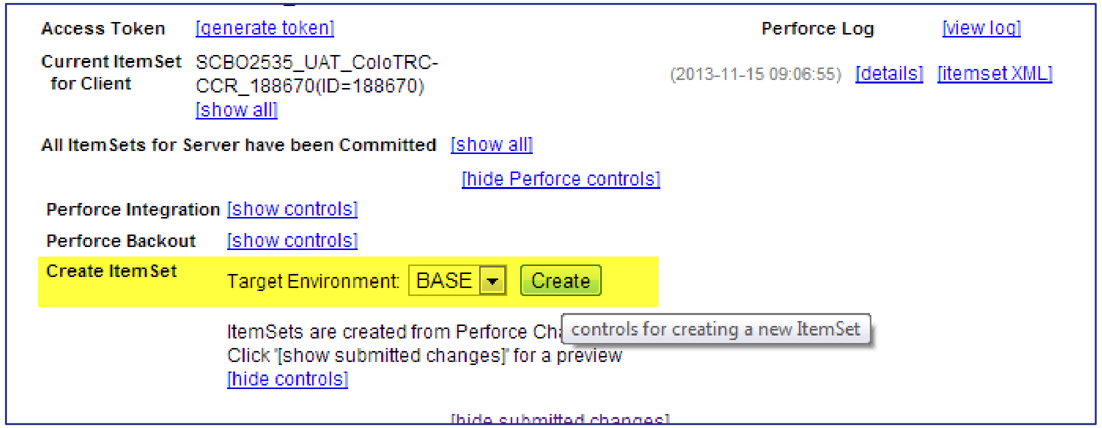
[hide Perforce controls]

Perforce Integration [show controls]

Perforce Backout [show controls]

Create ItemSet [show controls]

[show submitted changes]



Access Token [generate token] Perforce Log [view log]

Current ItemSet for Client SCBO2535_UAT_ColoTRC-CCR_188670(ID=188670) (2013-11-15 09:06:55) [details] [itemset XML] [show all]

All ItemSets for Server have been Committed [show all]

[hide Perforce controls]

Perforce Integration [show controls]

Perforce Backout [show controls]

Create ItemSet Target Environment: BASE Create

ItemSets are created from Perforce Ch controls for creating a new ItemSet
Click '[show submitted changes]' for a preview
[hide controls]

[hide submitted changes]

Quiz: Backing Out Changes

Question:

It is possible to backout changes from within the CCR application. True or False?

Answer: True.

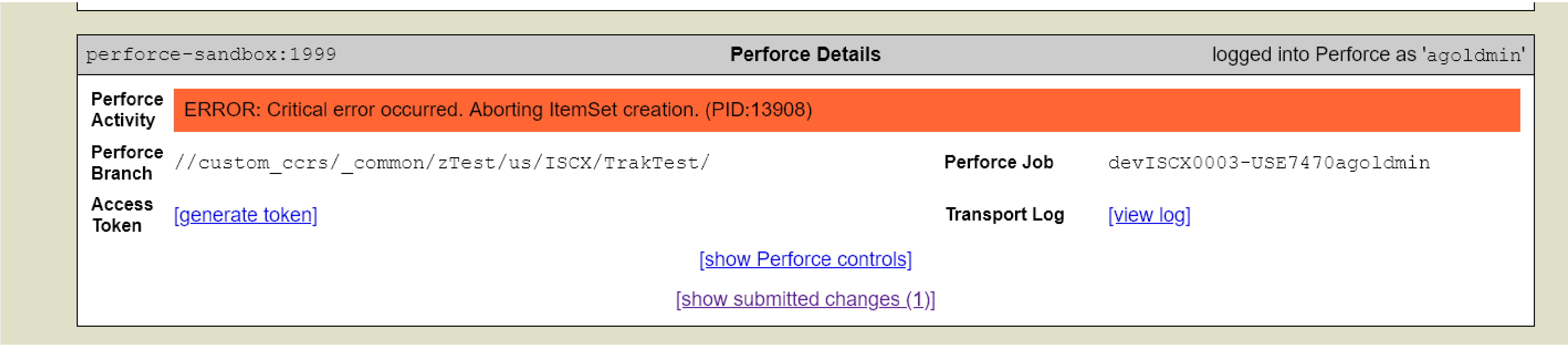
Frequently, changes can be backed out from within the CCR application when cancelling a CCR.

If it’s not possible, make sure to follow the previous steps to backout using Perforce.

Part 4: Resolving merge conflicts in P4V

Resolving merge conflicts in P4V

- Sometimes you will see integration errors when moving a CCR to the next phase.



perforce-sandbox:1999 **Perforce Details** logged into Perforce as 'agoldmin'

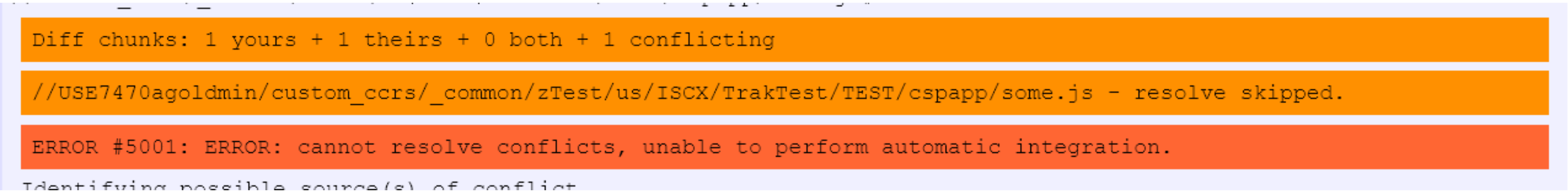
Perforce Activity	ERROR: Critical error occurred. Aborting ItemSet creation. (PID:13908)		
Perforce Branch	//custom_ccrs/_common/zTest/us/ISCX/TrakTest/	**Perforce Job**	devISCX0003-USE7470agoldmin
Access Token	[generate token]	**Transport Log**	[view log]

[show Perforce controls]

[show submitted changes (1)]

Resolving merge conflicts in P4V (cont.)

- You can check the Transport Log for details of the error.

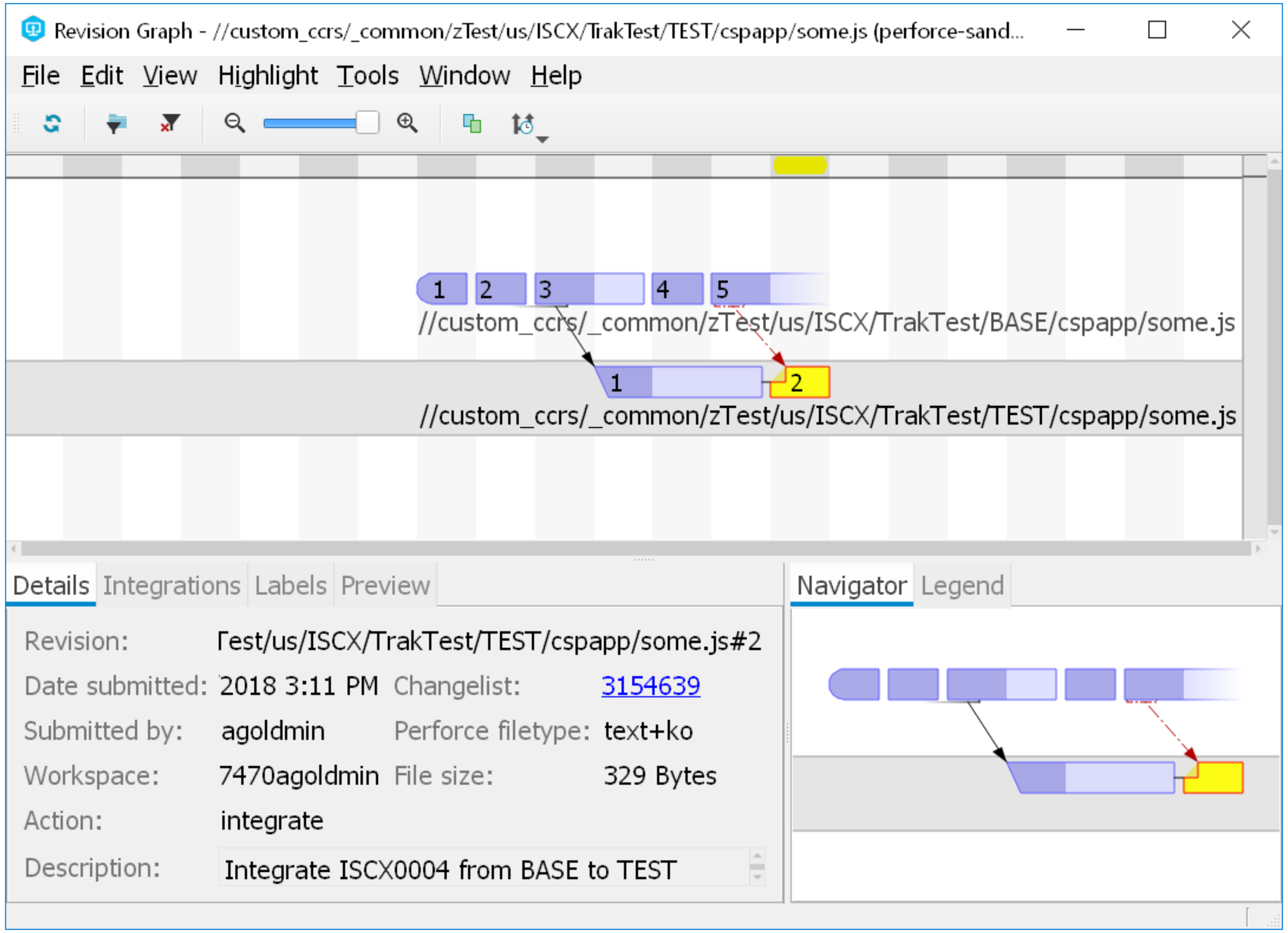


```
Diff chunks: 1 yours + 1 theirs + 0 both + 1 conflicting
//USE7470agoldmin/custom_ccrs/_common/zTest/us/ISCX/TrakTest/TEST/cspapp/some.js - resolve skipped.
ERROR #5001: ERROR: cannot resolve conflicts, unable to perform automatic integration.
Identifying possible source(s) of conflict
```

- This will tell you both what the error was, and which file caused the problem.

Resolving merge conflicts in P4V (cont.)

- Looking at the file’s integration history can be helpful to understand why there are conflicts.
- In this example, revision 5 needed to progress without revision 4.
  - Screenshot shows the state after manual integration.



Revision Graph - //custom_ccrs/_common/zTest/us/ISCX/TrakTest/TEST/cspapp/some.js (perforce-sand...
File Edit View Highlight Tools Window Help
1 2 3 4 5
//custom_ccrs/_common/zTest/us/ISCX/TrakTest/BASE/cspapp/some.js
1 2
//custom_ccrs/_common/zTest/us/ISCX/TrakTest/TEST/cspapp/some.js
Details Integrations Labels Preview
Navigator Legend

Revision: Test/us/ISCX/TrakTest/TEST/cspapp/some.js#2

Date submitted: 2018 3:11 PM Changelist: 3154639

Submitted by: agoldmin Perforce filetype: text+ko

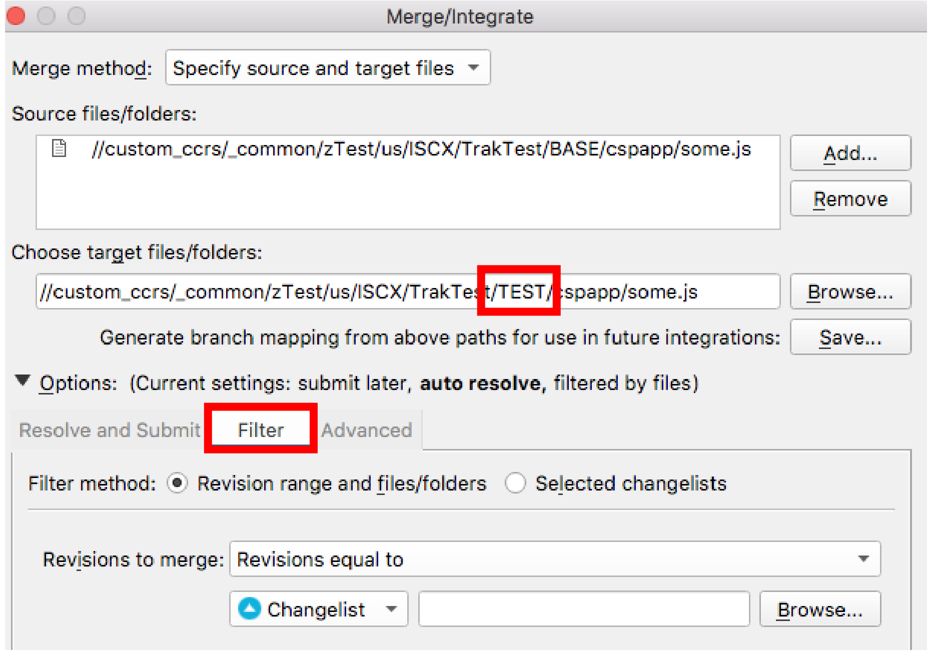
Workspace: 7470agoldmin File size: 329 Bytes

Action: integrate

Description: Integrate ISCX0004 from BASE to TEST

Resolving merge conflicts in P4V (cont.)

- Find the conflicting file(s) in question in the workspace view in P4V and select “Merge/Integrate” from the context menu.
  - NOTE – this manual step is only for conflicting files; after correcting them by hand use the CCR Integration Tool to manually integrate the remainder of the files.



Merge method: Specify source and target files

Source files/folders: //custom_ccrs/_common/zTest/us/ISCX/TrakTest/BASE/cspapp/some.js

Choose target files/folders: //custom_ccrs/_common/zTest/us/ISCX/TrakTest/TEST/cspapp/some.js

Generate branch mapping from above paths for use in future integrations:

Options: (Current settings: submit later, **auto resolve,** filtered by files)

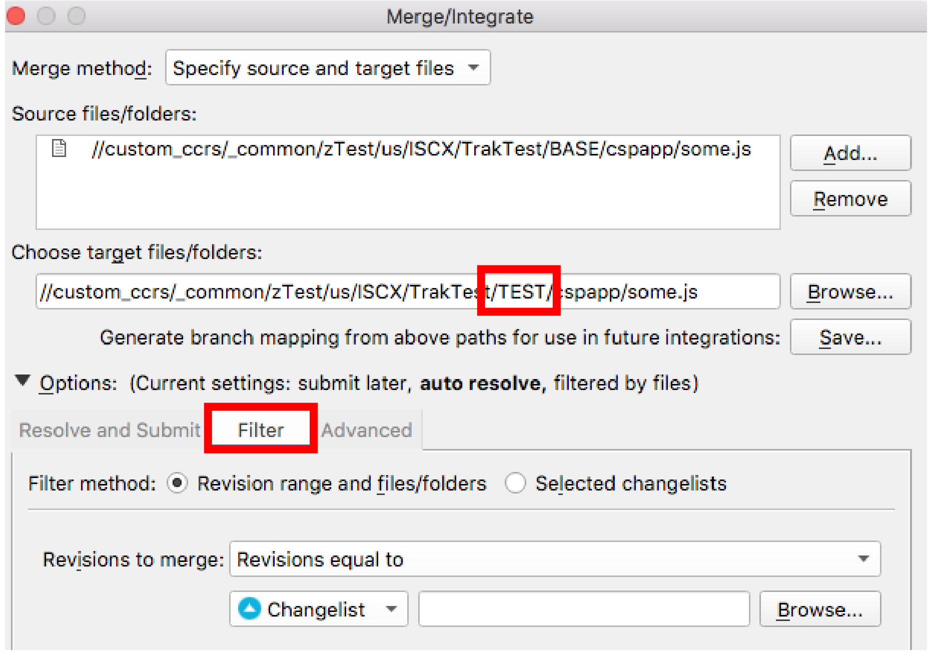
Filter method: ● Revision range and files/folders ○ Selected changelists

Revisions to merge: Revisions equal to

Changelist

Resolving merge conflicts in P4V (cont.)

- Make sure the target path is set to the corresponding file in the TEST branch.
  - Target is *case sensitive* (i.e. 'TEST' not 'test').
- On "Filter" tab, select just the changelists you want to integrate.
- NOTE: After finishing the merge, the merged file(s) will need to be resolved before they can be submitted.



Merge/Integrate

Merge method: Specify source and target files

Source files/folders:

//custom_ccrs/_common/zTest/us/ISCX/TrakTest/BASE/cspapp/some.js

Choose target files/folders:

//custom_ccrs/_common/zTest/us/ISCX/TrakTest/TEST/cspapp/some.js

Generate branch mapping from above paths for use in future integrations:

Options: (Current settings: submit later, **auto resolve,** filtered by files)

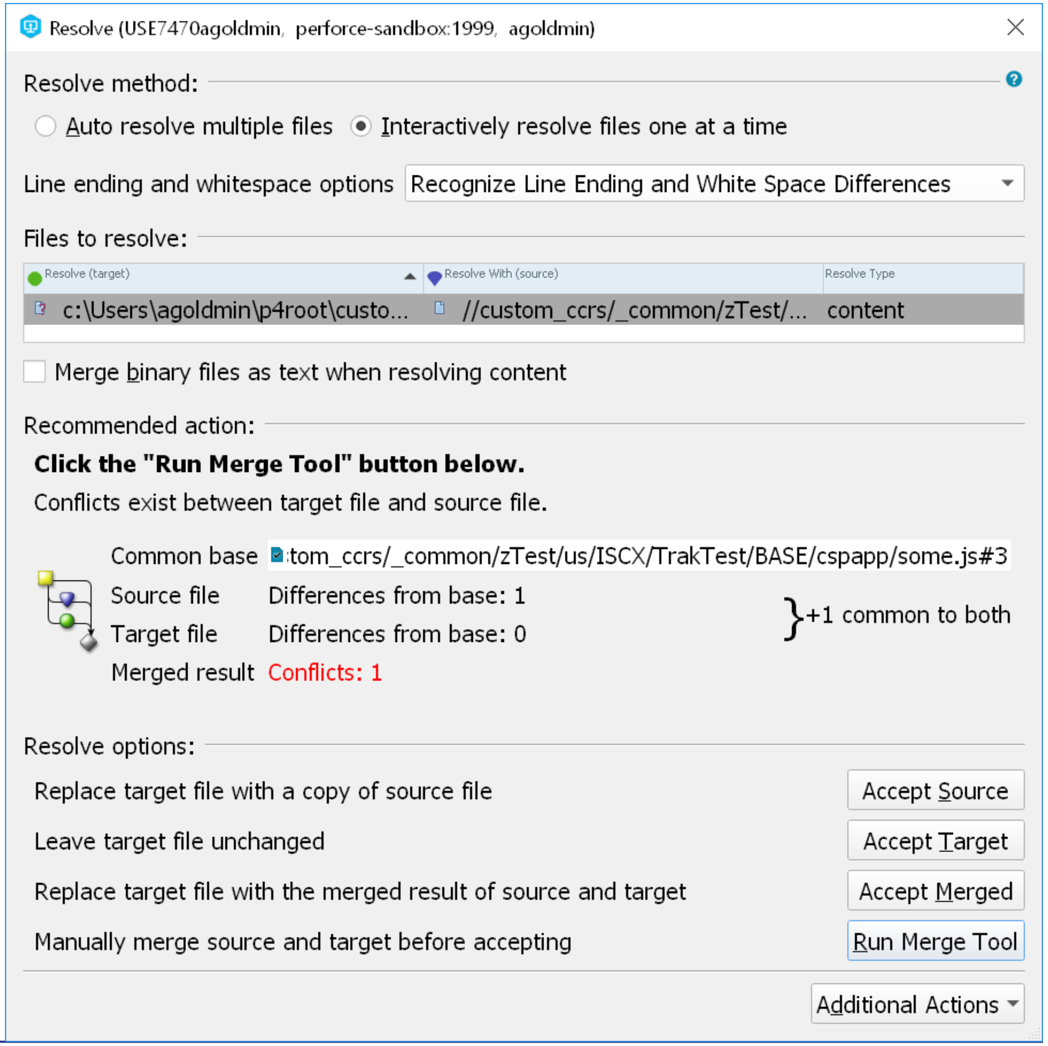
Filter method: ● Revision range and files/folders ○ Selected changelists

Revisions to merge: Revisions equal to

Changelist

Resolving merge conflicts in P4V (cont.)

- Find the file in the target branch and select “Resolve” from the context menu.
- You will see a summary of the changes in the source and target files and options to resolve the merge.



Resolve (USE7470agoldmin, perforce-sandbox:1999, agoldmin)

Resolve method:

○ Auto resolve multiple files ● Interactively resolve files one at a time

Line ending and whitespace options: Recognize Line Ending and White Space Differences

Files to resolve:

Resolve (target)	Resolve With (source)	Resolve Type
c:\Users\agoldmin\p4root\custo...	//custom_ccrs/_common/zTest/...	content

□ Merge binary files as text when resolving content

Recommended action:

Click the "Run Merge Tool" button below.

Conflicts exist between target file and source file.

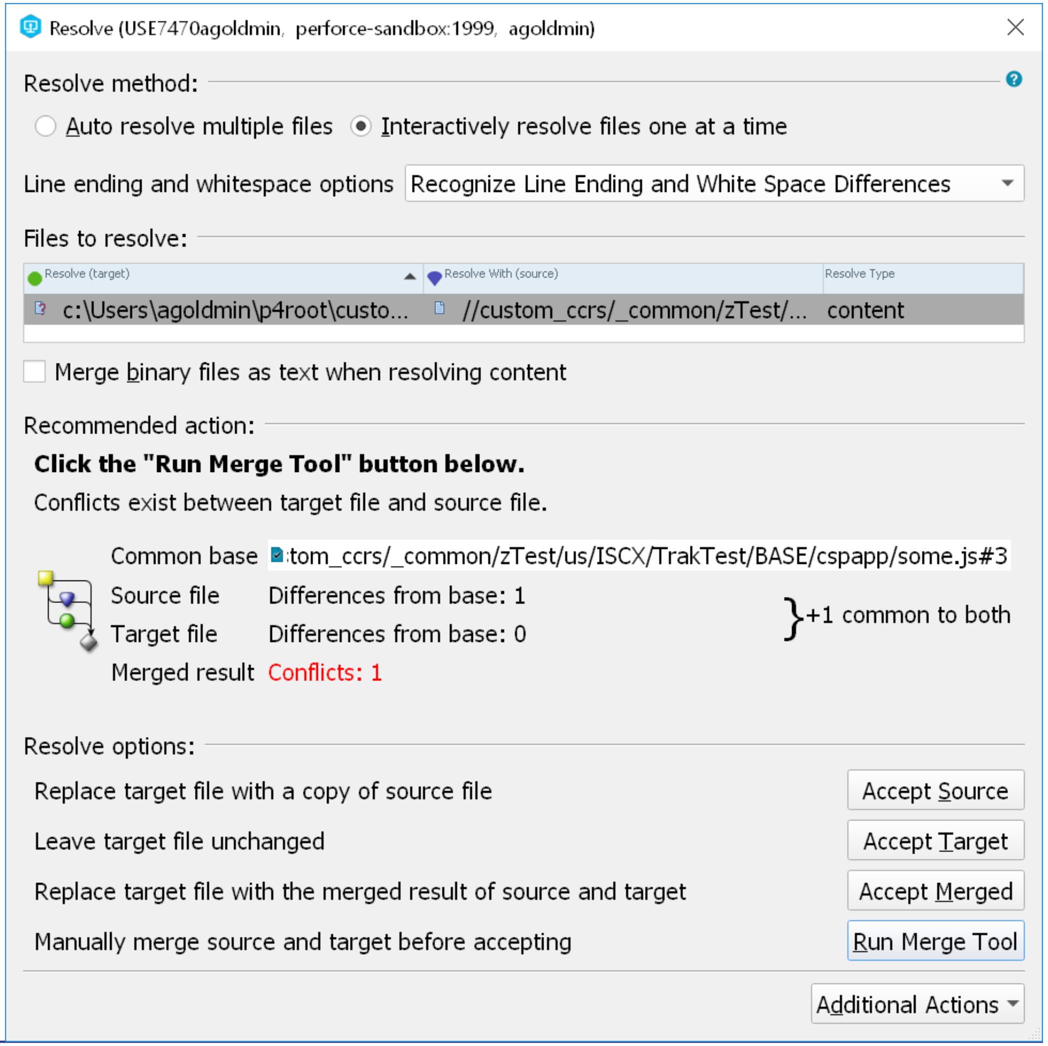
Common base	:tom_ccrs/_common/zTest/us/ISCX/TrakTest/BASE/cspapp/some.js#3	
Source file	Differences from base: 1	}+1 common to both
Target file	Differences from base: 0	
Merged result	Conflicts: 1	

Resolve options:

Replace target file with a copy of source file	Accept Source
Leave target file unchanged	Accept Target
Replace target file with the merged result of source and target	Accept Merged
Manually merge source and target before accepting	Run Merge Tool

Resolving merge conflicts in P4V (cont.)

- Run the merge tool to see a view of the file contents in the various versions of the file so that you can decide how best to merge.
- This requires some knowledge of what the
contents of the file mean to ensure that the result makes sense.



Resolve (USE7470agoldmin, perforce-sandbox:1999, agoldmin)

Resolve method:

○ Auto resolve multiple files ● Interactively resolve files one at a time

Line ending and whitespace options: Recognize Line Ending and White Space Differences

Files to resolve:

Resolve (target)	Resolve With (source)	Resolve Type
c:\Users\agoldmin\p4root\custo...	//custom_ccrs/_common/zTest/...	content

□ Merge binary files as text when resolving content

Recommended action:

Click the "Run Merge Tool" button below.

Conflicts exist between target file and source file.

Common base	:tom_ccrs/_common/zTest/us/ISCX/TrakTest/BASE/cspapp/some.js#3
Source file	Differences from base: 1
Target file	Differences from base: 0
Merged result	Conflicts: 1

}+1 common to both

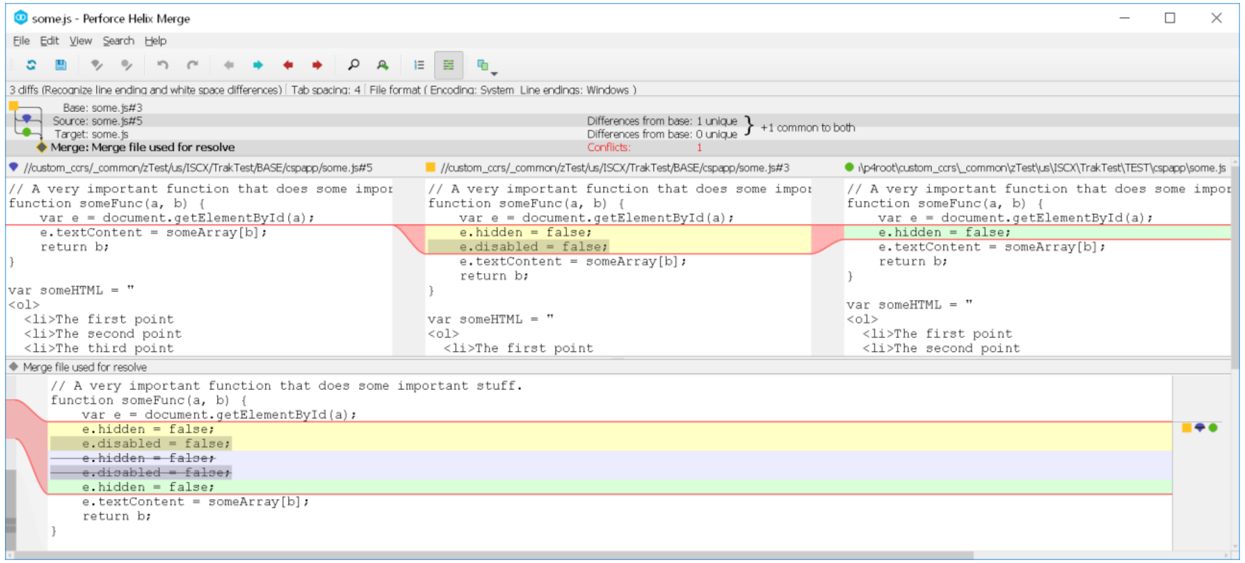
Resolve options:

Replace target file with a copy of source file	Accept Source
Leave target file unchanged	Accept Target
Replace target file with the merged result of source and target	Accept Merged
Manually merge source and target before accepting	Run Merge Tool

Additional Actions

Resolving merge conflicts in P4V (cont.)

- The merge tool on the right shows a view of the file contents in the base, source, and target revisions, as well as the diffs between them.
- It is basically a diff view, but with three files rather than two.
- Select the icons to the right of the conflicting lines to select the desired version.
- NOTE: Edits can also be made to get the desired result.



3 diffs (Recognize line ending and white space differences) | Tab spacing: 4 | File format (Encoding: System Line endings: Windows)

Base: some.js#3
Source: some.js#5
Target: some.js
Merge: Merge file used for resolve

Differences from base: 1 unique } +1 common to both
Differences from base: 0 unique
Conflicts: 1

//custom_ccrs/_common/zTest/us/ISCX/TrakTest/BASE/cspapp/some.js#5

```
// A very important function that does some impor
function someFunc(a, b) {
    var e = document.getElementById(a);
    e.textContent = someArray[b];
    return b;
}

var someHTML = "
<ol>
  <li>The first point
  <li>The second point
  <li>The third point
```

//custom_ccrs/_common/zTest/us/ISCX/TrakTest/BASE/cspapp/some.js#3

```
// A very important function that does some impor
function someFunc(a, b) {
    var e = document.getElementById(a);
    e.hidden = false;
    e.disabled = false;
    e.textContent = someArray[b];
    return b;
}

var someHTML = "
<ol>
  <li>The first point
```

\p4root\custom_ccrs_common\zTest\us\ISCX\TrakTest\TEST\cspapp\some.js

```
// A very important function that does some impor
function someFunc(a, b) {
    var e = document.getElementById(a);
    e.hidden = false;
    e.textContent = someArray[b];
    return b;
}

var someHTML = "
<ol>
  <li>The first point
  <li>The second point
```

Merge file used for resolve

```
// A very important function that does some important stuff.
function someFunc(a, b) {
    var e = document.getElementById(a);
    e.hidden = false;
    e.disabled = false;
    ~~e.hidden = false;~~
    ~~e.disabled = false;~~
    e.hidden = false;
    e.textContent = someArray[b];
    return b;
}
```

Resolving merge conflicts in P4V (cont.)

- Click the “Save” icon once the merge is complete.
- When prompted about replacing the item in your workspace with the merged file, answer “Yes.”
- Now the change is ready to be submitted to Perforce.
  - **Don’t forget to attach the Perforce job (i.e the CCR ID) to the changelist so that CCR can see the changelist and include it in the ItemSet.**

Resolving merge conflicts in P4V (cont.)

- If there are other files that didn’t have a conflict, they can be auto-integrated with the CCR Perforce Tools.
- Make sure an ItemSet is created; deploy it and continue moving the CCR forward.

Quiz: Resolving Merge Conflicts

Question:

If the integration of one file in the CCR needs to be merged by hand, then all files in that CCR must be manually merged as well. True or False?

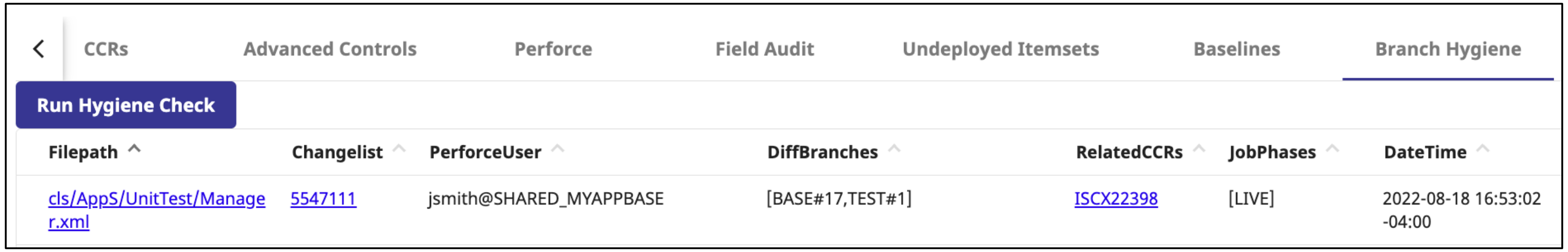
Answer: False.

Only the conflicting file must be manually merged; once the conflicting file is integrated, the other files can be integrated using the Perforce tools on the CCR.

Part 5:
Fixing integration history with p4 integrate -f

Branch Hygiene Tool

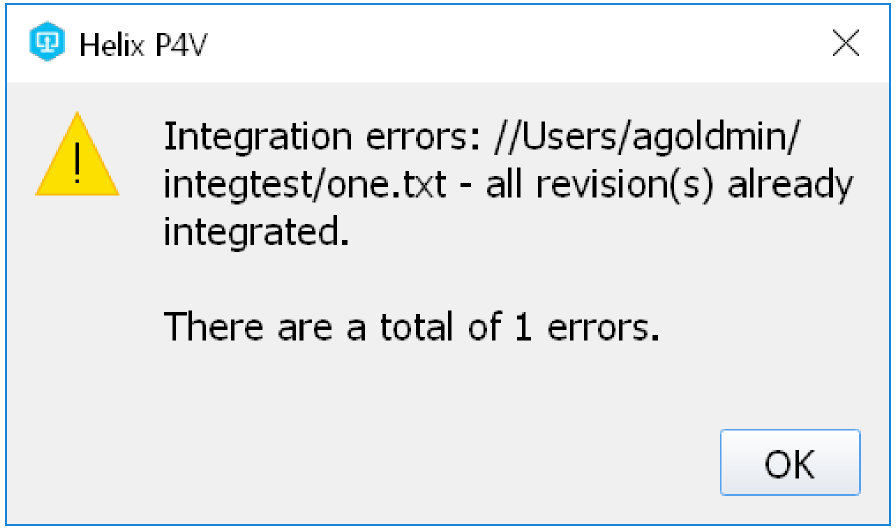
- Branch Hygiene tab on system definition shows files out of sync between branches.
  - Future changes to the files may have merge conflicts.
- To fix discrepancies:
  - For cancelled CCRs not backed out, back them out.
  - Use P4V to manually integrate changes.
    - Be sure to create a CCR and attach changelists.



Filepath	Changelist	PerforceUser	DiffBranches	RelatedCCRs	JobPhases	DateTime
cls/AppS/UnitTest/Manager.xml	5547111	jsmith@SHARED_MYAPPBASE	[BASE#17,TEST#1]	ISCX22398	[LIVE]	2022-08-18 16:53:02 -04:00

Fixing integration history with integrate -f

- Sometimes there are mistakes in p4 integrations.
  - E.g. resolving “accept target” instead of “accept source”.
  - Or integrating from the wrong source.

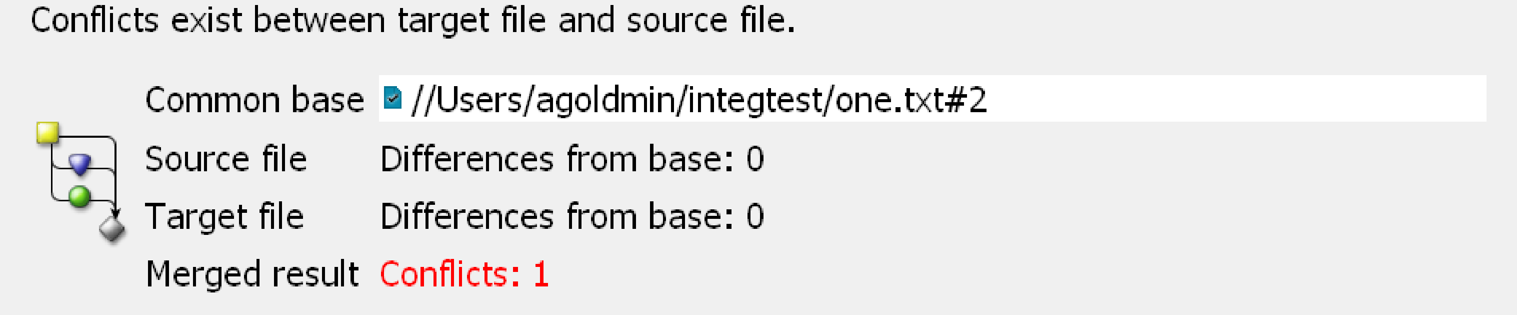


Integration errors: //Users/agoldmin/integtest/one.txt - all revision(s) already integrated.

There are a total of 1 errors.

Fixing integration history with integrate -f (cont.)

- Perforce won’t let you integrate the same change twice into the same file.
- Possibly more changes were checked into the source file, causing merge conflicts.

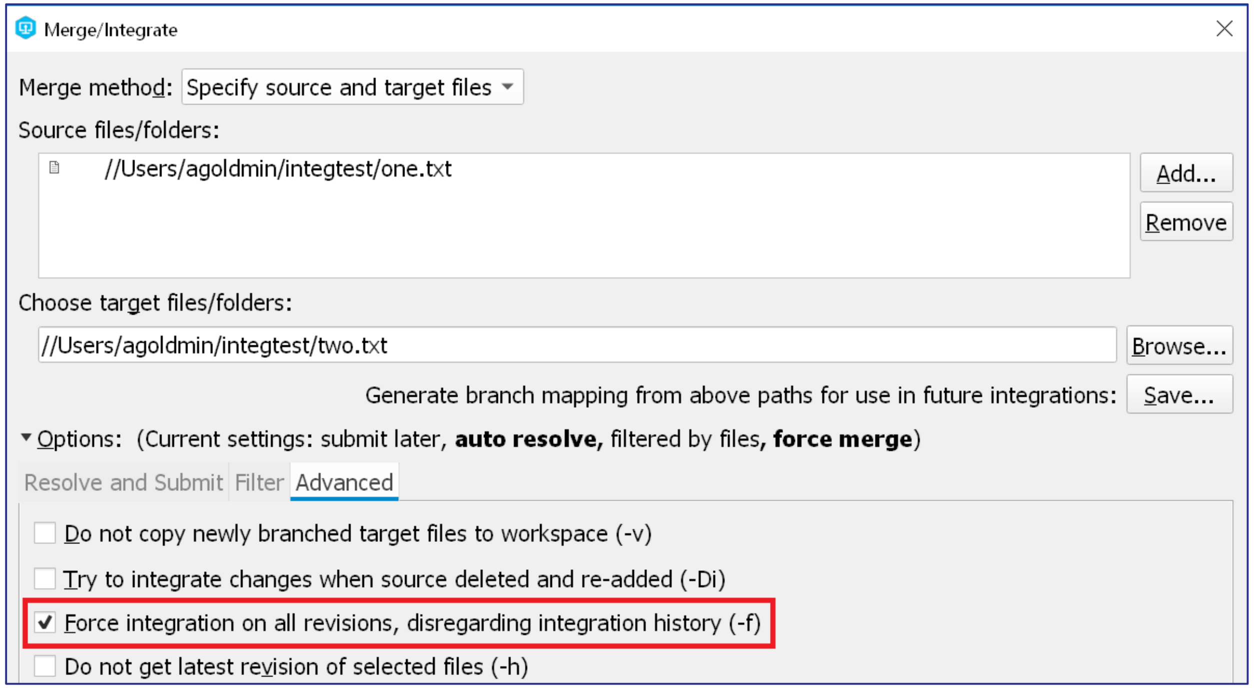


Conflicts exist between target file and source file.

Common base	//Users/agoldmin/integtest/one.txt#2
Source file	Differences from base: 0
Target file	Differences from base: 0
Merged result	Conflicts: 1

Fixing integration history with integrate -f (cont.)

- "p4 integrate -f" can help if you have a version of the file you want to copy forward to all branches.
- It will copy all the file contents and use it as a basis for all future integration history.



Merge/Integrate

Merge method: Specify source and target files

Source files/folders:

//Users/agoldmin/integtest/one.txt

Choose target files/folders:

//Users/agoldmin/integtest/two.txt

Generate branch mapping from above paths for use in future integrations: Save...

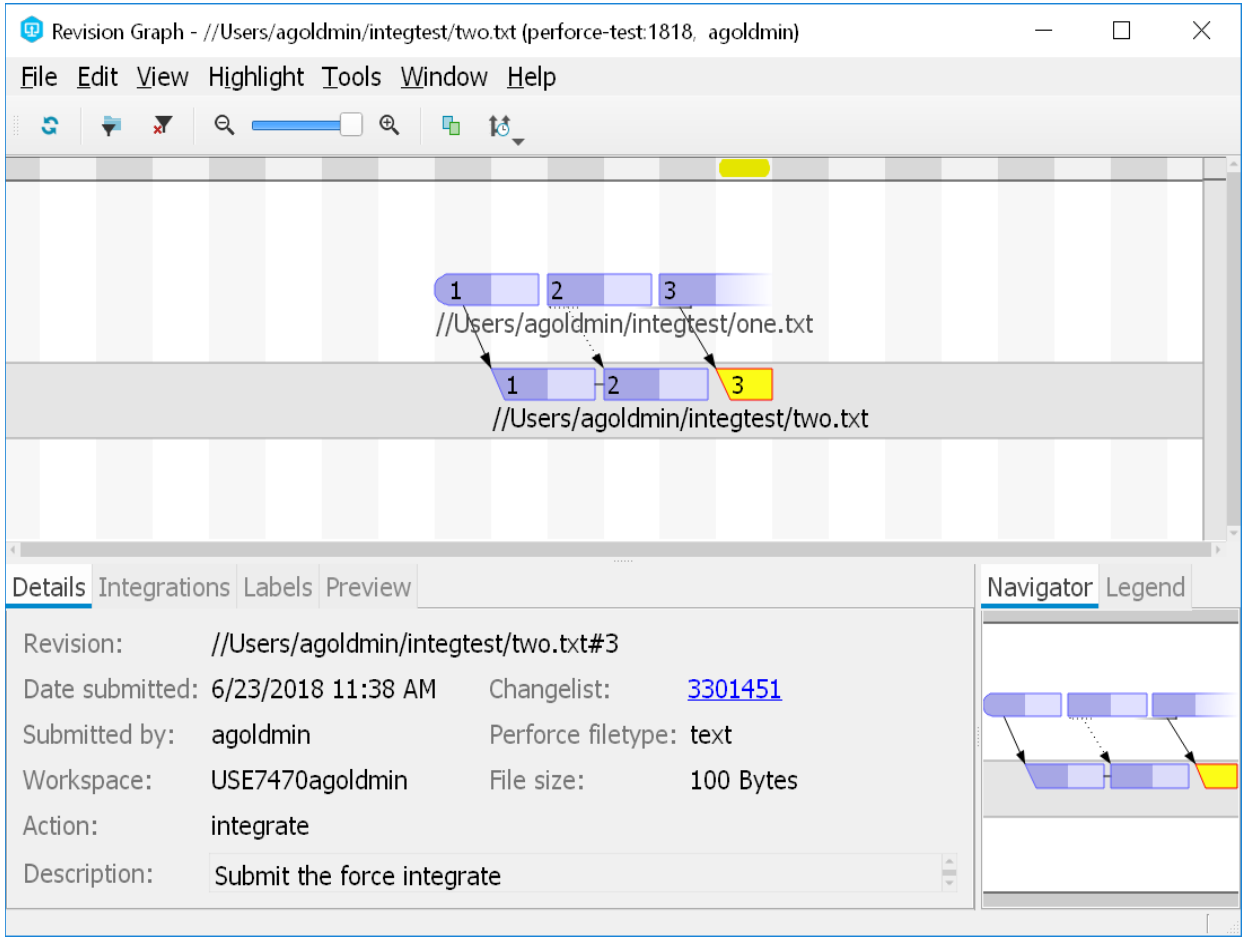
Options: (Current settings: submit later, **auto resolve,** filtered by files, **force merge**)

Resolve and Submit | Filter | Advanced

- ☐ Do not copy newly branched target files to workspace (-v)
- ☐ Try to integrate changes when source deleted and re-added (-Di)
- ☑ Force integration on all revisions, disregarding integration history (-f)
- ☐ Do not get latest revision of selected files (-h)

Fixing integration history with integrate -f (cont.)

- Results in Revision Graph like this:
- Note the diagonal corner on two.txt#3.
- The integration from one.txt#3 overwrites two.txt#2.


Revision Graph - //Users/agoldmin/integtest/two.txt (perforce-test:1818, agoldmin)
File Edit View Highlight Tools Window Help
1
2
3
//Users/agoldmin/integtest/one.txt
1
2
3
//Users/agoldmin/integtest/two.txt
Details Integrations Labels Preview
Navigator Legend

Revision: //Users/agoldmin/integtest/two.txt#3

Date submitted: 6/23/2018 11:38 AM Changelist: 3301451

Submitted by: agoldmin Perforce filetype: text

Workspace: USE7470agoldmin File size: 100 Bytes

Action: integrate

Description: Submit the force integrate

Part 6: Perforce Refresh

Perforce Refresh

- Refreshing the Perforce branches based on contents of LIVE:
  - To ensure that all configuration matches in each Perforce branch.
  - Clears integration history to prevent merge conflicts.
  - Can be done during working hours as it will have no impact on performance for users.
  - Before refresh ensure that:
    - Source Control Hooks (Tier 1) and/or Change Control Hooks (Tier 2) have been Locked.
    - Progress all CCR's to terminal state (Closed, Merged, or Cancelled).

Perforce Refresh (cont.)

- Refreshing the Perforce branches based on contents of LIVE:
  - Perforce refresh done as part of rebaseline.
  - See rebaselining in ICC520 for details on how to perform refresh as part of rebaseline.

Quiz: Perforce Refresh

Question:

A Perforce refresh should be done during off hours because it can affect performance for the client. True or False?

Answer: False.

Summary

- What are the key points for this module?

ICC 470: CCR Transport – Release Management

InterSystems
Creative data technology

Objectives

- Explain how release management can be used in CCR to manage the flow of common changes from an Edition out to a collection of different Child Systems.
- Demonstrate how to configure an Edition and one or more Edition Child Systems to receive releases from that Edition.

Part 1: Understanding Release Management

Releases

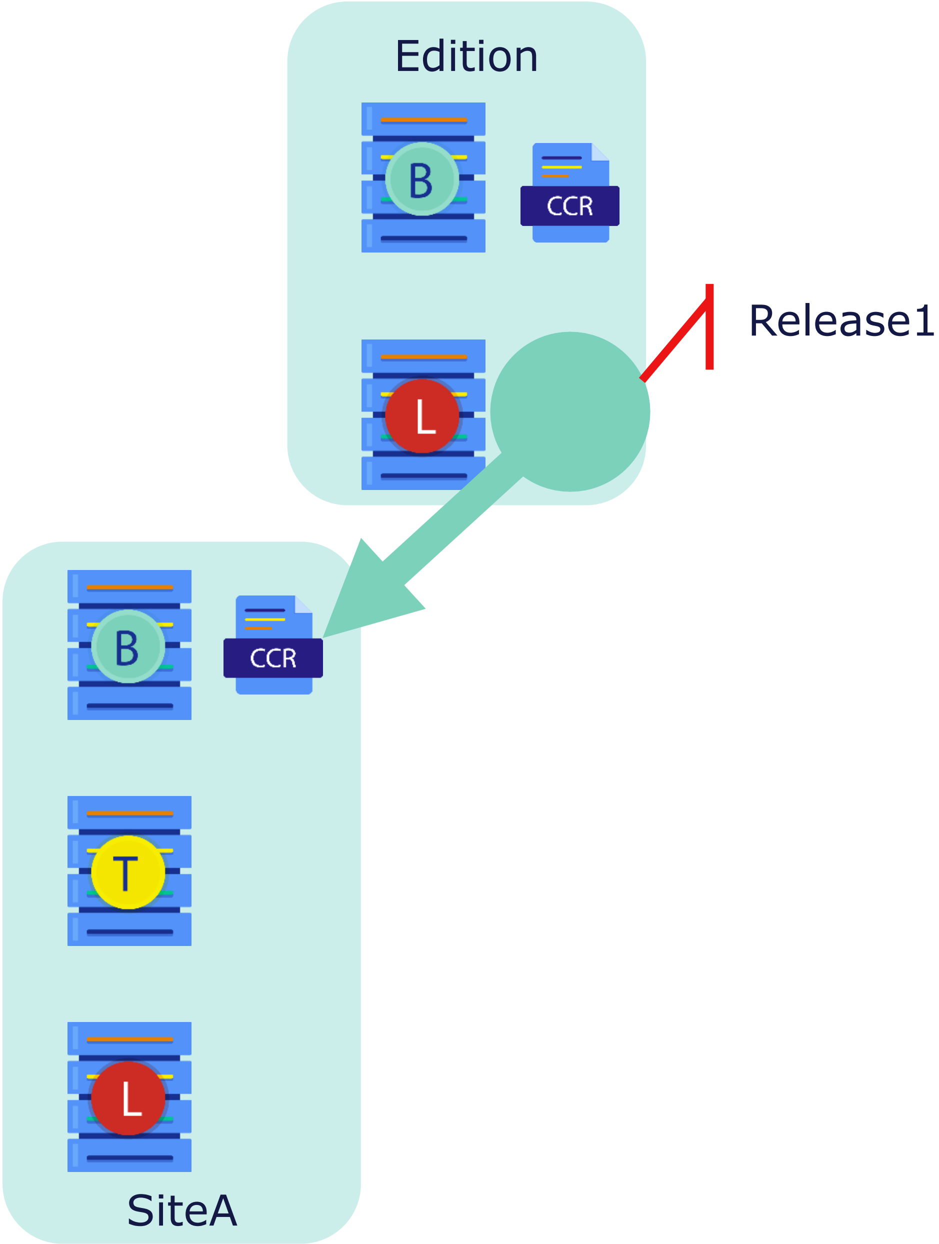
- CCR implemented functionality to automate release management:
  - Intended for Tier 1 and Tier 2 Systems.
  - Releases built from one Edition System.
  - Edition Child Systems configured to be tied to an Edition System.
    - An Edition Child is downstream from an Edition System and receives releases from it.
  - All Perforce integrations are automatic and merge all changes from the group of release CCRs from the Edition/LIVE branch to Child/BASE branch.
  - Releases provide automated tracking of deployment to each downstream Edition Child.

Release Use Cases

- CCR release management continues to be helpful when multiple customer Systems inherit customizations from a common upstream code-base.
  - Custom monitoring utilities.

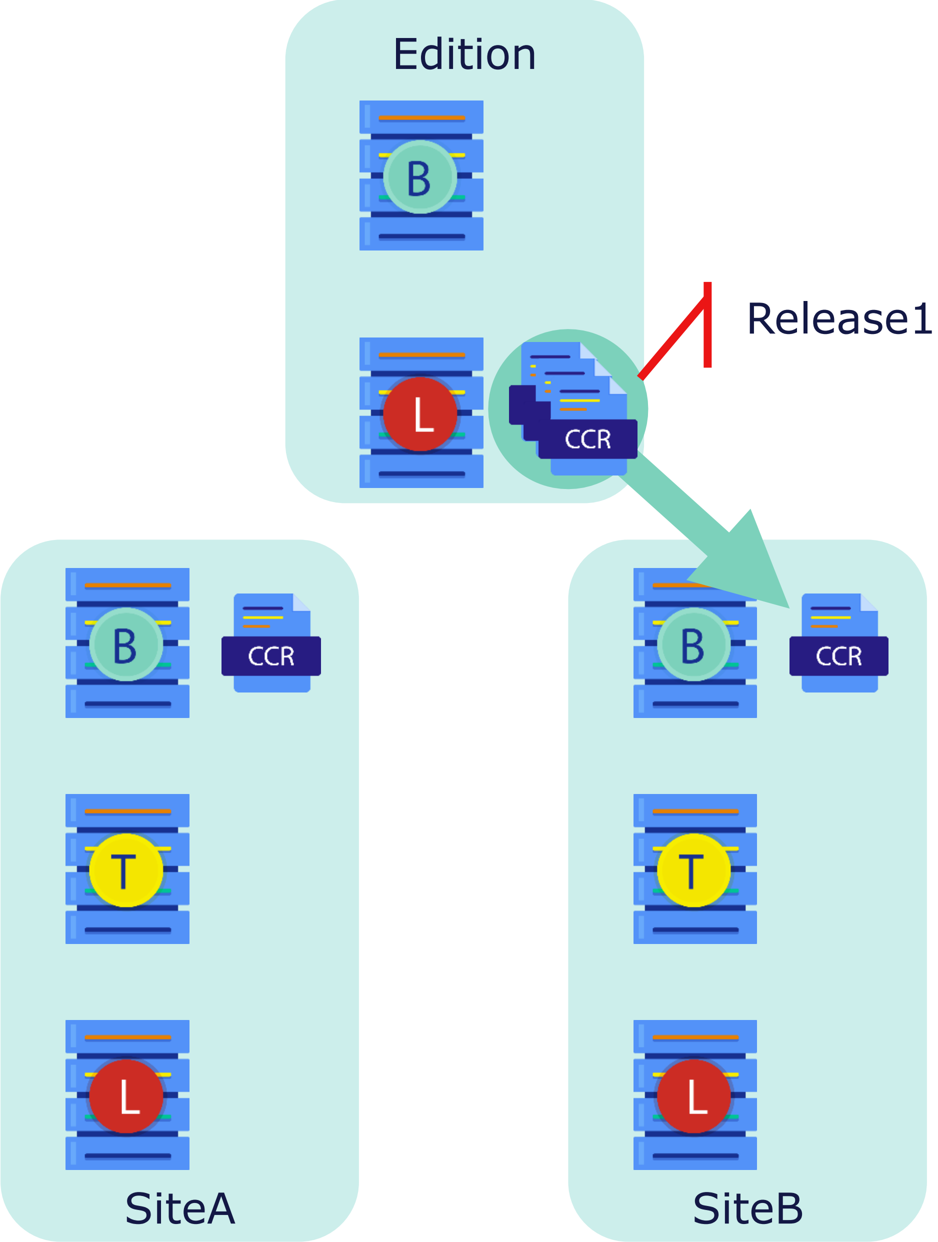
Release Flow

- CCRs flow from BASE to LIVE on Edition.
- Collection of Edition CCRs identified as Release1.
- SiteA is ready to accept Release1.
- All CCRs in Release1 integrated from Edition/LIVE to SiteA/BASE and a single CCR is created for SiteA.



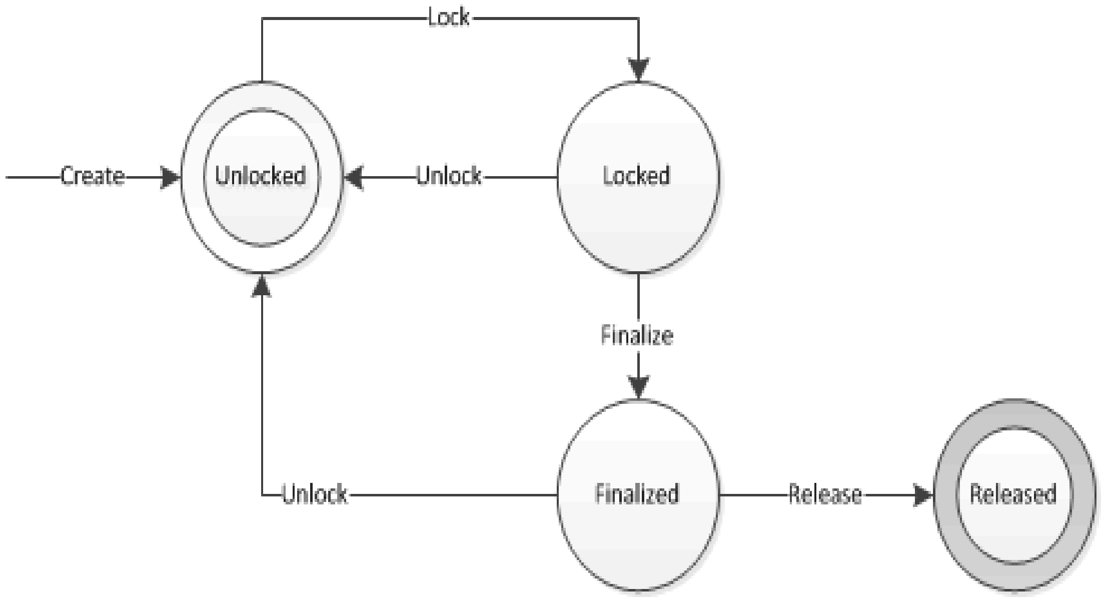
Release Flow (cont.)

- Release CCR moves through environments of SiteA
- SiteB is ready to accept Release1
- All CCRs in Release1 integrated from Edition/LIVE to SiteB/BASE and a single CCR is created for SiteB
- Release CCR moves through environments of SiteB



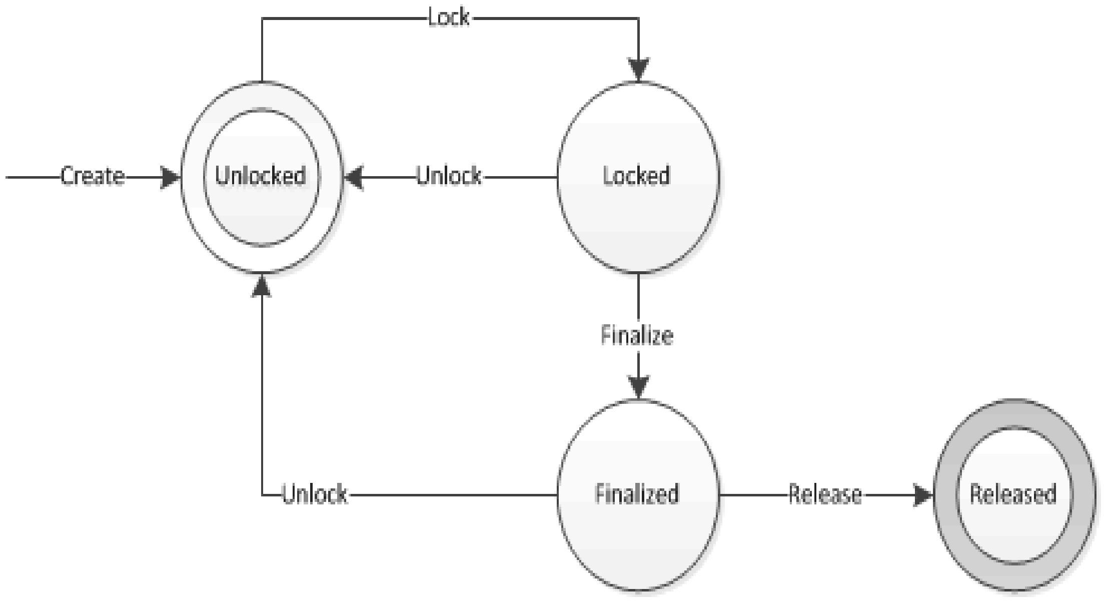
Release Lifecycle

- Unlocked.
  - Add CCRs to the Release.
- Locked.
  - Progress attached CCRs to terminal states.
- Finalized.
  - Assign the Release to an internal (testing) child system.
- Released.
  - Assign the release to customer child system(s).



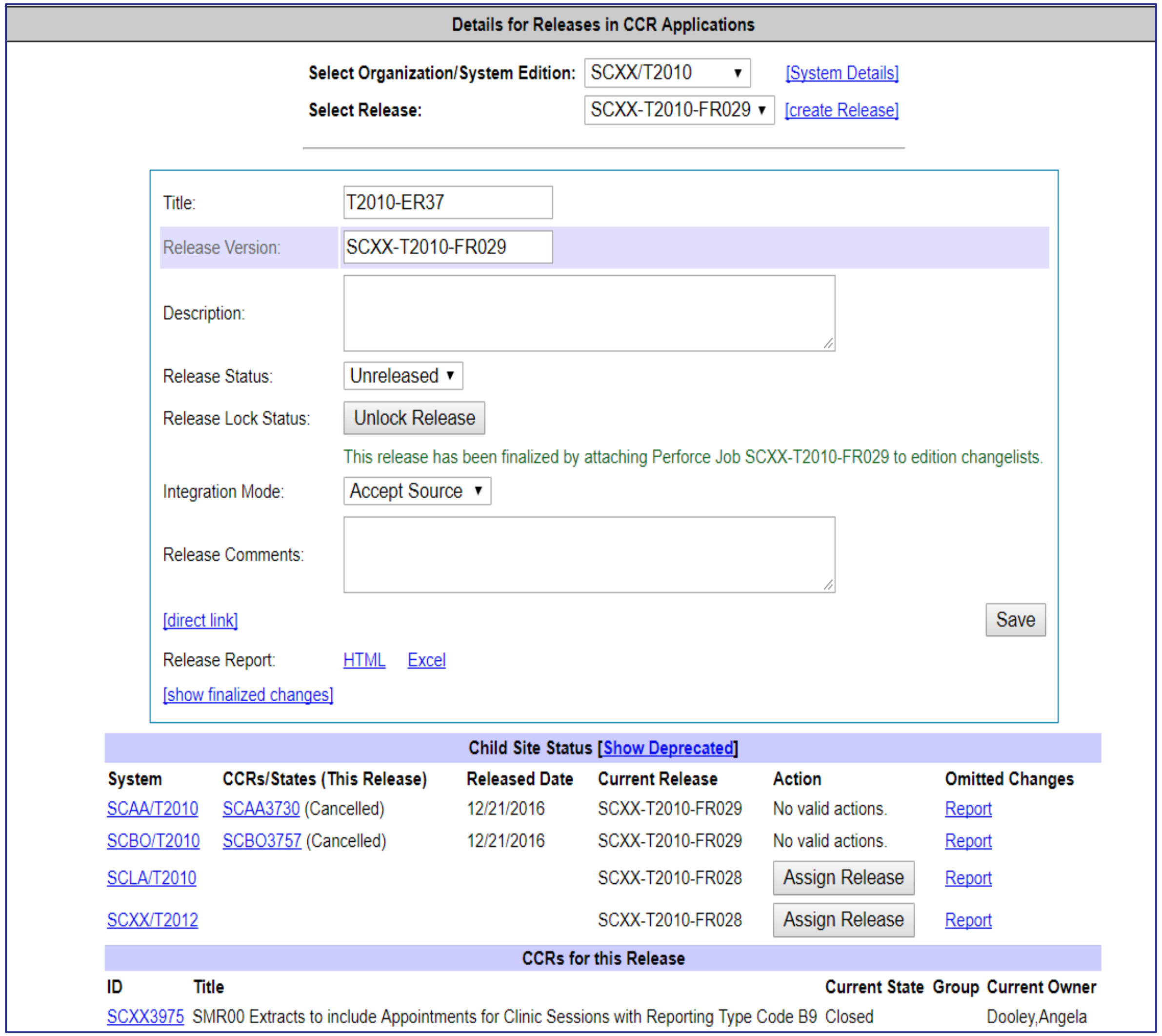
Release Lifecycle (cont.)

- See CCR online documentation on release management for more details on release states and release transitions.



Release User Interface

- Release Info menu option.
  - Create new releases.
  - Update Release status.
  - Lock Release to prevent changes.
  - Configure Integration Mode.
  - Download release notes report.
  - Omit changes and download omitted changes report.



Details for Releases in CCR Applications

Select Organization/System Edition: SCXX/T2010 ▾ [System Details]

Select Release: SCXX-T2010-FR029 ▾ [create Release]

Title:	T2010-ER37
Release Version:	SCXX-T2010-FR029
Description:	
Release Status:	Unreleased ▾
Release Lock Status:	Unlock Release This release has been finalized by attaching Perforce Job SCXX-T2010-FR029 to edition changelists.
Integration Mode:	Accept Source ▾
Release Comments:	

[direct link] Save

Release Report: HTML Excel

[show finalized changes]

Child Site Status [Show Deprecated]

System	CCRs/States (This Release)	Released Date	Current Release	Action	Omitted Changes
SCAA/T2010	SCAA3730 (Cancelled)	12/21/2016	SCXX-T2010-FR029	No valid actions.	Report
SCBO/T2010	SCBO3757 (Cancelled)	12/21/2016	SCXX-T2010-FR029	No valid actions.	Report
SCLA/T2010			SCXX-T2010-FR028	Assign Release	Report
SCXX/T2012			SCXX-T2010-FR028	Assign Release	Report

CCRs for this Release

ID	Title	Current State	Group	Current Owner
SCXX3975	SMR00 Extracts to include Appointments for Clinic Sessions with Reporting Type Code B9	Closed		Dooley,Angela

Quiz: Release Usage

Question:

Most CCR Systems worldwide should be set up to receive releases from an Edition System. True or False?

Answer:

False. TrakCare Edition management is now handled outside of CCR (by core Development processes); release management is an option for any region that attempts to create a locally managed HealthShare Edition, but most sites do not pull from an Edition.

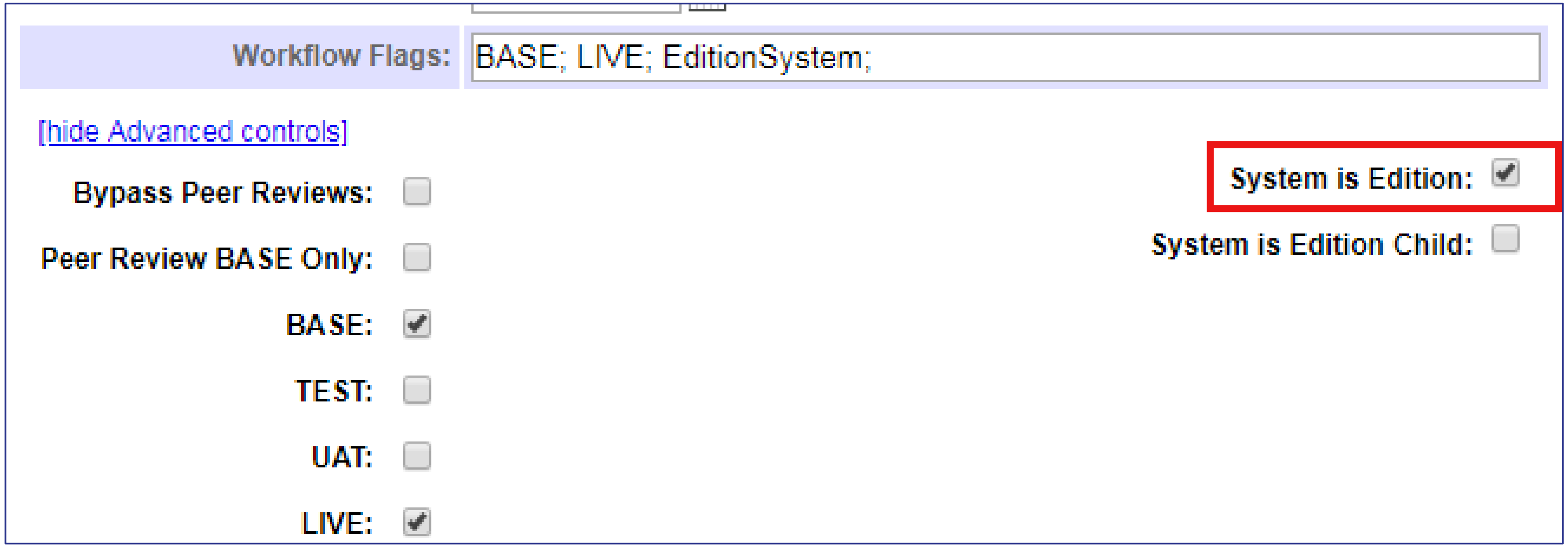
Part 2: Configuring Systems for Release Management

Configuration Overview

- Prior to using release management, System definitions must be configured.
  - 1 System configured as ‘Edition’ System on System menu in CCR.
  - 1+ Systems configured as ‘Edition Child’ System on System menu in CCR.
  - Systems can belong to different organizations.
  - Releases must be applied in sequential order.
- User needs “Release Manager” role in CCR to make changes to release Configuration or releases.

Edition System Configuration

1. Go to System Details for the Edition.
2. Click on “show Advanced Controls.”
3. Select “System is Edition.”
4. Click “Save.”



Workflow Flags: BASE; LIVE; EditionSystem;

[hide Advanced controls]

Bypass Peer Reviews: □

Peer Review BASE Only: □

BASE: ☑

TEST: □

UAT: □

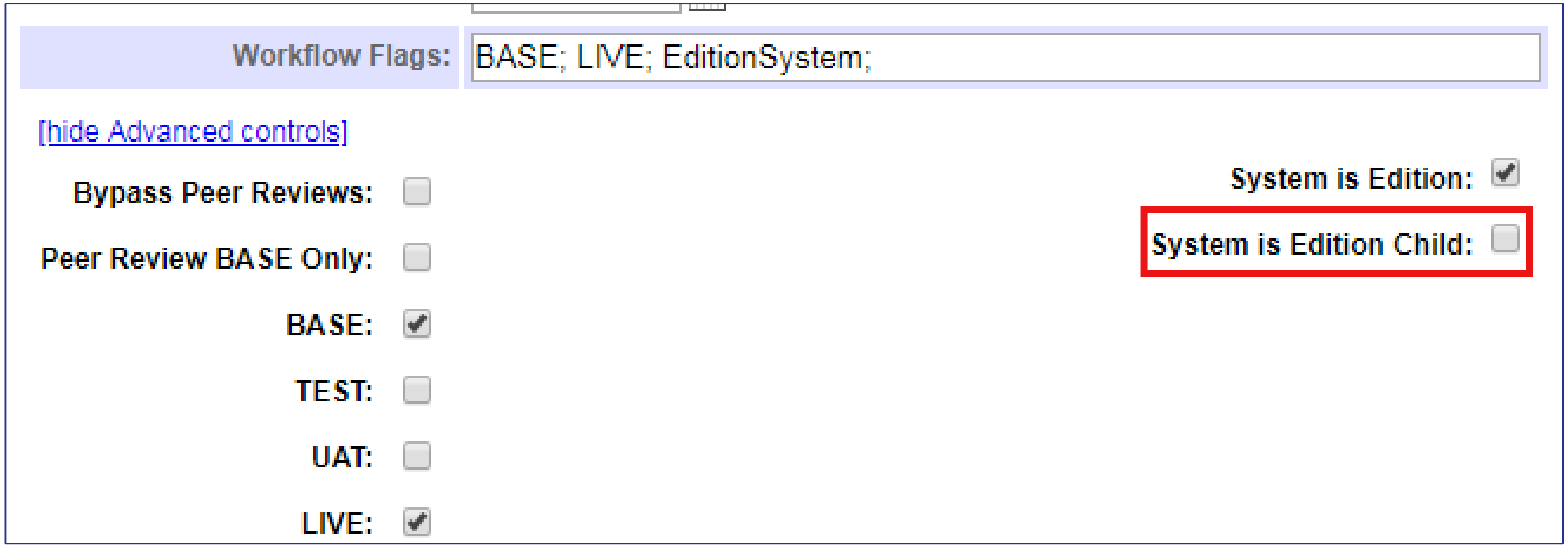
LIVE: ☑

System is Edition: ☑

System is Edition Child: □

Edition Child System Configuration

1. Go to System definition for System to be configured as child.
2. Click Advanced Controls tab.
3. Select System is Edition Child.
4. Save.



Workflow Flags: BASE; LIVE; EditionSystem;

[hide Advanced controls]

Bypass Peer Reviews: □

Peer Review BASE Only: □

BASE: ☑

TEST: □

UAT: □

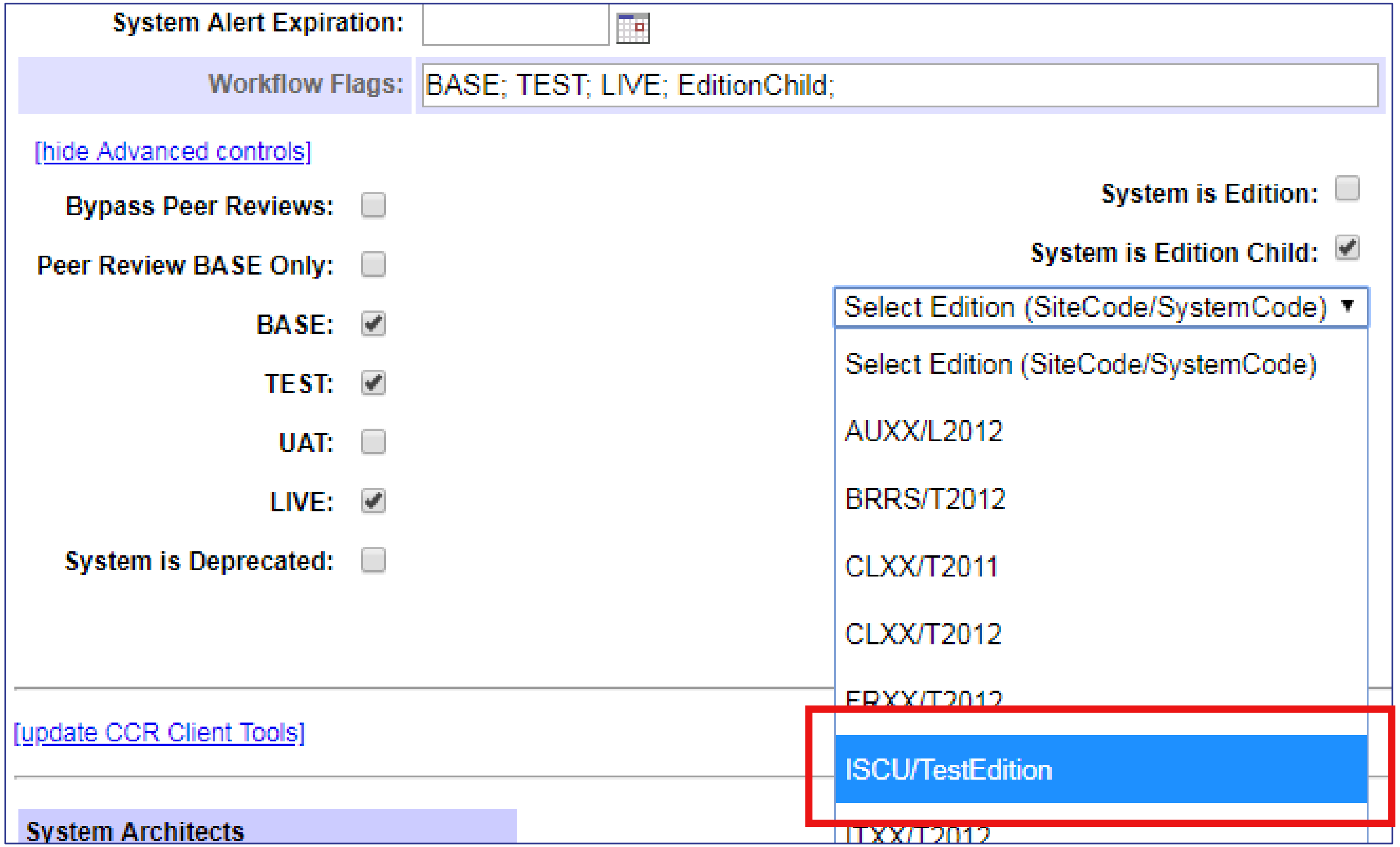
LIVE: ☑

System is Edition: ☑

System is Edition Child: □

Edition Child System Configuration (cont.)

5. After save, click Advanced Controls again.
6. Select Edition to be used as parent of this Edition Child.
7. Click Ok in confirmation dialog.



System Alert Expiration:	
Workflow Flags:	BASE; TEST; LIVE; EditionChild;

[hide Advanced controls]

Bypass Peer Reviews:	□
Peer Review BASE Only:	□
BASE:	☑
TEST:	☑
UAT:	□
LIVE:	☑
System is Deprecated:	□
System is Edition:	□
System is Edition Child:	☑

Select Edition (SiteCode/SystemCode) ▼

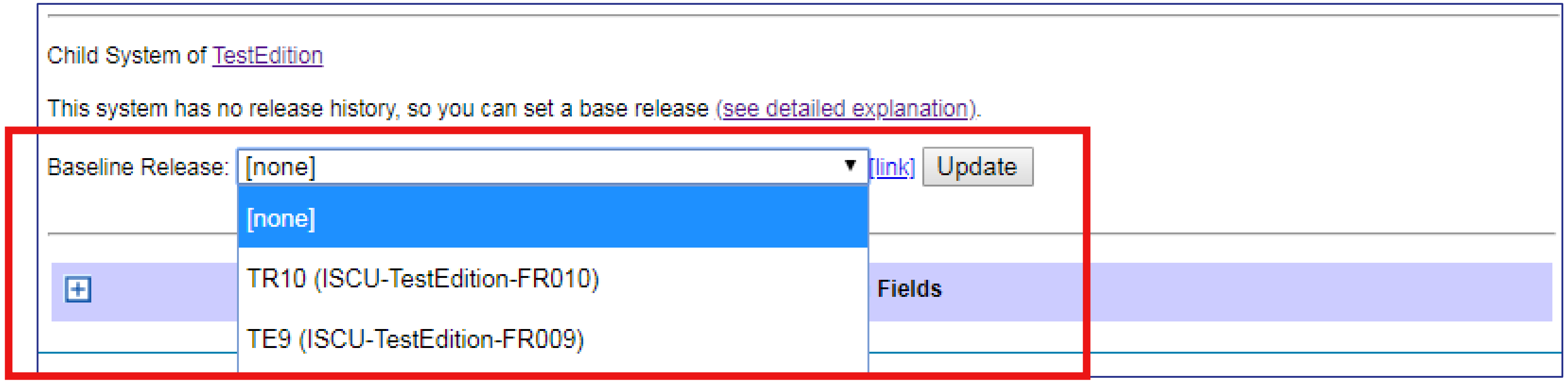
- Select Edition (SiteCode/SystemCode)
- AUXX/L2012
- BRRS/T2012
- CLXX/T2011
- CLXX/T2012
- ERXX/T2012
- ISCU/TestEdition
- ITXX/T2012

[update CCR Client Tools]

System Architects

Selecting Baseline Release for Edition Child

- After configuring a child, option to select a Baseline release appears.
  - Only applies if Edition has prior releases.
  - **Every** Release must be applied sequentially to a Child System.
    - Releases are incremental.
  - Releases prior to baseline Release need not be applied to child.



Child System of TestEdition

This system has no release history, so you can set a base release (see detailed explanation).

Baseline Release: [none]

Quiz: Edition and Children

Question:

Every Edition Child must be a System defined in the same organization as the Edition to which it is connected. True or False?

Answer:

False.

When release management is used, most often the Edition is a System internal to InterSystems and the Edition Child Systems are all defined for different organizations receiving the same code and configuration from the Edition.

Summary

- What are the key points for this module?

ICC 500:
Introduction to CCR Tier 1

InterSystems
Creative data technology

Objectives

- Explain at a high level what CCR Tier 1 means and when it is useful.
- Identify whether a deeper dive into CCR Tier 1 training material would be beneficial.

Part 1:

CCR Tier 1: Working with Custom Source Items

Review: Example types of changes for each CCR Tier

- CCR Tier 0:
  - File system path changes.
  - Memory settings.
  - User creation.
  - Patching.
  - Upgrades.
  - Any manual change to the environment or System.

- CCR Tier 1:
  - Classes.
  - Routines.
  - Reports.
  - CSP Pages.
  - JavaScript.
  - Integration logic.
  - DTL.
  - Productions.

Review: Example types of changes for each CCR Tier (cont.)

- CCR Tier 2:
  - Layouts.
  - Security groups.
  - Code table changes.
  - Workflow.

CCR Tier 1: The Fundamentals

- A “CCR Tier 1” Change Control Record tracks all of the change documentation recorded in a CCR Tier 0 record, and adds to it:
    - Recording of changes made to custom source items (typically edited via Studio or Atelier).
    - Source items automatically versioned within source control.
    - Delivery of items to or from the environments via the CCR Transport mechanism.

CCR Tier 1: The Fundamentals (cont.)

- A Perforce license is required in order to trigger the source control related actions which are part of a CCR Tier 1 workflow (specifically check-ins and integrations).
- Tier 1 CCRs are auto-populated with standard Implementation Plan and Backout Plan as the assumption is that the changes will be implemented via ItemSet deployment.
  - It is critical to change or augment those standard plans if manual steps are required.

CCR Tier 1: The Fundamentals (cont.)

- Any time changes are being made to source items, the Perforce branch *must* be kept in sync with the environments for TEST and LIVE; proper use of CCR process will do this.

CCR Tier 1: The Audience

- The courses focused on CCR Tier 1 (ICC5nn) are mostly focused on hands-on technical topics as Tier 1 involves changes to custom ‘source code’ which is itself a technical task.
  - Specifically, changes are usually made via an InterSystems IDE (e.g., Atelier or Studio) or the System Management Portal (e.g., to DeepSee items or DTL).
- Those taking these courses should already have a working knowledge of CCR Tier 0 usage and how CCR Transport works.

CCR Tier 1: The Audience (cont.)

- Additional CCR Tier 1 material relates to:
  - Baselining and building source code.
  - Topics dealing with HealthShare and Ensemble usage.
- While they should complete this Introduction course, non-technical users (e.g., TrakCare Application Specialists) will get little to no value from the other courses covering CCR Tier 1.

Quiz: CCR Tier 1 Applicability

Question:

Which groups of users are the CCR Tier 1 training materials applicable (select all that apply):

A. Users who use InterSystems Studio for writing custom integration code.

B. Users who spend all their time inside the InterSystems TrakCare application.

C. Users who work on custom DTL within InterSystems HealthShare.

Quiz: CCR Tier 1 Applicability (cont.)

Answer:

A. Users who use InterSystems Studio for writing custom integration code.

C. Users who work on custom DTL within InterSystems HealthShare.

TrakCare Application users should focus on CCR Tier 2 training, which covers how to manage changes to TrakCare configuration via CCR.

Quiz: CCR Tier 1 Functionality

Question:

Select all that distinguish a Tier 1 CCR from a Tier 0 CCR:

A. CCR Transport is used to move items to and from environments.

B. Items are versioned within the InterSystems Perforce server.

C. Changes made via Studio or Atelier are automatically tracked.

D. Configuration updates made through the TrakCare application are automatically tracked.

Quiz: CCR Tier 1 Functionality (cont.)

Answer:

A. CCR Transport is used to move items to and from environments.

B. Items are versioned within the InterSystems Perforce server.

C. Changes made via Studio or Atelier are automatically tracked.

TrakCare changes are tracked via Tier 2 CCRs.

Quiz: CCR Tier 1 Usage

Question:

Which of the following might be controlled via a Tier 1 CCR? Select all that apply.

A. Integration Logic.

B. Custom Classes.

C. TrakCare Security Groups.

D. Product Upgrades.

E. Changes to CSP Pages.

Quiz: CCR Tier 1 Usage (cont.)

Answer:

A. Integration Logic.

B. Custom Classes.

E. Changes to CSP Pages.

Changes to TrakCare Security Groups should be tracked in a Tier 2 CCR.

Product upgrades should be tracked in a Tier 0 CCR.

Areas for Further Study for CCR Tier 1

- ICC510 – Usage Basics.
- ICC520 – Baselining Source Code.
- ICC530 – Interoperability Components.
- ICC540 – HealthShare.

Summary

- What are the key points for this module?

ICC 510: CCR Tier 1 – Usage Basics

InterSystems
Creative data technology

Overview

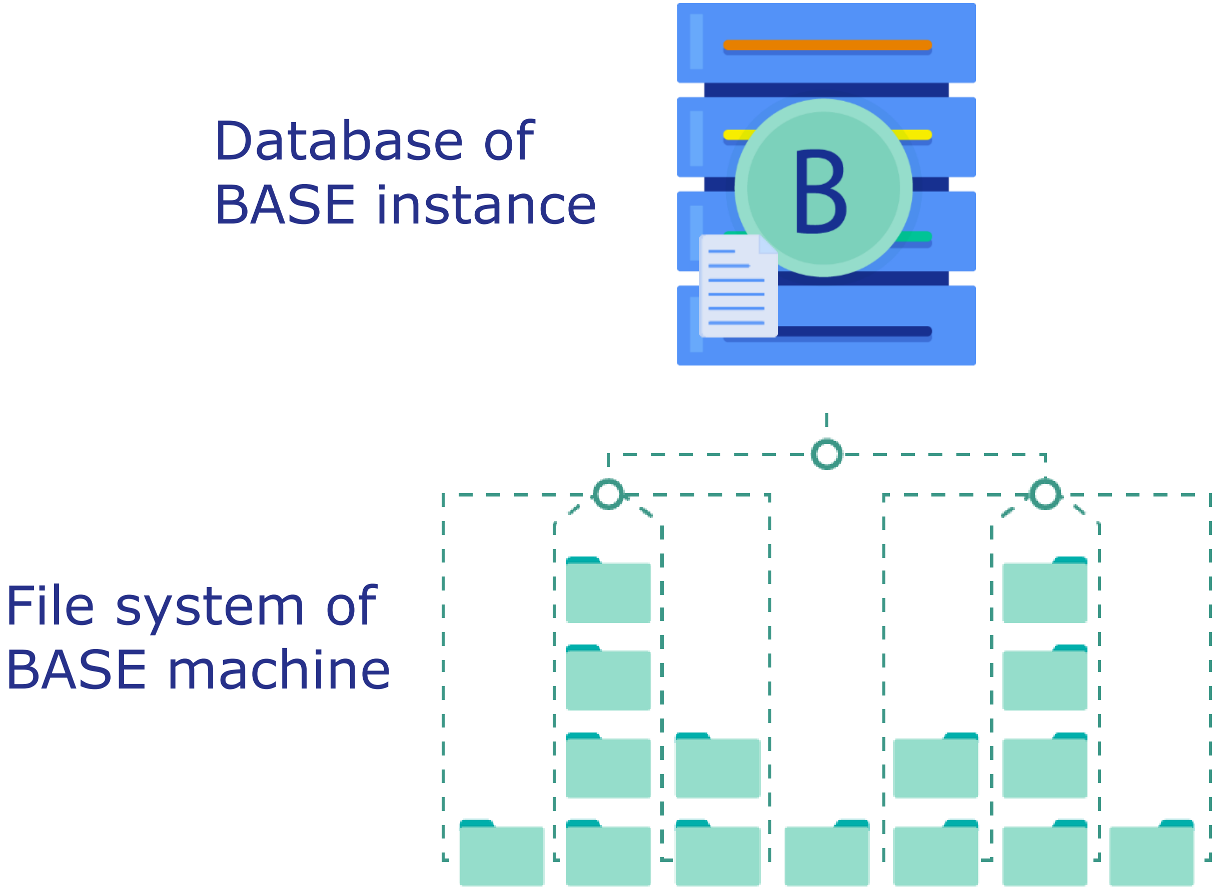
- Terminolgy.
  - Items.
  - ItemSets.
  - Source Workspace.
- Making Tier 1 changes in IDE.
- Submitting changes to CCR.
- Deploying changes to environments.
- Utility functions.
- Automating deployment tasks.

Terminology: Items and ItemSets

- Item: Individual representation of part of an application.
  - Examples include individual:
    - Class definitions.
    - Routines.
    - Rows of a table.
- ItemSet: File that combines one or more Items along with metadata for transporting Items.
  - Think of it like an envelope for sending mail.
  - Only used for transport purposes, not persisting change.

Source Workspace

- Directory in filesystem of each environment.
- Items exported as xml files to source workspace for transport to CCR server.

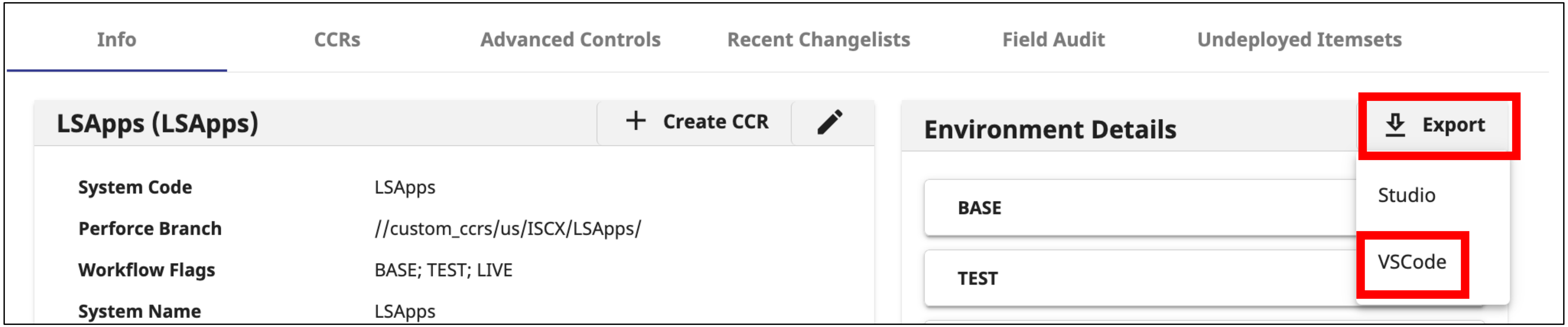


Connected vs Disconnected Mode

- 2 modes for BASE communication with CCR and Perforce.
  - Different options in source control menu.
- Disconnected.
  - BASE located outside InterSystems network.
    - BASE communicates directly with CCR.
  - All customer Environments are disconnected.
- Connected.
  - BASE located inside InterSystems network.
    - BASE communicates directly with Perforce.
  - Most internal applications configured for connected BASE.

Download VS Code Workspace

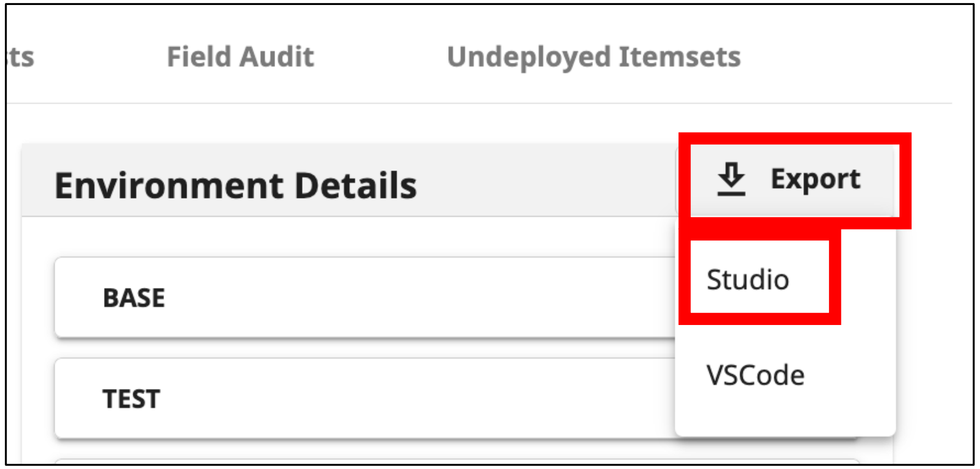
- Can download VS Code Workspace from Systems page.
  - Contains connection details for environments defined in System.
- Menu > Systems > Choose System > Export > VSCode.
  - Double click file to open in VS Code.

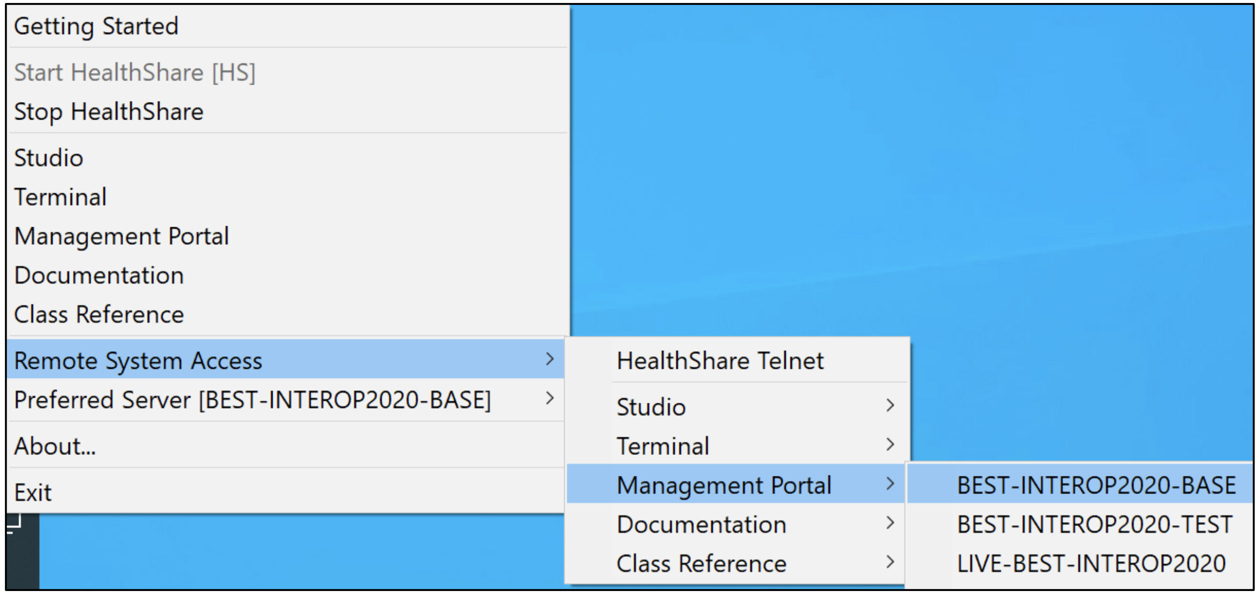


System Code	LSApps
Perforce Branch	//custom_ccrs/us/ISCX/LSApps/
Workflow Flags	BASE; TEST; LIVE
System Name	LSApps

Windows: Configure Launcher

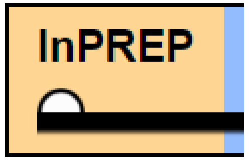
- Launcher provides remote system access to environments.
- Menu > Systems > Choose System > Export > Studio.
  - Run downloaded registry file.
    - Administrator credentials required.





Part 1: CCR PREP Phase

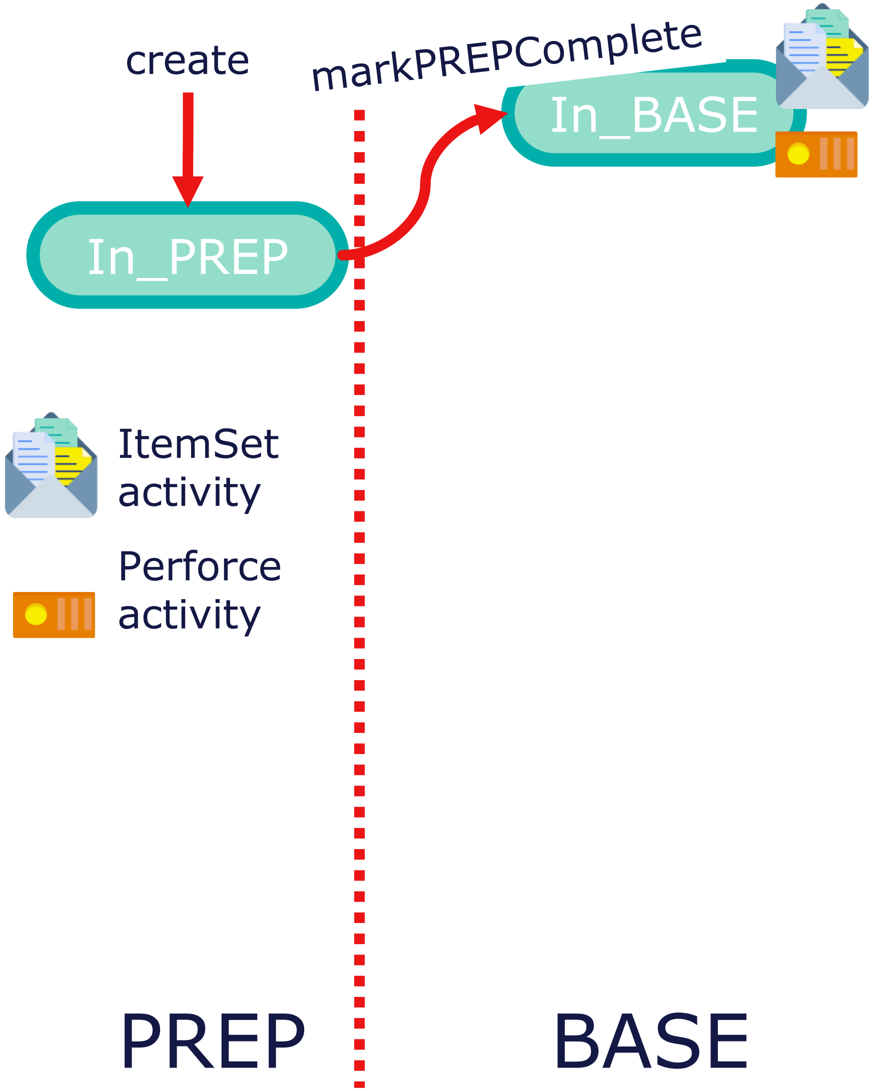
Tier 1 Considerations: In_PREP



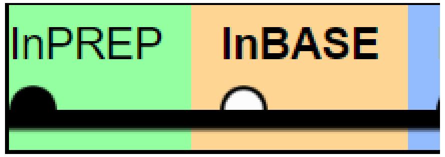
- Confirm Tier field set to “1 – Source Control.”
- Modified Items: Usually matches list of submitted changes.
- Perform markPREPComplete transition before checking out files.
  - Only make changes while in In_BASE state.

Part 2: CCR BASE Phase: Making Changes

Progressing a Tier 1 CCR…



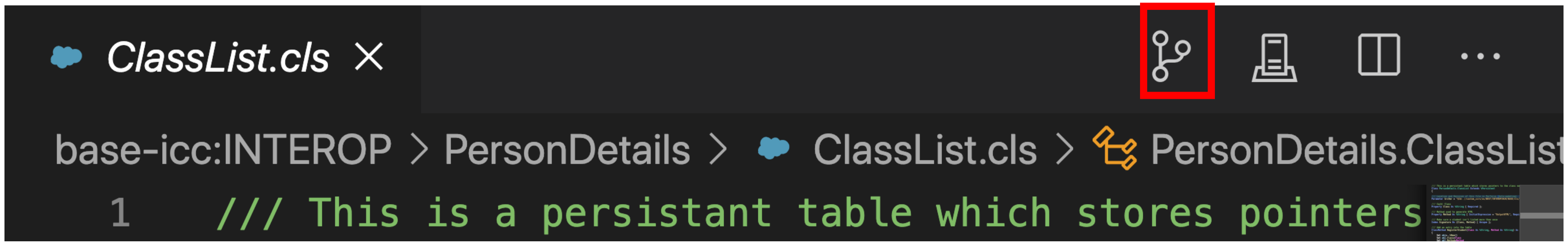
Tier 1 Considerations: In_BASE



- Work performed via IDE and/or the Management Portal.
- In order to progress, record must have either:
  - Changelist.
  - Uncommitted ItemSet.
- Implementation Plan and Backout Plan have default values
  - Only change if manual steps needed.

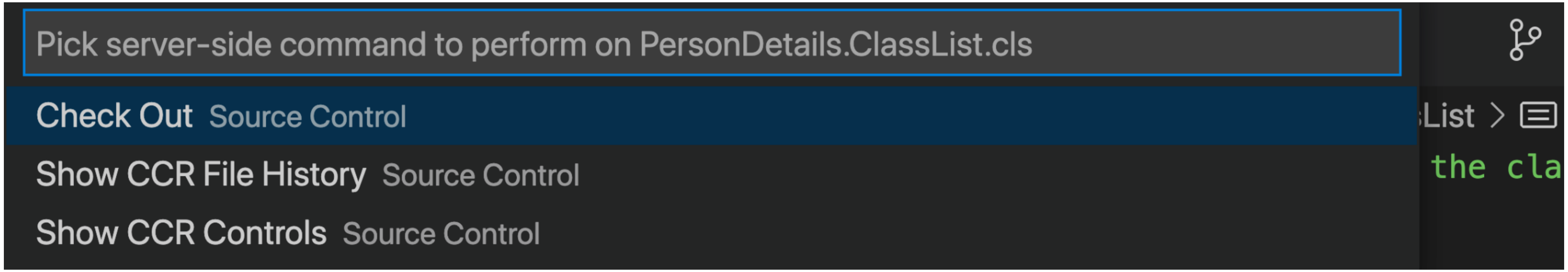
Using VS Code - ObjectScript with CCR

- Use server-side mode to connect to BASE.
  - Can download workspace definition from System Details page in CCR.
- Source Control button appears when viewing any item.



ClassList.cls
base-icc:INTEROP > PersonDetails > ClassList.cls > PersonDetails.ClassList

```
/// This is a persistant table which stores pointers
```

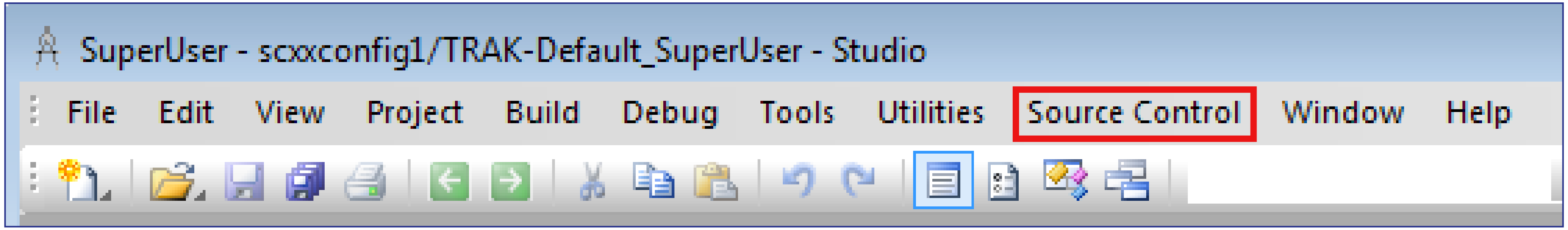


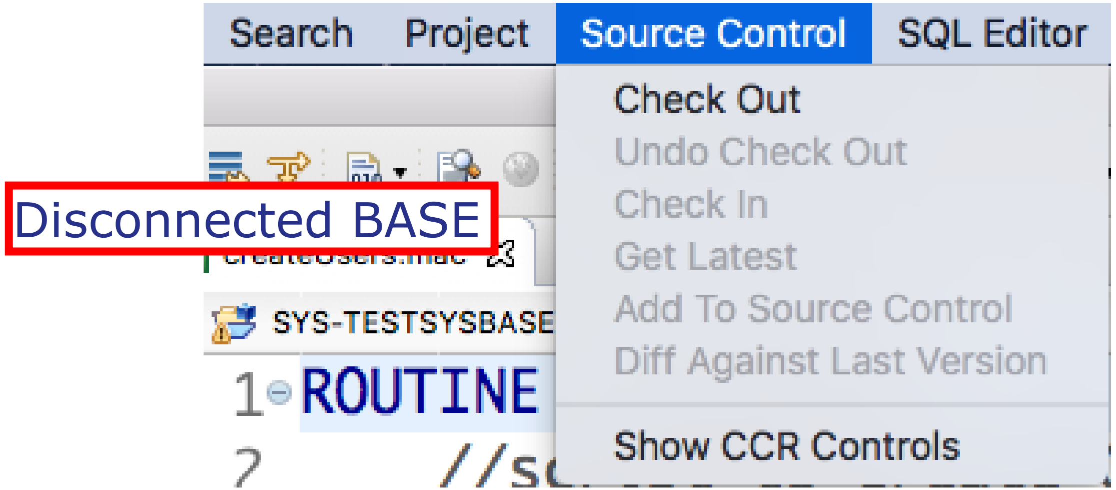
Pick server-side command to perform on PersonDetails.ClassList.cls

Check Out Source Control
Show CCR File History Source Control
Show CCR Controls Source Control

Using Studio with CCR

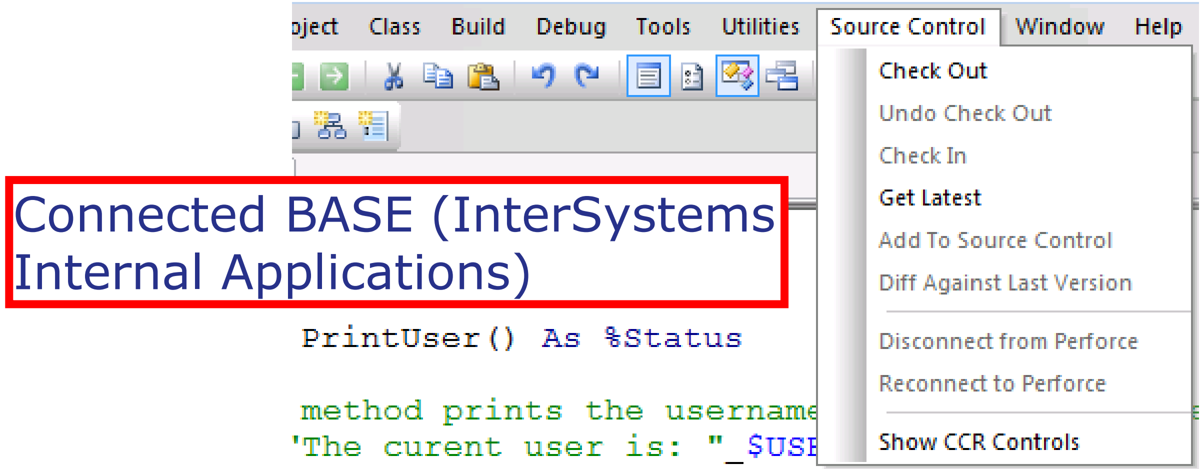
- Source Control menu available when connected to CCR configured namespace.





```
ROUTINE
//s
```

Disconnected BASE



```
PrintUser() As %Status

method prints the username
'The current user is: "_$USE
```

Connected BASE (InterSystems Internal Applications)

Working with Controlled Items

- Source Control menu used for interacting with source control hooks.
- Non-BASE environments locked.
  - Existing items read-only in IDE and cannot be checked out.
  - Cannot create new items in IDE.

Concurrency Control within Instance

- Must guarantee user has exclusive access to edit item.
  - Controlled by checking out an item before editing it.
    - Checkout status removed when submit changes.
- If multiple users could edit an item simultaneously:
  - All changes submitted to one CCR.
    - Changes from other users could be incomplete at time of upload.
  - Diff changes would show other user’s changes.
    - User who made change not highlighted.
    - User submitting change would need to notice diffs they did not cause.

Environment Source Control Summary

- Exports code as xml files to source workspace.
- System controls whether exported files are read-only or read/write.
  - Do not manually change R/W status of any files.
    - Will break source control integration.
- On checkout, duplicates file to create .bak file.
  - Used to:
    - Revert changes.
    - Diff changes.

Edit Existing Item

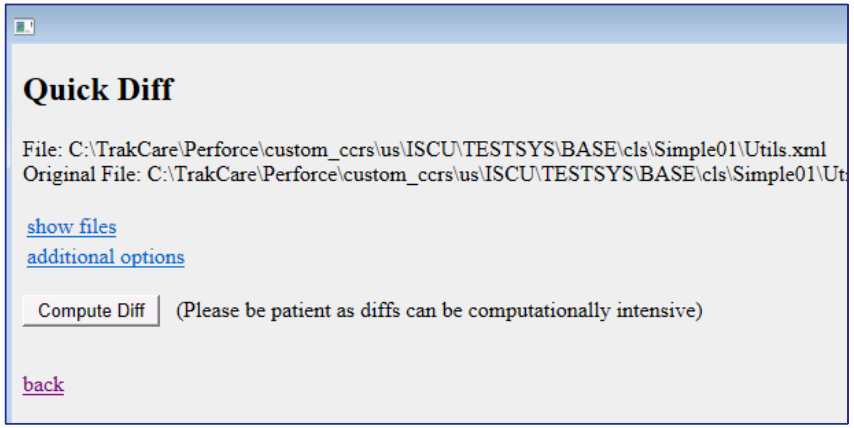
- Cannot checkout item checked-out to another username.
- 2 ways to checkout:
  - Start making change.
    - Popup message prompts to confirm checkout.
  - Open item, then use checkout menu option.
- Check out action:
  - Guarantees only your user can make changes.
  - Backup file created to enable revert.
  - For connected BASE, checks out item in Perforce as well.
    - InterSystems internal applications.

Identify Changes Made to Checked Out File

- Compares current exported file with .bak version, highlighting changed lines.
  - Important because indicates exactly what will be uploaded.
- If changes missing, most likely did not compile code.
  - Changes exported with successful compilation.

Diff Changes

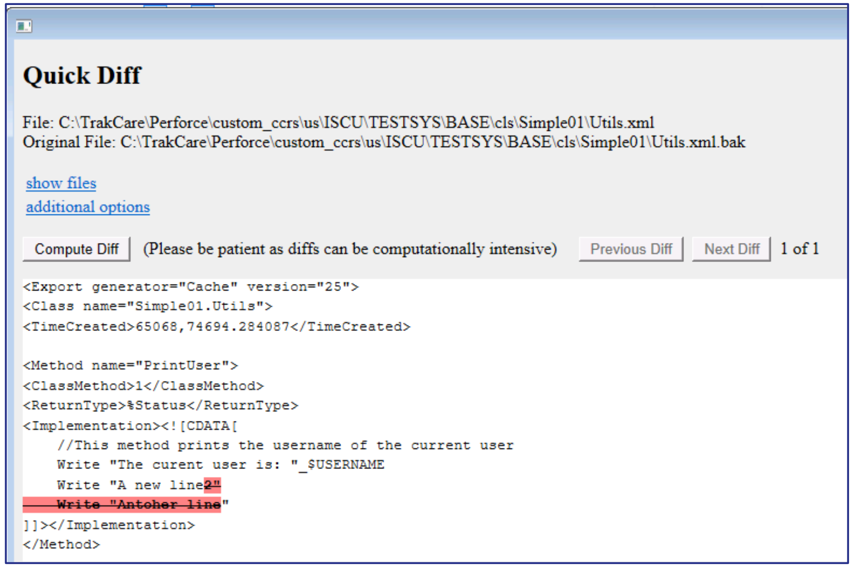
- 2 equivalent ways to reach Quick Diff page.
  - From IDE, Source Control Menu > Show CCR Controls > ItemSet Bundle > Click diff link for appropriate item.
  - With item open in IDE, go to Source Control Menu > Diff Against Last Version.



Quick Diff

File: C:\TrakCare\Perforce\custom_ccrs\us\ISCU\TESTSYS\BASE\cls\Simple01\Utils.xml
Original File: C:\TrakCare\Perforce\custom_ccrs\us\ISCU\TESTSYS\BASE\cls\Simple01\Ut

(Please be patient as diffs can be computationally intensive)



Quick Diff

File: C:\TrakCare\Perforce\custom_ccrs\us\ISCU\TESTSYS\BASE\cls\Simple01\Utils.xml
Original File: C:\TrakCare\Perforce\custom_ccrs\us\ISCU\TESTSYS\BASE\cls\Simple01\Utils.xml.bak

(Please be patient as diffs can be computationally intensive)

```
<Export generator="Cache" version="25">
<Class name="Simple01.Utils">
<TimeCreated>65068,74694.284087</TimeCreated>

<Method name="PrintUser">
<ClassMethod>1</ClassMethod>
<ReturnType>%Status</ReturnType>
<Implementation><![CDATA[
    //This method prints the username of the current user
    Write "The curent user is: "_$USERNAME
    Write "A new line~~2"~~
~~    Write "Antoher line~~"
]]></Implementation>
</Method>
```

Create New Item

- On first save, IDE prompts whether to add to source control.
  - Choose yes.
- Can also use Add To Source Control menu option.
- Either way, on add:
  - Item exported as writable XML.
    - Routines are exported as .rtn RO files for legacy reasons.
  - Item checked out to your username.
  - Item added to Uncommitted Queue with add action.

Delete Existing Item

- On delete from IDE, popup message prompts to remove from source control.
  - Choose yes.
- Deletes item from database and source workspace.
- Adds delete action to uncommitted queue.
  - Disconnected BASE: Bundle and upload to CCR.
  - Connected BASE: Submit to Perforce.

Revert Changes Not Uploaded to CCR

- To delete new item (not uploaded to CCR):
  - Open item then use Undo Add menu option.
  - Delete item in IDE.
    - Choose no in popup message asking to remove from source control.
- To undo all changes made to checked out file (and not yet uploaded):
  - Open item then use Undo Check Out menu option.

Quiz: Quick Diff

Question:

Which of the following is true about the Quick Diff functionality in the client tools in disconnected BASE environments?

A. It compares the Perforce head revision to the version in the IDE.

B. It compares the Perforce head revision to the exported version in the source workspace.

C. It compares the .bak version in the source workspace to the exported .xml version in the source workspace.

D. It compares the .bak version in the source workspace to the version in the IDE.

Quiz: Quick Diff (cont.)

Answer:

C. It compares the .bak version in the source workspace to the exported .xml version in the source workspace.

Quiz: Concurrency Controls

Question:

Who can edit, or bundle and upload a class definition that has been checked out to John Smith?

A. Only John Smith.

B. Whoever owns the CCR.

C. Anyone from John Smith’s organization.

D. John Smith and anyone with the %ALL role in the environment.

Quiz: Concurrency Controls (cont.)

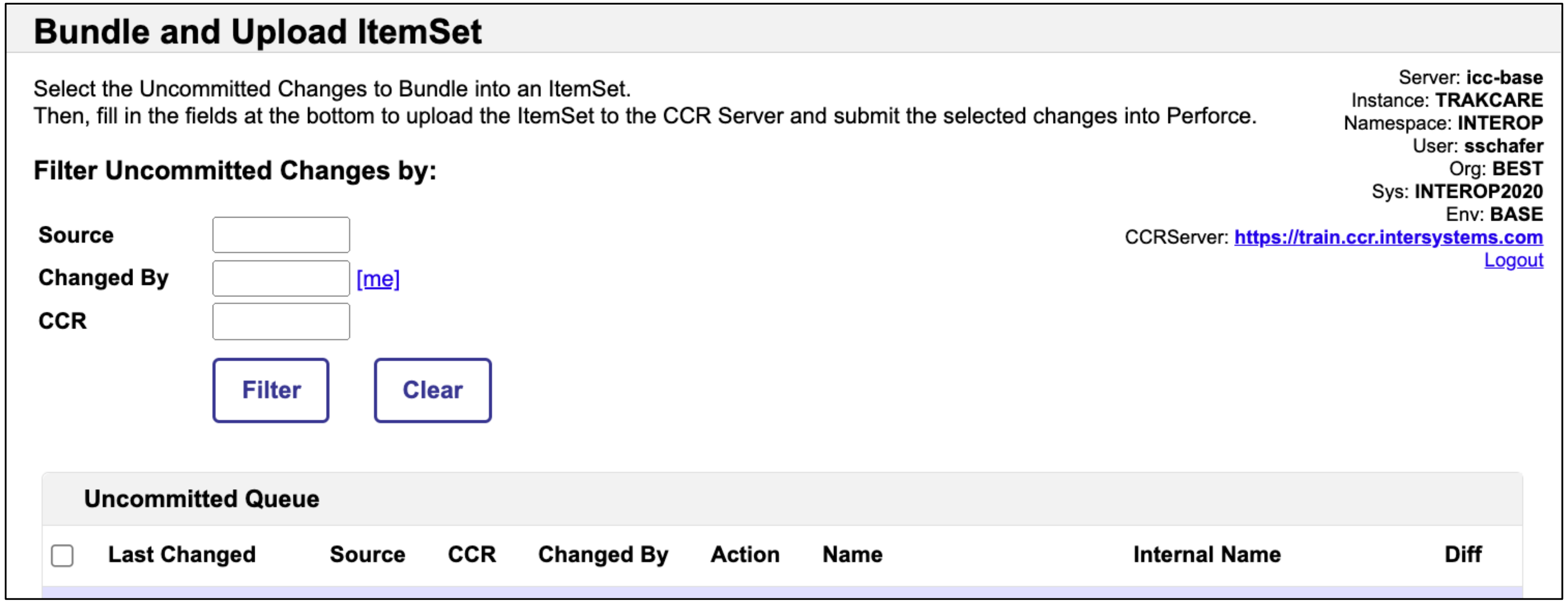
Answer:

A. Only John Smith.

Part 3: CCR BASE Phase: Submitting Changes

Bundle Screen

- From IDE:
  - Source Control > Commit Changes via ItemSet (Client tools 2024-Oct-16).
  - Source Control > Show CCR Controls > Bundle ItemSet.
- Filters up top have no effect on what is bundled.



Bundle and Upload ItemSet

Select the Uncommitted Changes to Bundle into an ItemSet.
Then, fill in the fields at the bottom to upload the ItemSet to the CCR Server and submit the selected changes into Perforce.

Server: **icc-base**
Instance: **TRAKCARE**
Namespace: **INTEROP**
User: **sschafer**
Org: **BEST**
Sys: **INTEROP2020**
Env: **BASE**
CCRServer: **https://train.ccr.intersystems.com**
Logout

Filter Uncommitted Changes by:

Source	
Changed By	[me]
CCR	

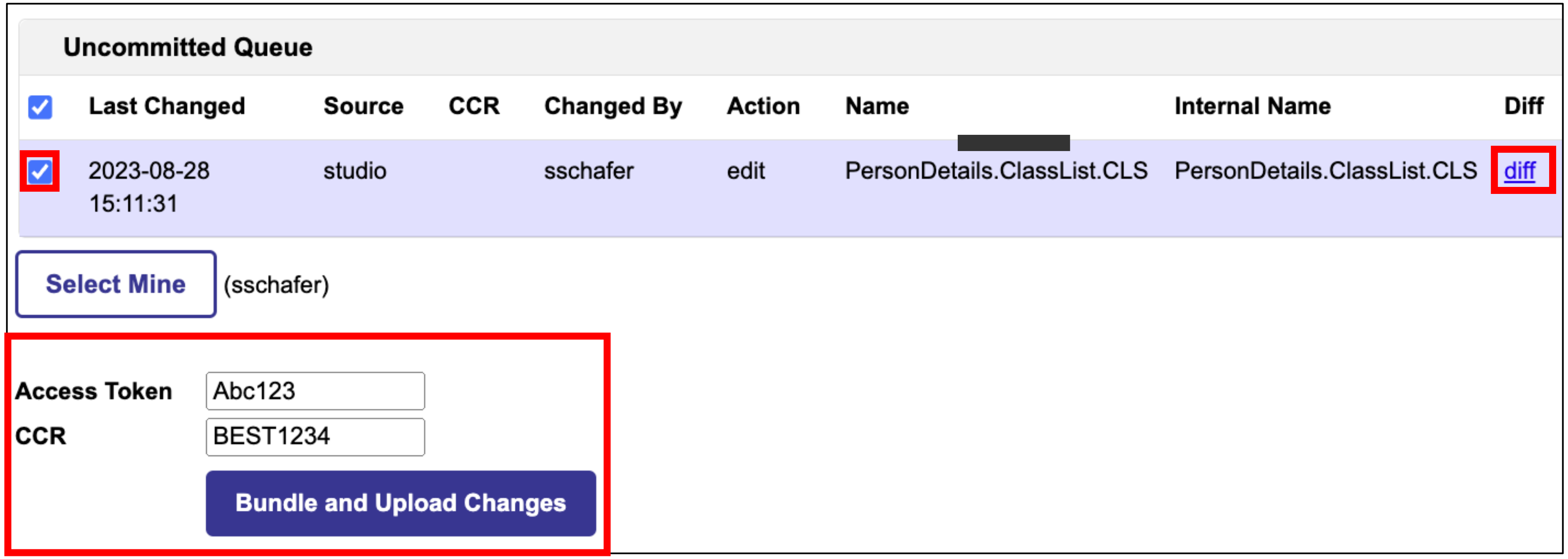
Filter Clear

Uncommitted Queue

□	Last Changed	Source	CCR	Changed By	Action	Name	Internal Name	Diff

Bundle Screen (cont.)

- Use diff link to diff changes.
- Select which items to bundle.
- Copy Access Token and CCR ID from CCR.
- Click Bundle and Upload Changes.



Uncommitted Queue

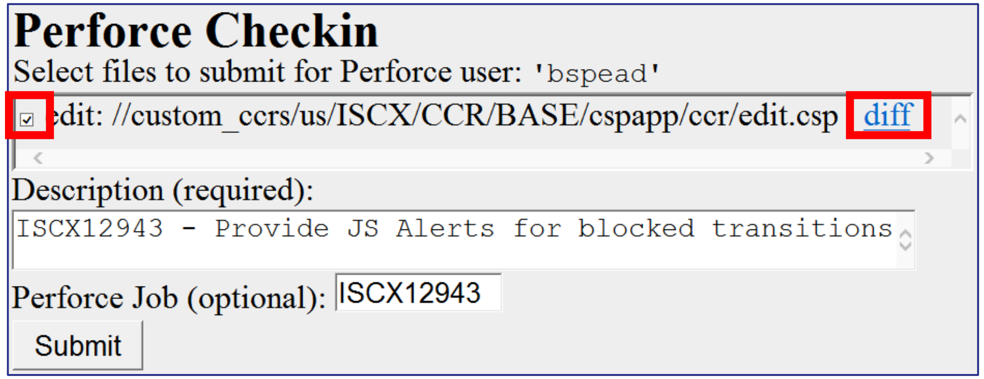
☑	Last Changed	Source	CCR	Changed By	Action	Name	Internal Name	Diff
☑	2023-08-28 15:11:31	studio		sschafer	edit	PersonDetails.ClassList.CLS	PersonDetails.ClassList.CLS	diff

(sschafer)

Access Token	Abc123
CCR	BEST1234

Connected: CCR Client → Perforce

- To submit changes from a Connected BASE:
  1. From IDE, Source Control > Check in.
  2. If prompted, enter Perforce password.
  3. Click diff link for each item to be submitted.
     - Always diff changes; know what changes you submit!
  4. Select item(s) to check in.



Perforce Checkin

Select files to submit for Perforce user: 'bspead'

☑ edit: //custom_ccrs/us/ISCX/CCR/BASE/cspapp/ccr/edit.csp diff

Description (required):

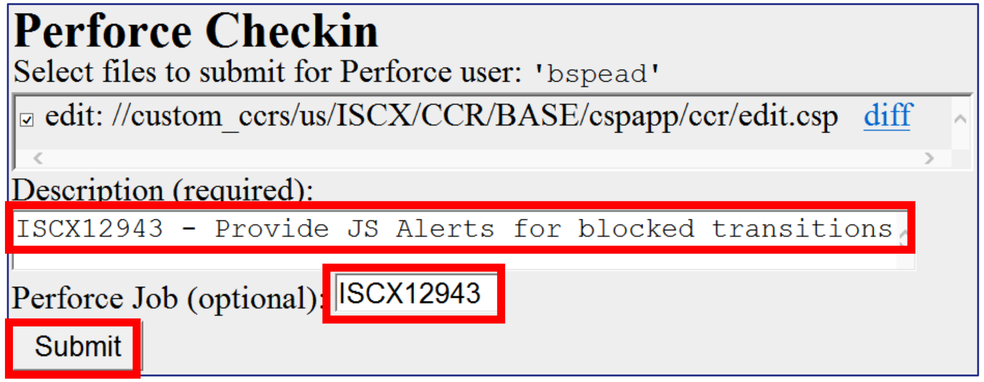
ISCX12943 - Provide JS Alerts for blocked transitions

Perforce Job (optional): ISCX12943

Submit

Connected: CCR Client → Perforce

- To submit changes from a Connected BASE:
  5. Copy and paste <CCR ID> - <Title> from CCR title bar to Description. Include additional details if desired.
  6. Copy CCR ID and paste into Job field (required).
  7. Click "Submit."



Perforce Checkin

Select files to submit for Perforce user: 'bspead'

☑ edit: //custom_ccrs/us/ISCX/CCR/BASE/cspapp/ccr/edit.csp diff

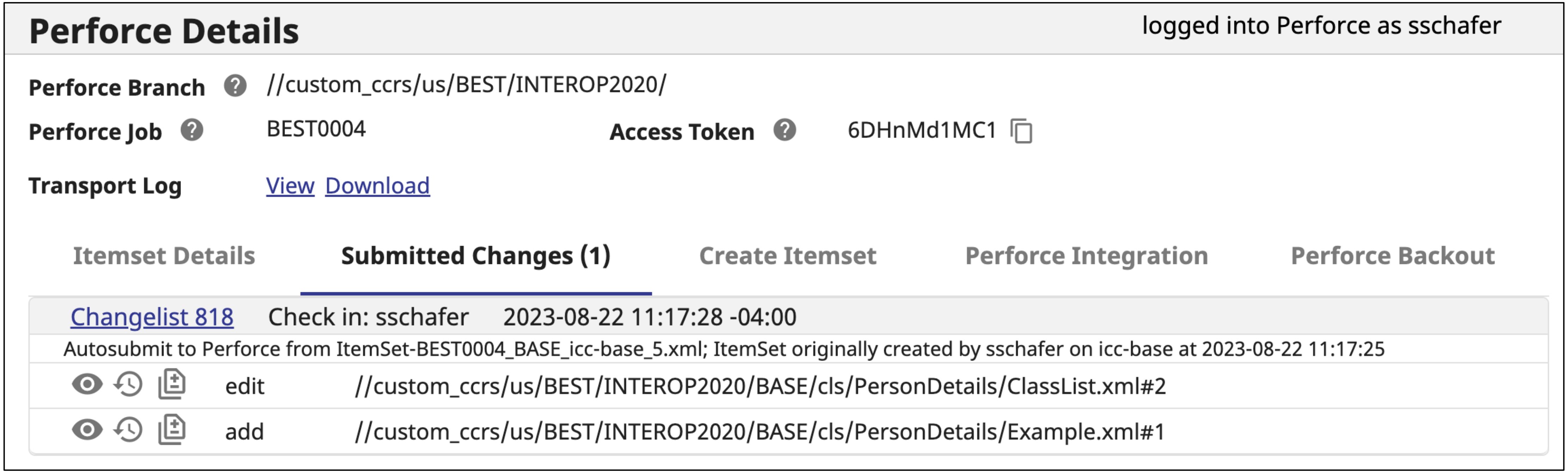
Description (required):

ISCX12943 - Provide JS Alerts for blocked transitions

Perforce Job (optional): ISCX12943

Verify Submitted Changes

- After submit changes, always verify correct items listed in CCR.
    - Under Perforce Details of CCR, click Submitted Changes tab.



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

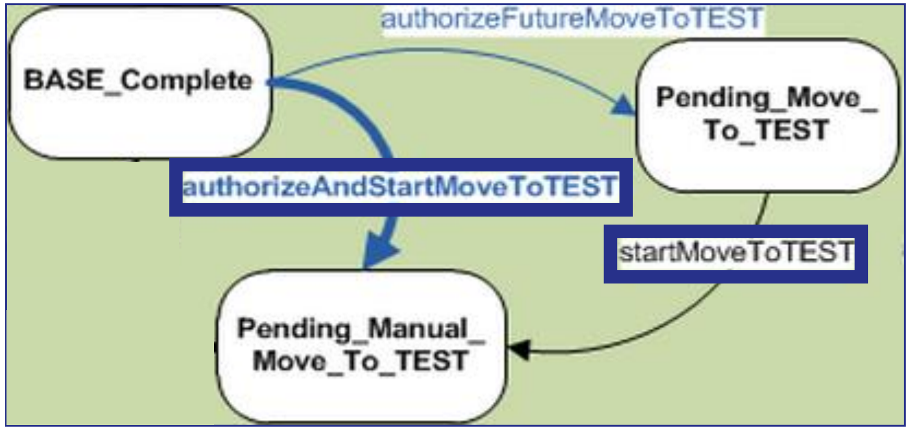
Autosubmit to Perforce from ItemSet-BEST0004_BASE_icc-base_5.xml; ItemSet originally created by sschafer on icc-base at 2023-08-22 11:17:25

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

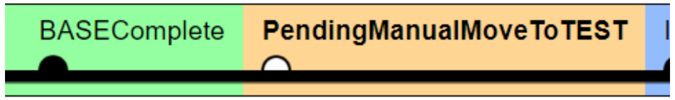
Part 4: Moving to TEST Environment

(authorizeAnd)StartMoveToTEST

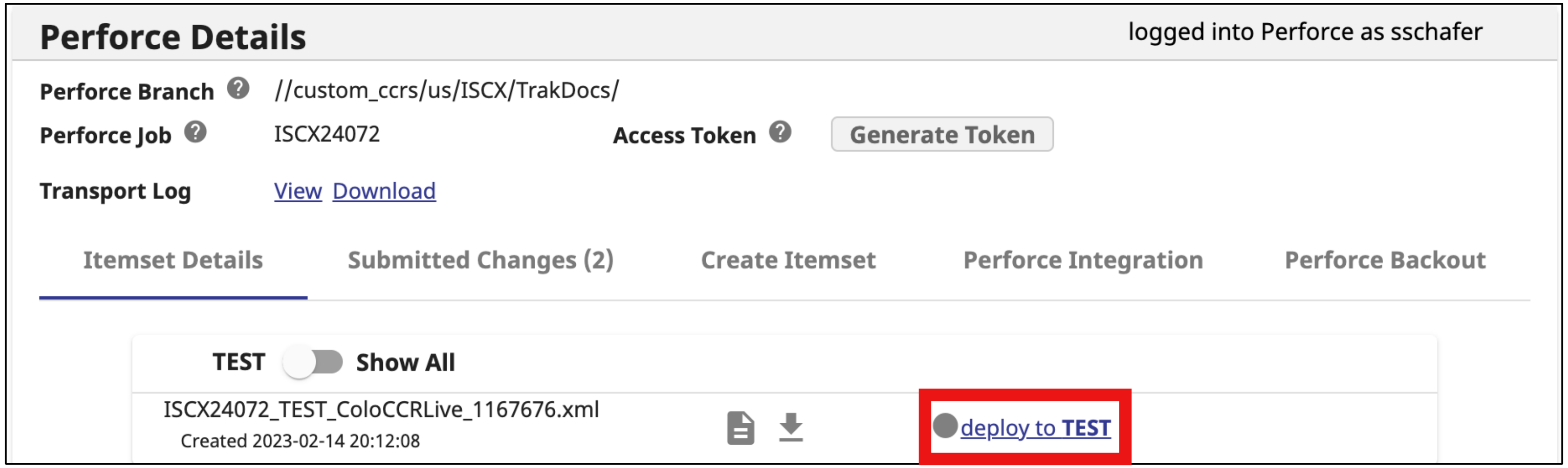
- Only perform authorizeAndStartMoveToTEST or startMoveToTEST when ready to move change.
- Immediately deploy ItemSet.
  - Delaying or abandoning ItemSet deployment will cause problems later.


BASE_Complete
authorizeFutureMoveToTEST
Pending_Move_To_TEST
authorizeAndStartMoveToTEST
startMoveToTEST
Pending_Manual_Move_To_TEST

Pending_Manual_Move_To_TEST



- ItemSet created by transition into this state.
- Deploy using deploy to TEST link under Perforce Details of CCR.
    - Opens web page on TEST Environment.
    - Log in using credentials for TEST Environment.
    - After log in, ItemSet download and item import process begins automatically.



Perforce Details logged into Perforce as sschafer

Perforce Branch //custom_ccrs/us/ISCX/TrakDocs/

Perforce Job ISCX24072

Access Token

Transport Log View Download

ISCX24072_TEST_ColoCCRLive_1167676.xml
Created 2023-02-14 20:12:08

Automatic markMoveToTESTComplete

- markMoveToTESTComplete automatically occurs if:
  - No changes made to default Implementation Plan.
  - ItemSet successfully deployed to all TEST environments that have Require ItemSet.
- Row in Transition History has note “{transition occurred automatically due to successful ItemSet load in TEST}”.

Quiz: Automatic Transitions

Question:

Tier 1 CCRs with a default (unedited) *Implementation Plan* will have the markManualMoveToTEST transition occur automatically once the ItemSet has been deployed to all environments in TEST which require ItemSets.

True or False?

Quiz: Automatic Transitions (cont.)

Answer:

True.

Editing the *Implementation Plan* to add manual steps will mean that the CCR user must perform the markManualMoveToTEST/LIVE manually after completing the manual pieces of the *Implementation Plan.*

Part 5: Test, Review, Deploy, Repeat

Progress CCR to LIVE

- Perforce integration and ItemSet creation workflow same as moving to TEST.
  - Integration from TEST to LIVE occurs on *StartMoveToLIVE transitions.
  - ItemSet automatically created and available for deployment under Perforce Details upon successful integration.

Exercise ICC510-1 through ICC510-6

Part 6: CCR BASE Phase: Tools for Developers

Advanced Tip: CCR Built-in Functions

- Allow your code to vary its behavior based on the current environment.
  - $$IsLIVE^%buildccr.
    - Prevent emails from being sent to actual users from BASE or TEST.
  - $$Env^%buildccr.
    - Display the environment in email footers, UI headers, etc (BASE, TEST, UAT or LIVE).
  - $$EnvName^%buildccr.
    - Test the environment name to perform customized logic (e.g. for TRAIN, which is a peer to TEST).

Advanced Tip: CCR Built-in Functions (cont.)

- Allow your code to vary its behavior based on the current environment.
  - $$Sys^%buildccr.
    - Put SystemCode in error logs.
  - $$Org^%buildccr.
    - Include organization details in email alerts to centralized monitoring system.

Automating Deployment Tasks

- ImplementCCR routines are per CCR custom ItemSet load logic.
  - Case-sensitive naming requirement: ImplementCCR.<CCR_ID>.
    - Example: CCRID of BEST0001 would require routine name ImplementCCR.BEST0001.
    - Client Tools automatically run routine during ItemSet deployment, if exists.
    - Run after importing and compiling all code in ItemSet.

Automating Deployment Tasks (cont.)

- CCR Event Handlers are per System custom ItemSet load logic.
    - Create class that extends %Studio.SourceControl.CCREventHandler.
    - Create callback methods.
        - See method stubs in %Studio.SourceControl.CCREventHandler.
    - Set ^SYS("SourceControl","EventHandler","Class")="MyClassName."
- Write statements directed to Transport Log.

Quiz: ImplementCCR Routines

Which of the following is true about ImplementCCR Routines.

A. They are executed automatically on ItemSet deployment.

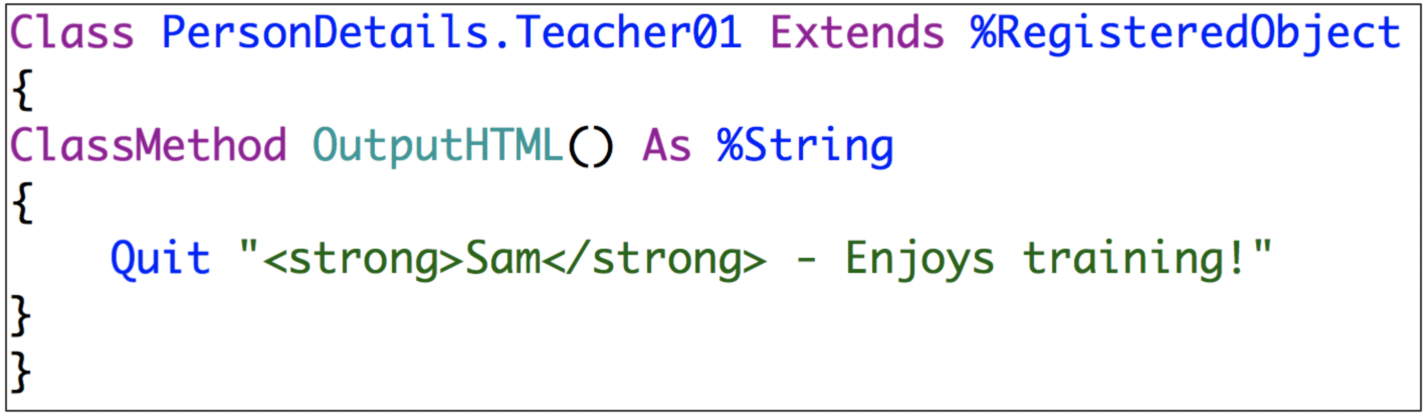
B. The name is not case-sensitive.

C. You will get an error if you try to upload a routine named ImplementCCR.BEST12345 to CCR BEST6789.

D. They cannot contain CCR built-in functions such as $$Env^%buildccr.

Answer: A

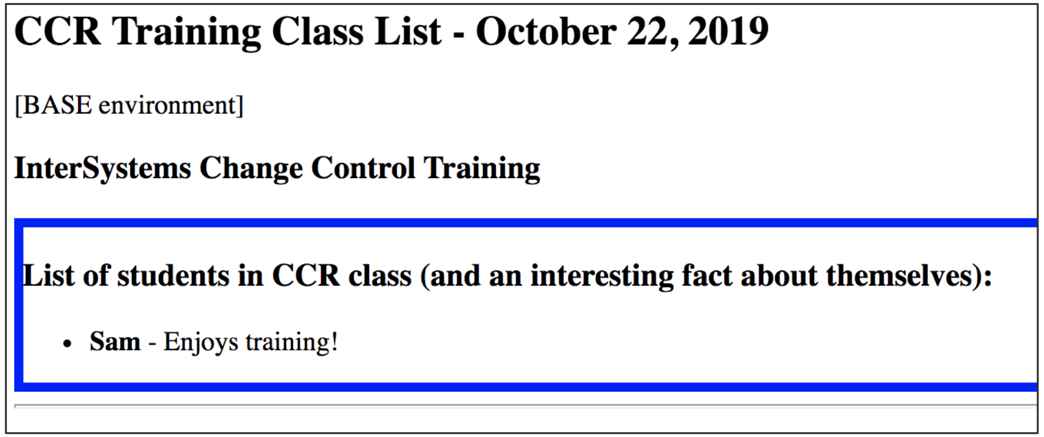
Class List Exercises



```
Class PersonDetails.Teacher01 Extends %RegisteredObject
{
ClassMethod OutputHTML() As %String
{
    Quit "<strong>Sam</strong> - Enjoys training!"
}
}
```

Class	Method
PersonDetails.Teacher01	OutputHTML

```
Do ##class(PersonDetails.ClassList).RegisterStudent(“Teacher01”, “OutputHTML”)
```



CCR Training Class List - October 22, 2019

[BASE environment]

InterSystems Change Control Training

List of students in CCR class (and an interesting fact about themselves):

- **Sam** - Enjoys training!

Exercise ICC510-7 through ICC510-10

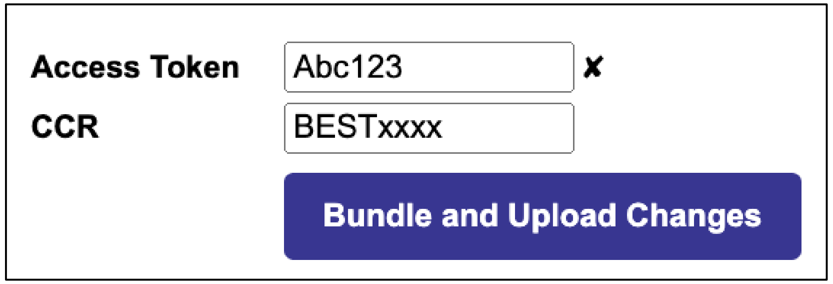
Part 7: Additional Information

Bundle and Upload Validation

- Bundle and Upload screen for Tier 1 systems validates:
  - Access key.
  - Whether CCR ID exists.
  - State of CCR.
  - System for CCR vs current namespace.
- Revalidation triggered by:
  - Changing access key.
  - Changing CCR ID.
  - Clicking ⟳ icon at end of error message.

Bundle and Upload: Invalid Access Token

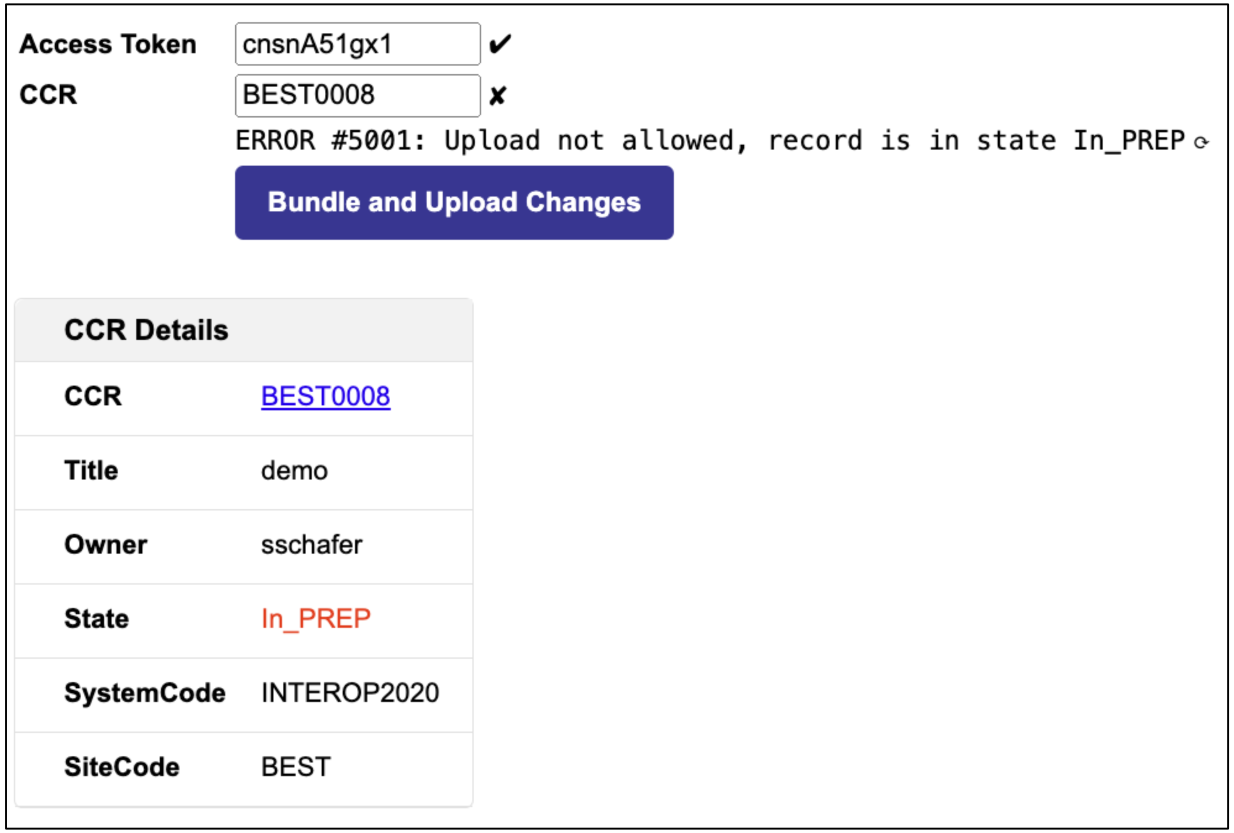
- Indicated by x next to Access Token field.
- Most likely a typo.



Access Token	Abc123 ✗
CCR	BESTxxxx

Bundle and Upload: Invalid State

- Valid states:
  - In_BASE.
  - BASE_Pending_Peer_Review.
- Solution:
  - Current state = In_PREP?
    - Perform markPREPComplete transition.
  - Current state past BASE_Pending_Peer_Review? Either:
    - Use catch-up CCR to progress change to original CCR (preferred method).
    - Use backwards transitions to return to In_BASE, backing out changes.



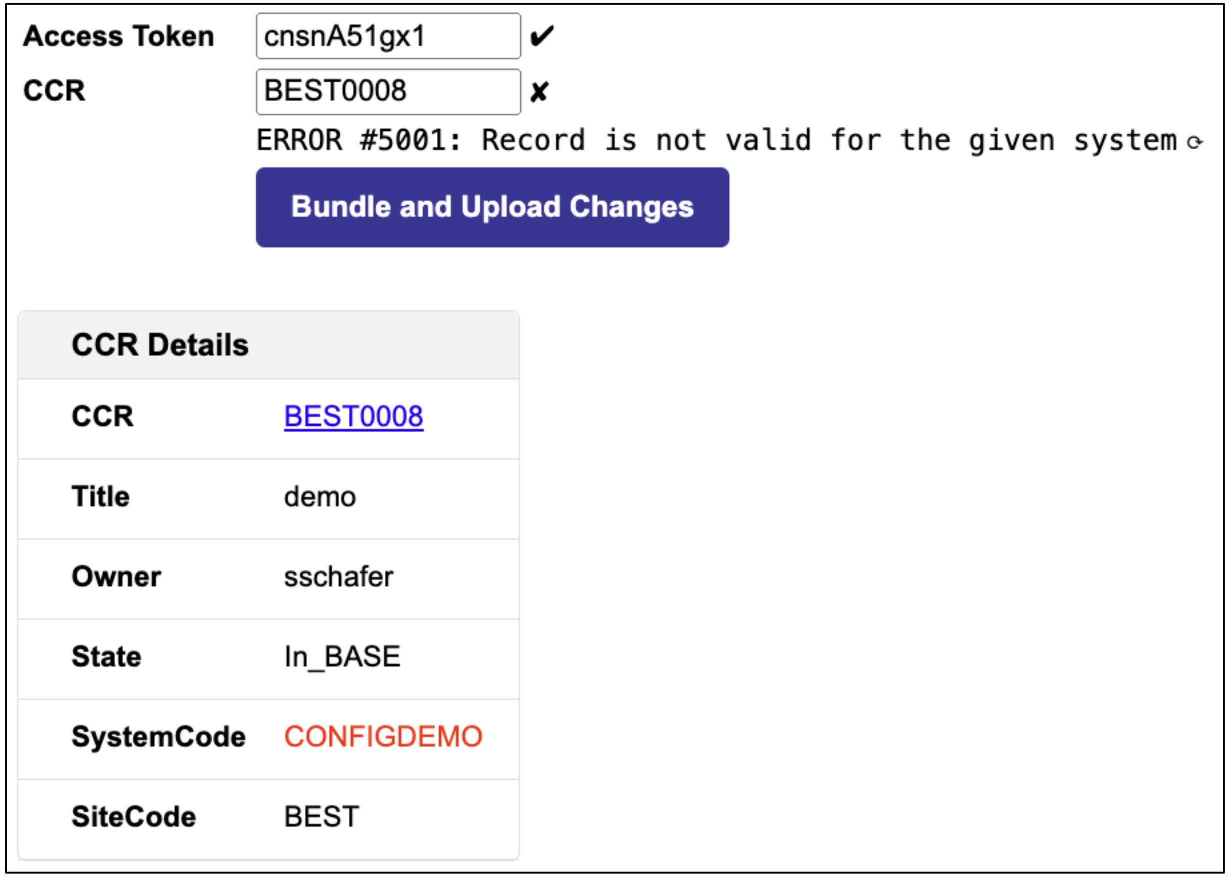
Access Token	cnsnA51gx1	✔
CCR	BEST0008	✘

```
ERROR #5001: Upload not allowed, record is in state In_PREP
```

CCR Details	
CCR	BEST0008
Title	demo
Owner	sschafer
State	In_PREP
SystemCode	INTEROP2020
SiteCode	BEST

Bundle and Upload: Invalid System

- System of CCR does not match System of environment.
- Causes and solution:
  - Wrong CCR.
    - Enter correct CCR ID.
  - Created CCR for wrong system.
    - System field editable until upload first ItemSet.
    - Edit CCR to change System.
  - Made change in wrong namespace.
    - Revert and make change in correct namespace.



Access Token cnsnA51gx1 ✔

CCR BEST0008 ✘

ERROR #5001: Record is not valid for the given system

CCR Details	
CCR	BEST0008
Title	demo
Owner	sschafer
State	In_BASE
SystemCode	CONFIGDEMO
SiteCode	BEST

Tier 1 Transport Load To Namespace Flow

- After Tier 1 items are loaded to the workspace, a number of additional things occur.
  1. If a CCR Event Handler is configured, the ItemSetAfterLoadToOS() method is executed.
  2. Any items in Tier 1 SubFolders are automatically loaded into the namespace:
     - /cls (class files).
     - /cspapp (web files e.g. JavaScript, CSP, CSS).
     - /inc (include files).
     - /prj (project files).
     - /rtn (routines).

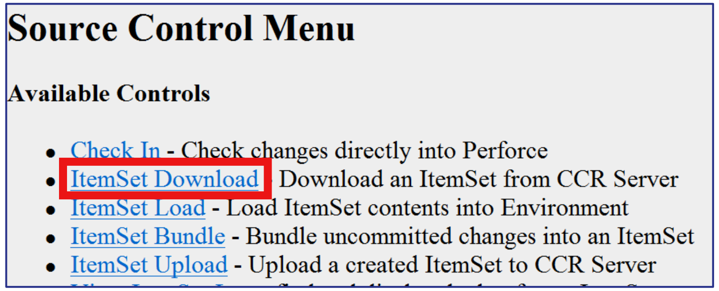
Tier 1 Transport Load To Namespace Flow (cont.)

- After Tier 1 items are loaded to the workspace, a number of additional things occur.

3. As they are loaded, all items are added to a project named after the ItemSet.
4. The project is compiled (which compiles all items in the proper order).
5. If an ImplementCCR routine exists for that CCR, it is executed.
6. If a CCR Event Handler is configured, the ItemSetAfterLoadToNS() method is executed.

Downloading an ItemSet via IDE

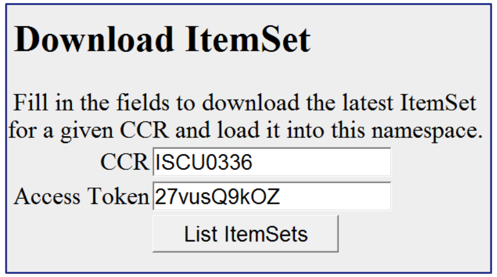
- If 'deploy' links are not configured, ItemSets can be downloaded via IDE:
  1. Select "Source Control" menu > Show CCR Controls.
  2. Click ItemSet Download link.
  3. Type the CCR ID into the Download ItemSet form.
     - Make sure it is accurate!



Source Control Menu

Available Controls

- Check In - Check changes directly into Perforce
- ItemSet Download - Download an ItemSet from CCR Server
- ItemSet Load - Load ItemSet contents into Environment
- ItemSet Bundle - Bundle uncommitted changes into an ItemSet
- ItemSet Upload - Upload a created ItemSet to CCR Server

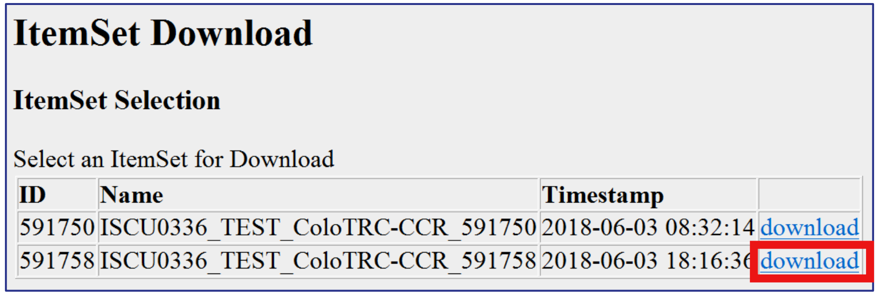


Download ItemSet

Fill in the fields to download the latest ItemSet for a given CCR and load it into this namespace.

CCR ISCU0336

Access Token 27vusQ9kOZ



ItemSet Download

ItemSet Selection

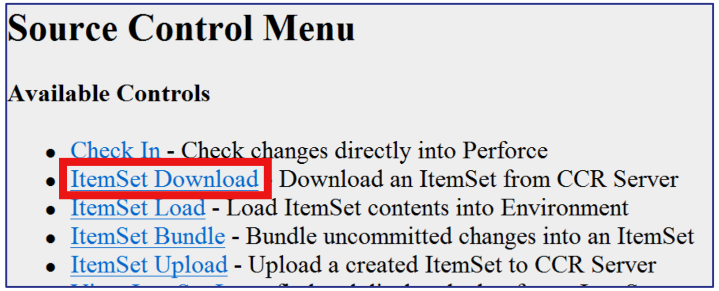
Select an ItemSet for Download

ID	Name	Timestamp	
591750	ISCU0336_TEST_ColoTRC-CCR_591750	2018-06-03 08:32:14	download
591758	ISCU0336_TEST_ColoTRC-CCR_591758	2018-06-03 18:16:36	download

Downloading an ItemSet via IDE (cont.)

- If 'deploy' links are not configured, ItemSets can be downloaded via IDE:

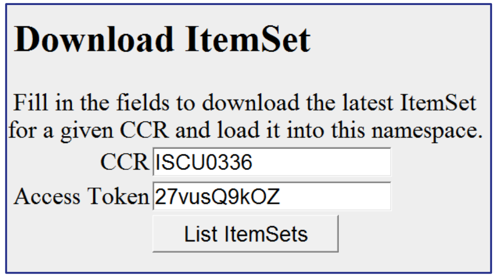
4. Copy the "Access Token" from the Perforce Pane on the CCR and paste it into the Download ItemSet form.
5. Click "List ItemSets."
6. A list of available ItemSets will appear, select "download" on the newest one (the last one in the list).
   - If only one exists, it will automatically begin to download.



Source Control Menu

Available Controls

- Check In - Check changes directly into Perforce
- ItemSet Download - Download an ItemSet from CCR Server
- ItemSet Load - Load ItemSet contents into Environment
- ItemSet Bundle - Bundle uncommitted changes into an ItemSet
- ItemSet Upload - Upload a created ItemSet to CCR Server

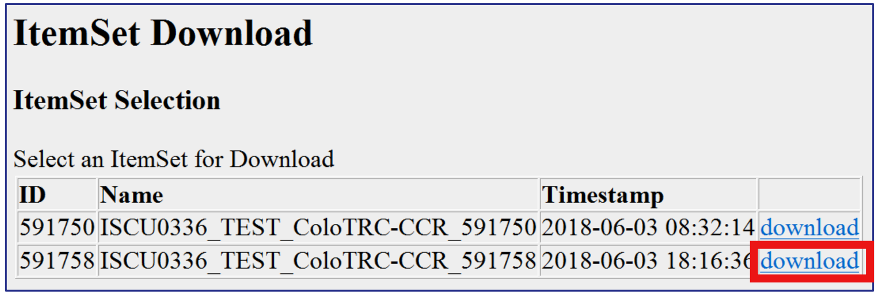


Download ItemSet

Fill in the fields to download the latest ItemSet for a given CCR and load it into this namespace.

CCR ISCU0336

Access Token 27vusQ9kOZ



ItemSet Download

ItemSet Selection

Select an ItemSet for Download

ID	Name	Timestamp	
591750	ISCU0336_TEST_ColoTRC-CCR_591750	2018-06-03 08:32:14	download
591758	ISCU0336_TEST_ColoTRC-CCR_591758	2018-06-03 18:16:36	download

Summary

- What are the key points for this module?

ICC 520: CCR Tier 1 - Baselining Source Code

InterSystems
Creative data technology

Overview

- Baselining Overview.
- Tier 1 Baseline tools.
- Re-baselining.

Part 1: Baselining Overview

Review: What is Baselining?

- Synchronizes contents of Perforce and all environments with a source of truth.
  - Source of truth usually current state of LIVE environment.
- Process steps include:
  - Export all source (Tier 1) and/or configuration (Tier 2) from environment.
  - Reconcile any content already in Perforce.
  - Populate other Perforce branches with reconciled export.
  - Populate source workspaces.
  - Refresh environments.

Review: Why is Baselining Important?

- Establishes starting point in Perforce.
  - Editing item in environment that isn’t in Perforce means:
    - Can’t diff change.
    - Can’t backout.
- Establishes clean integration history.

Re-baselining

- Performing a baseline again.
- Cleans up previous mistakes from poor CCR usage.
  - Abandoned changes.
  - Unlocked TEST or LIVE.
- Establishes clean integration history.
  - Prevents future merge conflicts.
- History of changes still maintained in Perforce.

Baselining Prerequisites

- Verify all productions use System Defaults.
  - Ensemble and HealthShare only.
  - See ICC530 for more information.
- Progress all CCRs to terminal state (Closed, Cancelled, Merged).
  - Baselining significantly simpler when no active changes in flight for System.
    - Avoids needing to reintroduce active changes.

Baselining Prerequisites (cont.)

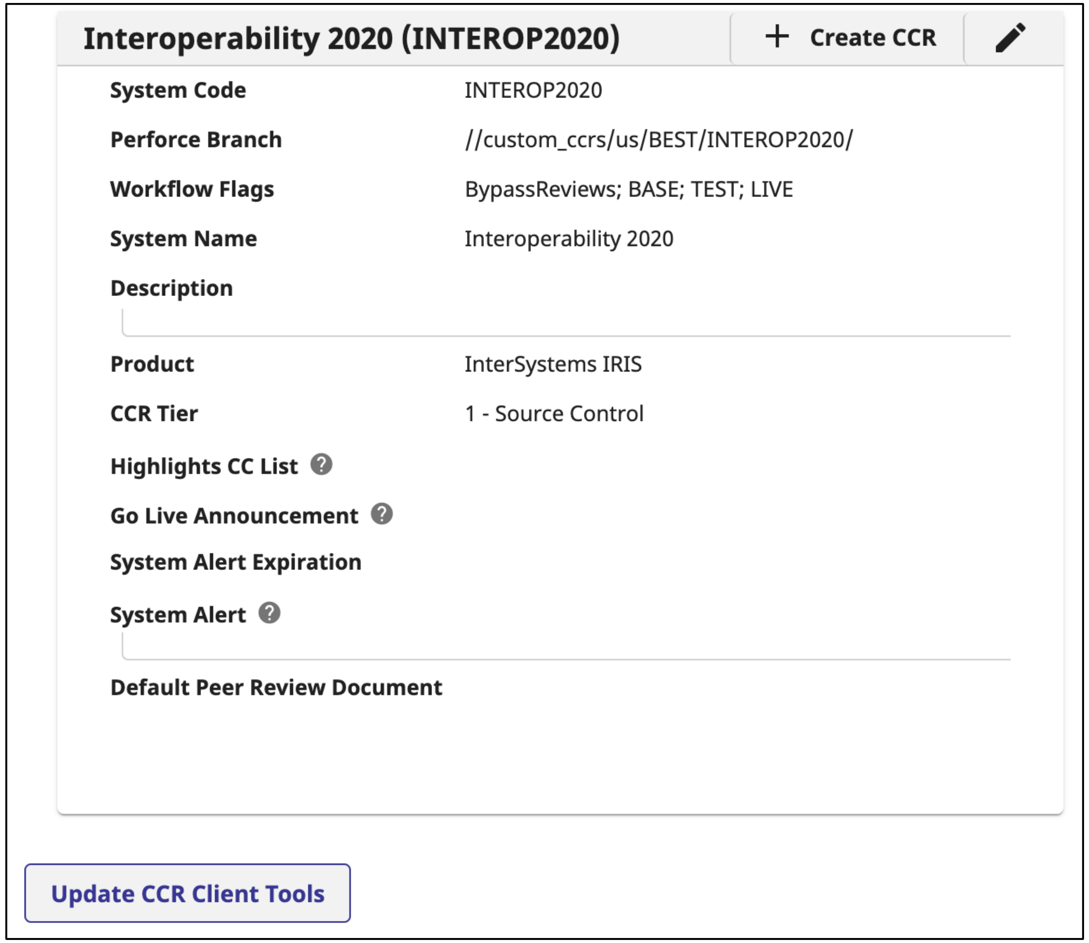
- If baselining multiple systems:
  - Each System will have to be frozen while baselining that System.
    - Baseline one at a time.
    - Prioritize which System to finish first.
- Clear uncommitted queue in BASE environment.
- Add system alert to warn everyone not to create CCRs.
  - See ICC320.
- Consider locking BASE to prevent accidental changes.
  - Use %Studio.SourceControl.ISC:Lock() method.

Baselining Prerequisites (cont.)

- Must first properly configure:
  - CCR System Details.
    - See ICC320.
  - Namespace.
    - See ICC430.
- For existing application with multiple Environments – choose which will be source of truth.
  - Usually LIVE.
    - Code in BASE, TEST, and UAT replaced with that of LIVE.

Baselining Prerequisites (cont.)

- Latest client tools have tooling to automate baseline process.
- Update client tools before baselining.
  - Create Tier 1 CCR using update CCR Client Tools button on System Details page.



Interoperability 2020 (INTEROP2020)

System Code	INTEROP2020
Perforce Branch	//custom_ccrs/us/BEST/INTEROP2020/
Workflow Flags	BypassReviews; BASE; TEST; LIVE
System Name	Interoperability 2020
Description	
Product	InterSystems IRIS
CCR Tier	1 - Source Control
Highlights CC List	
Go Live Announcement	
System Alert Expiration	
System Alert	
Default Peer Review Document	

Quiz: Purpose of Baselining

Question:

Which of the following is true about baselining? Select all that apply.

A. Rollback not possible without baselining?

B. Establishes starting point for every item managed via CCR.

C. Establishes clean integration history.

D. Required for diffing changes.

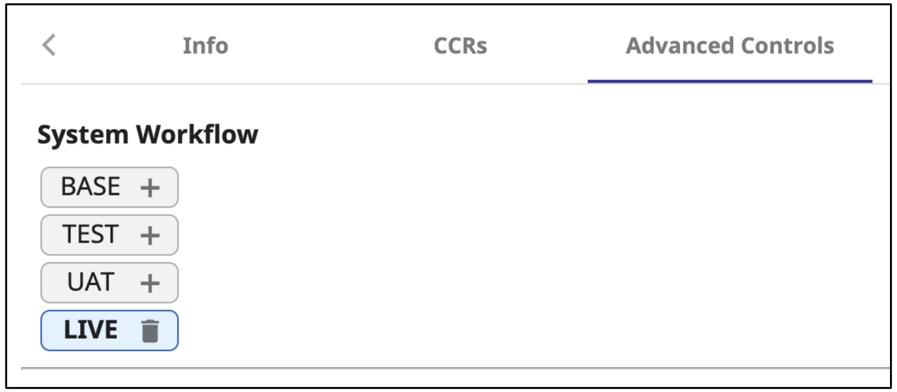
Answer:

All of the above.

Part 2: How To Baseline Existing Application

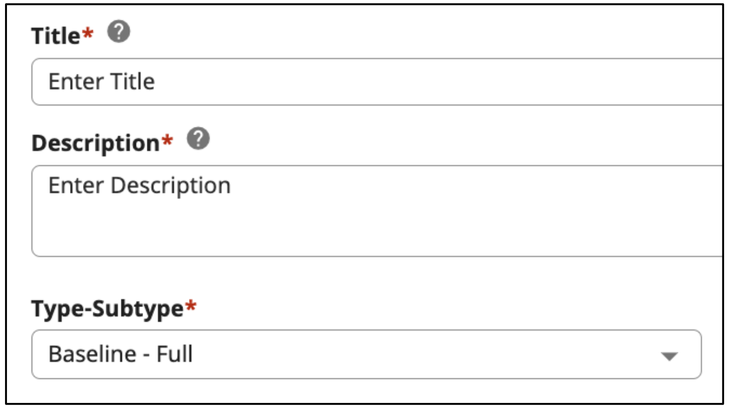
How To: Baseline Existing Application

1. Create Tier 1 CCR with Type – Subtype = Baseline – Full.
2. Remove all Environments except source of truth from System Definition.
3. Progress out of In_PREP.
   - Example: In_LIVE when using LIVE as source.



System Workflow

- BASE +
- TEST +
- UAT +
- LIVE 🗑



Title* ?

Enter Title

Description* ?

Enter Description

Type-Subtype*

Baseline - Full

How To: Baseline Existing Application (cont.)

4. Run Baseline utility in source Environment.
   - In Terminal session, run `Do Baseline^%buildccr`.
   - Answer prompts.
   - This will submit an ItemSet to your CCR.
5. Confirm CCR has submitted changes.
6. Add BASE to System Definition.
   - Select checkbox to generate Environment Initialization ItemSet.
     - Source Environment = LIVE.

How To: Baseline Existing Application (cont.)

7. Make note of ItemSet ID.
8. Download ItemSet through Terminal in BASE.
   - `Do Download^%buildccr.`
9. Add TEST (if exists) to System definition, generating an ItemSet.
   - Source = BASE.
10. Progress CCR to In_TEST.
11. Download ItemSet through Terminal in TEST.
    - `Do Download^%buildccr.`

How To: Baseline Existing Application (cont.)

12. Repeat steps 9-11 for UAT (if exists).
13. Progress CCR to Closed.

Part 3: Add New Environment

Add New Environment To Workflow

1. Clone existing VM.
   - Example: clone TEST to create LIVE
2. Run Configure^%buildccr to correct Environment.
3. Add new Environment to System definition in CCR.
   - Select checkbox to generate Environment Initialization ItemSet.
     - Source Environment should be highest existing Environment.
7. Make note of ItemSet ID.
8. In new Environment, download ItemSet through Terminal.
   - `Do Download^%buildccr.`

Summary

- What are the key points for this module?

ICC 530: CCR Tier 1 – Interoperability Components

InterSystems
Creative data technology

Overview

- Best practices.
- Handling differences between environments.
- Management Portal source control hooks.

Best Practices

- Use Tier 0 CCRs when not feasible to maintain item in source control.
  - Makes change discoverable.
  - Enforces workflow such as peer reviews.
- Use source control when possible.
  - Tier 1 CCR.

Developing Interfaces

- Use iterative process to configure new interfaces.
- Disable interfaces in development.
- Configure by progressing small, discrete CCRs to Closed.
- When interface ready for Go-LIVE, progress CCR enabling finished components.

Product Versions

- Versions in slides refer to releases for InterSystems IRIS®.
- HealthShare versions do not correlate to InterSystems IRIS versions.

Differences Between Environments

- Some configuration fields for Production components differ between environments.
  - File paths.
  - IP addresses.
- Production must be identical to integrate between environments.
  - Cannot make changes to Production directly in TEST, UAT, or LIVE.
- Solution is System Default Settings.

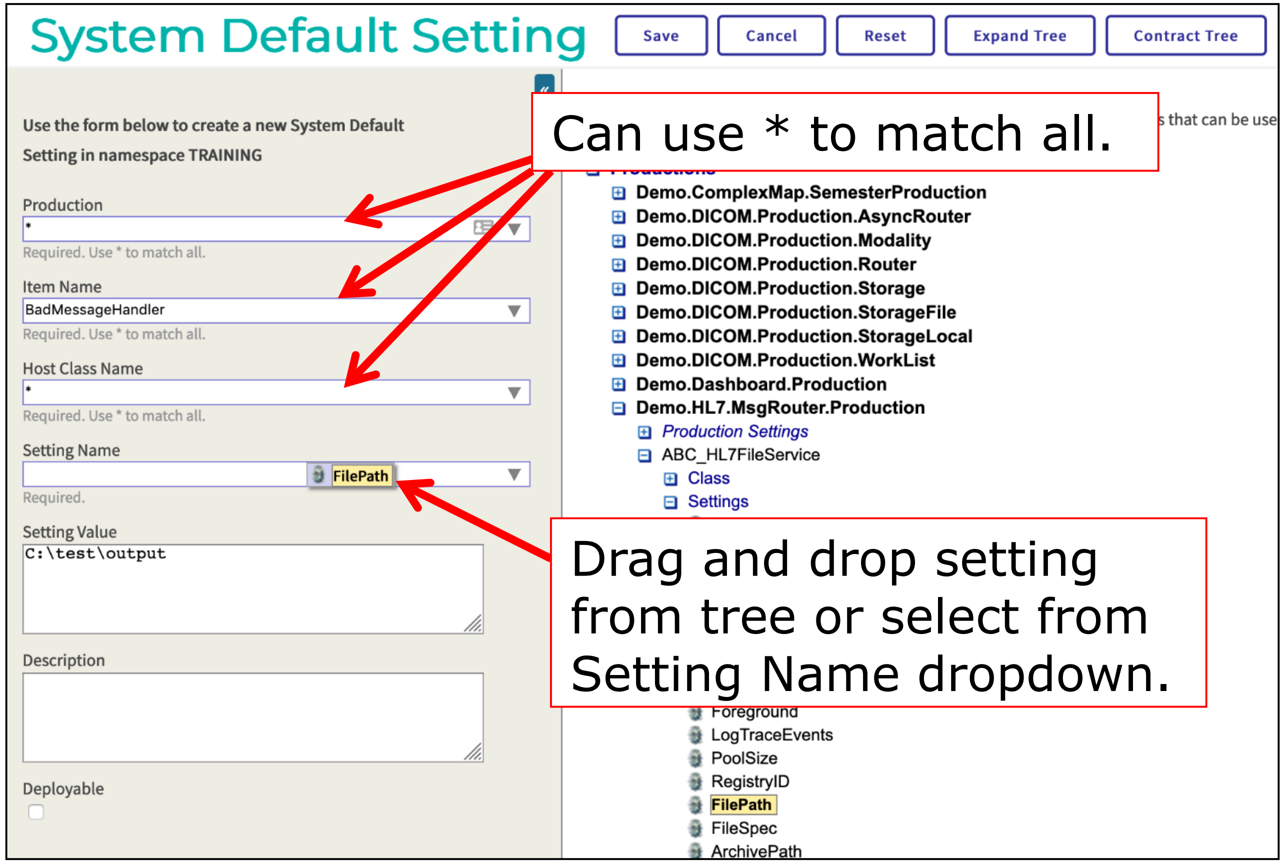
System Default Setting

- Use to configure default values for settings of business components.
  - Alternative to specifying values of settings in Production definition.
  - Configure unique sets of default values in each Environment.
- Allows Production definition to be identical in all Environments.

BASE	TEST
To_Lab	To_Lab
File Path = c:\BASE\\In	File Path = c:\TEST\\In
Failure Timeout = 15	Failure Timeout = 15

Creating System Default Settings

- Configure > System Default Settings.



System Default Setting
Save
Cancel
Reset
Expand Tree
Contract Tree

Use the form below to create a new System Default Setting in namespace TRAINING

Production
*
Required. Use * to match all.

Item Name
BadMessageHandler
Required. Use * to match all.

Host Class Name
*
Required. Use * to match all.

Setting Name
FilePath
Required.

Setting Value
```
C:\test\output
```

Description

Deployable
□

s that can be use
Demo.ComplexMap.SemesterProduction
Demo.DICOM.Production.AsyncRouter
Demo.DICOM.Production.Modality
Demo.DICOM.Production.Router
Demo.DICOM.Production.Storage
Demo.DICOM.Production.StorageFile
Demo.DICOM.Production.StorageLocal
Demo.DICOM.Production.WorkList
Demo.Dashboard.Production
Demo.HL7.MsgRouter.Production
Production Settings
ABC_HL7FileService
Class
Settings
Foreground
LogTraceEvents
PoolSize
RegistryID
FilePath
FileSpec
ArchivePath

Can use * to match all.

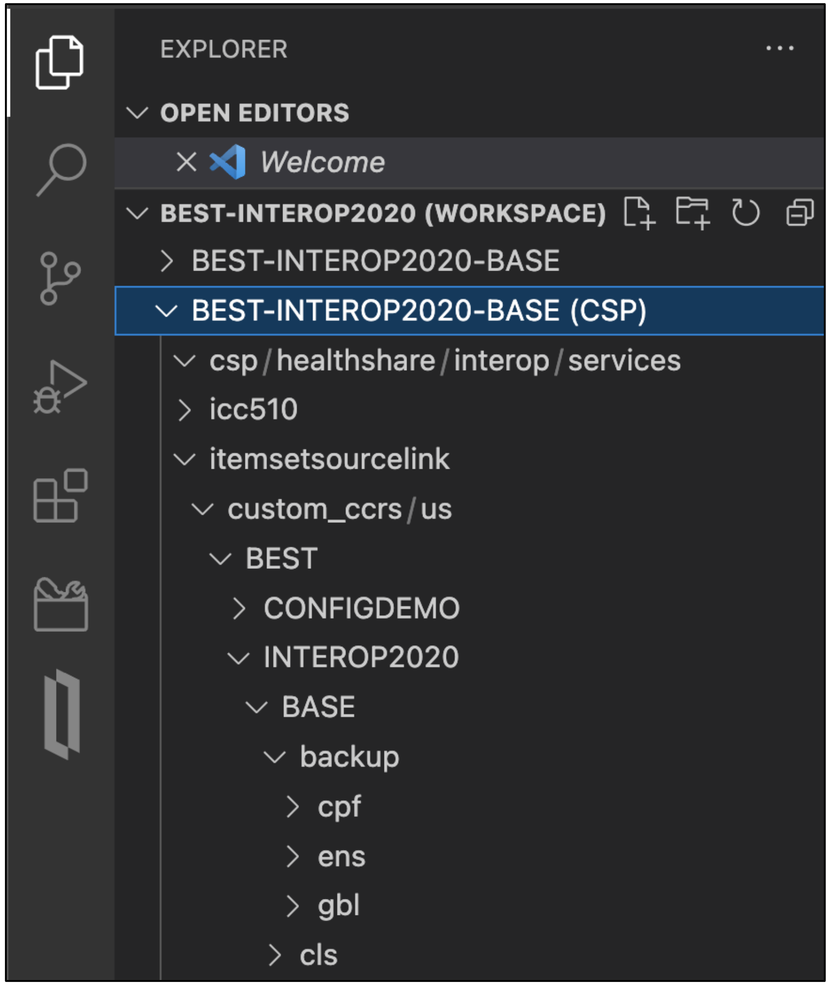
Drag and drop setting from tree or select from Setting Name dropdown.

Export System Default Settings

- Source control menu > CCR Controls > Export Default Settings > Choose Production > Export.
- Exports to /backup subdirectory.
- Can bundle and upload in BASE only.
  - Add to source control using itemsetsourcelink web application in IDE.

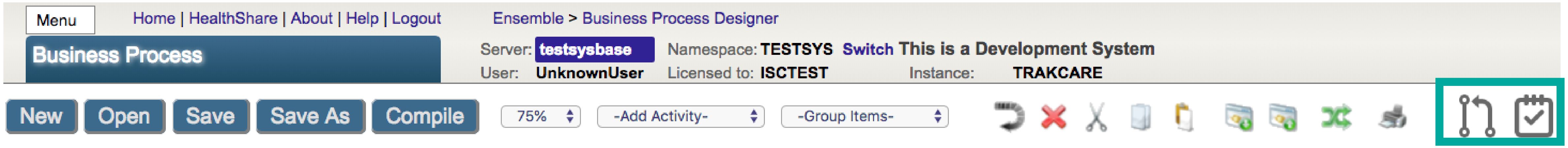
Add Exported File to Source Control

- Export file to source workspace.
- To find file in:
  - VS Code – ObjectScript:
    - From Explorer > itemsetsourcelink_*system* > custom_ccrs > …
  - In Studio:
    - From namespace view > CSP Files > itemsetsourcelink_*system* > custom_ccrs > …
- Right-click file > Source Control > Add To Source Control > Yes.
- Bundle and upload to a CCR.



Management Portal Source Control Hooks

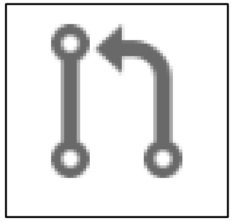
- Many pages in Management Portal have embedded hooks.
- Only visible in source control configured namespaces.
- Behavior same as source control menu in IDE.
- All items modified using Management Portal hooks automatically imported on ItemSet Deployment.


Menu
Home | HealthShare | About | Help | Logout
Ensemble > Business Process Designer
Business Process
New
Open
Save
Save As
Compile
75%
-Add Activity-
-Group Items-

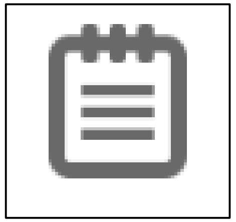
Server: testsysbase Namespace: TESTSYS Switch This is a Development System
User: UnknownUser Licensed to: ISCTEST Instance: TRAKCARE

Management Portal Source Control Hooks (cont.)

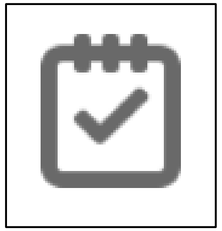
- Commands.
  - Add to source control.
  - Checkout.
  - Undo checkout.
  - CCR controls.
  - Diff.



- Outputs.
  - View unread output of control.

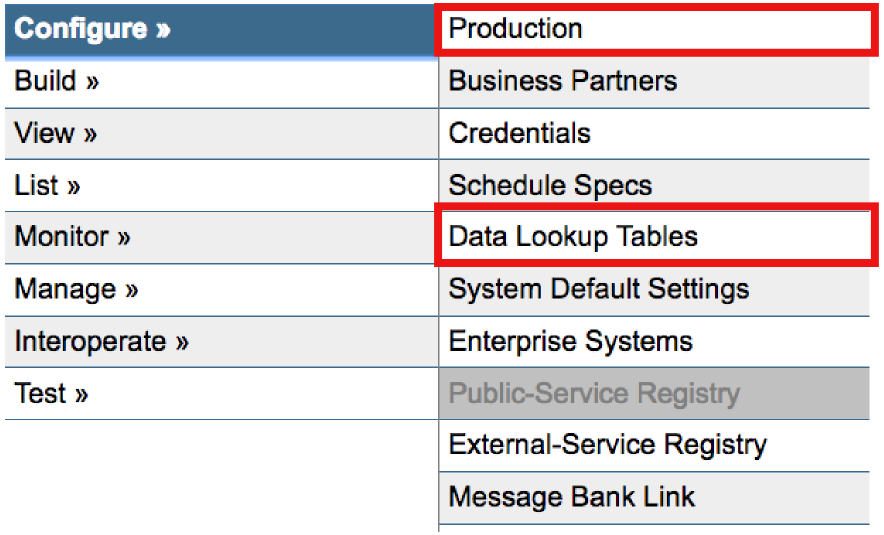


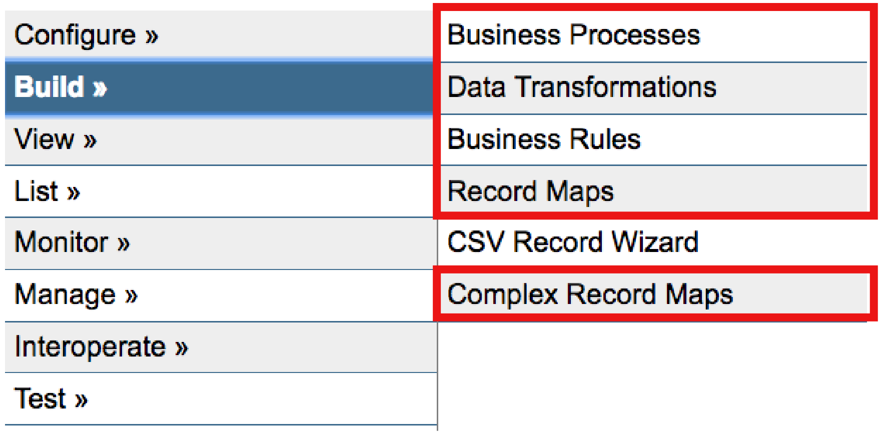
  - All source control output has been viewed.



Pages with Hooks

- Production.
- Data Lookup Tables.
- Business Processes.
- Data Transformations.
- Business Rules.
- Record Maps.
- Complex Record Maps.
- HL7 v2 Custom Schemas.
  - Only visible if privileged to edit.





Production Decomposition

- Allows users to checkout individual components of Production.
  - Without decomposition, only able to checkout entire Production.
- Cannot include custom code in Production class definition.
  - Instead, make Production inherit from super class that contains custom code.
- Can check out Production in IDE to edit XML.
  - Requires no components checked out.
- Recommend client tools updated since December 10, 2024.

How To: Enable Production Decomposition with Existing Productions

1. Progress any active CCRs changing Production to LIVE.
2. Create a Tier 1 CCR to document enabling Production Decomposition.
3. Implementation Plan:
   - BASE only:
     - `Do Configure^%buildccr`
     - Accept defaults until reach prompt: Decompose Productions in Source Control to handle Business Host and Production settings individually?
     - Enter Yes to Decompose Productions.
     - Enter Yes to the following 2 prompts to decompose existing Productions.
   - Non-BASE environments: Deploy ItemSet.

How To: Enable Production Decomposition with Existing Productions (cont.)

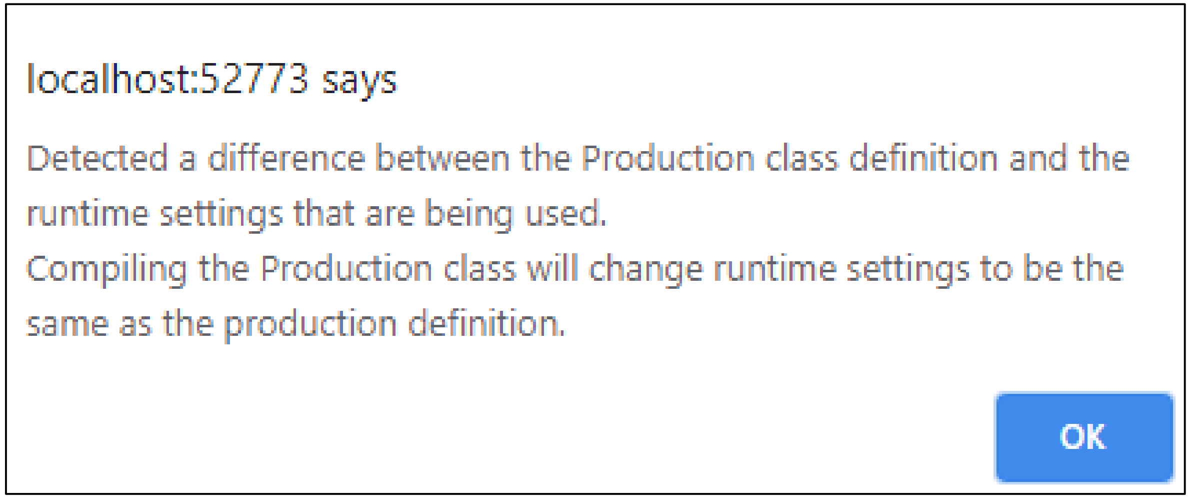
4. Bundle and upload changes to your CCR.
5. Progress CCR through to Closed.
   - Must be pushed to LIVE before changes to Production can begin.

How To: Enable Production Decomposition **without** Existing Productions

1. Progress any active CCRs changing Production to LIVE.
2. Create a Tier 0 CCR to document enabling Production Decomposition.
    - Implementation Plan for all Environments:
        - `Do Configure^%buildccr`
        - Accept defaults until reach prompt: Decompose Productions in Source Control to handle Business Host and Production settings individually?
        - Enter Yes to Decompose Productions.
        - Enter No to the following prompt to decompose existing Productions.
3. Progress CCR through to Closed.
    - Must be pushed to LIVE before changes to Production can begin.

Production Without Decomposition: Undo Checkout

- Steps:
  1. Click ⤺ > Undo Checkout.
  2. Compile Production class in IDE.
  3. Click Update button on Management Portal Production page.
- Change not reverted until these 2 additional steps completed!
- Only need to do this 3-step process for Productions.
  - Undo Checkout action sufficient for other components with hooks in Portal.

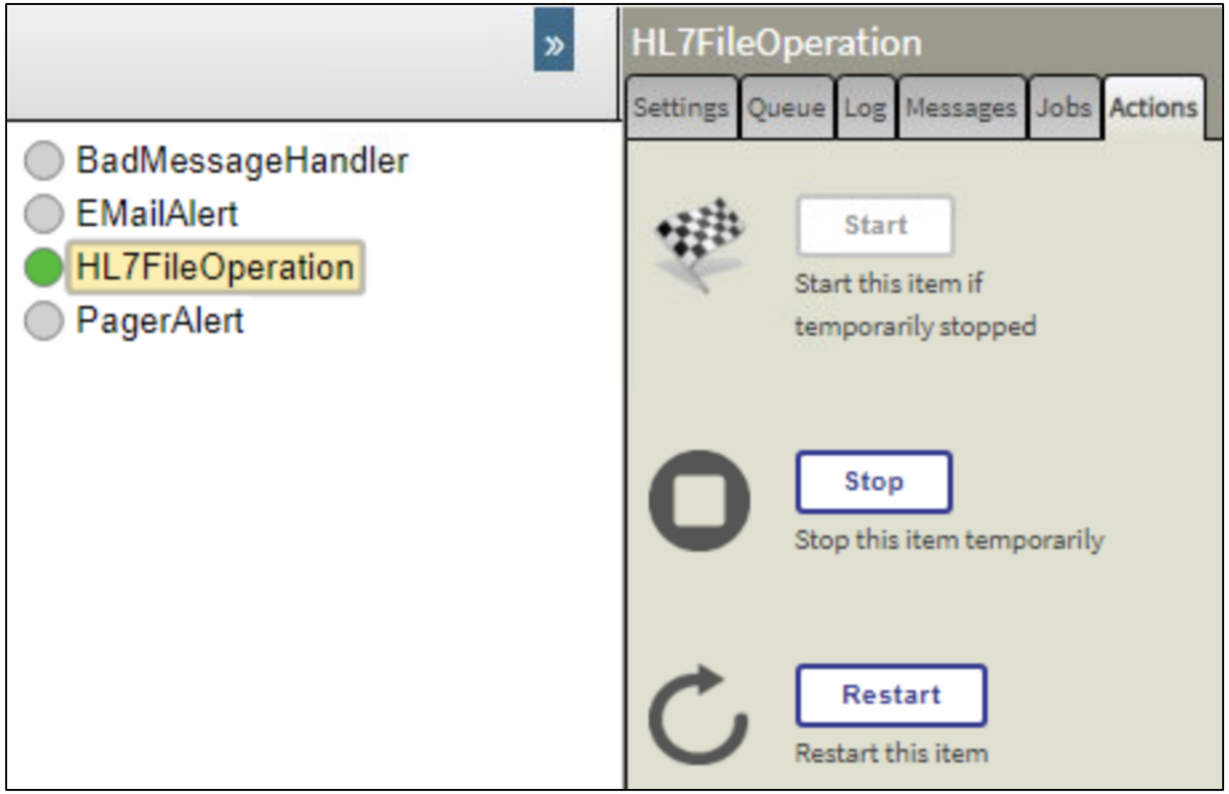


localhost:52773 says

Detected a difference between the Production class definition and the runtime settings that are being used.
Compiling the Production class will change runtime settings to be the same as the production definition.

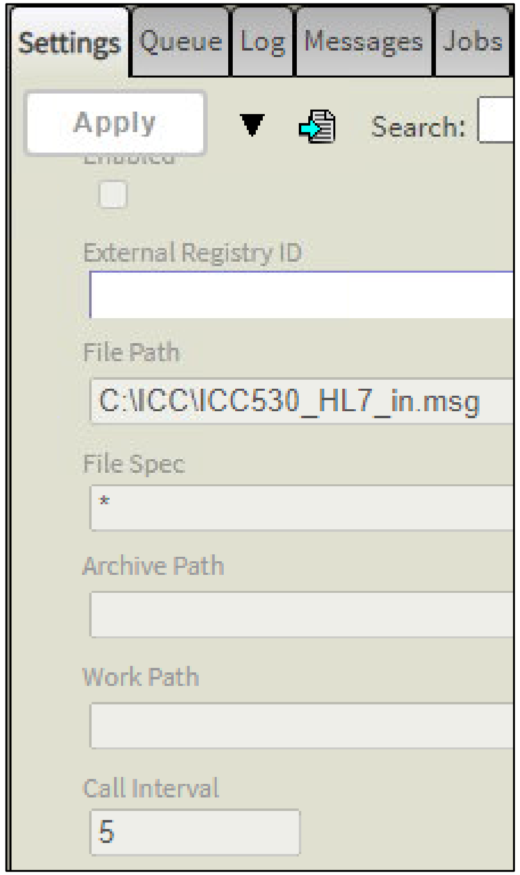
Productions: Temporarily Start, Stop, Restart

- Normally, double-click component to enable, disable, or restart.
  - Changes Production class definition, therefore:
    - Cannot do in TEST/UAT/LIVE.
    - Requires checkout in BASE.
- Able to temporarily stop, start, or restart components.
  - Select component > Actions.
  - Does not require unlocking environment or check out.
  - Cannot temporarily start if disabled.



Business Component Setting Labels (v2019.4 – v2021.1)

- Fields greyed out when Production read only.
- Must checkout Production in BASE to see color coding.
  - Color coding indicates source of value.
- Cannot see color coding in TEST/UAT/LIVE because cannot checkout Production.



Enabled
□

External Registry ID

File Path
C:\ICC\ICC530_HL7_in.msg

File Spec
*

Archive Path

Work Path

Call Interval
5

Summary

- What are the key points for this module?

Do Exercise 530-1 and 530-2

ICC 540: CCR Tier 1 – InterSystems HealthShare®

InterSystems
Creative data technology

Objectives

- Identify subtleties for using change control for HealthShare.
- Demonstrate best practices for using CCR for HealthShare changes.
- Identify how secondary environments are used in the CCR workflow.

Best Practices

- HealthShare registries do not support CCR Transport.
  - Use Tier 0 CCRs.
    - Makes change discoverable.
    - Enforces workflow such as peer reviews.
  - Consider Tier 1 CCR with ImplementCCR routine.
    - Call documented APIs.
- HealthShare Change Control (HCC) coming soon.
  - Enables export of configuration changes for CCR Transport automation.

Part 1: CCR for HealthShare Components

HealthShare Health Connect

- Many additional components candidates for Tier 0 CCR only:
    - Facility Registry.
    - HealthShare Service Registry.
    - OID Registry.
    - Assigning Authority Registry.
    - Configuration Registry.
    - XUA Configuration Registry.
    - Coded Entry Registry.
    - Exclude Audit Events table.
    - FHIR CSP Configuration.

HealthShare Unified Care Record: Candidates for Tier 0 Only

- Everything in Health Connect.
- User/Clinician Registry.
- Gateways Registry.
- Consent Registry (multiple tables).
- Program Registry.
- Relationship Registry.
- Clinician Group Registry.
- PIX Notification Registry.

HealthShare Unified Care Record: Candidates for Tier 0 Only (cont.)

- Query Instance Registry.
- Security Domains.
- Consent Groups.
- Push Notification settings (Delivery Management).
- Terminology settings (multiple tables).

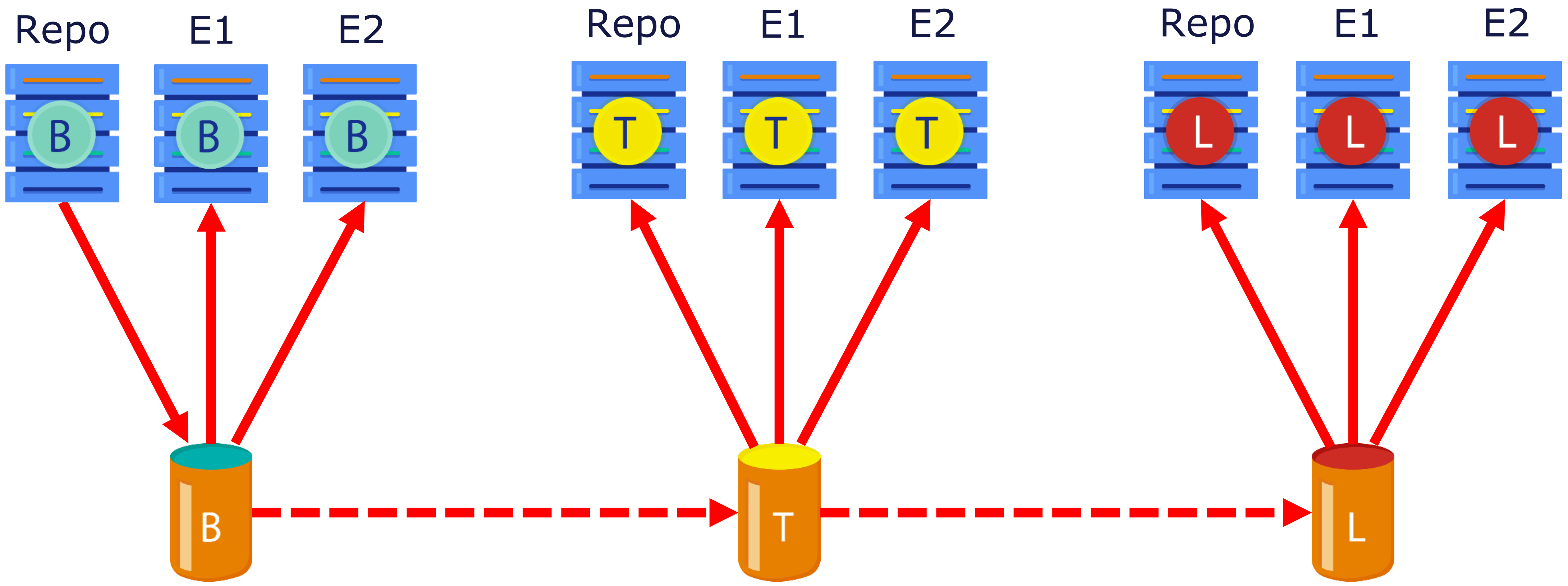
HealthShare Unified Care Record CCR System Architecture Options

- Separate CCR Systems for Edge Gateway, Access Gateway, and Registry.
  - Use Secondary Environments for multiple gateways.
- Instance-wide CCR.
  - Uses 1 CCR System with shared code database for all namespaces.

Secondary Environments

- Additional environments to receive changes at same time as corresponding primary environment.
  - Any primary environment can have 1+ secondary environments.
- Secondary environments for BASE require deployment of ItemSet during In_BASE state.
- ItemSet deployment opens new tab for each Environment.
  - Tabs automatically close on successful ItemSet deployment.
- See ICC440 for more information.

HealthShare Secondary Edge Gateways



HealthShare Health Insight

- Analytics Management Portal supports Tier 1 source control hooks.

HealthShare Patient Index: Linkage Definitions

- Maintain linkage definition class in source control using IDE.
  - Follow procedure described in ICC530 for no portal hooks but editable in IDE.
- Do not add generated classes to source control.
  - Will be regenerated on compilation.
- Consider configuring CCR Event Handler to handle additional required implementation steps:
  - Stop production.
  - Compile linkage definition.
  - Start production.
  - Rebuild linkage data.

Summary

- What are the key points for this module?

ICC 600:
Introduction to CCR Tier 2

InterSystems
Creative data technology

Objectives

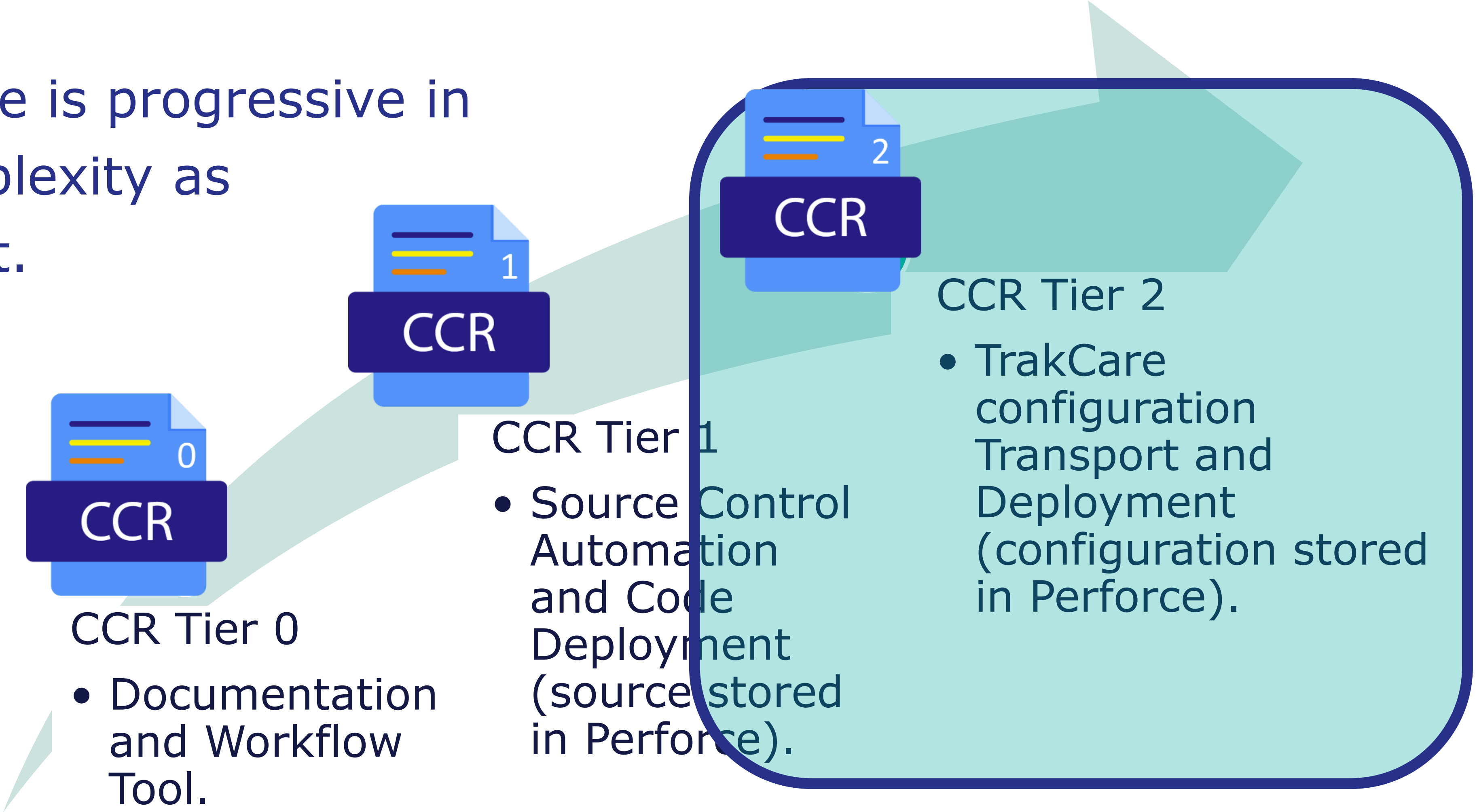
- Explain at a high level what CCR Tier 2 means and when it is useful.
- Explain how TCC, ElementXML, and GUIDs are related to CCR Tier 2.
- Identify whether a deeper dive into CCR Tier 2 training material would be beneficial.

Part 1: CCR Tier 2: Working with TrakCare Configuration

Review: CCR Usage Tiers

- Tiered usage is progressive in terms of complexity as well as benefit.

This series of courses focuses on CCR Tier 2.



CCR Tier 0

- Documentation and Workflow Tool.

CCR Tier 1

- Source Control Automation and Code Deployment (source stored in Perforce).

CCR Tier 2

- TrakCare configuration Transport and Deployment (configuration stored in Perforce).

CCR Tier 2: The Fundamentals

- Includes same documentation and workflow as Tier 0.
- Tracks changes made to TrakCare configuration items via TrakCare UI.
- TrakCare configuration versioned within source control.
- Movement of configuration to or from environments via CCR Transport mechanism.
- CCR Implementation and Backout Plans have default values.
    - Assumes plan is to use CCR Transport.
    - Must update plans if manual steps required.

CCR Tier 2: The Audience

- ICC6nn courses focus on Tier 2 for TrakCare.
- Prerequisite knowledge:
  - CCR Tier 0.
  - CCR Transport.

Quiz: CCR Tier 2 Usage

Question:

Which of the following would require a Tier 2 CCR (and could not be controlled by Tier 1)? Select all that apply.

A. Integration logic.

B. Custom TrakCare code.

C. TrakCare Security Groups.

D. Product upgrades.

E. Hospital configuration changes.

Quiz: CCR Tier 2 Usage (cont.)

Answer:

C. TrakCare Security Groups.

E. Hospital configuration changes.

Product upgrades should be tracked in Tier 0 CCR. All other changes in the above list can be included in a Tier 2 CCR, but only C and E have to be in Tier 2.

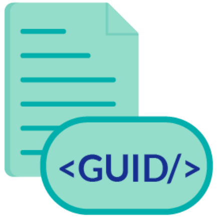
Part 2: TCC and ElementXML

TrakCare Change Control (TCC)

- TrakCare Change Control (TCC) handles import/export of TrakCare configuration.
- TrakCare made of >1000 persistent classes storing configuration.
  - User.*
    - Older classes; covers code tables and transactional data.
  - epr.*
    - Clincial data configuration; security groups; worklist and epr configuration.
  - websys.*
    - System data; screen layouts and translations; workflows; reports.

TrakCare Change Control (TCC) (cont.)

- Properties of these classes correspond to fields on page.
  - Some classes have 100+ properties.
- TCC provides standardised export format for these properties.



TrakCare Primary Keys

- Every configuration item has local ID.
  - Uniquely identifies that configuration item within Environment.
  - Also known as primary key.
  - Often sequential integer based on order in which configuration created.
- Not guaranteed unique across all Environments

PAC.Allergy
Primary Key: 1
Description: No Known Allergies…

PAC.Allergy
Primary Key: 2
Description: Pollen Allergen…

Globally Unique Identifiers (GUIDs)

- Uniquely identifies configuration items across Environments.
- Derived from timestamp when configuration created.
  - Not based on row id or TrakCare data.
  - Does not change when object updated.
- Carried through from BASE to TEST to LIVE when adding new object.
- `<User.INCStkBin GUID="EF0832ED-770F-4AF2-8FB9-9DF8F6798005">.`

GUID Example

BASE

Allergy
ID: 1
Description: No Known Allergies

Table	Row	GUID
User.PACAllergy	1	ABC1234

TEST

Allergy
ID: 20
Description: No Known Allergies

Table	Row	GUID
User.PACAllergy	20	ABC1234

New Configuration GUID Behavior

- Create configuration in BASE.
  - GUID assigned and associated with primary key within BASE environment.
  - Object exported for CCR Transport includes GUID (but no primary key) in XML.
- Deploy ItemSet to TEST.
  - Item deployed contains GUID.
  - TrakCare assigns primary key (may be different than BASE).
  - Primary key associated to that GUID within TEST environment.

Editing Configuration GUID Behavior

- Update configuration in BASE.
  - On export, TCC looks up GUID associated with that primary key.
  - Object exported for CCR Transport includes GUID (but no primary key) in XML.
- Deploy ItemSet to TEST.
  - Item deployed contains GUID.
  - TCC looks up primary key for that GUID.
  - Updates object based on primary key.

GUIDs: Impact on CCR Users

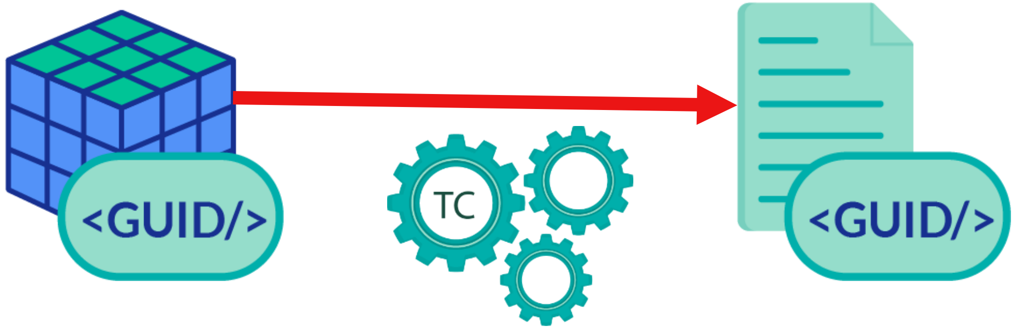
- GUIDs shown for configuration items in CCR UI.
- Some error messages reference GUID of configuration item.
  - Can then lookup GUID within TrakCare to find item in question.

ElementXML

- Export format used by TCC.
- XML representation of TrakCare code table and config data.
- All configuration items in TrakCare have an ElementXML representation.
- These may get complex.
  - Nested code tables.
  - References between configuration.
- Each XML file is one row in a code table.
  - May contain ”child” (dependent) rows.

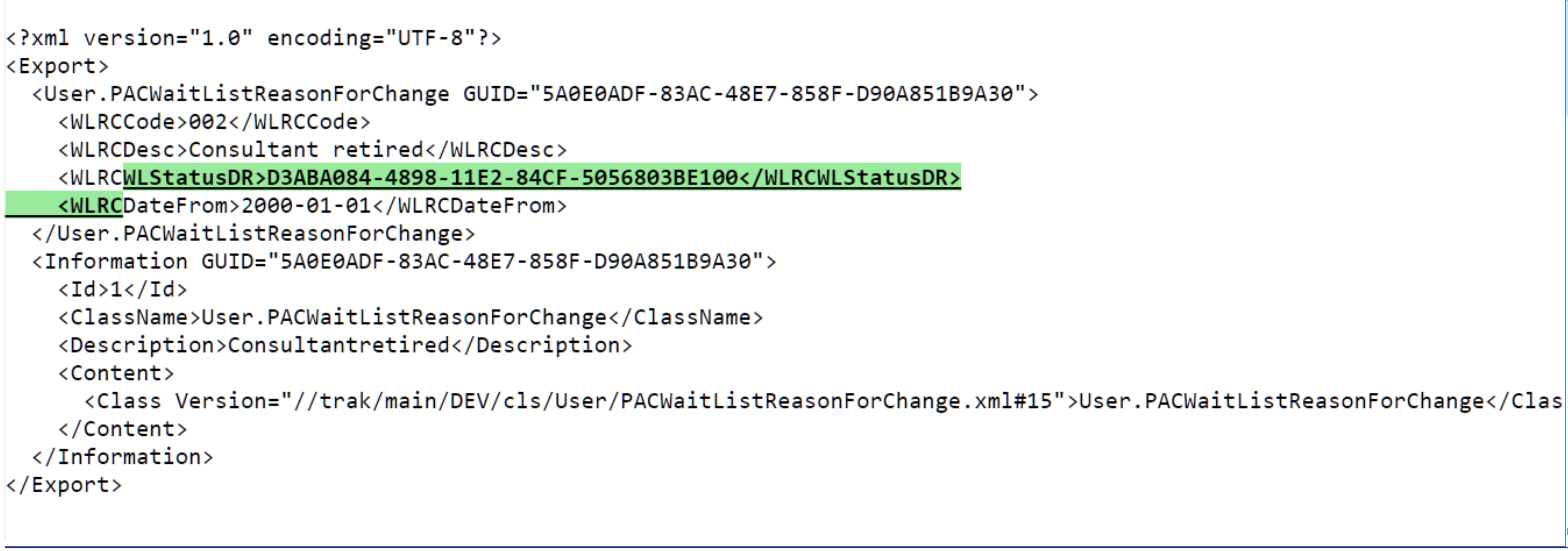
ElementXML and GUIDs

- Every piece of exportable configuration must have a GUID assigned.
  - GUIDs must be consistent across **all** environments in a **System**.
  - File name is GUID.
- ElementXML document imported into TEST/LIVE creates 1+ rows of TrakCare configuration.

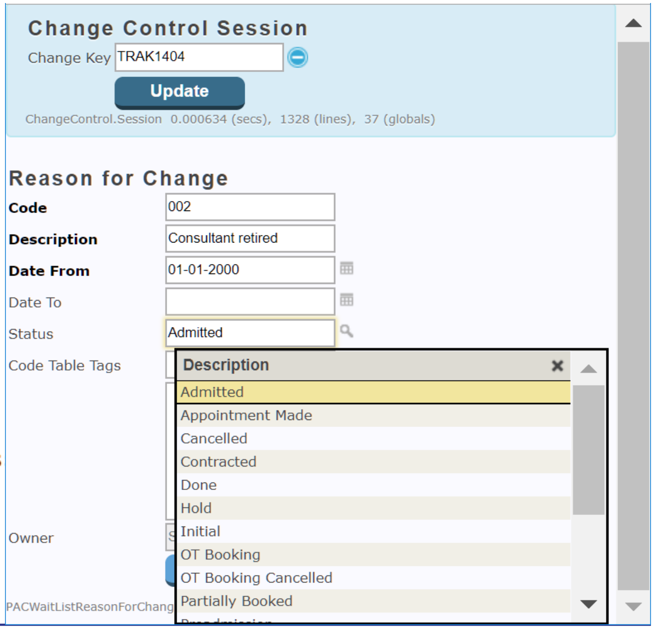


ElementXML Example

- Some code tables reference other tables.
- Use GUID in reference.
- Example: Reason for Change references Status.



```
<?xml version="1.0" encoding="UTF-8"?>
<Export>
  <User.PACWaitListReasonForChange GUID="5A0E0ADF-83AC-48E7-858F-D90A851B9A30">
    <WLRCCode>002</WLRCCode>
    <WLRCDesc>Consultant retired</WLRCDesc>
    <WLRCWLStatusDR>D3ABA084-4898-11E2-84CF-5056803BE100</WLRCWLStatusDR>
    <WLRCDateFrom>2000-01-01</WLRCDateFrom>
  </User.PACWaitListReasonForChange>
  <Information GUID="5A0E0ADF-83AC-48E7-858F-D90A851B9A30">
    <Id>1</Id>
    <ClassName>User.PACWaitListReasonForChange</ClassName>
    <Description>Consultantretired</Description>
    <Content>
      <Class Version="//trak/main/DEV/cls/User/PACWaitListReasonForChange.xml#15">User.PACWaitListReasonForChange</Clas
    </Content>
  </Information>
</Export>
```



Change Control Session

Change Key: TRAK1404

ChangeControl.Session 0.000634 (secs), 1328 (lines), 37 (globals)

Reason for Change

Code	002
Description	Consultant retired
Date From	01-01-2000
Date To	
Status	Admitted
Code Table Tags	
Owner	

Description
Admitted
Appointment Made
Cancelled
Contracted
Done
Hold
Initial
OT Booking
OT Booking Cancelled
Partially Booked

PACWaitListReasonForChang

Quiz: TCC

Question:

What does ‘TCC’ stand for?

A. Transporting Content via CCR.

B. TrakCare Configuration Changes.

C. Transport Change Controls.

D. TrakCare Change Control.

Quiz: TCC (cont.)

Answer:

D. TrakCare Change Control.

TCC is part of the TrakCare product and should be understood as a TrakCare feature. Issues with its functionality should follow normal reporting and escalation process for the TrakCare product.

Quiz: ElementXML

Question:

How does ElementXML relate to TCC?

A. ElementXML was a legacy feature replaced by TCC.

B. ElementXML is the XML export format used by TCC.

C. ElementXML is not related to TCC.

Quiz: ElementXML (cont.)

Answer:

B. ElementXML is the XML export format used by TCC.

ElementXML controls export of TrakCare configuration, and handles logic related to import. Historically, ‘ElementXML’ was used to refer to anything in TrakCare dealing with change control, but now ElementXML just refers to import/export mechanisms of TCC.

Quiz: Identifiers

Question:

What acts as a unique configuration identifier between different TrakCare instances?

A. OID.

B. ID.

C. TCC-ID.

D. GUID.

E. Code and Description.

Quiz: Identifiers (cont.)

Answer:

D. GUID.

GUIDs are unique between instances and are the critical piece for movement of TrakCare configuration between environments.

Areas for Further Study for CCR Tier 2

- ICC610 – Usage Basics.
- ICC615 – Debugging Basics.
- ICC620 – Baselining TrakCare Configuration.
- ICC630 – Debugging and Advanced Topics.

Summary

- What are the key points for this module?

ICC 610: CCR Tier 2 – Usage Basics

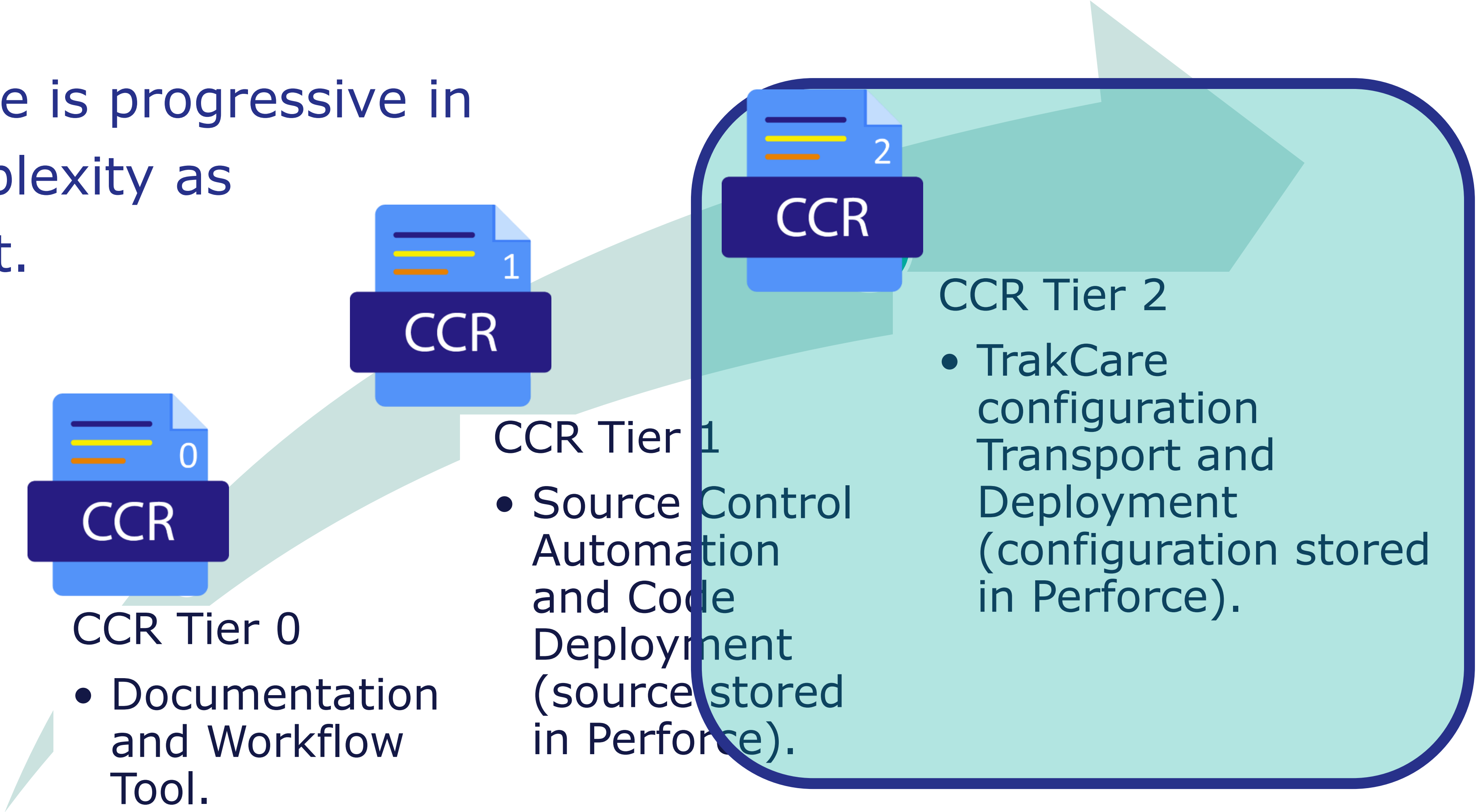
InterSystems
Creative data technology

Objectives

- Configure a TrakCare environment for change control.
- Create Tier 2 CCR.
- Progress basic Tier 2 CCR through BASE, TEST, LIVE workflow.

Review: CCR Usage Tiers

- Tiered usage is progressive in terms of complexity as well as benefit.



CCR Tier 0

- Documentation and Workflow Tool.

CCR Tier 1

- Source Control Automation and Code Deployment (source stored in Perforce).

CCR Tier 2

- TrakCare configuration Transport and Deployment (configuration stored in Perforce).

This series of courses focuses on CCR Tier 2

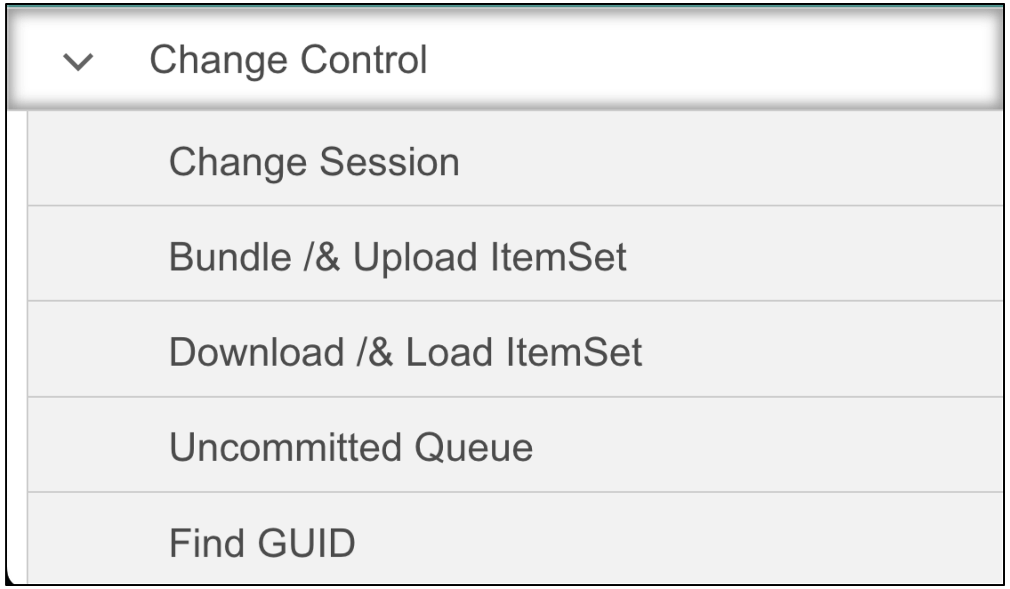
CCR & TrakCare (CCR Tier 2)

- Can use CCR to track ALL TrakCare changes, such as:
  - Layouts.
  - Security groups.
  - Code table changes.
  - Interface Workbench settings.
  - Order favorites.
- General rule:
  - Can export anything touched to change configuration.
    - Care providers.
  - Cannot (or should not) export transactional data.
    - Patient data.

Part 1: Change Control Configuration

Change Control Submenu

- CCR functionality available using Change Control submenu.
  - Tools > Change Control.
- Depending on privileges, menu links to:
  - Change Session.
  - Bundle & Upload ItemSet.
  - Download & Load ItemSet.
  - Recreate ItemSet.
  - Find GUID.
  - Configure.



TrakCare Exclusion Lists

- Exclusion lists disable change control for specific classes.
  - Tier 2 CCRs will NEVER be used for these items.
  - Need to change outside of CCR process.
  - Examples may include clinics, users, and local printers.
  - Strict CCR procedure would not permit timely and efficient changes.
- Exclusion list specific to each TrakCare environment.
- Recommend using Tier 0 CCR and standard changes to put control on changing excluded items.

TrakCare Exclusion Lists (cont.)

- No class should be added to the exclusion list without careful consideration being given to its dependencies.
  - Consult experts when considering changes to exclusion list.

TrakCare Exclusion Lists (cont.)

- Why do we want to do this?
  - Some tables contain generated data such as keywords.
    - Generated data should never be in source control.
  - Some classes have environment specific configuration such as printers.
    - CCR process depends on identical environments for changes progressed by a CCR.
  - Development may exclude some tables that should never be included in source control.

Possible TrakCare Exclusion List

- Please contact InterSystems for the most up to date exclusion list.

User.ARCItemKeywords	User.CTZip	User.PACNonGovOrg
epr.CannedText	User.CTZipKeywords	User.PACNonGovOrgKeywords
User.ARCItemKeywords	User.INCSysCounter	User.PACRefDoctorClinic
User.CFSM	User.LBCSystemCounter	User.PACRefDoctorKeywords
User.CFSystemFileDefinition	User.LBCTestSetRevisionReagent	User.PACWardAvailRestriction
User.CFSystemFilePath	User.MHCAgent	User.RBResource
User.CTCareProv (Maybe)	User.MRCICDDx	User.SSHL7
User.CTCity	User.MRCICDkeyw	User.SSUser
User.CTCityHCA	User.ORCOperation	websys.Configuration
User.CTHealthCareArea	User.PACClinic	websys.Print
User.CTHealthCareRegion	User.PACCounterType	websys.Printer
User.CTProvince	User.PACInstrText	websys.PrintService

Possible Pharmacy Exclusion List

- Please contact InterSystems for the most up to date exclusion list.

User.INCItm	User.INCItmLcDailyTotal	User.INCSterileCategory
User.INCItmBat	User.INCItmLoc	User.INCSterileItems
User.INCItmHosp	User.INCItmRcp	User.INCSterilePriority
User.INCItmHospVen	User.INCItmRecIngr	User.INCStorageBinGroup
User.INCItmHospVenCommItems	User.INCItmStockLocations	User.INCStorageBinType
User.INCItmHospVenDiscount	User.INCItmVen	User.INCStorageBinTypeStockItem
User.INCItmLcBt	User.INCItmVenCommItems	User.INCSysCounter
User.INCItmLcBtBin	User.INCItmVenDiscount	

Part 2: Change Session Usage

Change Sessions

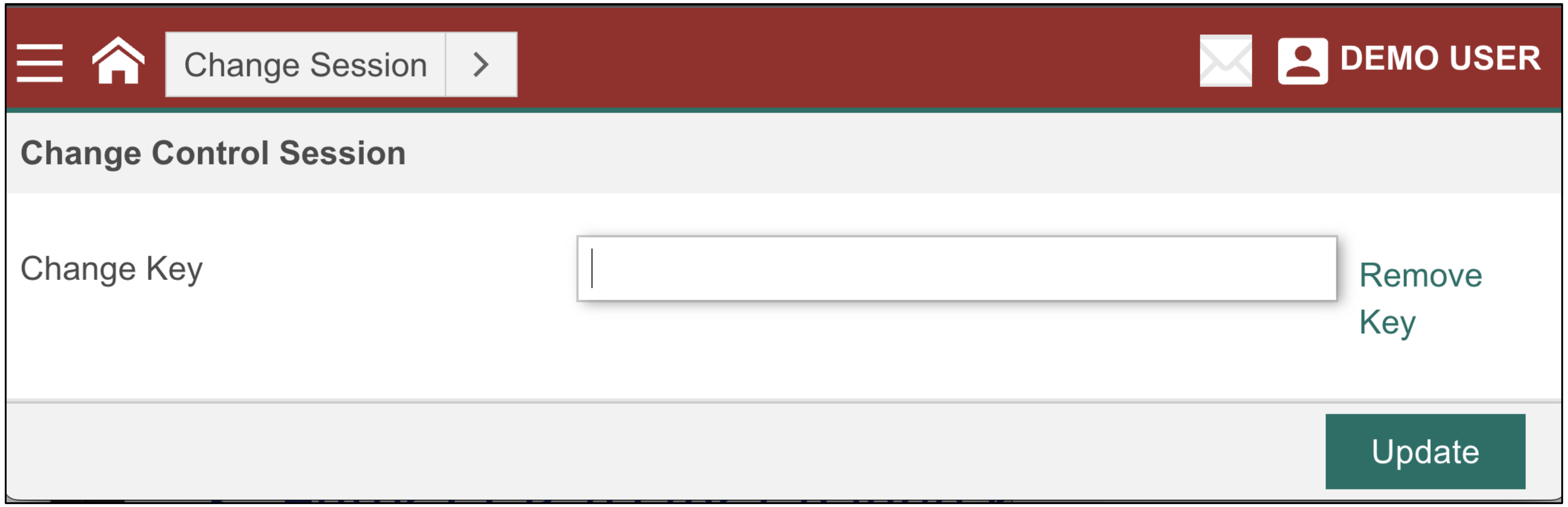
- Use change session to mark changed items with CCR ID to simplify change identification.
- Start a change session prior to making any configuration changes in TrakCare.
- Change session terminates:
  - When you log out of TrakCare.
  - After upload changes to Perforce (optional flag).
  - When click termination button on Change Session Menu.

Change Sessions (cont.)

- When change session active, all TrakCare Application changes are:
  - Exported via TCC to XML documents.
  - Added to the uncommitted queue.
  - Checked out to current user.

How To: Open a Change Session

1. CCR should already exist and be in In_BASE state.
2. Tools > Change Control > Change Session.
3. Enter CCR ID for Change Key.
4. Click Update.

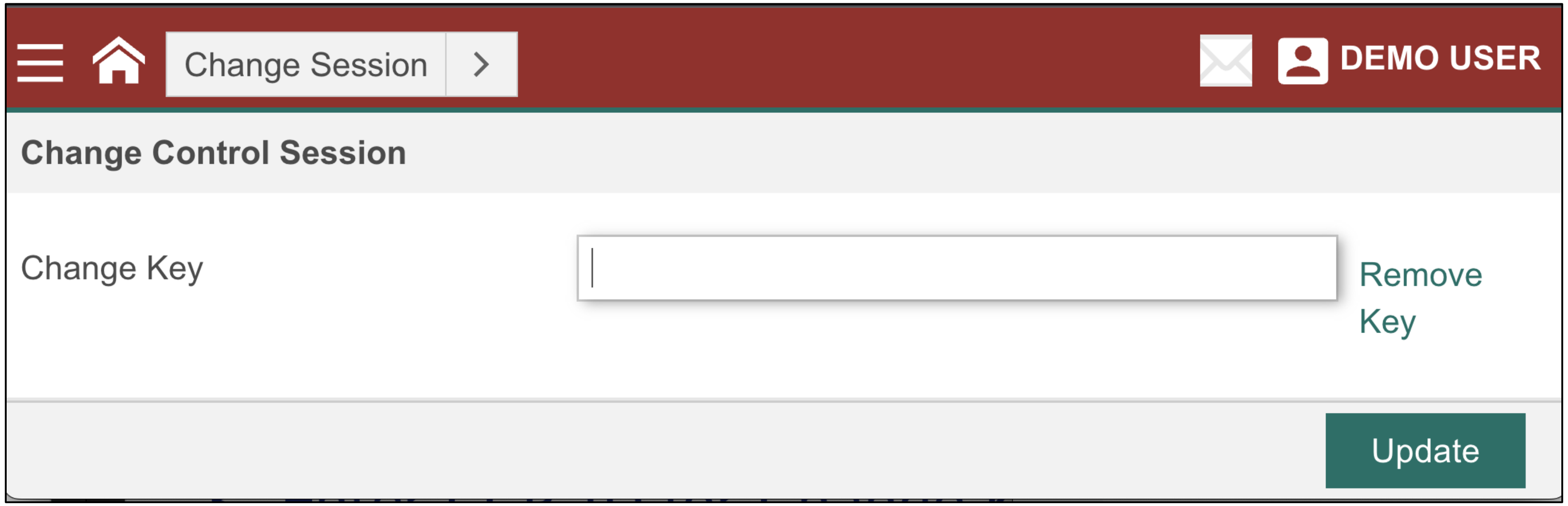


Change Control Session

Change Key

How To: Manually End Change Session

1. Tools > Change Control > Change Session.
2. Click Remove Key.



Change Session

DEMO USER

Change Control Session

Change Key

Exercise ICC610-1 through ICC610-2

Create a CCR and open a change session

Quiz: Change Session

Question:

What is the purpose of a change session? Select all that apply.

A. To ensure that a CCR has been created for the configuration that is taking place.

B. To prevent accidental changes from being made in the environment that can’t be accounted for.

C. To automate the tracking of changes made for a specific CCR and make it easier to upload those changes to CCR once the work is complete.

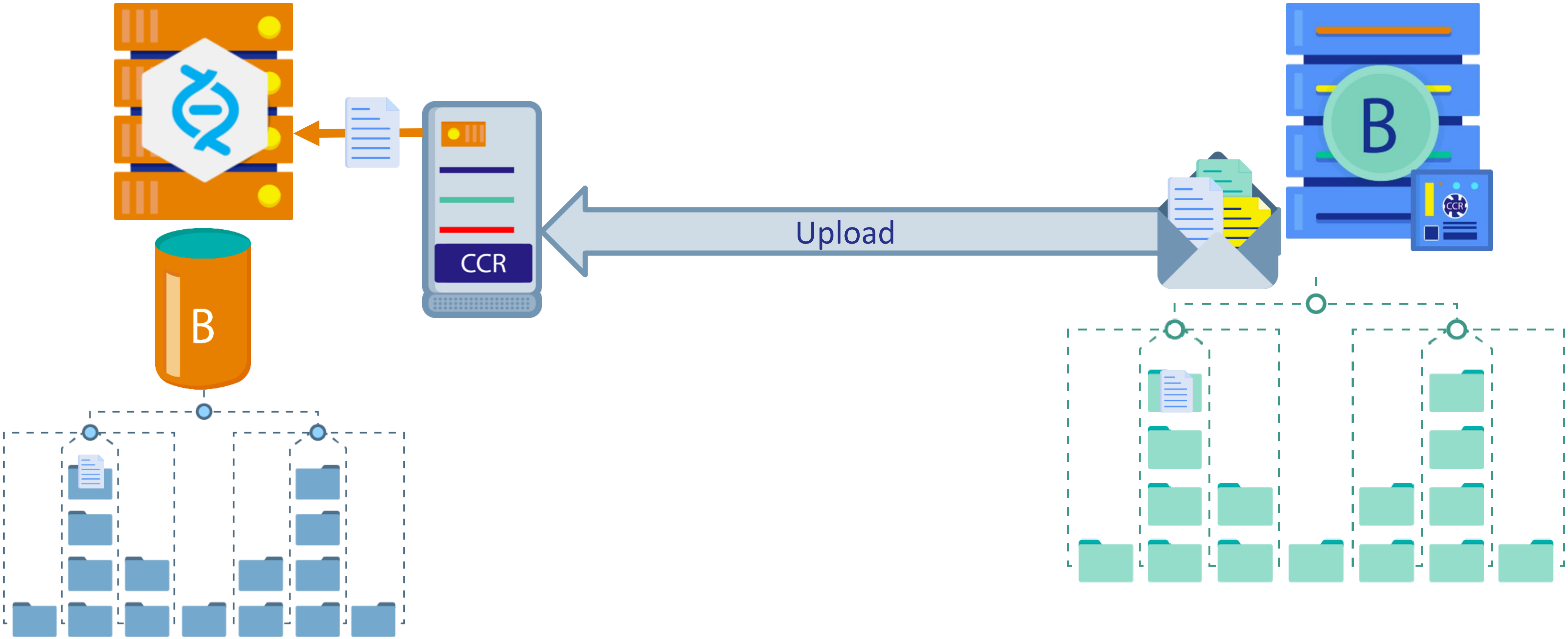
Quiz: Change Session

Answer:

All of the above.

Part 3: TrakCare Bundle & Upload ItemSet

Review: Sending Items from **Disconnected** BASE

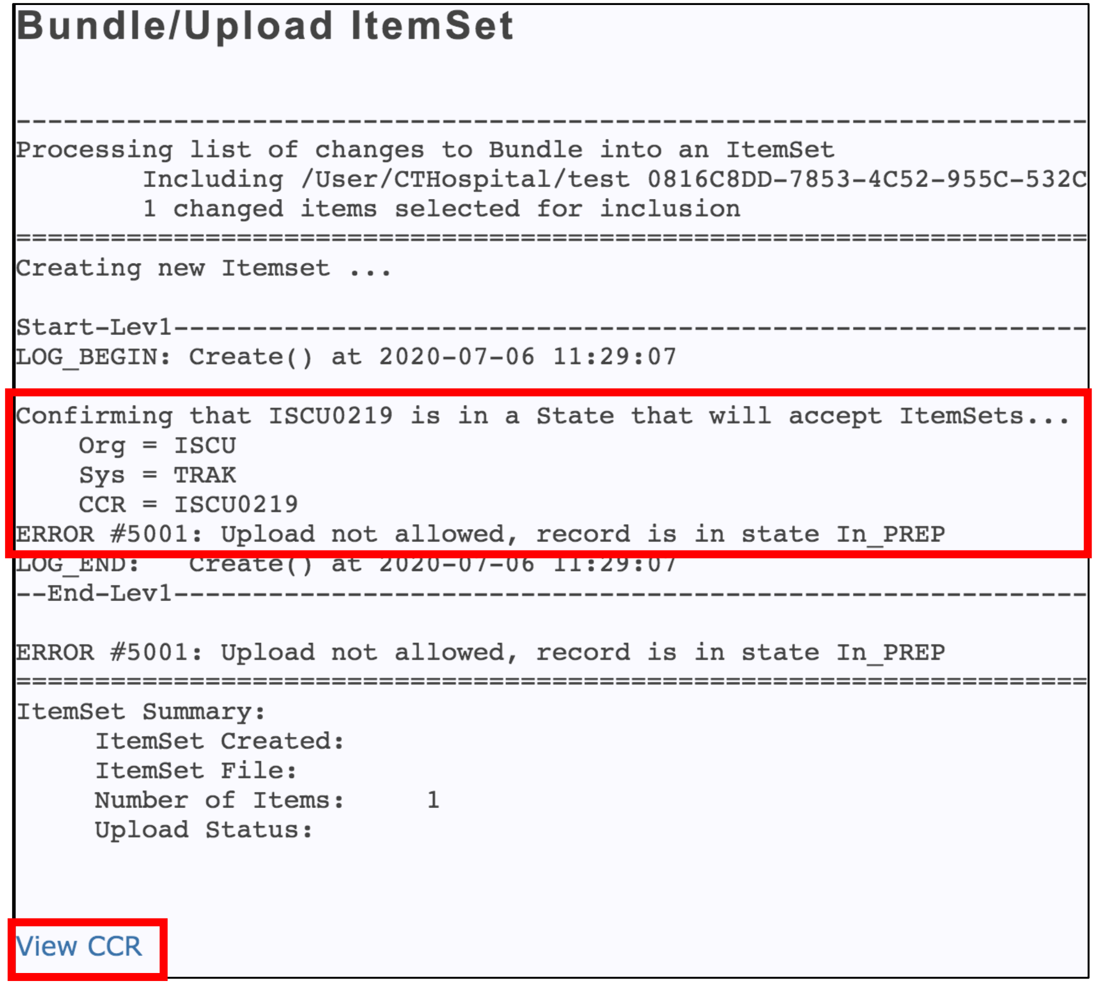


How To: Bundle & Upload ItemSet

- Generate and copy access token.
  - From Perforce Details section of your CCR, click Generate Token > copy.
- Complete Bundle and Upload form in TrakCare:
  - Tools > Change Control > Bundle & Upload ItemSet.
  - Select changed items.
  - Enter CCR ID.
  - Paste access token.
  - Select End Change Session.
- Click Upload.

Bundle & Upload ItemSet Restrictions (Client tools 2019-Dec-12)

- CCR validates ItemSet against following restrictions.
  - CCR state must either:
    - In_BASE.
    - BASE_Pending_Peer_Review.
  - System for CCR record and TrakCare environment must match.
- Returns error after clicking upload button and ItemSet fails to upload.
- Solve issue and redo bundle and upload process.

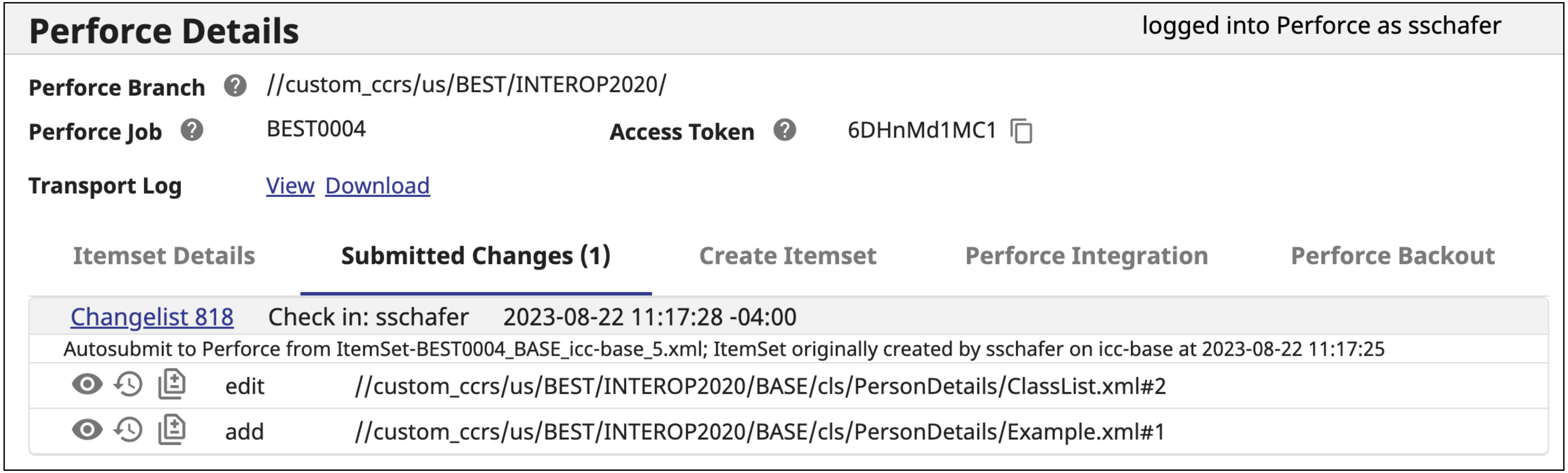


Bundle/Upload ItemSet

```
-----------------------------------------------------------------------
Processing list of changes to Bundle into an ItemSet
        Including /User/CTHospital/test 0816C8DD-7853-4C52-955C-532C
        1 changed items selected for inclusion
=======================================================================
Creating new Itemset ...

Start-Lev1-------------------------------------------------------------
LOG_BEGIN: Create() at 2020-07-06 11:29:07

Confirming that ISCU0219 is in a State that will accept ItemSets...
    Org = ISCU
    Sys = TRAK
    CCR = ISCU0219
ERROR #5001: Upload not allowed, record is in state In_PREP
LOG_END:   Create() at 2020-07-06 11:29:07
--End-Lev1-------------------------------------------------------------

ERROR #5001: Upload not allowed, record is in state In_PREP
=======================================================================
ItemSet Summary:
     ItemSet Created:
     ItemSet File:
     Number of Items:     1
     Upload Status:
```

Verify Submitted Changes

- After submit changes, always verify correct items listed in CCR.
  - Under Perforce Details of CCR, click Submitted Changes tab.



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

Quiz: Bundle & Upload ItemSet

Question:

What happens when items are bundled and uploaded? Select all that apply.

A. They are removed from the Uncommitted Queue.

B. Changes are integrated to the TEST branch of Perforce.

C. They become available for others to edit.

D. They are put into an ItemSet and sent to the CCR server.

E. The CCR server unpacks them from the ItemSet and checks them into Perforce.

Quiz: Bundle & Upload ItemSet (cont.)

Answer:

A. They are removed from the Uncommitted Queue.

C. They become available for others to edit.

D. They are put into an ItemSet and sent to the CCR server.

E. The CCR server unpacks them from the ItemSet and checks them into Perforce.

Note that E only occurs if a Perforce user does the upload; otherwise, the ItemSet sits on the CCR server in an uncommitted queue and are checked into Perforce by the next upload performed by a Perforce user.

Quiz: Bundle & Upload ItemSet

Question:

Different types of changes cannot be bundled into the same ItemSet. Different ItemSets must be used for each change type (config, layout, code tables, questionnaires, etc).

True or False?

Answer:

False. ItemSets can contain any type of changes desired (even Tier 1 and Tier 2 changes at the same time).

Quiz: Bundle & Upload ItemSet

Question:

The access token requested by the upload screen is obtained from Support. True or False?

Answer:

False. Access tokens are retrieved from the CCR to which the changes are to be uploaded via the [generate token] link in the Perforce Details pane.

Part 4: Predicting GUID Loads

Review: GUIDs

- Globally Unique Identifiers.
- Central to TCC’s ability to uniquely identify configuration items across different environments.
- Derived from current system time on server when new object is created.
- Does not change when object updated.
- Not based on row id.

Review: GUIDs (cont.)

- Not based on TrakCare data.
- Random, unique and system independent.
- “Fingerprint” or address of the object.
- `<User.INCStkBin GUID="EF0832ED-770F-4AF2-8FB9-9DF8F6798005">`

GUIDs and Elements

- Assigned to TrakCare elements.
  - Code Table entries.
  - Configuration.
  - New objects are assigned a GUID on creation.
  - Not patient or operational Data.
    - Only config data can be associated with a change session.
    - Patient and operational data can not be moved by CCR.

Quiz: GUIDs and Elements

Question:

When are GUIDs assigned to new objects?

A. At creation time.

B. After the initial edit under a change session.

C. After every edit.

Answer:

A. At creation time.

Quiz: Operational and Patient Data

Question:

ElementXML is never used to move operational or patient data

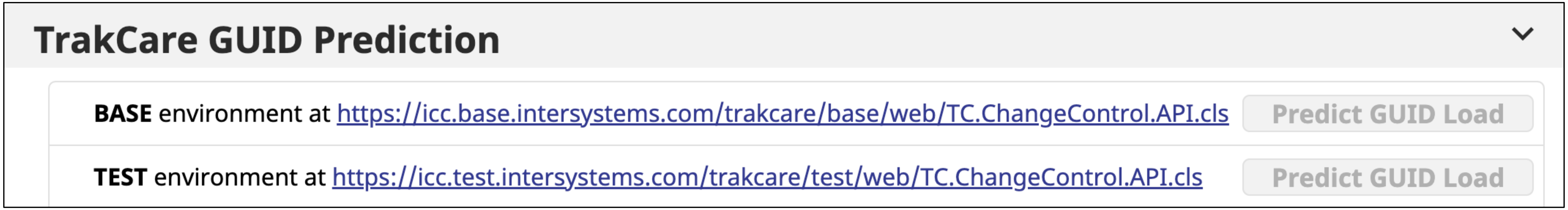
True or False?

Answer:

True. Transactional data (including operational or patient data) is not managed by TCC and should never be under change control (if it is, then it is a bug which should be reported).

GUID Predictions

- Verifies necessary GUIDs exist in target environment.
- TrakCare GUID Prediction section of CCR displayed when:
  - At least 1 item submitted.
  - At least 1 GUID service reachable from browser.
- Always predict GUID loads before performing authorizeAndStartMoveToXXXX



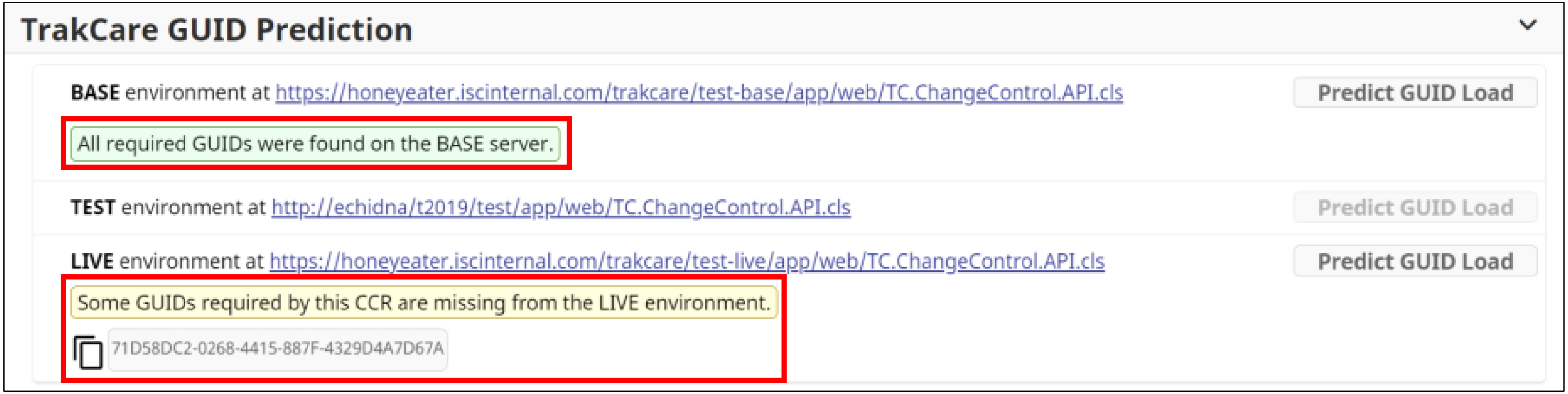
TrakCare GUID Prediction

BASE environment at https://icc.base.intersystems.com/trakcare/base/web/TC.ChangeControl.API.cls

TEST environment at https://icc.test.intersystems.com/trakcare/test/web/TC.ChangeControl.API.cls

GUID Prediction Messages

- See ICC615 for how to resolve missing GUIDs.



TrakCare GUID Prediction

BASE environment at https://honeyeater.iscinternal.com/trakcare/test-base/app/web/TC.ChangeControl.API.cls

All required GUIDs were found on the BASE server.

TEST environment at http://echidna/t2019/test/app/web/TC.ChangeControl.API.cls

LIVE environment at https://honeyeater.iscinternal.com/trakcare/test-live/app/web/TC.ChangeControl.API.cls

Some GUIDs required by this CCR are missing from the LIVE environment.

71D58DC2-0268-4415-887F-4329D4A7D67A

Quiz: GUID Predictions

Question:

How can a user determine whether missing GUID errors are likely to occur during an ItemSet load into the TEST environment?

A. Upon transitioning into BASE_Complete state, CCR will perform a preview check for missing GUIDs in TEST.

B. The user must click the Predict GUID Load button for TEST after uploading all changes to their CCR.

C. After the ItemSet has been created, the user must click Predict ItemSet Load to see any anticipated errors exist.

Quiz: GUID Predictions (cont.)

Answer:

B. The user must click the Predict GUID Load button for TEST after uploading all changes to their CCR.

A is wrong because CCR only checks for merge conflicts.

C is wrong because a Predict ItemSet Load button does not exist.

Exercise ICC610-3

Upload Changes; Run GUID Predictions; Peer Review

Part 5: TrakCare ItemSet Deployment

Deploy ItemSet to TrakCare

- 3 options:
  - Deploy links from CCR.
    - Easiest, least error prone, and most common.
  - Download and from TrakCare environment UI.
  - Download from TrakCare Terminal.

How To: Use CCR ItemSet Deploy Links

1. Log into TrakCare target environment.
2. Click deploy to XXXX link.

How To: Download & Load Itemset Via TrakCare

1. Open your CCR.
2. Ensure there is an ItemSet ready to download and marked as undeployed.
3. Generate and copy access token.

How To: Download & Load Itemset Via TrakCare (cont.)

4. Log into target TrakCare environment.
5. Go to Tools > Change Control > Download & Load ItemSet.
6. Enter CCR details.

- CCR number.
- Access token.

7. Click List ItemSets.
8. Select the Itemset and click Download.
    - All ItemSets for this namespace for this CCR shown.
    - Choose ItemSet with most recent TimeStamp if multiple.

Exercise ICC610-4 & ICC610-5

Downloading changes and Finishing off

Quiz: Download & Load ItemSet

Question:

Into which environment would you typically download ItemSets if the environment exists? Select all that apply.

A. BASE.

B. TEST.

C. UAT.

D. LIVE.

Quiz: Download & Load ItemSet (cont.)

Answer:

B. TEST.

C. UAT.

D. LIVE.

Part 6: CCR Best Practice Guidelines

CCR Best Practice Guidelines

- Never disable change control once baseline complete.
- Use separate CCR disabled environment as playground for experimental changes.
  - Use BASE for changes that you intend to progress.

Summary

- What are the key points for this module?

ICC 615: CCR Tier 2 – Debugging Basics

InterSystems
Creative data technology

Objectives

- Undo changes.
- Cancel Tier 2 CCR.
- Anatomy of GUIDs and ElementXML.
- Introduction to debugging Tier 2 CCRs.
- Understand meaning and resolution for:
  - Uploading to incorrect system.
  - Missing GUID.
  - GUID Conflict.

Objectives (cont.)

- Recreate ItemSet for server.
- iService guidelines.

Part 1: Undo Change Not Bundled and Uploaded

Undo a Change Not Bundled and Uploaded

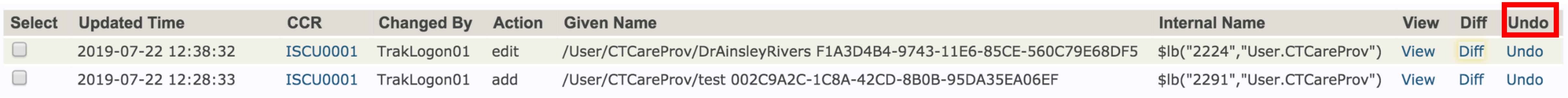
- Useful for unintentional changes in BASE.
- Irreversible actions.
- Undo will:
  - Revert all changes to affected item.
    - Added items are deleted.
  - Remove item from uncommitted queue.
  - Remove check-out status on item.

Undo a Change Not Bundled and Uploaded (cont.)

- Careful – may cause broken pointers. For example:
  - Add 2 items where item A references item B.
  - Perform undo on item B.
  - Item A still references item B and the undo on B might not throw an error.
  - Solution: make dummy change to item A to force re-export.

How To: Undo a Change Not Bundled and Uploaded

1. If not in active change session, start change session.
2. Go to Tools > Change Control > Bundle & Upload ItemSet.
3. Click Undo for appropriate Item.
4. Click OK to confirm Undo.



Select	Updated Time	CCR	Changed By	Action	Given Name	Internal Name	View	Diff	Undo
☐	2019-07-22 12:38:32	ISCU0001	TrakLogon01	edit	/User/CTCareProv/DrAinsleyRivers F1A3D4B4-9743-11E6-85CE-560C79E68DF5	$lb("2224","User.CTCareProv")	View	Diff	Undo
☐	2019-07-22 12:28:33	ISCU0001	TrakLogon01	add	/User/CTCareProv/test 002C9A2C-1C8A-42CD-8B0B-95DA35EA06EF	$lb("2291","User.CTCareProv")	View	Diff	Undo

Part 2: Cancel Transition for Tier 2

Review: Note on Cancelling CCRs Properly

- Only cancel CCRs that you no longer wish to have in any environment.
  - For example, started creating a care provider and received noticed that they don’t need to be added.
- Make sure that cancelled CCRs are properly backed out.
  - Backed out of the Perforce branch.
  - Backed out of the environments.

Notes on Backing Out Tier 2 Change

- Verify no add actions under [show Submitted changes] in CCR.
  - New items should be end dated rather than backed out from Perforce.
  - A Perforce backout will cause item deletion!
- If all items in BASE have add action:
  - Backout plan is to end date those added items.
  - Do not use the Perforce backout tools.

Notes on Backing Out Tier 2 Change (cont.)

- If all actions are edit, use Perforce backout tools as described in ICC450.
- If there is a mix of add and edit actions for BASE:
  - Contact InterSystems Support for help.
  - Added items need to be end dated.
  - Edited items need to be backed using P4V (an InterSystems task).

Part 3: GUIDs and ElementXML

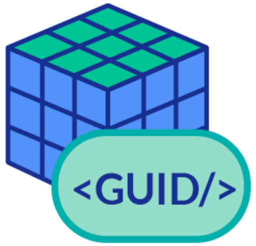
Review: GUIDs and Elements

- Assigned to TrakCare elements.
  - Code Table entries.
  - Configuration.
  - New objects are assigned a GUID on creation.
  - Not patient or operational data.
    - Only config data can be associated with a change session.
    - Patient and operational data can not be moved by CCR.

GUIDs and ElementXML

- GUID form basis of ElementXML export filenames.
- ElementXML files use GUIDs, not embedded row IDs.
- Cannot use row IDs because they may differ between environments.
- Cannot use Code or Desc because those values may be subject to change.
- Deleting and recreating a configuration item creates new object with new GUID.

Anatomy of a GUID



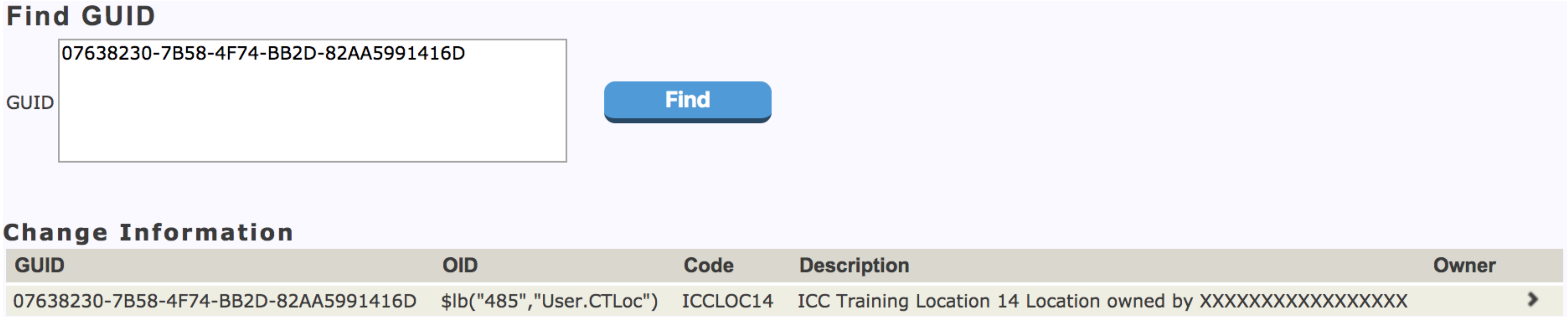
- Assigned on object creation.
  - format 09290496-0BAD-11E2-9090-F403C32D8A93.
- Queryable with the %GUID SQL property.
- No relation to content of TrakCare configuration.
- Composed of 2 data elements; ^OBJ.GUID is split into 2 branches.
  - 0 subscript maps GUID to OID (Object Identity).
  - 1 subscript maps OID to GUID.
  - These should always match. Mismatches can cause strange behaviour.

GUID Integrity (cont.)

- Question: If I deleted ^OBJ.GUID, what effect would a TrakCare user see?
  - No effect to running TrakCare application.
  - Can no longer make any changes through change control.
  - GUIDs only used when importing and exporting config.
  - They are not used for any functional purpose in TrakCare, all references are decoded when loading configuration.

Find GUID

- GUIDs are random; they do not contain any information about configuration.
- Tools > Change Control > Find GUID.
- Relates the GUID back to the configuration item.
- Provides change history for configuration item.



Find GUID

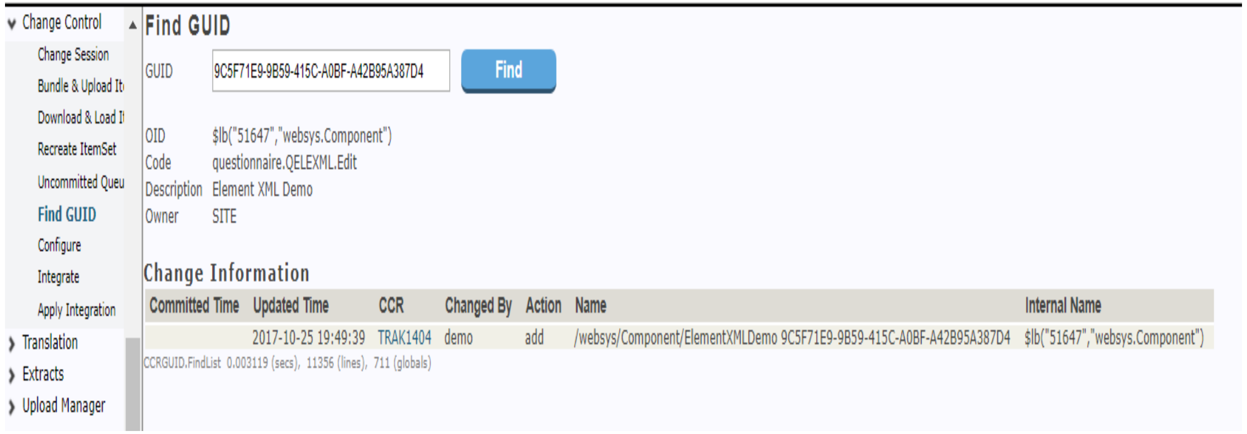
GUID: 07638230-7B58-4F74-BB2D-82AA5991416D

Change Information

GUID	OID	Code	Description	Owner
07638230-7B58-4F74-BB2D-82AA5991416D	$lb("485","User.CTLoc")	ICCLOC14	ICC Training Location 14 Location owned by XXXXXXXXXXXXXXXX	›

Find GUID (cont.)

- This screen will provide all information about the referenced object.
  1. Class and ID (OID).
  2. Code value (where available).
  3. Description.
  4. Owner.
  5. Change history (note no Committed Time on this one).



Find GUID

GUID 9C5F71E9-9B59-415C-A0BF-A42B95A387D4

OID	$lb("51647","websys.Component")
Code	questionnaire.QELEXML.Edit
Description	Element XML Demo
Owner	SITE

Change Information

Committed Time	Updated Time	CCR	Changed By	Action	Name	Internal Name
	2017-10-25 19:49:39	TRAK1404	demo	add	/websys/Component/ElementXMLDemo 9C5F71E9-9B59-415C-A0BF-A42B95A387D4	$lb("51647","websys.Component")

CCRGUID.FindList 0.003119 (secs), 11356 (lines), 711 (globals)

Part 4: Basic Debugging: TCC Errors

“CCR is Broken”

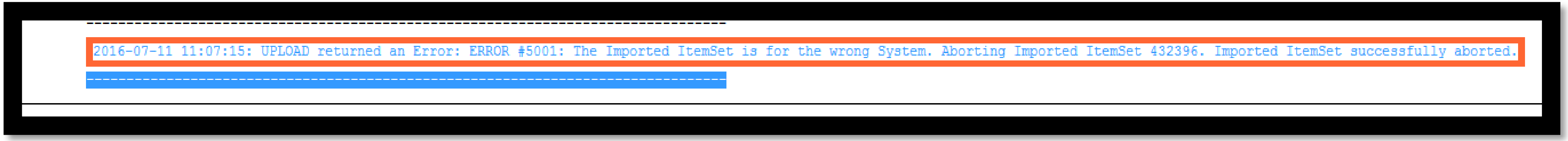
- This is the most common misused phrase in TrakCare Implementation/Support.
- CCR is an exceptional stable codebase.
    - Bug in the CCR server or CCR Client Tools code very rare.
- Change control bugs are almost certainly TrakCare Change Control (TCC) issues.
- Improper use of CCR can lead to unexpected (but not technically erroneous) behaviours.

General Error Notes (cont.)

- Most error messages include how to successfully progress the change.
- If you think you’ve found a CCR bug report it immediately to Support.

Incorrect System Area

- When uploading an Itemset from BASE the below error is displayed.
  - Users may select the incorrect system, for example they have selected T2016 instead of T2018 during an upgrade project.
- Resolution.
  - Create a new CCR with the correct system, change list needs to be moved to the new CCR and the Itemset needs to be re-submitted. It may require you to email CCR users and CCR dev to request for the change lists to be moved across to the new CCR or to make changes to the ItemSet status to enable re-uploading.
    - NOTE: This must be done ASAP as there is risk that future changes will step over the failed upload and therefore configuration could potentially be reverted.



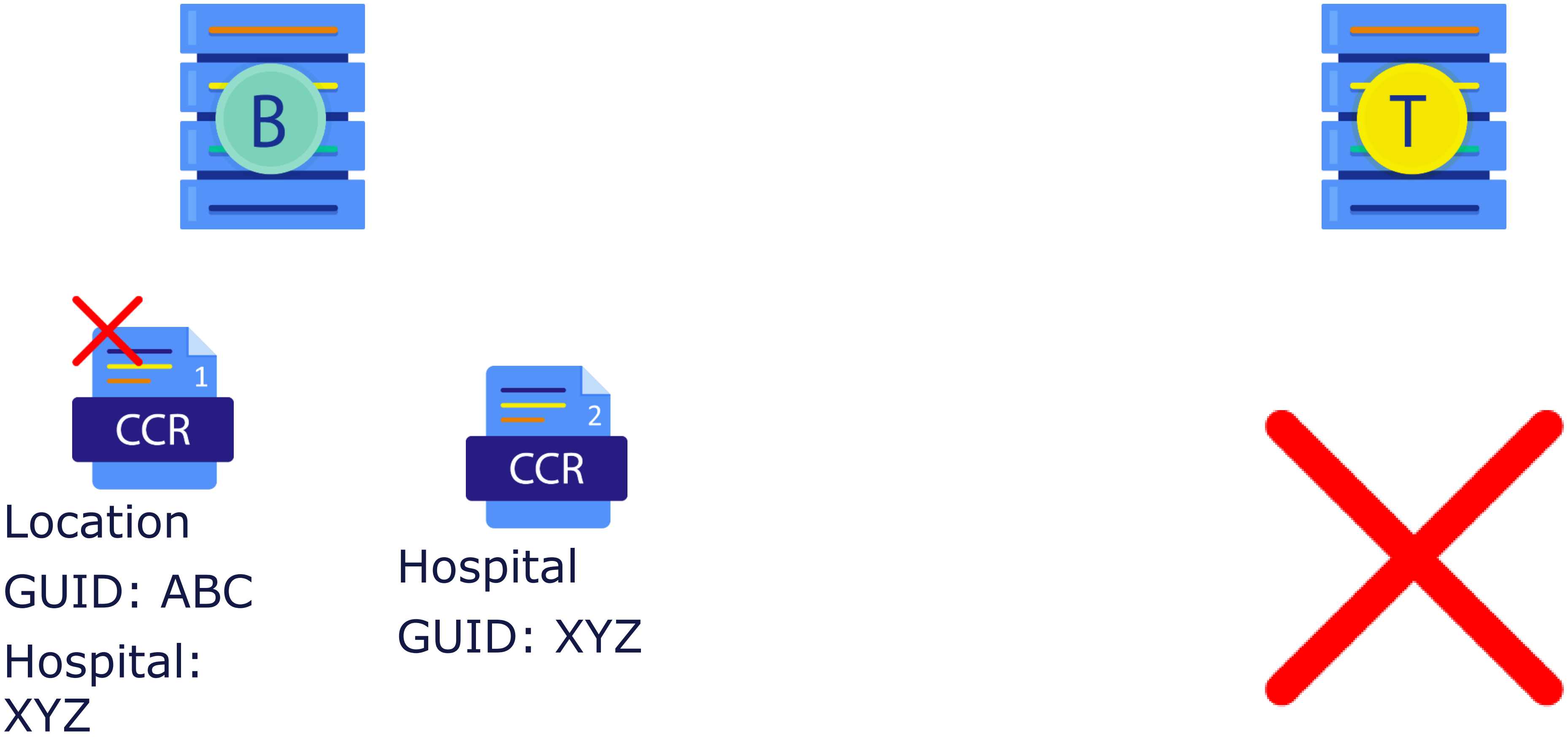
```
--------------------------------------------------------------------------------
2016-07-11 11:07:15: UPLOAD returned an Error: ERROR #5001: The Imported ItemSet is for the wrong System. Aborting Imported ItemSet 432396. Imported ItemSet successfully aborted.
--------------------------------------------------------------------------------
```

Part 5: Missing GUID

Missing GUID

- This error occurs on download of TrakCare configuration.
- TCC will not load incomplete configuration by default.
  - A missing GUID error means that the GUID lookup did not return a match.
  - TCC cannot link to a non-existent object.
- This will cause the load to fail and the whole transaction to roll back.

Missing GUID Example



Location

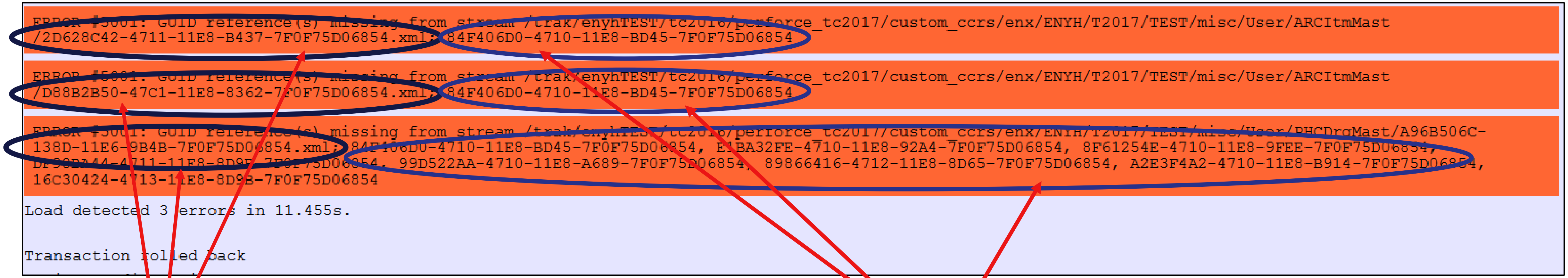
GUID: ABC

Hospital:
XYZ

Hospital

GUID: XYZ

Example Missing GUID Error



```
ERROR #5001: GUID reference(s) missing from stream /trak/enyhTEST/tc2016/perforce_tc2017/custom_ccrs/enx/ENYH/T2017/TEST/misc/User/ARCItmMast
/2D628C42-4711-11E8-B437-7F0F75D06854.xml: 84F406D0-4710-11E8-BD45-7F0F75D06854

ERROR #5001: GUID reference(s) missing from stream /trak/enyhTEST/tc2016/perforce_tc2017/custom_ccrs/enx/ENYH/T2017/TEST/misc/User/ARCItmMast
/D88B2B50-47C1-11E8-8362-7F0F75D06854.xml: 84F406D0-4710-11E8-BD45-7F0F75D06854

ERROR #5001: GUID reference(s) missing from stream /trak/enyhTEST/tc2016/perforce_tc2017/custom_ccrs/enx/ENYH/T2017/TEST/misc/User/PHCDrgMast/A96B506C-
138D-11E6-9B4B-7F0F75D06854.xml: 84F406D0-4710-11E8-BD45-7F0F75D06854, E1BA32FE-4710-11E8-92A4-7F0F75D06854, 8F61254E-4710-11E8-9FEE-7F0F75D06854,
DF98BA44-4711-11E8-8D9B-7F0F75D06854, 99D522AA-4710-11E8-A689-7F0F75D06854, 89866416-4712-11E8-8D65-7F0F75D06854, A2E3F4A2-4710-11E8-B914-7F0F75D06854,
16C30424-4713-11E8-8D9B-7F0F75D06854
Load detected 3 errors in 11.455s.

Transaction rolled back
```

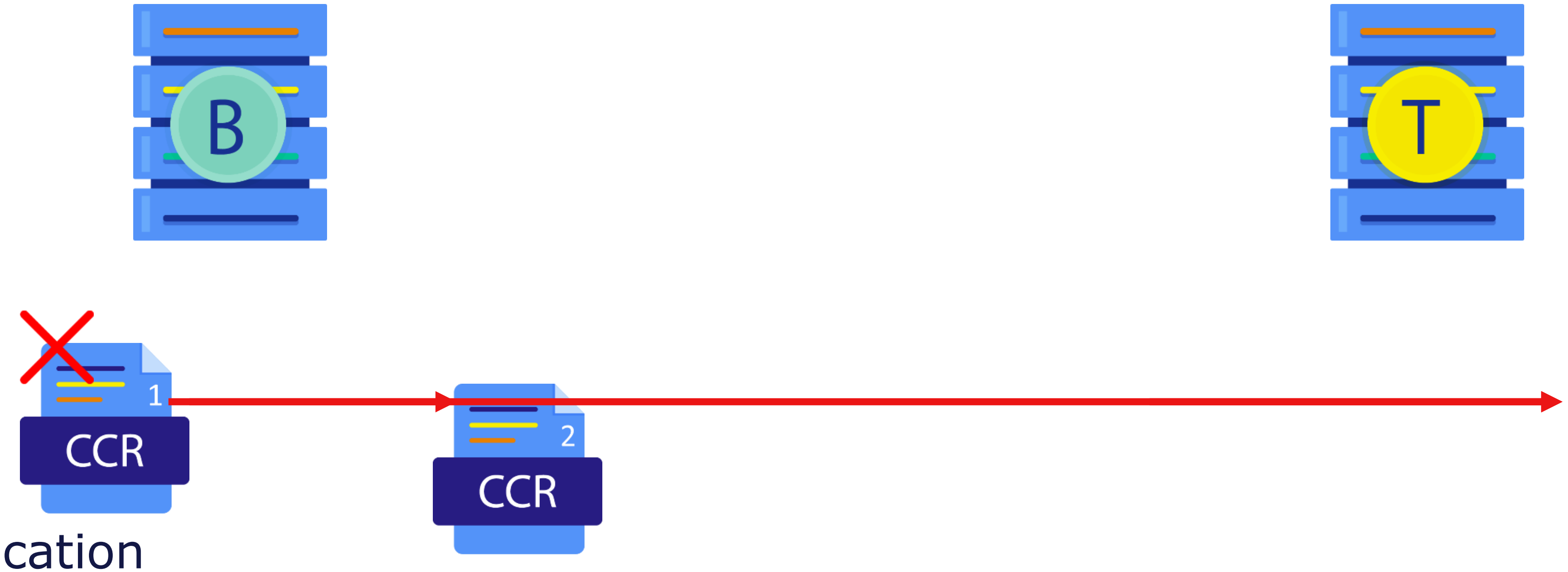
Configuration Files with missing GUID references

GUIDS that are missing from the target environment

Resolve Missing GUID

- Steps below assume BASE-TEST-LIVE workflow and error occurred deploying to TEST.
  1. Use Find GUID in BASE to find object associated with missing GUID.
     - If object missing in TEST added by this CCR, most likely a different error is causing ItemSet load to fail. See Secondary Error discussion later.
     - If object missing in TEST added by different CCR still in BASE phase, continue to step 2.
  2. Specify CCR adding missing object as prerequisite CCR on CCR showing error.
  3. Progress CCR adding missing object to TEST.
  4. Deploy ItemSet that originally errored out.
  5. Progress both CCRs through normal workflow to Closed.

Missing GUID Example: Resolution



Location

GUID: ABC

Hospital:
XYZ

Hospital

GUID: XYZ

Circular Reference

- CCR1 adds configuration that depends on configuration added by CCR2.
- CCR2 adds configuration that depends on configuration added by CCR1.
- Solution: merge the 2 CCRs.
- Avoid situation by:
  - Keeping change small and discrete.
  - Progressing changes quickly.

Secondary Errors

- Missing GUIDs are the easiest warning to resolve.
- If multiple errors, most likely want to resolve missing GUID last.
- Example:
  - If 3 items are loaded in 1 ItemSet:
    - GUID A is OK.
    - GUID B fails with a data validation issue.
    - GUID C fails as it refers to B.
  - If B is resolved, then the missing GUID warning will resolve.
- Remember to check if your GUID is in the current download.

Quiz: Missing GUIDs

Question:

Preview integrations will catch missing GUID errors.

True or False?

Answer:

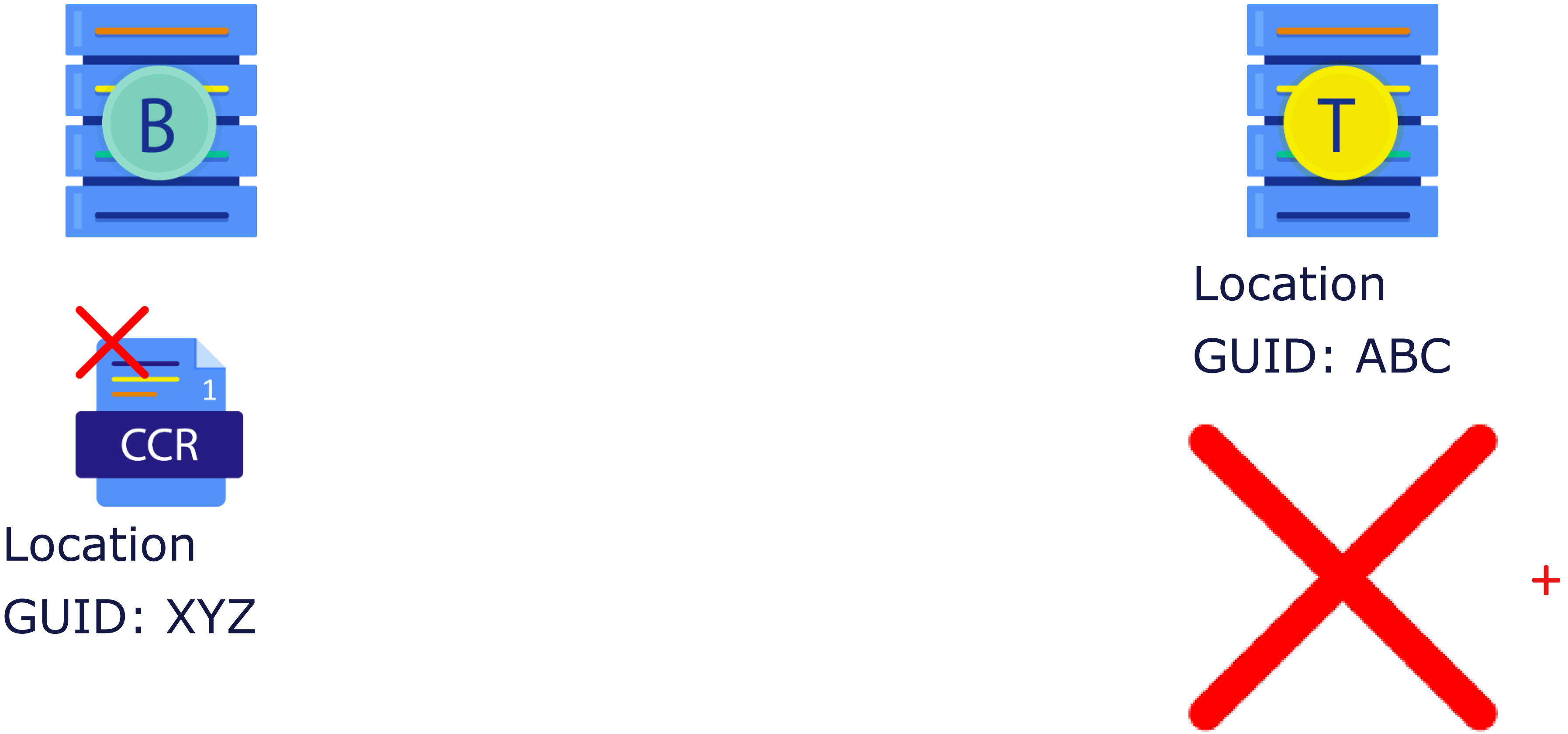
False. Perforce treats files as plain text. Perforce cannot interpret GUIDs.

Part 6: GUID Conflict

GUID Conflict

- The XML Load uses the GUID to determine what to do when loading configuration.
  - GUID found – Update that object with the contents of the XML.
  - GUID not present – Insert a new record.
- GUIDs must be unique over the whole system.
- In most tables, configuration must also be unique based on code and description.
- GUID assigned on creation, and NOT linked to the object content.

GUID Conflict Example



Location

GUID: XYZ

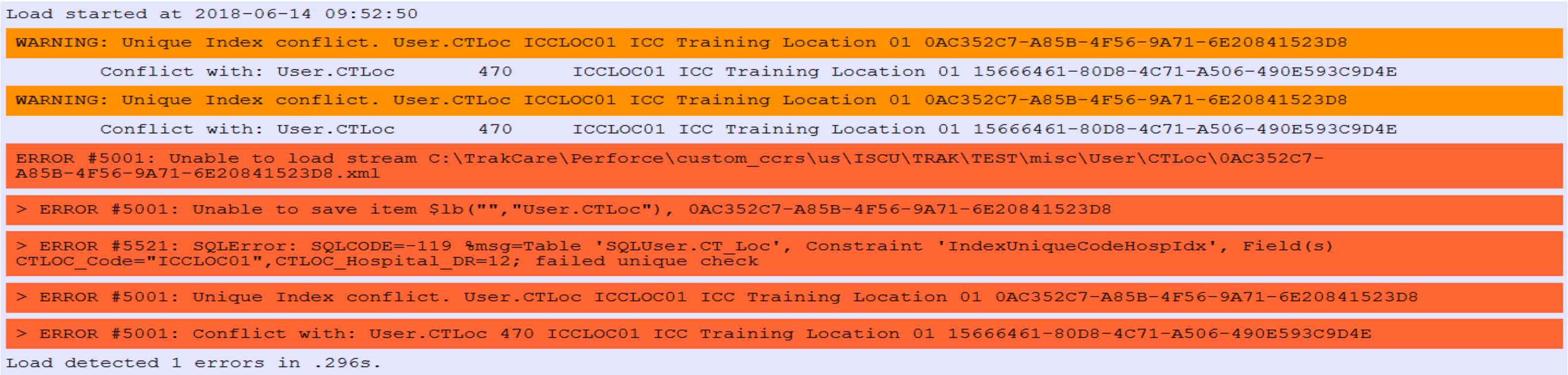
Location

GUID: ABC

Part 7: Wrong GUID

Wrong GUID on Object

- There will be a conflict on Code and Description (or other unique keys) if cannot find GUID.
- Recent TrakCare versions will report which Row ID is conflicting.
    - Older versions require the user to decode the key and determine the source of the conflict.



```
Load started at 2018-06-14 09:52:50
WARNING: Unique Index conflict. User.CTLoc ICCLOC01 ICC Training Location 01 0AC352C7-A85B-4F56-9A71-6E20841523D8
        Conflict with: User.CTLoc       470      ICCLOC01 ICC Training Location 01 15666461-80D8-4C71-A506-490E593C9D4E
WARNING: Unique Index conflict. User.CTLoc ICCLOC01 ICC Training Location 01 0AC352C7-A85B-4F56-9A71-6E20841523D8
        Conflict with: User.CTLoc       470      ICCLOC01 ICC Training Location 01 15666461-80D8-4C71-A506-490E593C9D4E
ERROR #5001: Unable to load stream C:\TrakCare\Perforce\custom_ccrs\us\ISCU\TRAK\TEST\misc\User\CTLoc\0AC352C7-
A85B-4F56-9A71-6E20841523D8.xml
> ERROR #5001: Unable to save item $lb("","User.CTLoc"), 0AC352C7-A85B-4F56-9A71-6E20841523D8
> ERROR #5521: SQLError: SQLCODE=-119 %msg=Table 'SQLUser.CT_Loc', Constraint 'IndexUniqueCodeHospIdx', Field(s)
CTLOC_Code="ICCLOC01",CTLOC_Hospital_DR=12; failed unique check
> ERROR #5001: Unique Index conflict. User.CTLoc ICCLOC01 ICC Training Location 01 0AC352C7-A85B-4F56-9A71-6E20841523D8
> ERROR #5001: Conflict with: User.CTLoc 470 ICCLOC01 ICC Training Location 01 15666461-80D8-4C71-A506-490E593C9D4E
Load detected 1 errors in .296s.
```

How do we get these conflicts?

- Remember, a GUID is assigned on creation, and is NOT linked to the object content.
- Deleting a layout and recreating the “same” layout does not result in the export of the same object.
  - This will create a new object (with a new GUID), having the same key as an existing one.
- If the delete doesn’t progress through, this will cause a conflict.
  - This happens a lot in Edition.
- **Fix instead of delete and recreate.**

Resolution: Wrong GUID on Object

- This will require a GUID reassign.
- Remember, the GUID is not used for anything other than referencing elements in XML.
- Always update the target environment, to match the export from BASE.

Resolution: Wrong GUID on Object (cont.)

- GUID reassign cheatsheet:
  - `Set GUID = “correct GUID”.`
  - `Set OID = $lb(“id”, “class”).`
  - `Set wrongGUID = ^OBJ.GUID(1,oid).`
  - `Kill ^OBJ.GUID(0, wrongGUID).`
  - `Set ^OBJ.GUID(0,GUID) =OID.`
  - `Set ^OBJ.GUID(1,OID) =GUID.`
- Or use ^SSADMIN, Option 5 – Change Control, then Option 7 - Replace GUID.

Quiz: GUID conflicts

Question:

Deleting and recreating an object is a likely cause of GUID conflict when moving between environments. True or False?

Answer:

True. This will create a new object (with a new GUID), having the same key as an existing one.

Part 10: iService Guidelines

Change Control Related Incidents within iService (cont.)

- iService contains Application Areas and Closure Types for incidents which are designed to capture both the respective change control area as well as the root cause.
  - It is important to understand the classifications and make sure they are used properly.
  - An accurate capture of the data will result in better reporting and actionable analysis.
    - This is especially important when a new site goes LIVE with CCR for the first time; we want to catch training or understanding gaps as quickly as possible.

Change Related Incident Classification within iService

- Application Areas:
  - CCR – CCR Online.
  - CCR – TrakCare Change Control.

Change Related Incident Classification within iService (cont.)

- Closure Types:
  - GUID Fix.
  - Perforce Intervention.
  - User Training Required.
  - Documentation Deficiency.
  - User Error.
  - Product Fix.

Summary

- What are the key points for this module?

ICC 620: CCR Tier 2 – Baselining TrakCare Configuration

InterSystems

Creative data technology

Objective

- Explain which tools can be used to perform a Tier 2 baseline of a TrakCare instance.
- Demonstrate the ability to use the baseline export tools.
- Identify places to go for more details and instructions on performing a Tier 2 baseline.

Part 1: Baselining Overview

Review: What is Baselining?

- Synchronizes contents of Perforce and all environments with a source of truth.
    - Source of truth usually current state of LIVE environment.
- Process steps include:
    - Export all source (Tier 1) and/or configuration (Tier 2) from environment.
    - Reconcile any content already in Perforce.
    - Populate other Perforce branches with reconciled export.
    - Populate source workspaces.
    - Refresh environments.

Review: Why is Baselining Important?

- Establishes starting point in Perforce.
  - Editing item in environment that isn’t in Perforce means:
    - Can’t diff change.
    - Can’t backout.
- Establishes clean integration history.

Re-baselining

- Performing a baseline again.
- Cleans up previous mistakes from poor CCR usage.
  - Abandoned changes.
  - Unlocked TEST or LIVE.
- Establishes clean integration history.
  - Prevents future merge conflicts.
- History of changes still maintained in Perforce.

Baselining Prerequisites

- Verify all productions use System defaults.
    - Ensemble and HealthShare only.
    - See ICC530 for more information.
- Progress all CCRs to terminal state (Closed, Cancelled, Merged).
    - (re)baselining significantly simpler when no active changes in flight for System.
        - Avoids needing to reintroduce active changes.

Baselining Prerequisites (cont.)

- If (re)baselining multiple systems:
  - Each System will have to be frozen while (re)baselining that System.
    - (re)baseline one at a time.
    - Prioritize which System to finish first.

Baselining Process Notes

- InterSystems internal task.
  - Requires access to Perforce using p4 client.
- Must first properly configure:
  - CCR System Details.
    - ICC320.
  - Namespace.
    - ICC430.
- **Always** create a CCR to document process.
- Consider connecting via Remote Desktop to Cambridge for Perforce updates to improve efficiency.

Baselining Process Notes (cont.)

- Add system alert to warn everyone not to create CCRs.
  - See ICC320.
- Consider locking BASE to prevent accidental changes.
  - Use Tools > Change Control > Configure > Lock Change Control.
- Verify Bundle Queue empty.
  - Tools > Change Control > Bundle and Upload ItemSet.
- In environment from which exporting configuration, run:
  - XML.Element:AssignGUIDs(1).
  - XML.Element:CheckGUIDs(1).

Quiz: Purpose of Baselining

Question:

Which of the following is true about baselining? Select all that apply.

A. Rollback not possible without baselining.

B. Establishes starting point for every item managed via CCR.

C. Establishes clean integration history.

D. Required for diffing changes.

Answer:

All of the above.

Quiz: Baselining Procedure

Question:

Which of the following is required for baselining? Select all that apply.

A. Machine with Perforce client (P4) installed.

B. Valid Perforce user.

C. Machine with access to InterSystems internal network.

D. Environment with change control enabled.

Answer:

All of the above.

Manual Baseline

1. Export configuration:
   - `Set st = ##class(TC.ChangeControl.CCR).BaselineExport().`
2. Upload to InterSystems FTP2 site.
3. Download on RDP server with P4V client.
4. Manually create workspace, load files and branch were required.
5. Typically takes 3 days to achieve and full-time equivalent resource.

Part 2: AutoBaseline Script For TrakCare Configuration

Automation Overview

- This course discusses an automation baselining script created to decrease time and minimize errors when doing a CCR Tier 2 (TrakCare configuration) baseline.
- The script covers the process of reconciling a baseline against a Perforce branch and automatically resolving and submitting the contents of the new baseline on top of existing branch content.

Automation Overview (cont.)

- This is an InterSystems-internal task which requires access to a machine with a Perforce client (p4) where the script can be run.
- Scripts can be found within Perforce at:
  - //custom_ccrs/_common/tools/AutoBaselineScripts/.

References

- Latest version of AutoBaseline script can be found here:
  - //custom_ccrs/_common/tools/AutoBaselineScripts/client/cls/Custom/Utility/Baseline/BaselineHelper.xml.
  - //custom_ccrs/_common/tools/AutoBaselineScripts/client/inc/Custom_Utility_Baseline_Main_inc.xml.
- Latest version of Perforce AutoScript can be found here:
  - //custom_ccrs/_common/tools/AutoBaselineScripts/internal/bat/p4BaselineAutoScript.bat.

References (cont.)

- Optional.
  - Latest version of TrakCare SSADMIN Regional Menu for baselining can be found here:
    - //custom_ccrs/_common/tools/AutoBaselineScripts/client/cls/Custom/Utility/Baseline/Menu.xml.

Summary of Baseline Steps Using Scripts

1. Configure CCR system details and CCR environments appropriately.
2. Use AutoBaseline script to export source from BASE and upload to FTP2 site.
3. Use Perforce baseline batch script to download and submit export to Perforce.
4. Copy Perforce branches to environment source workspaces.
5. Perform refresh of environment.

Steps in AutoBaseline Script

1. Export XML files to temp location.
2. Compress files.
3. Create .ini configuration file.
4. Upload zip of files and configuration file to InterSystems FTP2 site.

How To: Run AutoBaseline Script

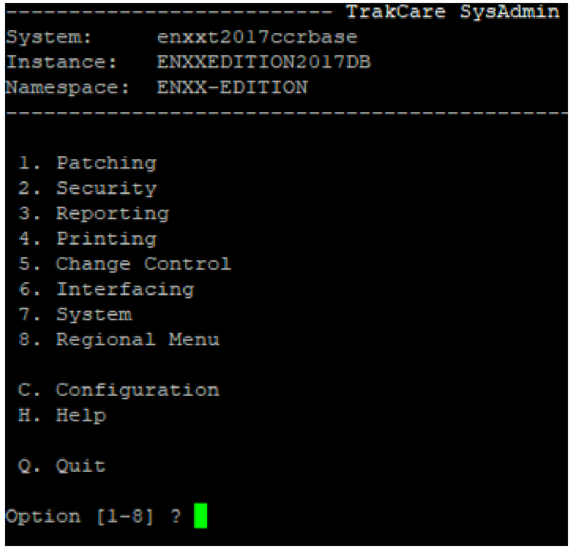
1. On Windows, 7Zip installed in C:\Program Files\7-Zip\7z.exe.
2. Verify using latest version of script.
   - Use Get Latest action in p4v.
   - Verify Id tag in files.
3. Import include file into environment.
   - Custom_Utility_Baseline_Main_inc.xml.
4. Import BaselineHelper into environment.
   - Custom.Utility.Baseline.BaselineHelper.xml.
5. Use SSADMIN tool to run baseline.
   - Alternatively, call methods directly.

How To: Setup SSADMIN Utility for Baselines

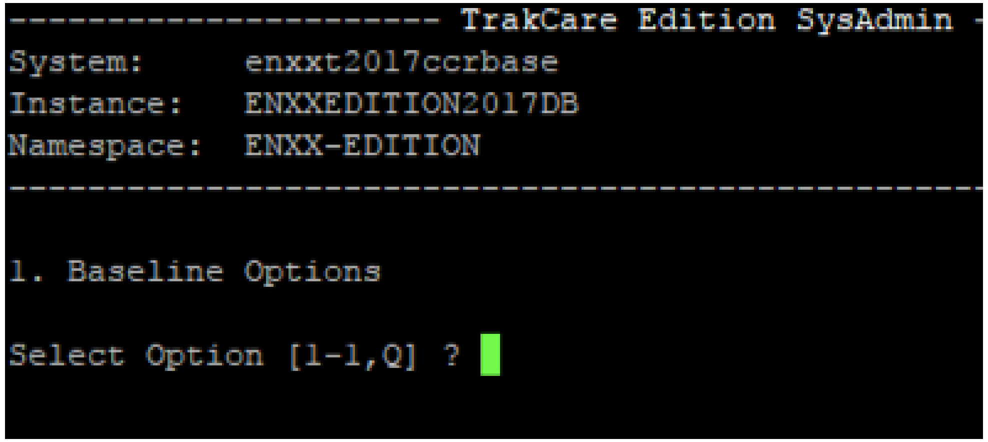
1. Get latest version of menu from Perforce:
   - //custom_ccrs/_common/tools/AutoBaselineScripts/client/cls/Custom/Utility/Baseline/Menu.xml.
2. Import Menu.xml into namespace.
   - May require Tier 1 CCR.
3. Run `do ##class(Custom.Utility.Baseline.Menu).SetupMenu().`
   - Once installed, prompt enters into SSADMIN tool.

How To: Use SSADMIN Utiltiy for Baselines

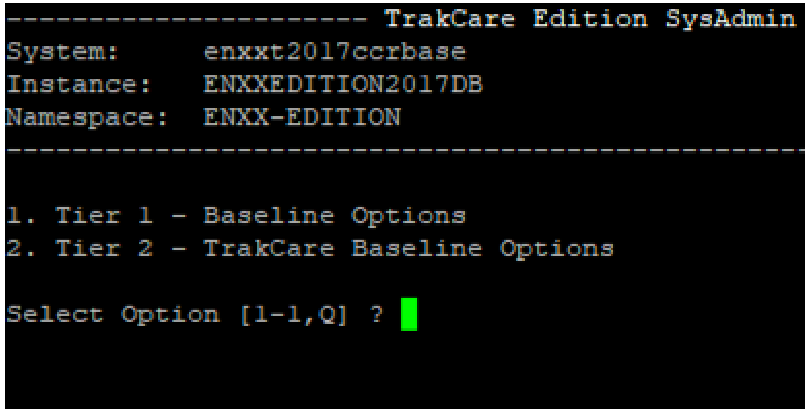
1. Enter SSADMIN utility.
   - Automatically entered into utility after installation.
   - Call `do ^SSADMIN` to run at any time.
2. Choose 8 > 1 > 2 to access Tier 2 baseline options.



```
------------------------- TrakCare SysAdmin
System:      enxxt2017ccrbase
Instance:    ENXXEDITION2017DB
Namespace:   ENXX-EDITION
--------------------------------------------

 1. Patching
 2. Security
 3. Reporting
 4. Printing
 5. Change Control
 6. Interfacing
 7. System
 8. Regional Menu

 C. Configuration
 H. Help

 Q. Quit

Option [1-8] ?
```



```
------------------------ TrakCare Edition SysAdmin -
System:      enxxt2017ccrbase
Instance:    ENXXEDITION2017DB
Namespace:   ENXX-EDITION
----------------------------------------------------

1. Baseline Options

Select Option [1-1,Q] ?
```



```
---------------------- TrakCare Edition SysAdmin
System:      enxxt2017ccrbase
Instance:    ENXXEDITION2017DB
Namespace:   ENXX-EDITION
-------------------------------------------------

1. Tier 1 - Baseline Options
2. Tier 2 - TrakCare Baseline Options

Select Option [1-1,Q] ?
```

Using SSADMIN tool - Options

- Select from the baseline options for Tier 2:
  - Full TrakCare Baseline.
    - Exports all tables from TrakCare usual after a refresh, patching or upgrade.
  - Selective Baseline Tables.
    - If only a subset of tables need to be baselined then this option can be used. It requires configuration set as a one-off task.
  - Selective Baseline GUID List(s).
    - Upload scripts or GUID syncs may provide the ability to detect the rows and classes which are being updated. It maybe quicker to export just the list of GUIDs rather than full tables to provide faster turn around on baselines.

<GUID/>

Using SSADMIN tool – Options (cont.)

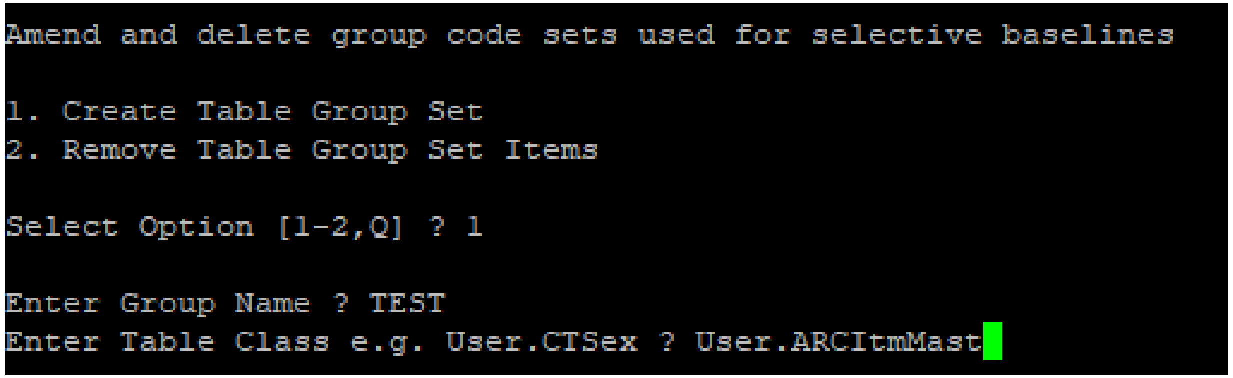
- TrakCare Baseline CCR.
  - The ability to export a list of changes in the uncommitted queue based on CCR number rather than using CCR. This can be handy for large items and especially when dealing with CT_Overrides.
- Configuration options.
  - This provides the mechanism to group a common list of classes under a tag. This tag just needs setup once and allows for quick exports when using option ‘Selective Baseline Tables’ rather than setting up the logic every time it needs run. More details on this can be found later.

Using SSADMIN tool – Options (cont.)

- For selective baseline tables you need to create a group set code which can be achieved under option 5 'Configuration options.’

Using SSADMIN tool - Configuration options

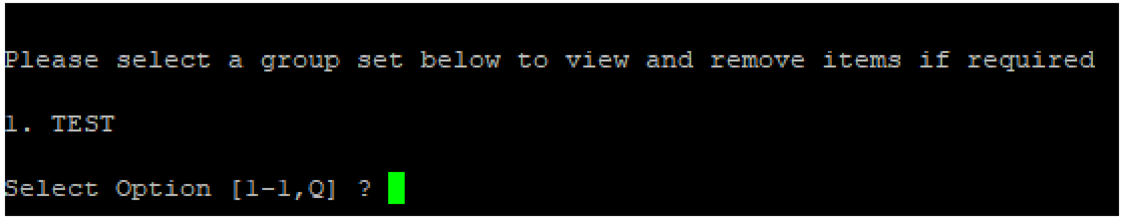
- To create group set codes for selective baseline tables then configure this under option 5 'Configuration options.’
- To create a new group code set select option 1 and add code and class. To add multiple select same option again and use the same code.



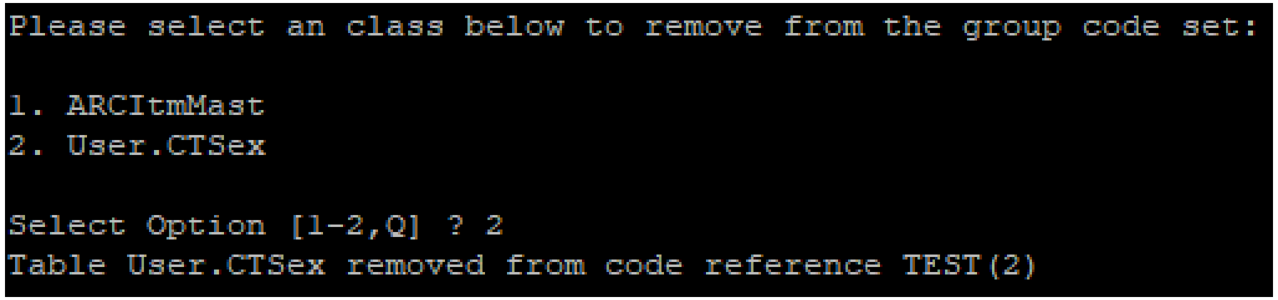
```
Amend and delete group code sets used for selective baselines

1. Create Table Group Set
2. Remove Table Group Set Items

Select Option [1-2,Q] ? 1

Enter Group Name ? TEST
Enter Table Class e.g. User.CTSex ? User.ARCItmMast
```

Using SSADMIN tool - Configuration options (cont.)

- To remove any items select option 2 'Remove Table Group Set Items'. This will display a list of the group code sets.
- Select group code to amend. This will display all the classes associated with this code set.
- Select the option number and this will be removed.



```
Please select a group set below to view and remove items if required

1. TEST

Select Option [1-1,Q] ?
```



```
Please select an class below to remove from the group code set:

1. ARCItmMast
2. User.CTSex

Select Option [1-2,Q] ? 2
Table User.CTSex removed from code reference TEST(2)
```

Full Baseline Details

- Using the script 'Custom.Utility.Baseline.BaselineHelper.xml' at site this will do the following items:
  - Run final CCR setup (exclusion list, itemset configuration source global).
  - Lock environment if required.
  - Assign GUIDs if required.
  - Export the files.

Full Baseline Details (cont.)

- Using the script 'Custom.Utility.Baseline.BaselineHelper.xml' at site this will do the following items:
  - Compress the file to tar.gz.
    - File name displayed at end of script.
  - Create configuration ini file with the same name as compressed file.
  - FTP to ftp2.intersystems.com.
  - NOTE: This can run from any environments.

Selective Baseline

- Using the script 'Custom.Utility.Baseline.BaselineHelper.xml' at site this will do the following items:
  - Assign GUIDs if required.
  - Export the selected tables.
  - Compress the file to tar.gz.
    - Filename displayed at end of script.
  - Create configuration ini file with the same name as compressed file.
  - FTP to ftp2.intersystems.com.

GUID List Export

- Using the script above 'Custom.Utility.Baseline.BaselineHelper.xml' at site this will do the following items:
  - Export list of GUIDS based in a file or files separated by a semi colon.
  - Compress the file to tar.gz.
    - Filename displayed at end of script.
  - Create configuration ini file with the same name as compressed file.
  - FTP to ftp2.intersystems.com.

Baseline CCR List

- Using the script above 'Custom.Utility.Baseline.BaselineHelper.xml' at site this will do the following items:
  - Export list of changes by CCR number where items are in the bundle queue.
  - Compress the file to tar.gz.
    - Name of file displayed at end of script.2
  - Create configuration ini file with the same name as compressed file.
  - FTP to ftp2.intersystems.com.
  - Removes items from the uncommitted queue for the given CCR, sets the file permissions back to read only and removes backup files.

Running Baselines Without SSADMIN Utility

- Can directly run method called by SSADMIN.
- For each, run method and verify returned value equals 1.
- Can use `write $system.Status.DisplayError(sc)` to get error text.

How to: Run Full Baseline Method

-
  ```
  set sc = ##class(Custom.Utility.Baseline.BaselineHelper).Baseline(FinalCCRSetup, LockEnv, AssignGUIDS).
  ```
- Method Parameters:
  - All passed as 0 for false or 1 for true.
  - FinalCCRSetup = Checks exclusion list and sets this up if not present and assigns global ^SYS("SourceControl","ItemSetConfig","Source") if not present.
  - LockEnv - Lock environment down as part of the baseline.
  - AssignGUIDS - runs the AssignGUIDs and checkGUIDS commands.

How to: Run Selective Baseline Method

1. Assign Code Set.
   - To setup selective baseline, need code set ID allocated against set of tables.
   - `set sc = ##class(Custom.Utility.Baseline.BaselineHelper).AssignBaseRefCode(ExportCodeRef,ClassTable).`
     - ExportCodeRef = code to assign.
     - ClassTable = Table to associate with ExportCodeRef.
   - To remove a table:
     `set sc = ##class(Custom.Utility.Baseline.BaselineHelper).RemoveBaseRefCode(ExportCodeRef,ClassTable).`

How to: Run Selective Baseline Method (cont.)

2. Run baseline.
   - `set sc = ##class(Custom.Utility.Baseline.BaselineHelper).BaselineTables(ExportCodeRef, AssignGUIDS).`
   - Method parameters:
     - ExportCodeRef = The code for the code set used linked to selected tables for baseline.
     - AssignGUIDS = Run the AssignGUIDs and checkGUIDS commands. Pass as 0 or 1.

How To: Run GUID List Export Method

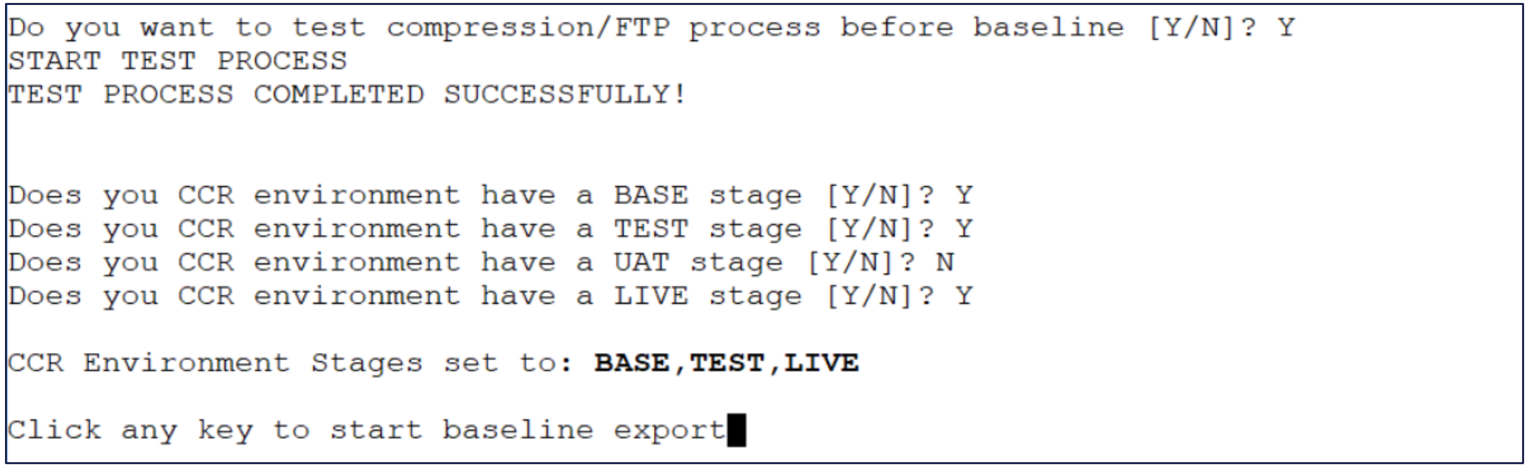
- `set sc = ##class(Custom.Utility.Baseline.BaselineHelper).BaselineGUIDs(RefCode,FileList).`
- Method Parameters:
  - RefCode = used for the file name to make unique.
  - FileList = list of filenames including full path separated by semi colon.
- ^zCustom.Baseline.Log has log details.

How To: Run Baseline CCR List Method

- `set sc = ##class(Custom.Utility.Baseline.BaselineHelper).BaselineCCR(CCRNumber).`

BaselineHelper Script Prompts

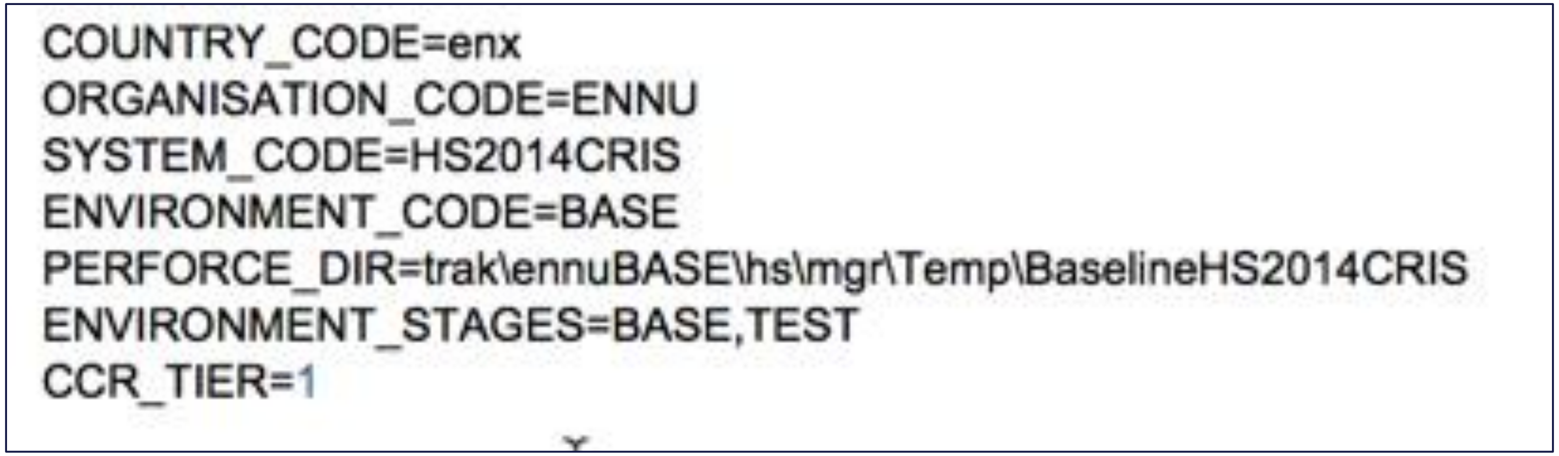
- No matter which of the above options chosen, script will ask:
  - Do you want to test compression/FTP process before baseline [Y/N]?
    - Recommend Y.
  - If previously run the script, asks whether System changed.
  - If never run or respond Y to System changed, prompts Y/N for each possible environment.



```
Do you want to test compression/FTP process before baseline [Y/N]? Y
START TEST PROCESS
TEST PROCESS COMPLETED SUCCESSFULLY!


Does you CCR environment have a BASE stage [Y/N]? Y
Does you CCR environment have a TEST stage [Y/N]? Y
Does you CCR environment have a UAT stage [Y/N]? N
Does you CCR environment have a LIVE stage [Y/N]? Y

CCR Environment Stages set to: BASE,TEST,LIVE

Click any key to start baseline export
```

Configuration Ini File

- Created by script.
- Contains information about System.
- Used by Perforce batch script discussed in next section.



```
COUNTRY_CODE=enx
ORGANISATION_CODE=ENNU
SYSTEM_CODE=HS2014CRIS
ENVIRONMENT_CODE=BASE
PERFORCE_DIR=trak\ennuBASE\hs\mgr\Temp\BaselineHS2014CRIS
ENVIRONMENT_STAGES=BASE,TEST
CCR_TIER=1
```

Quiz: TrakCare AutoBaseline Script

Question:

Which of these are TrakCare baseline options? Select all that apply.

A. Full TrakCare Baseline.

B. Selective Baseline Tables.

C. Selective GUID list(s).

D. TrakCare Baseline CCR.

Answer:

All of the above.

Quiz: TrakCare Baseline Script

Question:

The TrakCare AutoBaseline script can only run from the BASE environment. True or False?

Answer:

False. The TrakCare AutoBaseline script can run from any environment in a System.

Quiz: TrakCare AutoBaseline Script

Question:

What does the TrakCare AutoBaseline script do? Select all that apply.

A. Exports the XML files to temp location.

B. Compresses file to tar.gz.

C. Creates a configuration file to hold CCR baseline information.

D. Uploads the compressed file and configuration ini file to InterSystems FTP2 site.

Quiz: TrakCare AutoBaseline Script (cont.)

Answer:

All of the above.

Part 3: Perforce Baseline Batch File Script

Perforce Baseline Batch File Script

- Automates process after running AutoBaseline script.
- Unofficial testing shows takes 1 hour.
- Requires identical perforce username and SSO username.
  - Contact appservices@intersystems.com if not possible.
- Maintained in Perforce.
  - //custom_ccrs/_common/tools/AutoBaselineScripts/internal/bat/p4BaselineAutoScript.bat.

Steps in Batch File

1. Create temp subfolder of script directory.
2. Download tar.gz compressed file and .ini configuration file from InterSystems FTP2 to temp folder in directory of batch file.
3. Unzip file.
4. Read configuration ini file for default variable settings.
5. Prompt for any settings still empty.
6. Create workspace.
   - Username_CCR_BASELINE.

Steps in Batch File (cont.)

7. Copy new files from zip file.
8. Reconcile files in BASE in batches.
    - For Tier 2: only files under /misc branch.
        - Example: //custom_ccrs/us/ISCU/TRAK/BASE/misc/…
9. Integrate and resolve changelists for BASE to other branches.
10. Delete workspace if no pending changelists present.
11. Delete temp files and workspace.

How To: Run Perforce Batch Script

1. RDP to us2wwperforce.
2. Log into Perforce using P4V.
   - Stores credentials for use by script.
3. Get latest version of script from Perforce.
   - Must have connected to Perforce on this machine before running script.
   - Always use current version of script!

How To: Run Perforce Batch Script (cont.)

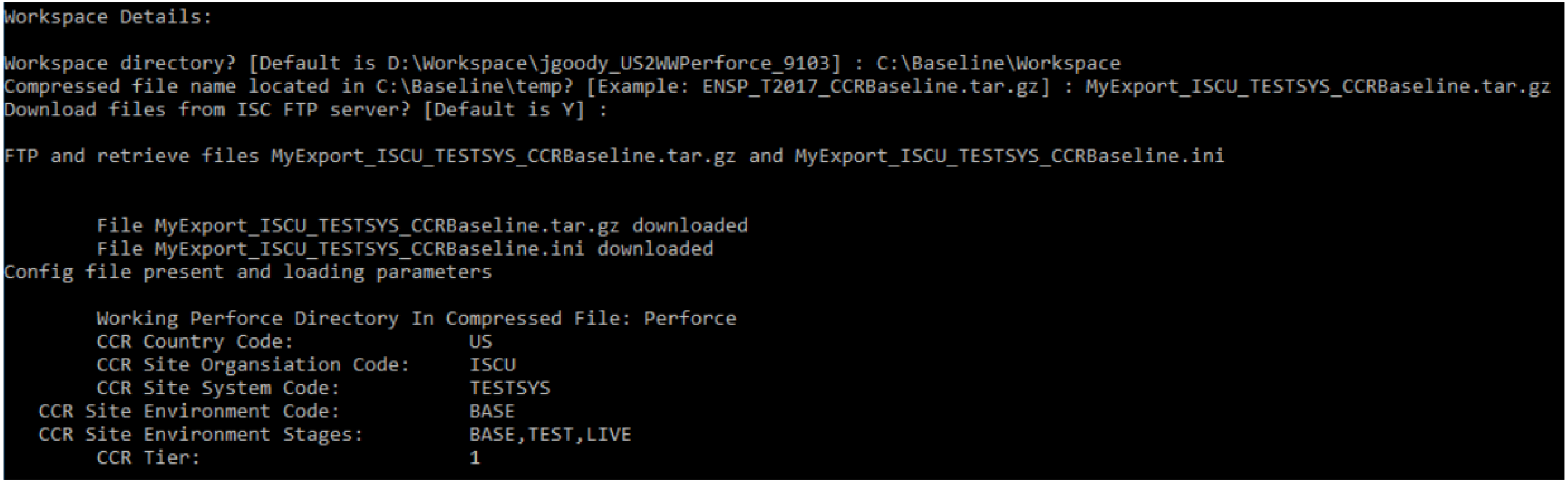
4. Create local filesystem location for temporary Perforce workspace.
   - Directory, but not workspace, must exist before running script.
   - Script will delete all files at end of script.
     - Do not use directory with pre-existing contents.
   - Script prompts for directory.
5. Double click batch script to run.

Perforce Batch File Prompts

- Workspace directory?
  - Default D:\Workspace\jgoody_US2WWPerforce_9103.
  - See notes on previous slide.
  - Cannot have trailing slash.
- Name of compressed file?
  - Specify name of file generated by AutoBaseline script.
- Download files from FTP server?
  - Default: Y.
  - Enter N when FTP not accessible.
    - Put zipped files in temp subfolder of script directory.

Perforce Batch File Prompts (cont.)

- Environment Details.
  - Uses values in configuration file to skip prompts.



```
Workspace Details:

Workspace directory? [Default is D:\Workspace\jgoody_US2WWPerforce_9103] : C:\Baseline\Workspace
Compressed file name located in C:\Baseline\temp? [Example: ENSP_T2017_CCRBaseline.tar.gz] : MyExport_ISCU_TESTSYS_CCRBaseline.tar.gz
Download files from ISC FTP server? [Default is Y] :

FTP and retrieve files MyExport_ISCU_TESTSYS_CCRBaseline.tar.gz and MyExport_ISCU_TESTSYS_CCRBaseline.ini


        File MyExport_ISCU_TESTSYS_CCRBaseline.tar.gz downloaded
        File MyExport_ISCU_TESTSYS_CCRBaseline.ini downloaded
Config file present and loading parameters

        Working Perforce Directory In Compressed File: Perforce
        CCR Country Code:               US
        CCR Site Organsiation Code:     ISCU
        CCR Site System Code:           TESTSYS
   CCR Site Environment Code:           BASE
   CCR Site Environment Stages:         BASE,TEST,LIVE
        CCR Tier:                       1
```

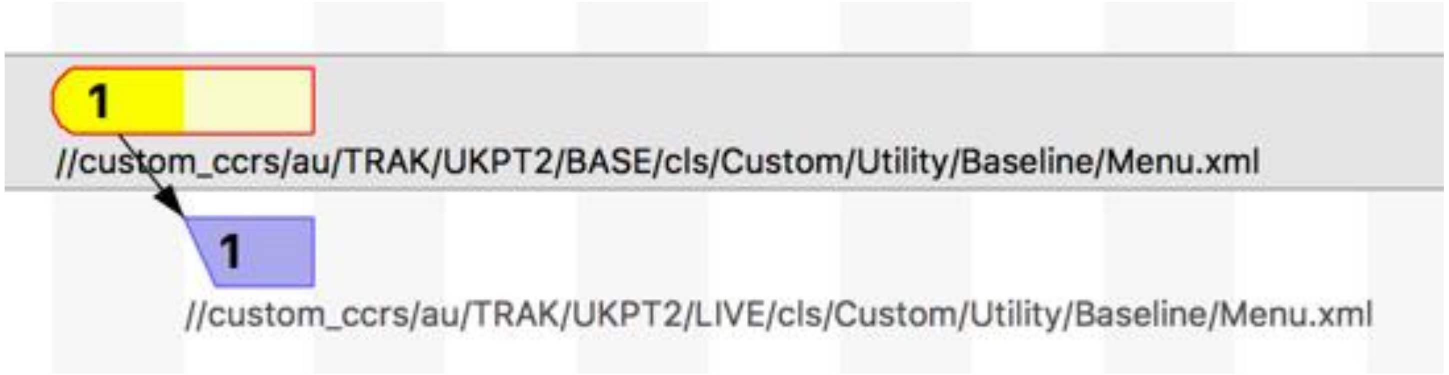
Log files

- Several log files created during script execution.
- Upon script completion:
  - Adds output.log and error.log to log.log.
  - Deletes all except log.log.

Log file	Description	Deleted
log.log.	Overall log of full process.	No.
output.log.	Temp log for each process step.	Yes.
error.log.	Temp error log for each process step.	Yes.
submitlog.log.	Submitted changelists for workspace, user, branch and date/time between merges.	Yes.

Review Changes

- Review:
  - Log file.
  - Branches.
    - Ensure folders present and merges executed correctly.


1
//custom_ccrs/au/TRAK/UKPT2/BASE/cls/Custom/Utility/Baseline/Menu.xml
1
//custom_ccrs/au/TRAK/UKPT2/LIVE/cls/Custom/Utility/Baseline/Menu.xml

Quiz: Baseline Batch script

Question:

Which of the following is true about the configuration ini file?

A. Contains environment default information required for baselining.

B. Speeds up the process of baselining.

C. Contains the CCR site organization code.

D. Contains the CCR site stage environments.

Answer:

All of the above.

Quiz: Perforce Baseline Batch File Script

Question:

Which of the following is true about the Perforce baseline batch file script? Select all that apply.

A. Requires access to a machine with a Perforce client (p4).

B. Running the script is an InterSystems-internal task.

C. The perforce username must match the ISC active directory username running the script.

Answer:

All of the above.

Quiz: Perforce Baseline Batch File Script

Question:

What differences relating to file types and processes are only involved when running a Tier 2 baselining script? Select all that apply.

A. Classes only imported.

B. Classes and configuration items are imported.

C. Configuration items imported only.

D. Only looks at the misc branch.

Quiz: Perforce Baseline Batch File Script (cont.)

Answer:

C. Configuration items imported only.

D. Only looks at the misc branch.

Quiz: Perforce Baseline Batch File Script

Question:

A Perforce user account is required to run the Perforce baseline batch file script. True or False?

Answer:

True.

Part 4: Refresh Environment

Refresh Environment

- Need to update environments to match Perforce branches.
- Refresh^%buildccr only imports Tier 1 source.
- Need to either clone VM used for export or copy code databases and TrakCare files.
- If clone VM, make sure to:
  - Update source workspace path.
    - Source workspace path contains environment type (BASE/TEST/UAT/LIVE).
  - Run Configure^%buildccr to update environment type and lock/unlock appropriately.

Part 5: Re-baselining

How To: Re-Baseline

- Same prerequisites and notes from earlier apply.
- Usually export from LIVE instead of BASE.

How To: Re-Baseline (cont.)

1. Get Latest on BASE branch.
2. Delete all files in BASE folder from your Perforce workspace.
3. Copy files from export into BASE folder of Perforce workspace.
4. In p4v, right-click BASE branch > Reconcile Offline Work.
   - Reconcile Offline Work compares files in workspace to files in depot.
5. Choose just the edits > New Changelist > Reconcile.
6. Repeat steps 4 and 5 for Adds, adding to a new changelist.

How To: Re-Baseline (cont.)

7. Submit changelists.
   - Don’t forget to attach CCR number as a job.
8. Integrate changelists from BASE to TEST.
   - Add both integrations to same changelist.
   - Resolve changelist, choosing Accept Source.
   - Submit changelist.
9. Repeat step 8, integrating TEST to next environment through LIVE.
10. Refresh environments.

Baseline Conclusion

- All branches and environments in sync.
- Branches and environments will remain in sync provided:
  - Best practices followed regarding ItemSet deployment and backing out.
  - All changes made via CCR.
  - CCR never disabled.
  - Locks on non-BASE environment never removed.

Additional Resources

- Confluence: CCR Baseline Guidelines

Summary

- What are the key points for this module?

ICC 630: CCR Tier 2 – Debugging and Advanced Topics

InterSystems

Creative data technology

Objectives

- Explain an advanced understanding of TCC and ElementXML.
- Discuss the complex data structures used in TCC.
- Debug and resolve advanced issues encountered with TCC.
- Implement CCR for TrakCare Analytics.

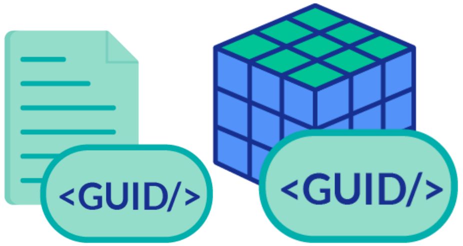
Part 1: TCC Background and Review

TrakCare Architecture

- TrakCare is made of many Persistent classes, split into packages.
  - User.* - older classes; 75% of stored info; Covers code table and Transactional Data.
  - epr.* - Clincial data configuration; security groups; worklist and epr configuration; 15% of data.
  - websys.* - System data; screen layouts and translations; workflows; reports; 10% of data.
  - There are close to 1000 classes considered to be “Configuration” which are controlled by TCC.

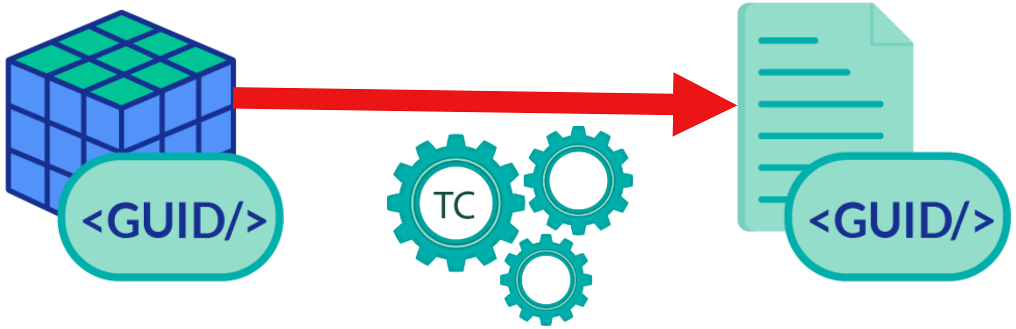
TrakCare Architecture (cont.)

- Each of these classes stores properties which correspond to fields on the page.
- Classes can have any number of properties (some classes have 100+).
  - TCC provides a standardized export format for these properties.



Review: ElementXML

- All TrakCare configuration is exported via TCC in ElementXML format.



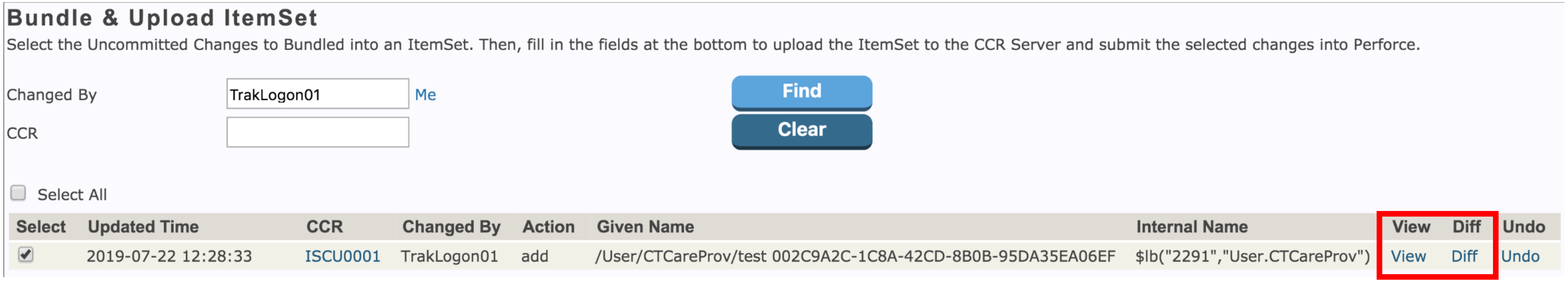
- Every piece of exportable configuration must have a GUID assigned to it.
  - GUIDs (Globally Unique Identifiers) must be consistent across all environments in a System.

Review: ElementXML (cont.)

- An ElementXML document imported into TrakCare will create one or more rows of TrakCare configuration.
  - GUIDs are used in ElementXML documents to point to referenced configuration.
  - If referenced GUIDs are not present then TCC will not load the ElementXML document.

How To: View ElementXML

- ElementXML can be viewed:
  - Before upload: Use view or diff links on Bundle & Upload page.
  - After upload: Use view, diff or history links under [show Submitted Changes] on CCR.



Bundle & Upload ItemSet

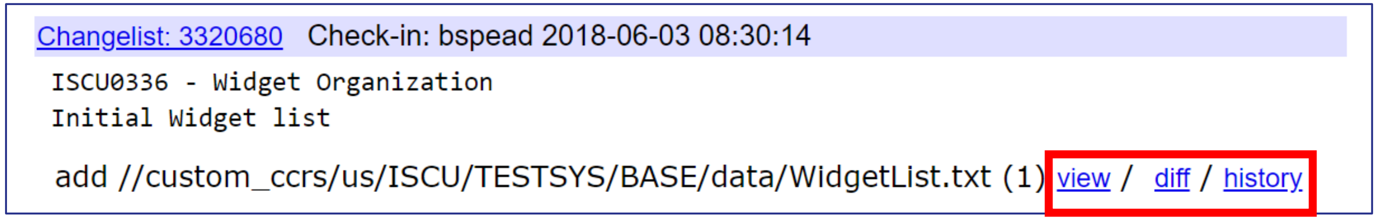
Select the Uncommitted Changes to Bundled into an ItemSet. Then, fill in the fields at the bottom to upload the ItemSet to the CCR Server and submit the selected changes into Perforce.

Changed By: TrakLogon01 Me

CCR:

□ Select All

Select	Updated Time	CCR	Changed By	Action	Given Name	Internal Name	View	Diff	Undo
☑	2019-07-22 12:28:33	ISCU0001	TrakLogon01	add	/User/CTCareProv/test 002C9A2C-1C8A-42CD-8B0B-95DA35EA06EF	$lb("2291","User.CTCareProv")	View	Diff	Undo



Changelist: 3320680 Check-in: bspead 2018-06-03 08:30:14

```
ISCU0336 - Widget Organization
Initial Widget list
```

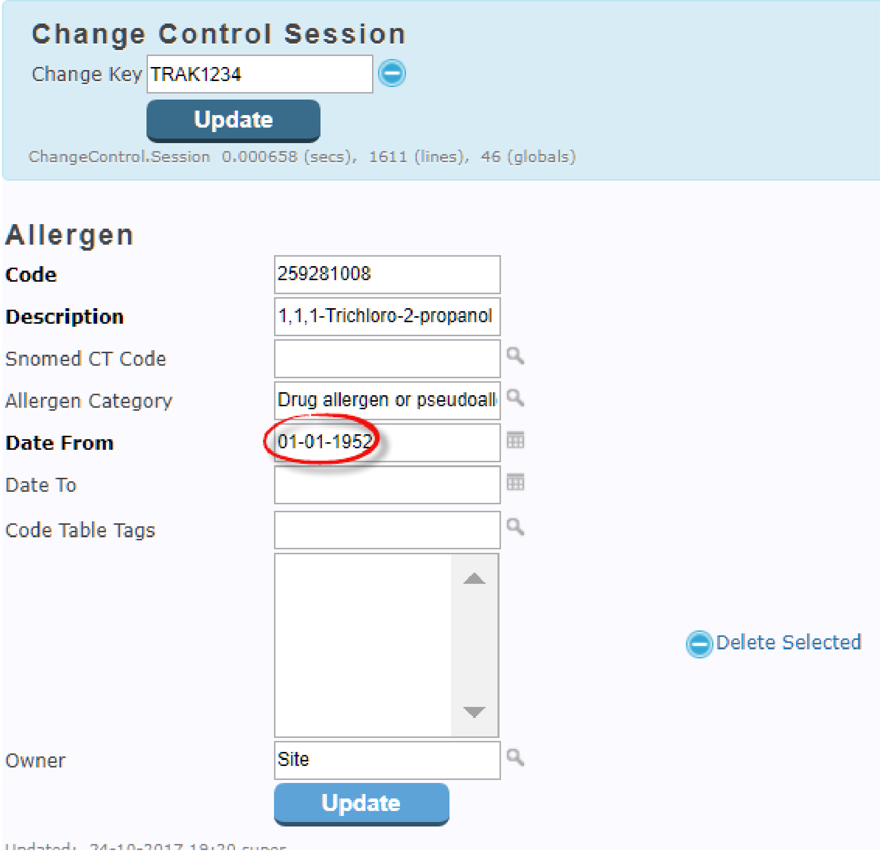
add //custom_ccrs/us/ISCU/TESTSYS/BASE/data/WidgetList.txt (1) view / diff / history

ElementXML – Additional Details

- One change in the web UI should generally mean one line changed on XML.
- Each simple property occupies one line on the export.
  - Each line consists of <Open Tag>Content</CloseTag>.
- Complex properties are exported sequentially, and their properties are exported using the same rules.
  - The Open and Close tags occupy their own lines, and do not mix with content.
    - This allows the Perforce diff to operate.

Example – simple change in XML

- Updated Date From:



Change Control Session

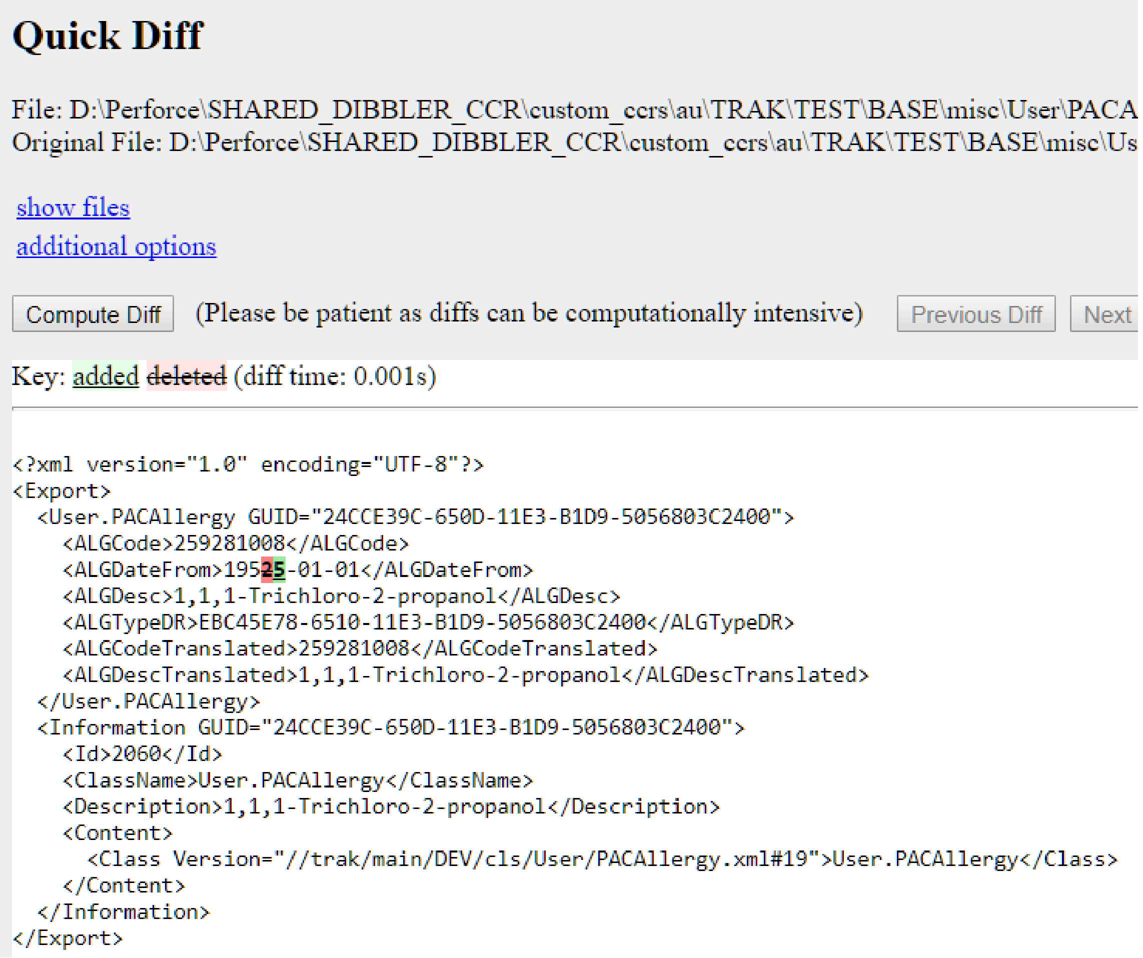
Change Key	TRAK1234

ChangeControl.Session 0.000658 (secs), 1611 (lines), 46 (globals)

Allergen

Code	259281008
Description	1,1,1-Trichloro-2-propanol
Snomed CT Code	
Allergen Category	Drug allergen or pseudoall
Date From	01-01-1952
Date To	
Code Table Tags	
Owner	Site

Updated: 24-10-2017 19:20 super



Quick Diff

File: D:\Perforce\SHARED_DIBBLER_CCR\custom_ccrs\au\TRAK\TEST\BASE\misc\User\PACA

Original File: D:\Perforce\SHARED_DIBBLER_CCR\custom_ccrs\au\TRAK\TEST\BASE\misc\Us

(Please be patient as diffs can be computationally intensive)

Key: added ~~deleted~~ (diff time: 0.001s)

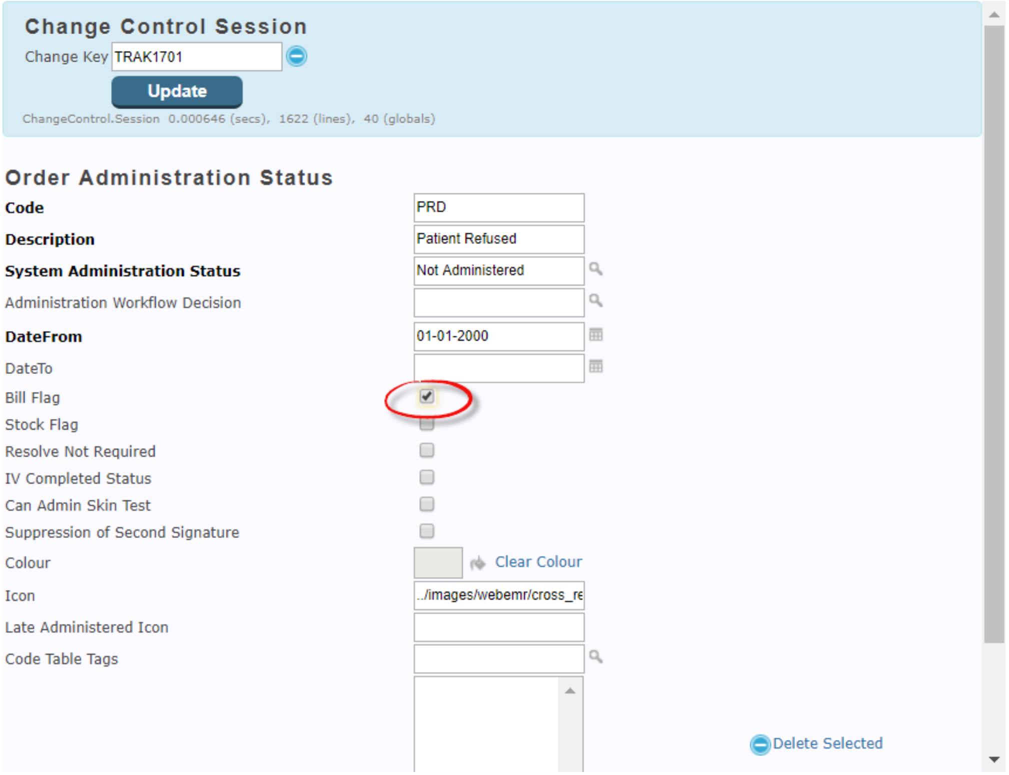
```
<?xml version="1.0" encoding="UTF-8"?>
<Export>
  <User.PACAllergy GUID="24CCE39C-650D-11E3-B1D9-5056803C2400">
    <ALGCode>259281008</ALGCode>
    <ALGDateFrom>19525-01-01</ALGDateFrom>
    <ALGDesc>1,1,1-Trichloro-2-propanol</ALGDesc>
    <ALGTypeDR>EBC45E78-6510-11E3-B1D9-5056803C2400</ALGTypeDR>
    <ALGCodeTranslated>259281008</ALGCodeTranslated>
    <ALGDescTranslated>1,1,1-Trichloro-2-propanol</ALGDescTranslated>
  </User.PACAllergy>
  <Information GUID="24CCE39C-650D-11E3-B1D9-5056803C2400">
    <Id>2060</Id>
    <ClassName>User.PACAllergy</ClassName>
    <Description>1,1,1-Trichloro-2-propanol</Description>
    <Content>
      <Class Version="//trak/main/DEV/cls/User/PACAllergy.xml#19">User.PACAllergy</Class>
    </Content>
  </Information>
</Export>
```

Quiz: Change Validity

Question:

This shows a valid change. True or False?

```
<User.OECOrderAdminStatus GUID="CA41377A-4898-11E2-B7A0-5056803BE100">
  <STATCode>PRD</STATCode>
  <STATDesc>Patient Refused</STATDesc>
  <STATBill>NY</STATBill>
  <STATDeductStock>N</STATDeductStock>
  <STATIVCompleted>N</STATIVCompleted>
  <STATCanAdminSkinTest>N</STATCanAdminSkinTest>
  <STATIcon>../images/webemr/cross_red.png</STATIcon>
  <STATDateFrom>2000-01-01</STATDateFrom>
  <STATResolveNotReq>N</STATResolveNotReq>
  <STATSecondSignature>N</STATSecondSignature>
  <STATSystemAdminStatus>NOT</STATSystemAdminStatus>
</User.OECOrderAdminStatus>
```



Change Control Session

Change Key TRAK1701

ChangeControl.Session 0.000646 (secs), 1622 (lines), 40 (globals)

Order Administration Status

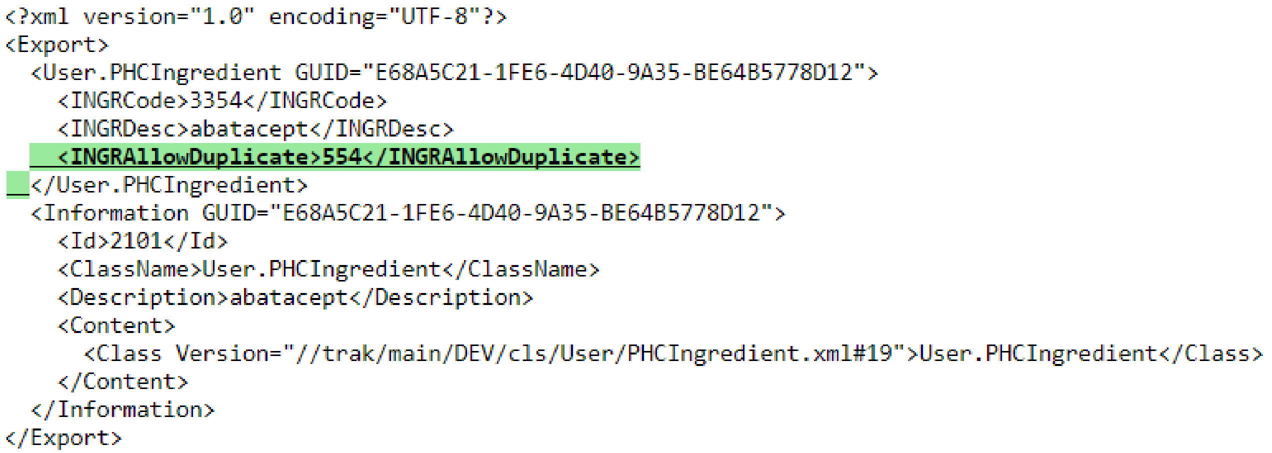
Code	PRD
Description	Patient Refused
System Administration Status	Not Administered
Administration Workflow Decision	
DateFrom	01-01-2000
DateTo	
Bill Flag	☑
Stock Flag	☐
Resolve Not Required	☐
IV Completed Status	☐
Can Admin Skin Test	☐
Suppression of Second Signature	☐
Colour	Clear Colour
Icon	../images/webemr/cross_re
Late Administered Icon	
Code Table Tags	

Answer:

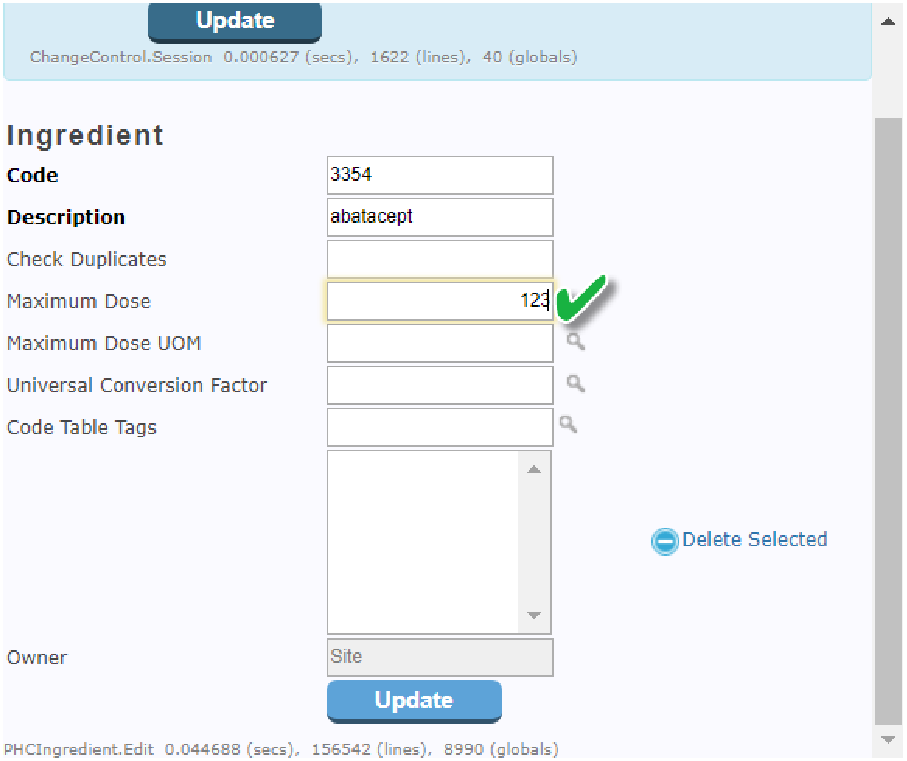
True.

Quiz: Change Validity

Question: This shows a valid change. True or False?



```
<?xml version="1.0" encoding="UTF-8"?>
<Export>
  <User.PHCIngredient GUID="E68A5C21-1FE6-4D40-9A35-BE64B5778D12">
    <INGRCode>3354</INGRCode>
    <INGRDesc>abatacept</INGRDesc>
    <INGRAllowDuplicate>554</INGRAllowDuplicate>
  </User.PHCIngredient>
  <Information GUID="E68A5C21-1FE6-4D40-9A35-BE64B5778D12">
    <Id>2101</Id>
    <ClassName>User.PHCIngredient</ClassName>
    <Description>abatacept</Description>
    <Content>
      <Class Version="//trak/main/DEV/cls/User/PHCIngredient.xml#19">User.PHCIngredient</Class>
    </Content>
  </Information>
</Export>
```



ChangeControl.Session 0.000627 (secs), 1622 (lines), 40 (globals)

Ingredient

Code	3354
Description	abatacept
Check Duplicates	
Maximum Dose	123 ✓
Maximum Dose UOM	
Universal Conversion Factor	
Code Table Tags	
Owner	Site

PHCIngredient.Edit 0.044688 (secs), 156542 (lines), 8990 (globals)

Answer:

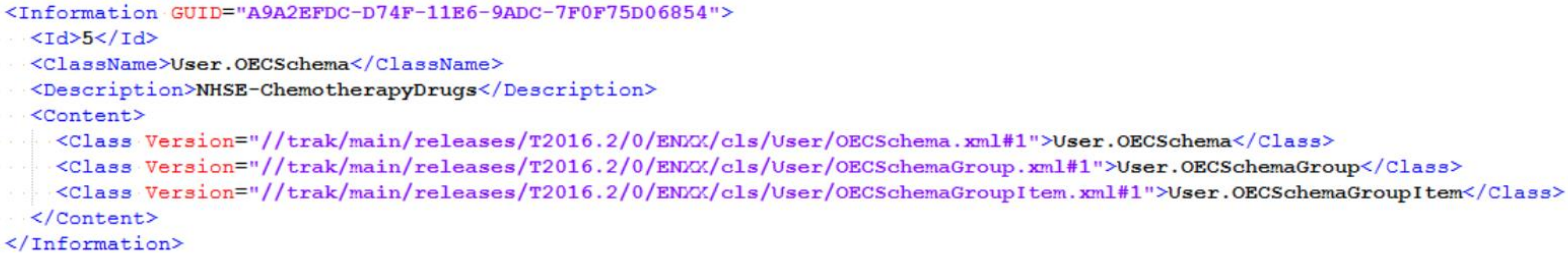
False.

The change in the XML doesn’t match the entry on the screen.

Part 2: Export Information Section

Information Section (T2011+)

- Each ElementXML export contains an Information section.
  - This is not processed by the XML parser at all.
  - It is there for your benefit only.



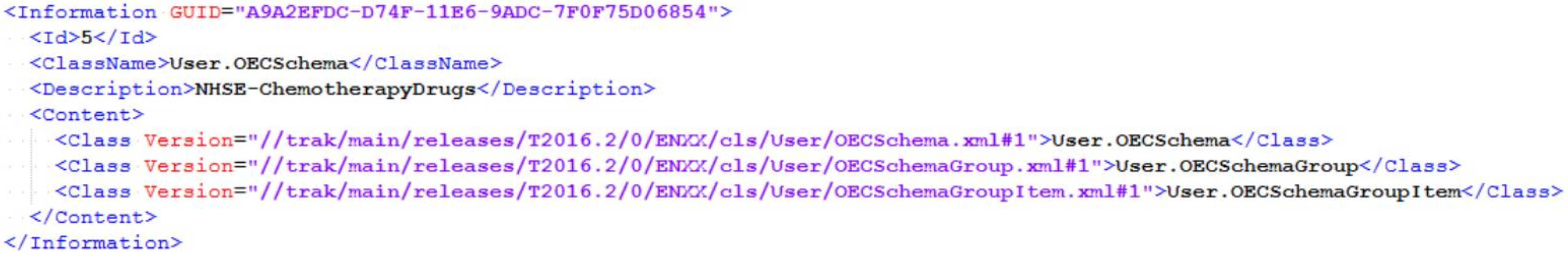
```xml
<Information GUID="A9A2EFDC-D74F-11E6-9ADC-7F0F75D06854">
  <Id>5</Id>
  <ClassName>User.OECSchema</ClassName>
  <Description>NHSE-ChemotherapyDrugs</Description>
  <Content>
    <Class Version="//trak/main/releases/T2016.2/0/ENXX/cls/User/OECSchema.xml#1">User.OECSchema</Class>
    <Class Version="//trak/main/releases/T2016.2/0/ENXX/cls/User/OECSchemaGroup.xml#1">User.OECSchemaGroup</Class>
    <Class Version="//trak/main/releases/T2016.2/0/ENXX/cls/User/OECSchemaGroupItem.xml#1">User.OECSchemaGroupItem</Class>
  </Content>
</Information>
```

Information Section (cont.)

- The information section is generated from the export process and includes:
  - Id: Row ID of the record when exported.
  - ClassName: Class of object exported.
  - Description: Description property (if present).
  - Content: All classes (including children), and the exact version of the TrakCare class.

Information Section (cont.)

- Why is this useful?



```
<Information GUID="A9A2EFDC-D74F-11E6-9ADC-7F0F75D06854">
  <Id>5</Id>
  <ClassName>User.OECSchema</ClassName>
  <Description>NHSE-ChemotherapyDrugs</Description>
  <Content>
    <Class Version="//trak/main/releases/T2016.2/0/ENXX/cls/User/OECSchema.xml#1">User.OECSchema</Class>
    <Class Version="//trak/main/releases/T2016.2/0/ENXX/cls/User/OECSchemaGroup.xml#1">User.OECSchemaGroup</Class>
    <Class Version="//trak/main/releases/T2016.2/0/ENXX/cls/User/OECSchemaGroupItem.xml#1">User.OECSchemaGroupItem</Class>
  </Content>
</Information>
```

- It enables you to track if any JIRAs have been subsequently released to fix any bugs you see.

Information Section (cont.)

- Able to tell if the export was from a different version of TrakCare than you are importing to.
  - Which can explain erroneous behavior if the download fails.
- You can tell whether the export took place before a recent patch, and the change just hasn’t been moved.
- Possible to see all invoked classes and their exact versions and use Perforce client to see if there are any subsequent changes.

Quiz: Information Section

Question:

Which of the following is true about the information section of ElementXML? Select all that apply.

A. The Information Section contains configuration information.

B. The information section is processed by the XML parser.

C. The information section is useful during debugging and troubleshooting errors.

Quiz: Information Section (cont.)

Answer:

C. The information section is useful during debugging and troubleshooting errors.

It enables you to track exactly whether any JIRAs have been subsequently released to fix any bugs you see.

Part 3: Complex Data

Review Complex data - References

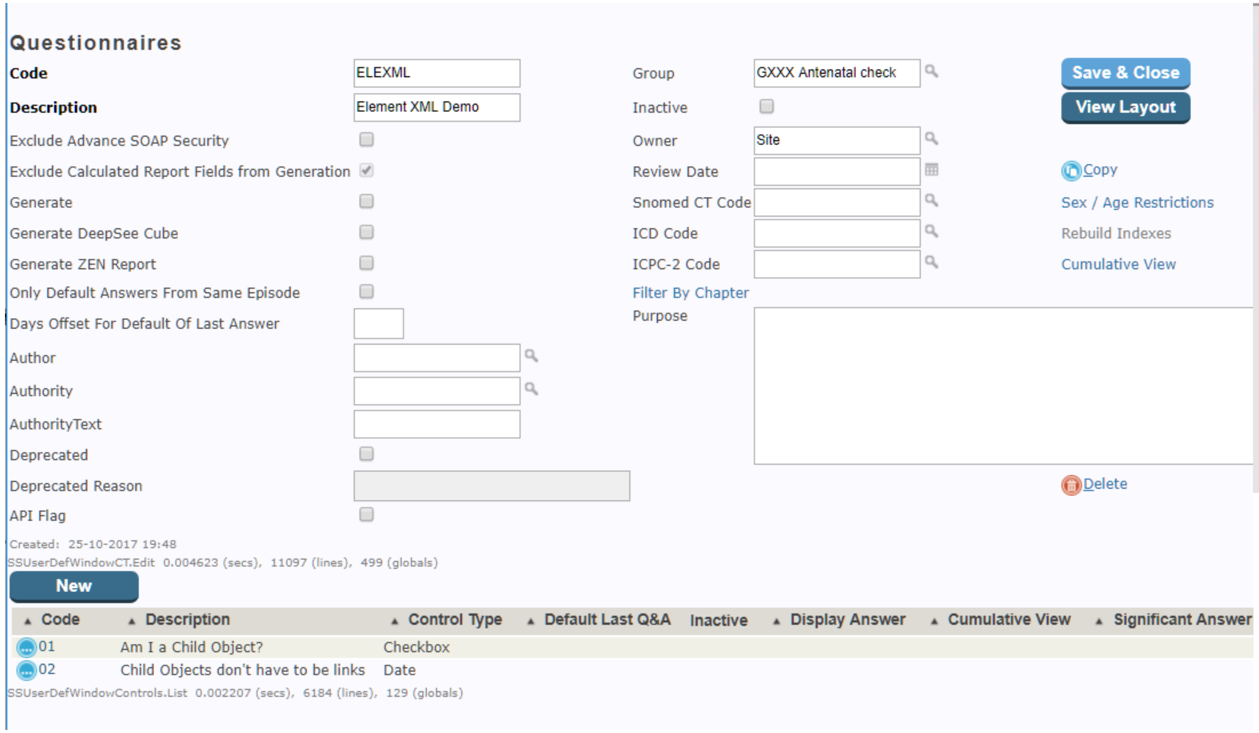
- When linking a different table or type, we should always export it as a GUID reference.
- Why would we not use code or description?
  - Code or description can change which would mess up referential integrity of pointers in the exports.
- If you see a corruption when moving configuration, check the format of the export.
  - References to other classes should ALWAYS use GUID references.
  - These issues can be intermittent, as the Row IDs may match between environments.
  - The risk increases as time passes from your last refresh.

Complex data - Children

- Code tables can group related configuration as Child tables.
- These will typically appear as a link, or table on the code table page.
- Children are typically exported with the parent object. This can cause:
  1. Large export files, which can cause performance issues.
  2. Locking contention, editing a child will lock the parent and all other children.
  3. Additional GUID synchronisation requirements.

Complex data – Children (cont.)

- Child objects can have Child objects.
- Questionnaires have many possible layers of Child objects.



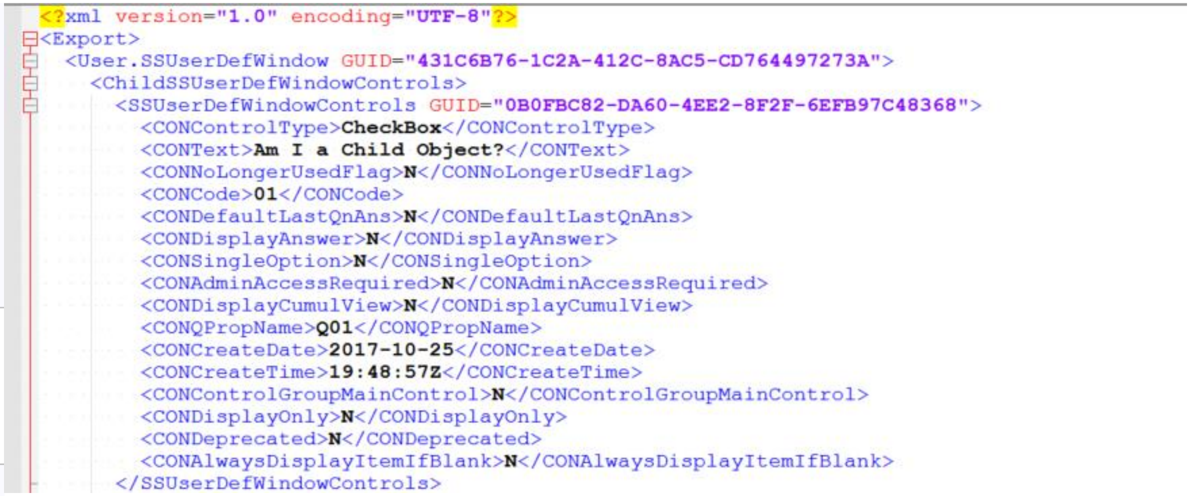
Questionnaires

Code	ELEXML	Group	GXXX Antenatal check
Description	Element XML Demo	Inactive	□
Exclude Advance SOAP Security	□	Owner	Site
Exclude Calculated Report Fields from Generation	☑	Review Date	
Generate	□	Snomed CT Code	
Generate DeepSee Cube	□	ICD Code	
Generate ZEN Report	□	ICPC-2 Code	
Only Default Answers From Same Episode	□	Filter By Chapter	
Days Offset For Default Of Last Answer		Purpose	
Author			
Authority			
AuthorityText			
Deprecated	□		
Deprecated Reason			
API Flag	□		

Created: 25-10-2017 19:48
SSUserDefWindowCT.Edit 0.004623 (secs), 11097 (lines), 499 (globals)

Code	Description	Control Type	Default Last Q&A	Inactive	Display Answer	Cumulative View	Significant Answer
01	Am I a Child Object?	Checkbox					
02	Child Objects don't have to be links	Date					

SSUserDefWindowControls.List 0.002207 (secs), 6184 (lines), 129 (globals)



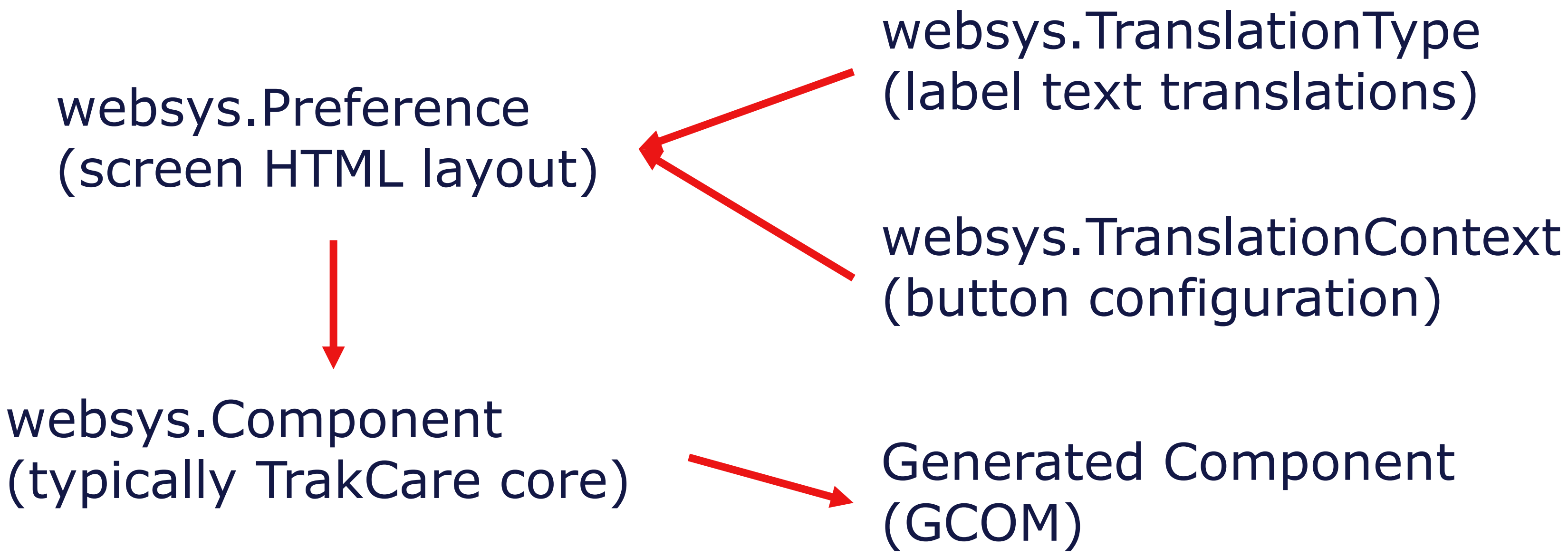
```
<?xml version="1.0" encoding="UTF-8"?>
<Export>
  <User.SSUserDefWindow GUID="431C6B76-1C2A-412C-8AC5-CD764497273A">
    <ChildSSUserDefWindowControls>
      <SSUserDefWindowControls GUID="0B0FBC82-DA60-4EE2-8F2F-6EFB97C48368">
        <CONControlType>CheckBox</CONControlType>
        <CONText>Am I a Child Object?</CONText>
        <CONNoLongerUsedFlag>N</CONNoLongerUsedFlag>
        <CONCode>01</CONCode>
        <CONDefaultLastQnAns>N</CONDefaultLastQnAns>
        <CONDisplayAnswer>N</CONDisplayAnswer>
        <CONSingleOption>N</CONSingleOption>
        <CONAdminAccessRequired>N</CONAdminAccessRequired>
        <CONDisplayCumulView>N</CONDisplayCumulView>
        <CONQPropName>Q01</CONQPropName>
        <CONCreateDate>2017-10-25</CONCreateDate>
        <CONCreateTime>19:48:57Z</CONCreateTime>
        <CONControlGroupMainControl>N</CONControlGroupMainControl>
        <CONDisplayOnly>N</CONDisplayOnly>
        <CONDeprecated>N</CONDeprecated>
        <CONAlwaysDisplayItemIfBlank>N</CONAlwaysDisplayItemIfBlank>
      </SSUserDefWindowControls>
```

Complex data – Children (cont.)

- Each child object has a GUID.
- These need to be consistent between environments.
- Conflicts can occur at a child level, but will report only as a failure at the parent.
- All child GUIDs should consistently refer to a parent.

Complex data – Special Cases

- Layouts – TrakCare UI (Responsive UI handles this differently).
- Layouts are compiled routines based on various other pieces of configuration, for a defined context.



websys.Preference
(screen HTML layout)

websys.TranslationType
(label text translations)

websys.TranslationContext
(button configuration)

websys.Component
(typically TrakCare core)

Generated Component
(GCOM)

Complex data – Special Cases (cont.)

- If no configuration exists at the current Save Level/Context for a layout, it will look at a higher level.
- A change to any piece of a layout should trigger a rebuild of the layout.
  - The source data is ONLY used to build the GCOM.
  - Deleting the TranslationType will not remove translations, until the next regenerate of the layout.
  - Translation* objects are difficult to change once saved, since their ID Key is based on data they contain.

Complex data – Special Cases (cont.)

- Typically it is necessary to delete and recreate these in the event of errors.
  - NOTE: This is a special case where this is recommended. Deleting and recreating an object results in an new object with a new GUID which in most other scenarios causes inconsistencies.

Quiz: Complex data

Question:

GUIDs should always be used when referencing other objects. True or False?

Answer:

True.

References to other classes should ALWAYS use GUID references.

Part 4: Questionnaires

Questionnaires

- Questionnaires are the most problematic configuration for TCC!
- A questionnaire is a very complex object, with nested layers of Children, all of which can refer to other tables.
- Questionnaires also generate other configuration, some of which is exported and some is not.
  - A component is generated and exported so that layout configuration can reference it.
    - However, the loaded configuration is not used, it is always regenerated.
  - Dictionary entries are generated for translation.
  - At least one layout save is generated.
  - A Caché class is generated to store the data.

Questionnaires (cont.)

- A questionnaire has many moving parts.
- If any are not moved with the initial load of a questionnaire, they will be generated on download.
- This will create "new" objects, which will be guaranteed to cause future GUID conflicts.
- It is important to verify that all configuration is progressed when creating a questionnaire.
  - Repair is tedious and time consuming, especially when the error is made in Edition.

Questionnaires (cont.)

- Why does TCC export the component, if it is going to generate the content again?
  - To maintain GUID synchronisation.
  - Layouts refer to the component, so it needs parity on all environments.
    - This also means it has all of the complexity of layout management when managing questionnaires.

Questionnaire Configuration

- A typical questionnaire should create all of the following configuration.
  - SSUserDefWindow – this is the questionnaire configuration.
  - SSUserDefWinGroup (optional) – questionnaire’s group.
  - websys.Component – the component code generated from the questionnaire configuration.
  - websys.Peference – the default screen layout.
  - websys.TranslationContext – the default control setup on the layout.
  - websys.TranslationType (optional) – text label translations.
  - websys.Dictionary – a dictionary file to allow system level translations.

Quiz: TrakCare Configuration

Question:

What are the most complex configuration object to export in TrakCare?

A. Tables.

B. Layouts.

C. Questionnaires.

D. Worklists.

Answer:

C. Questionnaires.

Quiz: Perforce and TCC

Question:

Perforce gets updated with structural changes to TCC’s ElementXML in TrakCare. True or False?

Answer:

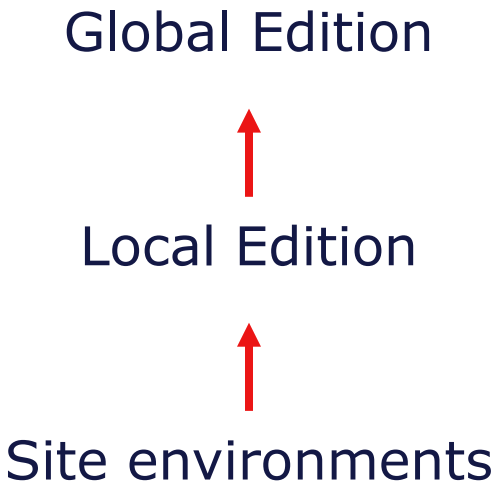
False.

Perforce doesn’t know anything about the content.

Part 5: Missing GUID Advanced Topics

GUID Without Change History

- There may be no change history if the object was not created on that System.
- Check up through the Edition model.
  - Configuration may come through an Edition patch, check that patch levels match.



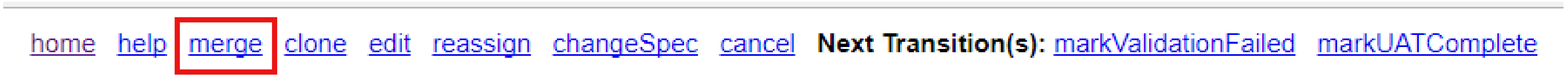
Global Edition

Local Edition

Site environments

Missing GUID Complications

- Missing GUIDs are usually very simple to resolve.
- Poor planning can lead to GUID deadlock.
  - This typically happens with very large changes, or changes left for a long period of time.
  - Example:
    - CCR A implements an item that CCR B’s questionnaire references.
    - CCR B implements a Chart Profile which references the item from CCR A.
- To resolve this, the CCRs must be merged, with the CCR merge function.



Quiz: Missing GUIDs

Question:

What information can you infer from looking at just a GUID?

A. The type of object associated with the GUID.

B. The source environment of the GUID.

C. The Edition.

D. No inference can be made from just a GUID.

Answer:

D. No inference can be made from just a GUID.

Exercise ICC630-1 through ICC630-7

Create a CCR & Debug

Quiz: Perforce Integration conflict

Question: To cancel a CCR all you need to do is click Cancel.

True or False?

Answer:

False.

You also need to backout the changes from the CCR and download the ItemSets to remove it. **Do not** leave a car crash in the lane you wish to drive down in the future.

Part 6: TCC Debugging – Incorrect Configuration

Imported configuration incorrect

- TrakCare and TCC are not perfect. There are bugs.
- Bugs can happen during export.
  - Class not marked as XMLConfigItem.
  - Field changes not captured.
  - References not exported as GUID.
  - Encoding issues (XML/HTML encoding, $lb).
  - Child record marked as XMLConfigItem.

Imported configuration incorrect (cont.)

- Bugs can happen during import.
  - Inappropriate data constraints.
  - Complex fields not processed correctly.
  - Zero node pointers not updating.
  - Generation not triggered (questionnaires, layouts).
- It is also possible that the CCR does not have a full version of the changes.
  - Remember, each CCR moves diff chunks through.
  - The CCR in question may be reliant on another CCR's changes.
    - Use the History button to check.

Export Bugs

- Each instance of this will present differently.
- This was a very common problem in the early days of CCR adoption.
- Development are now focused on ensuring TCC support is appropriate when creating new functionality.
- We are likely to see issues on new functionality, but also older functionality which hasn’t been used with change control (Stock is a good example of this).
- These should be reproduced in TEST, and raised as a critical JIRA.
  - JIRA should have the component “Change Control.”

Export Bugs – Missing Property

- The easiest (but relatively uncommon) issue to diagnose is the failure to export a property.
- To reproduce:
  1. Open a change session.
  2. Make the change to that field only.
  3. Open Bundle and Upload and perform a Diff to see the changes.
  4. If no change is present in the XML (it will highlight), then there is an issue with the field export.
- Ensure a JIRA is logged with “Change Control” as one of the selected components.
  - Include documentation of the steps to replicate.

Export Bugs – Row IDs

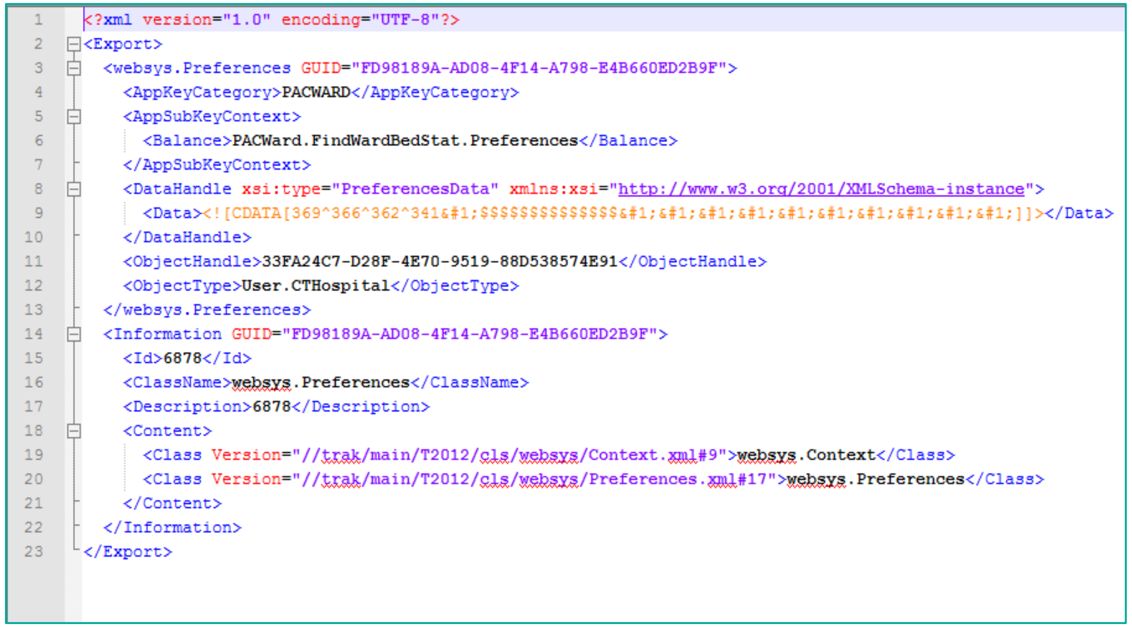
- The more common failure is that an external reference is exposed as a Row Id instead of a GUID.
  - This is more common for Collection types, as they require extra code to export.
  - These issues can fly under the radar if the environments are well aligned for RowIDs.

Export Bugs – Row IDs (cont.)

- The process is very similar to the previous:
  1. Open a change session, and make the change.
  2. Check the Diff.
  3. If you see numbers or text instead of a GUID in the changed field, this is a bug.
- This should also be raised as a critical JIRA (Component = "Change Control").

Export Bugs – Row IDs (cont.)

- The Data tags encode references as IDs.
- The DataHandler must encode these values as GUIDs.
- Why are these issues so painful?
  - They generally have no workaround.
  - They can be difficult to notice, and will mildly corrupt configuration.
  - They will not present the same on all envs (TEST can be different to LIVE).
  - They will require direct changes in the environments to fix.



```
<?xml version="1.0" encoding="UTF-8"?>
<Export>
  <websys.Preferences GUID="FD98189A-AD08-4F14-A798-E4B660ED2B9F">
    <AppKeyCategory>PACWARD</AppKeyCategory>
    <AppSubKeyContext>
      <Balance>PACWard.FindWardBedStat.Preferences</Balance>
    </AppSubKeyContext>
    <DataHandle xsi:type="PreferencesData" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <Data><![CDATA[369^366^362^341&#1;$$$$$$$$$$$$$$&#1;&#1;&#1;&#1;&#1;&#1;&#1;&#1;&#1;&#1;]]></Data>
    </DataHandle>
    <ObjectHandle>33FA24C7-D28F-4E70-9519-88D538574E91</ObjectHandle>
    <ObjectType>User.CTHospital</ObjectType>
  </websys.Preferences>
  <Information GUID="FD98189A-AD08-4F14-A798-E4B660ED2B9F">
    <Id>6878</Id>
    <ClassName>websys.Preferences</ClassName>
    <Description>6878</Description>
    <Content>
      <Class Version="//trak/main/T2012/cls/websys/Context.xml#9">websys.Context</Class>
      <Class Version="//trak/main/T2012/cls/websys/Preferences.xml#17">websys.Preferences</Class>
    </Content>
  </Information>
</Export>
```

Export bugs - Ownership

- The Edition model introduced the concept of ownership of tables and records.
- The owner can be defined as SYS, REGION, SITE. This controls what can be edited and exported from each environment. It is possible to set a table to allow ownership from all 3 levels, but each individual entry is also subject to ownership.

Export bugs – Ownership (cont.)

- This means a table owned by SYS and SITE, can have entries belonging to each owner.
  - There will be no Region entries in this table.
  - SYS can only be edited in a Global Edition env, SITE ownership of the table allows creation and maintenance of SITE entries only.

Export bugs – Ownership (cont.)

- Ownership is typically simple, each code table row will define a user directly.
- Table ownership is defined in CT_EditionManagement.
- Some configuration inherits their ownership from referenced objects.

Export bugs – Ownership (cont.)

- Questionnaires.
  - Layouts, refer to components, refer to the questionnaire which defines ownership.
  - The layout is also saved at a Save Level, and this is factored into the rules.
    - It is logically impossible to export a SITE level save from Edition.
- This process is new, and while stable, has not been bug-free.

Import bugs - Generation

- Some configuration generated related configuration when saved.
  - Questionnaires (component, storage class).
  - Order items (Keywords).
  - Layout configuration (GCOM code).
- The XML Load should automatically trigger any additional processing.
- This is an infrequently occurring issue, and will generally require technical input.

Import bugs - Constraints

- Development guidelines state that most data validation should be applied at the Application layer.
- Data constraints can cause issues with downloads of linked data.
- Example: Sequenced configuration in Lab (ref: TC-115317).
  - A constraint on the sequence number prevented these being reordered.
  - Import sequence 1 conflicted with existing sequence 1, which is now sequence 3.

Import bugs – Constraints (cont.)

- These errors should be reproduced in TEST-LIVE, and raised as a JIRA with change control selected as one of the components.

Quiz: Debugging Incorrect configuration

Question:

Which of the following would result in bugs during import of configuration?

A. Class not marked as XMLConfigItem.

B. References not exported as GUID.

C. Encoding issues (XML/HTML encoding, $lb).

D. Complex fields not processed correctly.

Answer:

All of the above.

Quiz: Debugging Incorrect configuration

Question:

The Edition model introduced the concept of ownership of tables and records. What are the different defined owners in this model? Select all that apply.

A. System.

B. Region.

C. Site.

D. None of the above.

Quiz: Debugging Incorrect configuration (cont.)

Answer:

A. System.

B. Region.

C. Site.

Part 7: TCC Debugging - Key not Unique

Key not Unique Index Resolution

- CCR Error:
  - ERROR #5001: Unable to load file /trak/sclo/UAT/DATAFILES/perforce/custom_ccrs/scx/LUHT/T2012/UAT/misc/websys/Preferences/PAWaitingListInquiry.List/A90DCD56-2B51-11E3-AC06-590A8ECE23D5.xml.
  - > ERROR #5001: Unable to save item $lb("","websys.Preferences"), A90DCD56-2B51-11E3-AC06-590A8ECE23D5.
  - > ERROR #5808: Key not unique: Index.
- Cause of the Error:
  - Workflow Context no longer exists within the application which the preferences are assigned to.

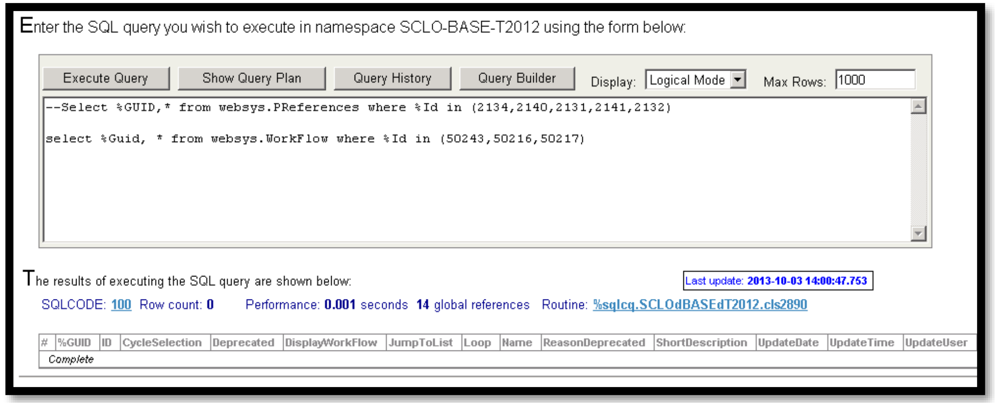
Key not Unique Index Resolution (cont.)

- Investigation steps:
  - Find GUID’s and RowID’s in BASE and UAT using FIND GUID within the TrakCare.

1. Open System Management Portal.
2. Run the following SQL Statement within SMP for BASE and UAT:
   - Select %GUID,* from Websys.Preferences where %ID in (Enter in your row ID’s).
3. Ensure they match in both environments.

CCR - Key not Unique Index Resolution (cont.)

- Investigation steps:
  4. Then run the following SQL query in both environments:
     - Select %Guid, * from websys.WorkFlow where %Id in (your row ID's which you find within the AppKeyContext from your previous query).
     - NOTE: no rows return means they have been deleted.



Enter the SQL query you wish to execute in namespace SCLO-BASE-T2012 using the form below:

Display: Logical Mode   Max Rows: 1000

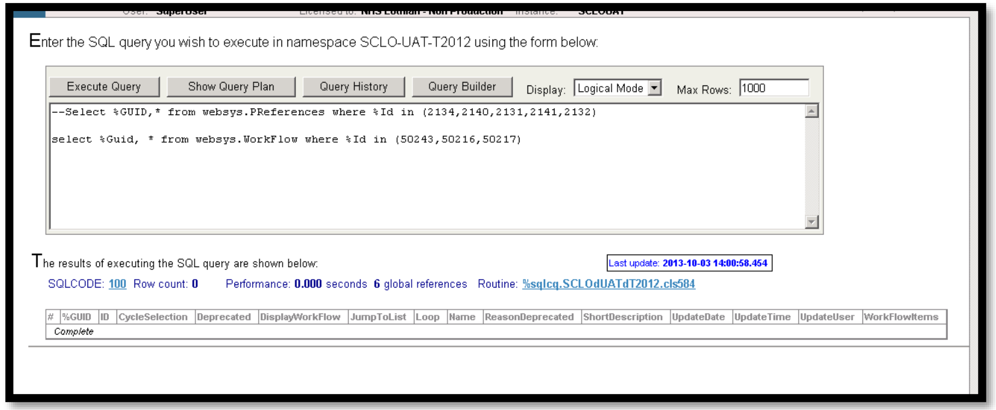
```
--Select %GUID,* from websys.PReferences where %Id in (2134,2140,2131,2141,2132)

select %Guid, * from websys.WorkFlow where %Id in (50243,50216,50217)
```

The results of executing the SQL query are shown below:

Last update: **2013-10-03 14:00:47.753**

SQLCODE: 100 Row count: **0**   Performance: **0.001** seconds **14** global references   Routine: %sqlcq.SCLOdBASEdT2012.cls2890

#	%GUID	ID	CycleSelection	Deprecated	DisplayWorkFlow	JumpToList	Loop	Name	ReasonDeprecated	ShortDescription	UpdateDate	UpdateTime	UpdateUser
Complete													



User: SuperUser   Licensed to: NHS Lothian - Non Production   Instance: SCLOUAT

Enter the SQL query you wish to execute in namespace SCLO-UAT-T2012 using the form below:

Display: Logical Mode   Max Rows: 1000

```
--Select %GUID,* from websys.PReferences where %Id in (2134,2140,2131,2141,2132)

select %Guid, * from websys.WorkFlow where %Id in (50243,50216,50217)
```

The results of executing the SQL query are shown below:

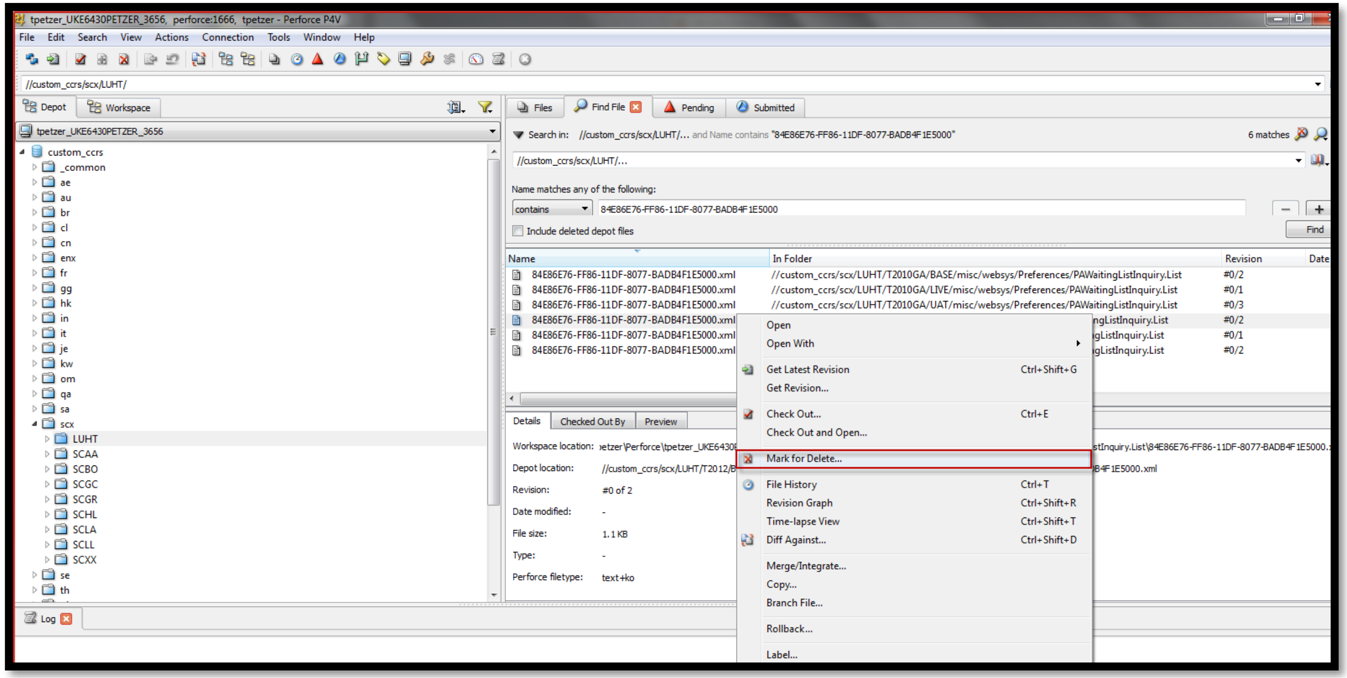
Last update: **2013-10-03 14:00:58.454**

SQLCODE: 100 Row count: **0**   Performance: **0.000** seconds **6** global references   Routine: %sqlcq.SCLOdUATdT2012.cls584

#	%GUID	ID	CycleSelection	Deprecated	DisplayWorkFlow	JumpToList	Loop	Name	ReasonDeprecated	ShortDescription	UpdateDate	UpdateTime	UpdateUser	WorkFlowItems
Complete														

CCR - Key not Unique Index Resolution (cont.)

- Investigation steps:
  5. Login Into Perforce.
  6. Ensure you are in Depot tab.
  7. Select your Client.
  8. Within Find File > Find your first GUID.
  9. Right Click > Mark for Delete.



Search in: //custom_ccrs/scx/LUHT/... and Name contains "84E86E76-FF86-11DF-8077-BADB4F1E5000" 6 matches

//custom_ccrs/scx/LUHT/...

Name matches any of the following:

contains 84E86E76-FF86-11DF-8077-BADB4F1E5000

☐ Include deleted depot files

Name	In Folder	Revision	Date
84E86E76-FF86-11DF-8077-BADB4F1E5000.xml	//custom_ccrs/scx/LUHT/T2010GA/BASE/misc/websys/Preferences/PAWaitingListInquiry.List	#0/2	
84E86E76-FF86-11DF-8077-BADB4F1E5000.xml	//custom_ccrs/scx/LUHT/T2010GA/LIVE/misc/websys/Preferences/PAWaitingListInquiry.List	#0/1	
84E86E76-FF86-11DF-8077-BADB4F1E5000.xml	//custom_ccrs/scx/LUHT/T2010GA/UAT/misc/websys/Preferences/PAWaitingListInquiry.List	#0/3	
84E86E76-FF86-11DF-8077-BADB4F1E5000.xml	ngListInquiry.List	#0/2	
84E86E76-FF86-11DF-8077-BADB4F1E5000.xml	gListInquiry.List	#0/1	
84E86E76-FF86-11DF-8077-BADB4F1E5000.xml	gListInquiry.List	#0/2	

Workspace location: etzer\Perforce\tpetzer_UKE6430 stInquiry.List\84E86E76-FF86-11DF-8077-BADB4F1E5000.x

Depot location: //custom_ccrs/scx/LUHT/T2012/B B4F1E5000.xml

Revision: #0 of 2

Date modified: -

File size: 1.1 KB

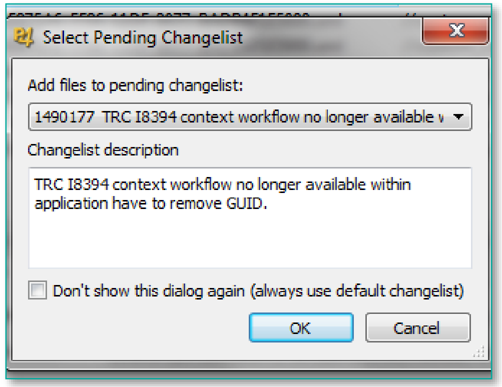
Type: -

Perforce filetype: text+ko

CCR - Key not Unique Index Resolution (cont.)

- Investigation steps:

10. Select New Pending Change List.
11. Add your TRC number and details of why you are deleting the GUID’s.
12. Click on OK.



Add files to pending changelist:

1490177 TRC I8394 context workflow no longer available w

Changelist description

TRC I8394 context workflow no longer available within application have to remove GUID.

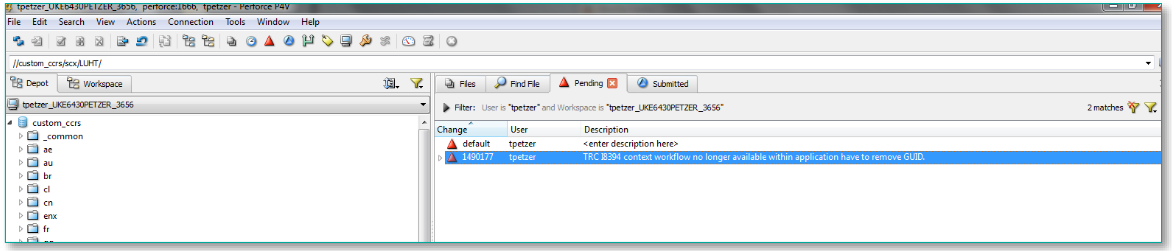
□ Don't show this dialog again (always use default changelist)

13. Repeat this process for the rest of the GUID’s you need to delete.
    - This time, select the pre-existing changelist instead of New that you have created above for each.

CCR - Key not Unique Index Resolution (cont.)

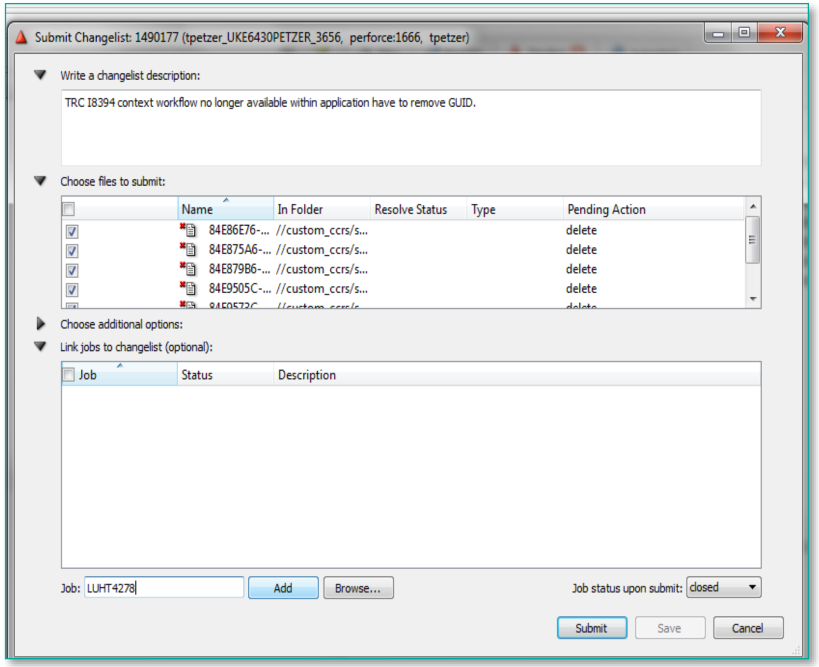
- Investigation steps:

14. Once all the GUID’s have been marked for Delete Select your Pending Tab.


tpetzer_UKE6430PETZER_3656, perforce:1666, tpetzer - Perforce P4V
File Edit Search View Actions Connection Tools Window Help
//custom_ccrs/scx/LUHT/
Depot
Workspace
tpetzer_UKE6430PETZER_3656
custom_ccrs
_common
ae
au
br
cl
cn
enx
fr
Files
Find File
Pending
Submitted
Filter: User is "tpetzer" and Workspace is "tpetzer_UKE6430PETZER_3656"
2 matches

Change	User	Description
default	tpetzer	<enter description here>
1490177	tpetzer	TRC I8394 context workflow no longer available within application have to remove GUID.

15. Your changelist will appear within the list.
16. Right click > Select Submit.
17. Add in the CCR number to the Job field.
18. Click on Add.
19. Click on Submit.


Submit Changelist: 1490177 (tpetzer_UKE6430PETZER_3656, perforce:1666, tpetzer)
Write a changelist description:
Choose files to submit:
Choose additional options:
Link jobs to changelist (optional):
Add
Browse...
Submit
Save
Cancel

TRC I8394 context workflow no longer available within application have to remove GUID.

☐	Name	In Folder	Resolve Status	Type	Pending Action
☑	84E86E76-...	//custom_ccrs/s...			delete
☑	84E875A6-...	//custom_ccrs/s...			delete
☑	84E879B6-...	//custom_ccrs/s...			delete
☑	84E9505C-...	//custom_ccrs/s...			delete
☑	84E9573C...	//custom_ccrs/s...			delete

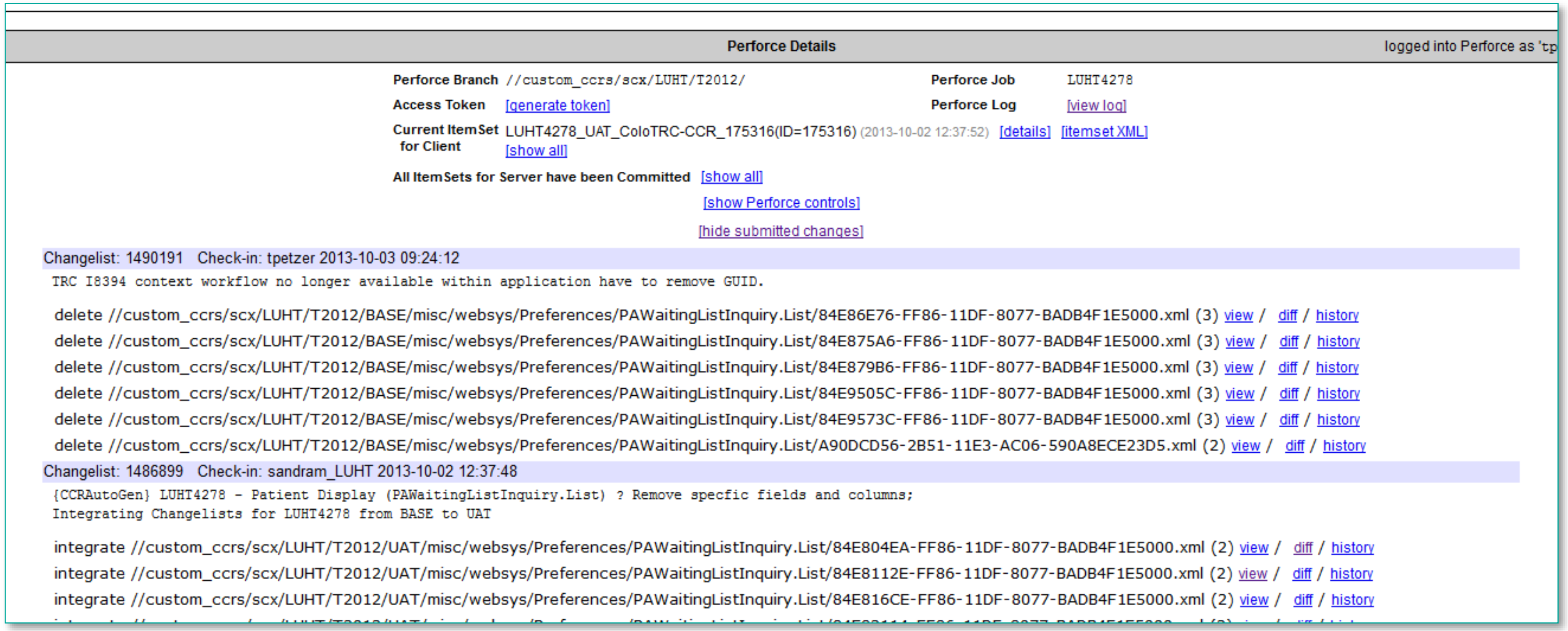
☐ Job	Status	Description

Job: LUHT4278

Job status upon submit: closed

CCR - Key not Unique Index Resolution (cont.)

- Investigation steps:
  19. Return to CCR Online application.
  20. Your new change list of deletes will appear under show Submitted changes.



Perforce Details logged into Perforce as 'tp

Perforce Branch //custom_ccrs/scx/LUHT/T2012/ **Perforce Job** LUHT4278

Access Token [generate token] **Perforce Log** [view log]

Current ItemSet for Client LUHT4278_UAT_ColoTRC-CCR_175316(ID=175316) (2013-10-02 12:37:52) [details] [itemset XML] [show all]

All ItemSets for Server have been Committed [show all]

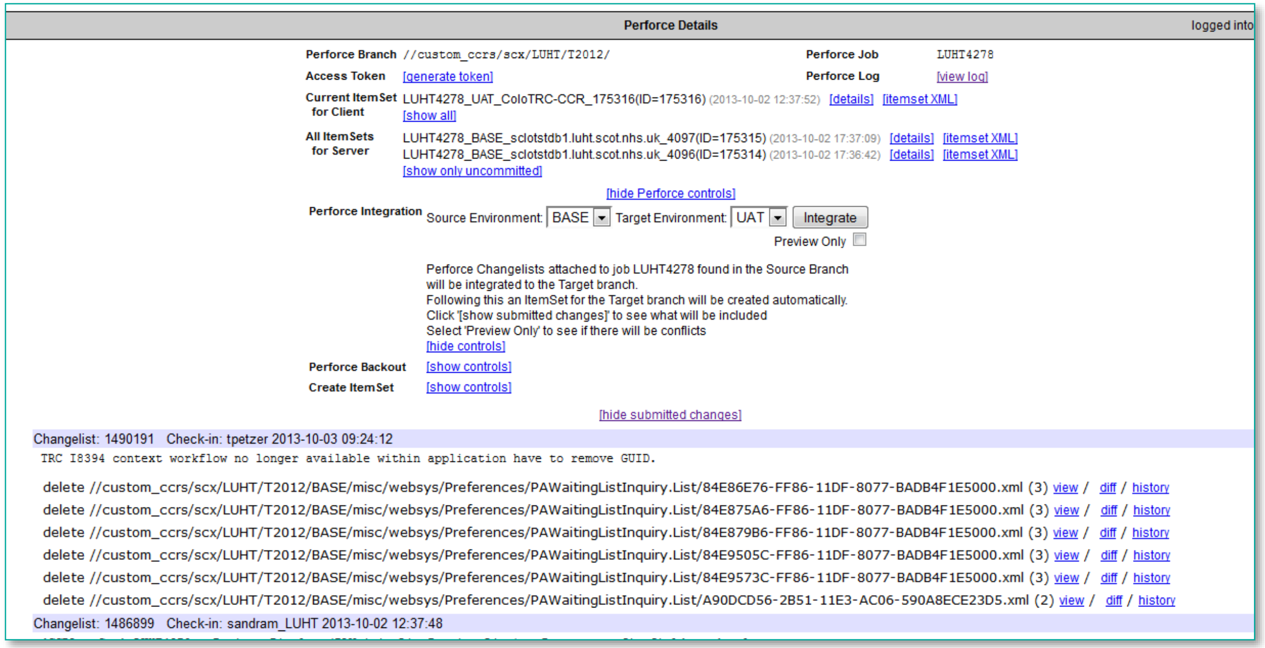
[show Perforce controls]

[hide submitted changes]

```
Changelist: 1490191   Check-in: tpetzer 2013-10-03 09:24:12
 TRC I8394 context workflow no longer available within application have to remove GUID.

 delete //custom_ccrs/scx/LUHT/T2012/BASE/misc/websys/Preferences/PAWaitingListInquiry.List/84E86E76-FF86-11DF-8077-BADB4F1E5000.xml (3) view /  diff / history
 delete //custom_ccrs/scx/LUHT/T2012/BASE/misc/websys/Preferences/PAWaitingListInquiry.List/84E875A6-FF86-11DF-8077-BADB4F1E5000.xml (3) view /  diff / history
 delete //custom_ccrs/scx/LUHT/T2012/BASE/misc/websys/Preferences/PAWaitingListInquiry.List/84E879B6-FF86-11DF-8077-BADB4F1E5000.xml (3) view /  diff / history
 delete //custom_ccrs/scx/LUHT/T2012/BASE/misc/websys/Preferences/PAWaitingListInquiry.List/84E9505C-FF86-11DF-8077-BADB4F1E5000.xml (3) view /  diff / history
 delete //custom_ccrs/scx/LUHT/T2012/BASE/misc/websys/Preferences/PAWaitingListInquiry.List/84E9573C-FF86-11DF-8077-BADB4F1E5000.xml (3) view /  diff / history
 delete //custom_ccrs/scx/LUHT/T2012/BASE/misc/websys/Preferences/PAWaitingListInquiry.List/A90DCD56-2B51-11E3-AC06-590A8ECE23D5.xml (2) view /  diff / history
Changelist: 1486899   Check-in: sandram_LUHT 2013-10-02 12:37:48
 {CCRAutoGen} LUHT4278 - Patient Display (PAWaitingListInquiry.List) ? Remove specfic fields and columns;
 Integrating Changelists for LUHT4278 from BASE to UAT

 integrate //custom_ccrs/scx/LUHT/T2012/UAT/misc/websys/Preferences/PAWaitingListInquiry.List/84E804EA-FF86-11DF-8077-BADB4F1E5000.xml (2) view /  diff / history
 integrate //custom_ccrs/scx/LUHT/T2012/UAT/misc/websys/Preferences/PAWaitingListInquiry.List/84E8112E-FF86-11DF-8077-BADB4F1E5000.xml (2) view /  diff / history
 integrate //custom_ccrs/scx/LUHT/T2012/UAT/misc/websys/Preferences/PAWaitingListInquiry.List/84E816CE-FF86-11DF-8077-BADB4F1E5000.xml (2) view /  diff / history
```

CCR - Key not Unique Index Resolution (cont.)

- Investigation steps:
  21. Select Show Perforce Controls hyperlink.
  22. Select Show Controls for Perforce Integration.
  23. Ensure the Preview checkbox is unchecked.
  24. Click Integrate.
  25. The new ItemSet is now available for the user to download into their environment.



Perforce Details | logged into

Perforce Branch //custom_ccrs/scx/LUHT/T2012/ | Perforce Job LUHT4278

Access Token [generate token] | Perforce Log [view log]

Current ItemSet for Client: LUHT4278_UAT_ColoTRC-CCR_175316(ID=175316) (2013-10-02 12:37:52) [details] [itemset XML]
[show all]

All ItemSets for Server:
LUHT4278_BASE_sclotstdb1.luht.scot.nhs.uk_4097(ID=175315) (2013-10-02 17:37:09) [details] [itemset XML]
LUHT4278_BASE_sclotstdb1.luht.scot.nhs.uk_4096(ID=175314) (2013-10-02 17:36:42) [details] [itemset XML]
[show only uncommitted]

[hide Perforce controls]

Perforce Integration: Source Environment: BASE Target Environment: UAT Integrate
Preview Only □

Perforce Changelists attached to job LUHT4278 found in the Source Branch
will be integrated to the Target branch.
Following this an ItemSet for the Target branch will be created automatically.
Click '[show submitted changes]' to see what will be included
Select 'Preview Only' to see if there will be conflicts
[hide controls]

Perforce Backout [show controls]

Create ItemSet [show controls]

[hide submitted changes]

Changelist: 1490191 Check-in: tpetzer 2013-10-03 09:24:12

```
TRC I8394 context workflow no longer available within application have to remove GUID.
```

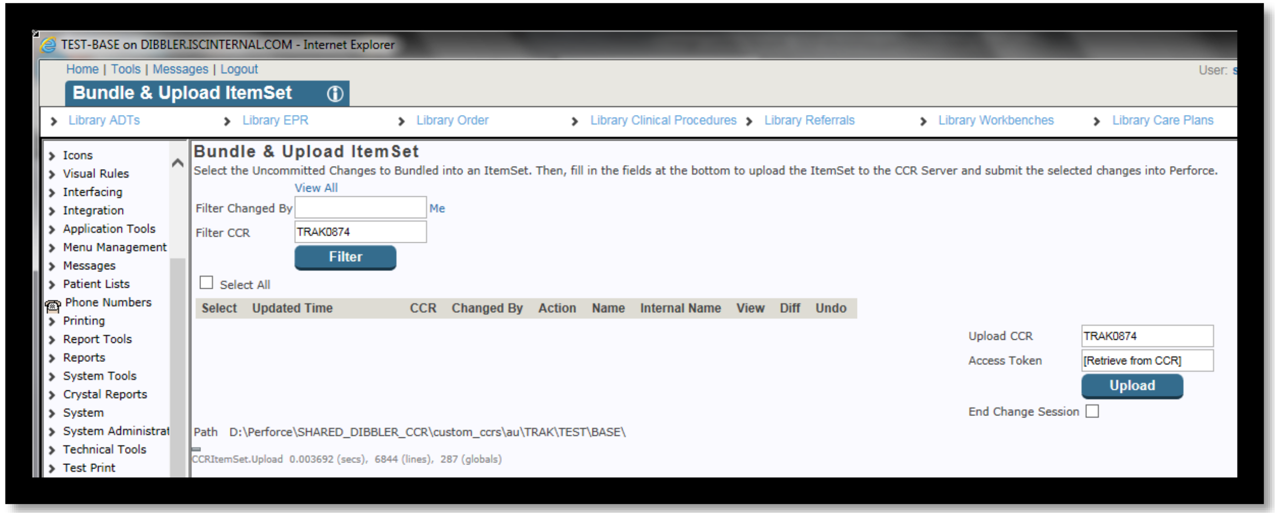
```
delete //custom_ccrs/scx/LUHT/T2012/BASE/misc/websys/Preferences/PAWaitingListInquiry.List/84E86E76-FF86-11DF-8077-BADB4F1E5000.xml (3) view / diff / history
delete //custom_ccrs/scx/LUHT/T2012/BASE/misc/websys/Preferences/PAWaitingListInquiry.List/84E875A6-FF86-11DF-8077-BADB4F1E5000.xml (3) view / diff / history
delete //custom_ccrs/scx/LUHT/T2012/BASE/misc/websys/Preferences/PAWaitingListInquiry.List/84E879B6-FF86-11DF-8077-BADB4F1E5000.xml (3) view / diff / history
delete //custom_ccrs/scx/LUHT/T2012/BASE/misc/websys/Preferences/PAWaitingListInquiry.List/84E9505C-FF86-11DF-8077-BADB4F1E5000.xml (3) view / diff / history
delete //custom_ccrs/scx/LUHT/T2012/BASE/misc/websys/Preferences/PAWaitingListInquiry.List/84E9573C-FF86-11DF-8077-BADB4F1E5000.xml (3) view / diff / history
delete //custom_ccrs/scx/LUHT/T2012/BASE/misc/websys/Preferences/PAWaitingListInquiry.List/A90DCD56-2B51-11E3-AC06-590A8ECE23D5.xml (2) view / diff / history
```

Changelist: 1486899 Check-in: sandram_LUHT 2013-10-02 12:37:48

Part 8: TCC Debugging – Additional issues

Changes are not appearing in the bundle queue

- Check the CCR exclusion list to ensure that the class has not been excluded from CCR.
- Check to make sure that you haven’t uploaded the changes to the incorrect CCR number.
- Network errors or invalid tokens blocking the changes uploading to CCR.



Bundle & Upload ItemSet

Select the Uncommitted Changes to Bundled into an ItemSet. Then, fill in the fields at the bottom to upload the ItemSet to the CCR Server and submit the selected changes into Perforce.

Filter Changed By
Filter CCR TRAK0874

□ Select All

Select	Updated Time	CCR	Changed By	Action	Name	Internal Name	View	Diff	Undo

Upload CCR TRAK0874
Access Token [Retrieve from CCR]
End Change Session □

Path D:\Perforce\SHARED_DIBBLER_CCR\custom_ccrs\au\TRAK\TEST\BASE\

```
CCRItemSet.Upload  0.003692 (secs),  6844 (lines),  287 (globals)
```

Changes will not undo within the bundle queue

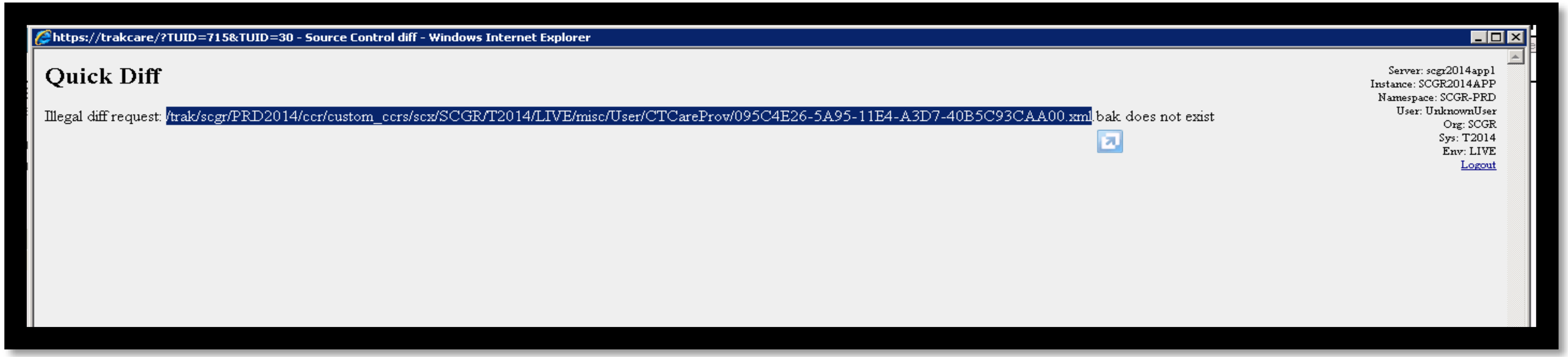
- It is suggested that undo within TrakCare should be attempted with caution as not all items can easily be “undone”. If you are going to attempt it, then try the following:
  - Ensure any changes you want to remove are not used within TrakCare, e.g., a new location is not being used by a patient you added for testing. If it is, then it cannot be undone as its still in use.
  - Undo in the reverse order.

Changes will not undo within the bundle queue (cont.)

- If you do run into issues its likely that it is down to a parent table being undone before child table or the changes are undone in the incorrect order.
  - First try adding an active CCR session with the same CCR number.

Changes will not undo within the bundle queue (cont.)

- Resolution if this fails:
  - Check the diff to see if it states illegal diff:



https://trakcare/?TUID=715&TUID=30 - Source Control diff - Windows Internet Explorer

Quick Diff

Illegal diff request: /trak/scgr/PRD2014/ccr/custom_ccrs/scx/SCGR/T2014/LIVE/misc/User/CTCareProv/095C4E26-5A95-11E4-A3D7-40B5C93CAA00.xml.bak does not exist

Server: scgr2014app1
Instance: SCGR2014APP
Namespace: SCGR-PRD
User: UnknownUser
Org: SCGR
Sys: T2014
Env: LIVE
Logout

  - This can be down to a .bak file not being generated when the change was made. The .bak file is the file that is used to revert the changes, in this case it has nothing to revert back to which is why the undo failed.

Changes will not undo within the bundle queue (cont.)

- Resolution if this fails:
  - Attempt to upload the changes to a CCR, then cancel and backout that CCR and download the new Itemset to remove changes. This is the preferred approach for multiple changes.

Quiz: Debugging Incorrect configuration

Question:

Which is the best way to undo changes from a bundle list?

A. Delete the changes from the change control table.

B. Undo in reverse order in the bundle list.

C. Undo in any order, as it doesn’t make any difference.

D. Upload to the CCR and follow the backout process.

Quiz: Debugging Incorrect configuration

Answer:

B. Undo in reverse order in the bundle list.

If you can, but recommended approach is:

D. Upload to the CCR and follow the backout process.

Part 9: Analytics

TrakCare Analytics – T2018

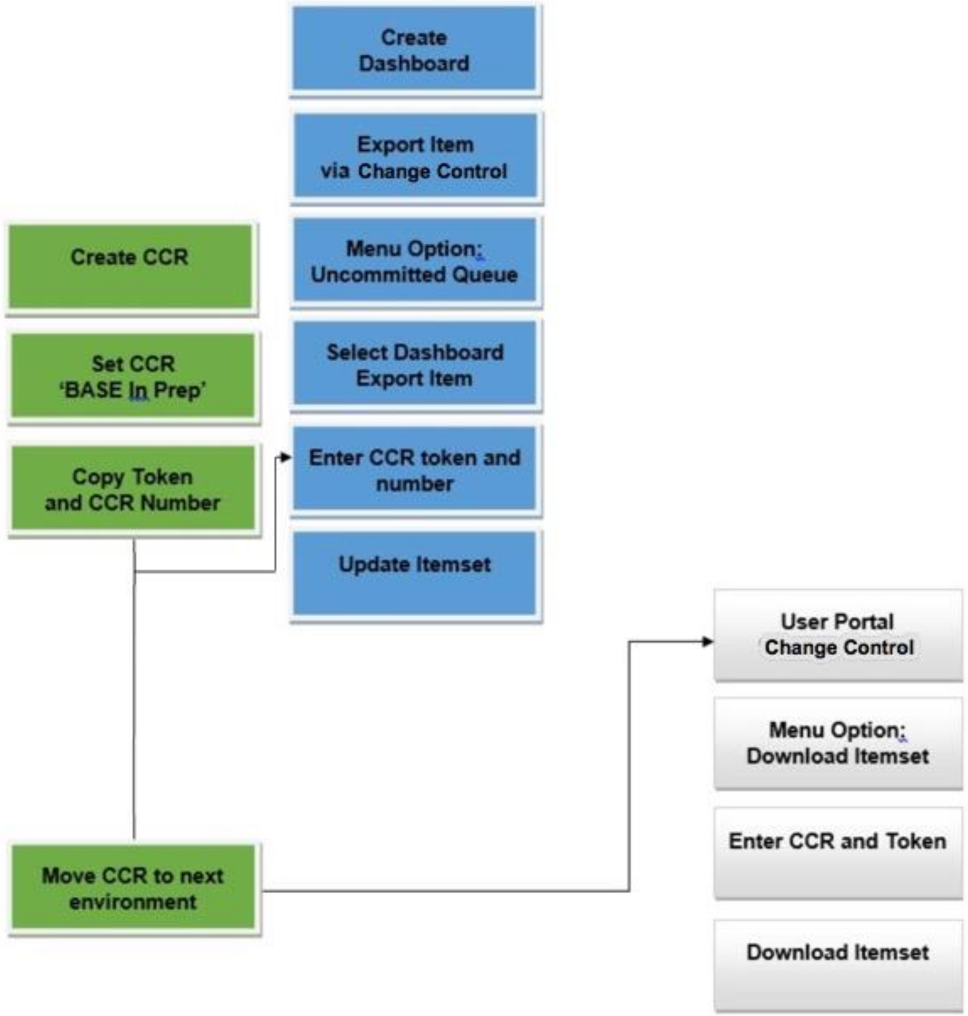
- TrakCare Analytics fully supports change control from T2018.
- Both SYS and Editions manage change control now via JIRA.
- Clients (extended analytics license) can manage change control via CCR.
- Process involves:
  - Configuration exported via new User Portal page.
  - Generates classes which appear in bundle queue.
  - Bundle and upload to promote to next environment.

TrakCare Analytics – T2018 (cont.)

- All DeepSee configuration is exposed as classes which supports.
  - Tier 1 CCR process.
  - Tier 1 baseline.

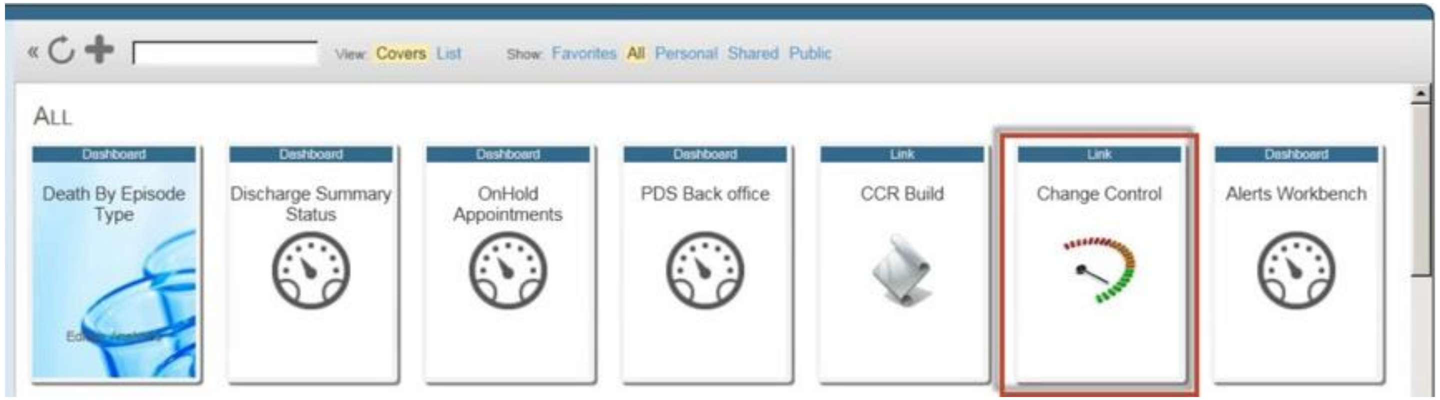
TrakCare Analytics – T2018

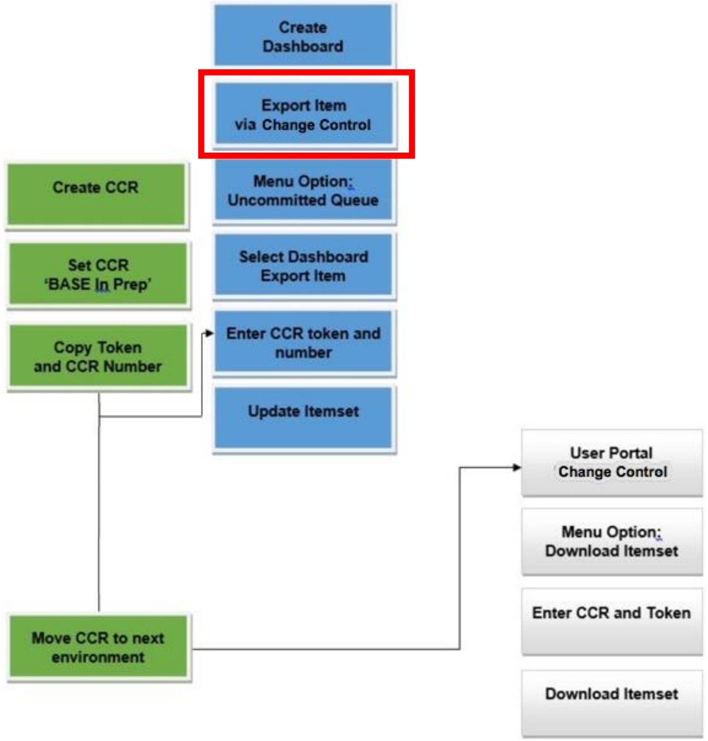
- Workflow process.
  - Create CCR.
    - Separate CCR System to TrakCare.
  - Create/Amend Analytics Configuration.
  - Export and move through CCR.


Create CCR
Set CCR 'BASE In Prep'
Copy Token and CCR Number
Move CCR to next environment
Create Dashboard
Export Item via Change Control
Menu Option: Uncommitted Queue
Select Dashboard Export Item
Enter CCR token and number
Update Itemset
User Portal Change Control
Menu Option: Download Itemset
Enter CCR and Token
Download Itemset

TrakCare Analytics – T2018 (cont.)

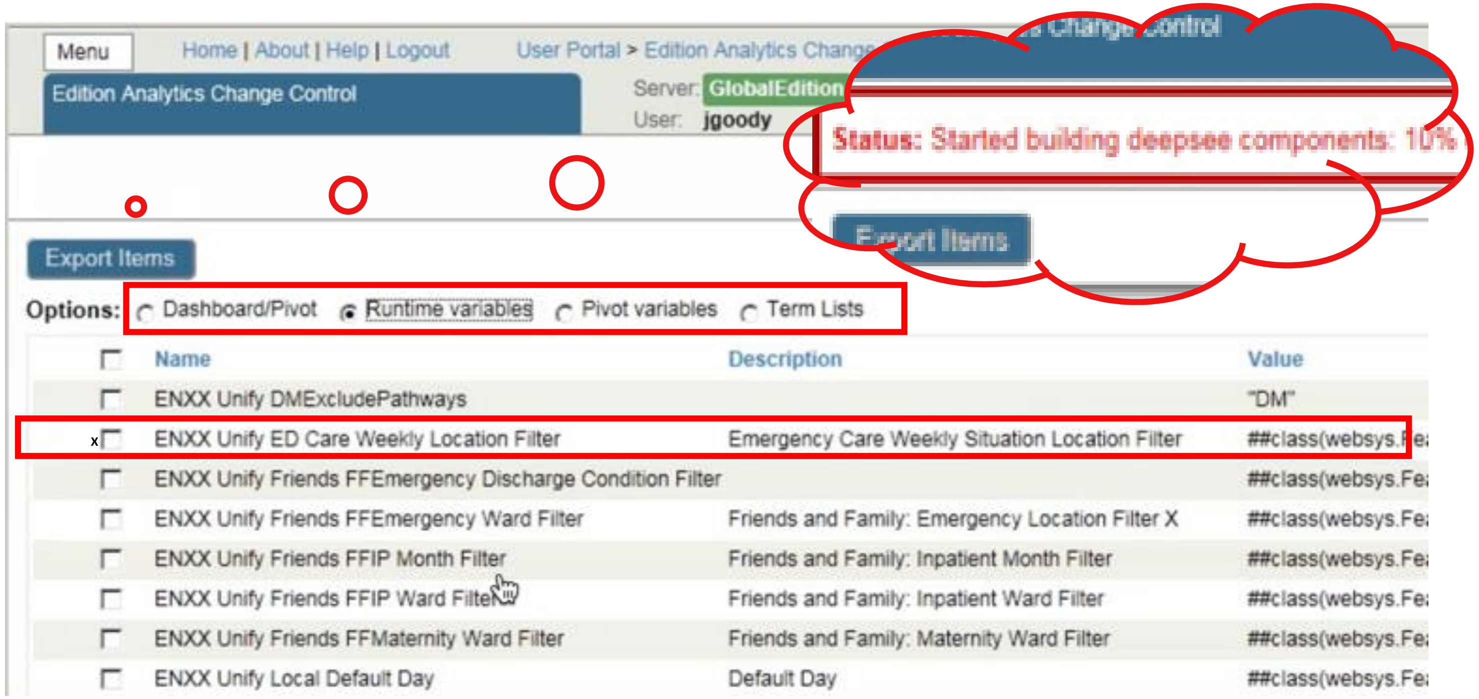
- Access DeepSee User Portal on Management Portal.
- Click on ‘Change Control’ menu.




Create Dashboard
Export Item via Change Control
Create CCR
Menu Option: Uncommitted Queue
Set CCR 'BASE In Prep'
Select Dashboard Export Item
Copy Token and CCR Number
Enter CCR token and number
Update Itemset
User Portal Change Control
Menu Option: Download Itemset
Move CCR to next environment
Enter CCR and Token
Download Itemset

TrakCare Analytics – T2018

- Select type from radio buttons.
- Select items from table to export.
- Click Export button – status shown.



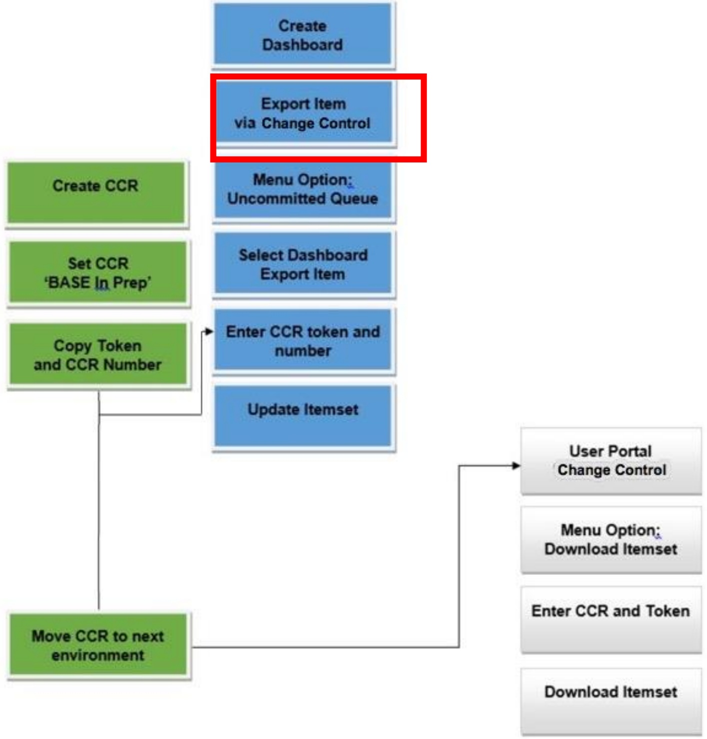
Menu
Home | About | Help | Logout
User Portal > Edition Analytics Change
Edition Analytics Change Control
Change Control

Server: GlobalEdition
User: jgoody

Status: Started building deepsee components: 10%

Options: ○ Dashboard/Pivot ● Runtime variables ○ Pivot variables ○ Term Lists

	Name	Description	Value
□	ENXX Unify DMExcludePathways		"DM"
x□	ENXX Unify ED Care Weekly Location Filter	Emergency Care Weekly Situation Location Filter	##class(websys.Fe
□	ENXX Unify Friends FFEmergency Discharge Condition Filter		##class(websys.Fe
□	ENXX Unify Friends FFEmergency Ward Filter	Friends and Family: Emergency Location Filter X	##class(websys.Fe
□	ENXX Unify Friends FFIP Month Filter	Friends and Family: Inpatient Month Filter	##class(websys.Fe
□	ENXX Unify Friends FFIP Ward Filter	Friends and Family: Inpatient Ward Filter	##class(websys.Fe
□	ENXX Unify Friends FFMaternity Ward Filter	Friends and Family: Maternity Ward Filter	##class(websys.Fe
□	ENXX Unify Local Default Day	Default Day	##class(websys.Fe



Create Dashboard

Export Item via Change Control

Menu Option: Uncommitted Queue

Select Dashboard Export Item

Enter CCR token and number

Update Itemset

Create CCR

Set CCR 'BASE In Prep'

Copy Token and CCR Number

Move CCR to next environment

User Portal Change Control

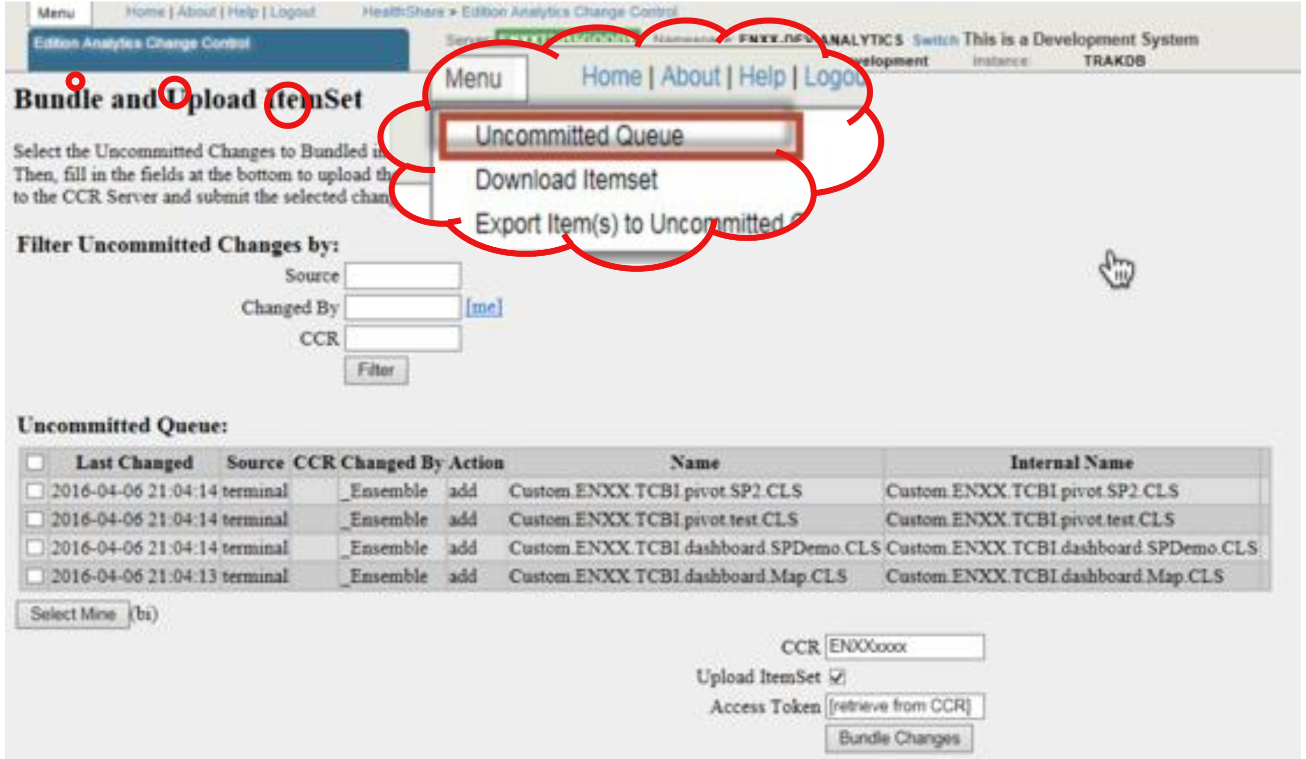
Menu Option: Download Itemset

Enter CCR and Token

Download Itemset

TrakCare Analytics – T2018

- Select ‘Uncommitted queue’ from menu.


Menu
Home | About | Help | Logout
HealthShare » Edition Analytics Change Control
Edition Analytics Change Control
ENXX-DEV-ANALYTICS
Switch
This is a Development System
TRAKDB
Menu
Home | About | Help | Logou
Uncommitted Queue
Download Itemset
Export Item(s) to Uncommitted

Bundle and Upload ItemSet

Select the Uncommitted Changes to Bundled i
Then, fill in the fields at the bottom to upload th
to the CCR Server and submit the selected chan

Filter Uncommitted Changes by:

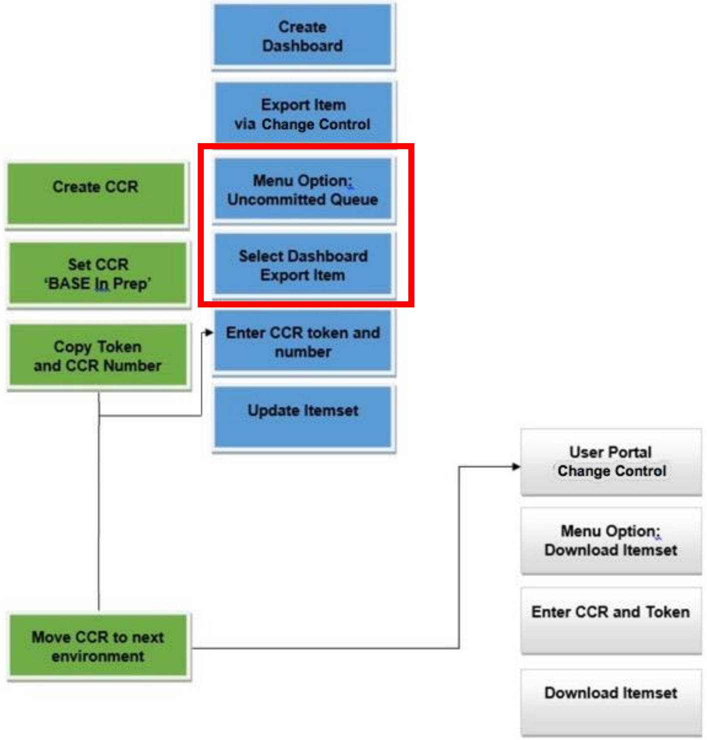
Source
Changed By [me]
CCR
Filter

Uncommitted Queue:

□	Last Changed	Source	CCR	Changed By	Action	Name	Internal Name
□	2016-04-06 21:04:14	terminal		_Ensemble	add	Custom.ENXX.TCBI.pivot.SP2.CLS	Custom.ENXX.TCBI.pivot.SP2.CLS
□	2016-04-06 21:04:14	terminal		_Ensemble	add	Custom.ENXX.TCBI.pivot.test.CLS	Custom.ENXX.TCBI.pivot.test.CLS
□	2016-04-06 21:04:14	terminal		_Ensemble	add	Custom.ENXX.TCBI.dashboard.SPDemo.CLS	Custom.ENXX.TCBI.dashboard.SPDemo.CLS
□	2016-04-06 21:04:13	terminal		_Ensemble	add	Custom.ENXX.TCBI.dashboard.Map.CLS	Custom.ENXX.TCBI.dashboard.Map.CLS

Select Mine (bi)

CCR ENXXxxxx
Upload ItemSet ☑
Access Token [retrieve from CCR]
Bundle Changes

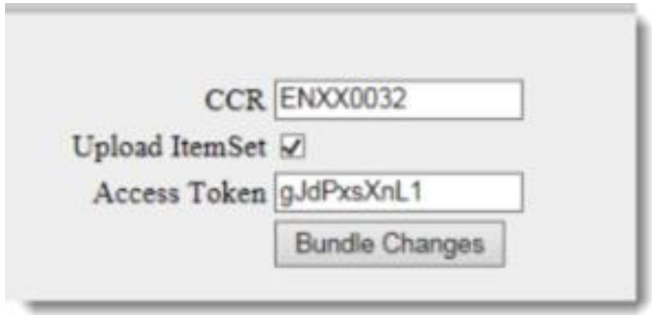


Create Dashboard
Export Item via Change Control
Menu Option: Uncommitted Queue
Select Dashboard Export Item
Create CCR
Set CCR ‘BASE In Prep’
Copy Token and CCR Number
Enter CCR token and number
Update Itemset
Move CCR to next environment
User Portal Change Control
Menu Option: Download Itemset
Enter CCR and Token
Download Itemset

- Select items to bundle to CCR.

TrakCare Analytics – T2018

- Enter CCR and access token.

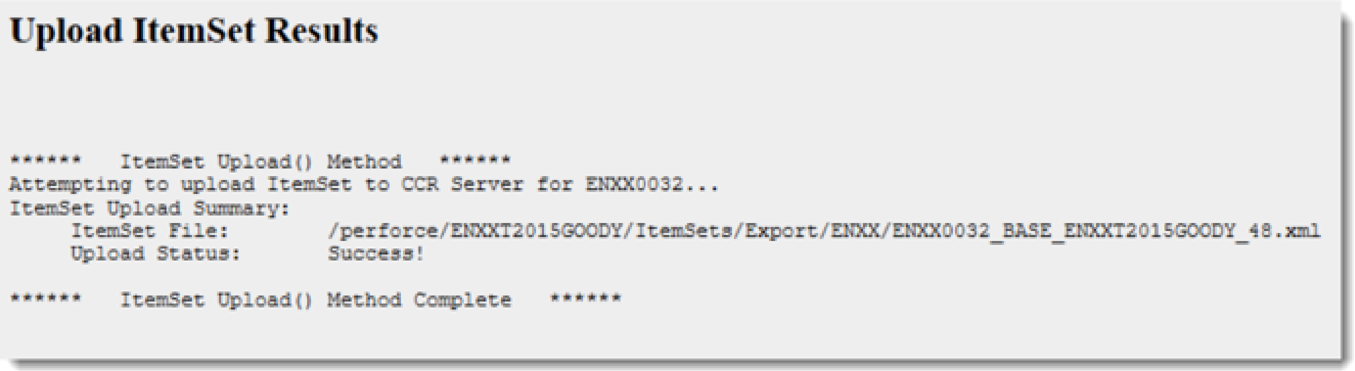


CCR ENXX0032

Upload ItemSet ☑

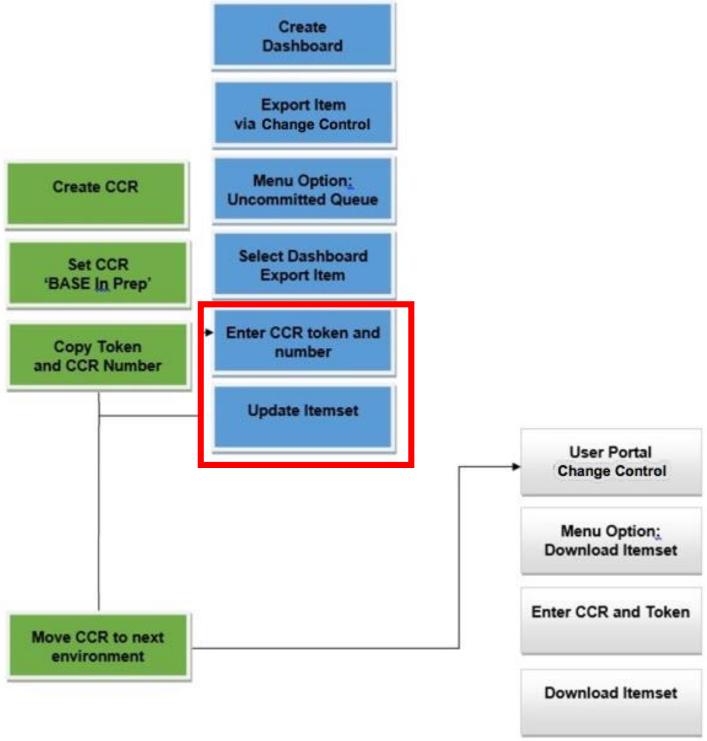
Access Token gJdPxsXnL1

- Click ’Bundle Changes.’



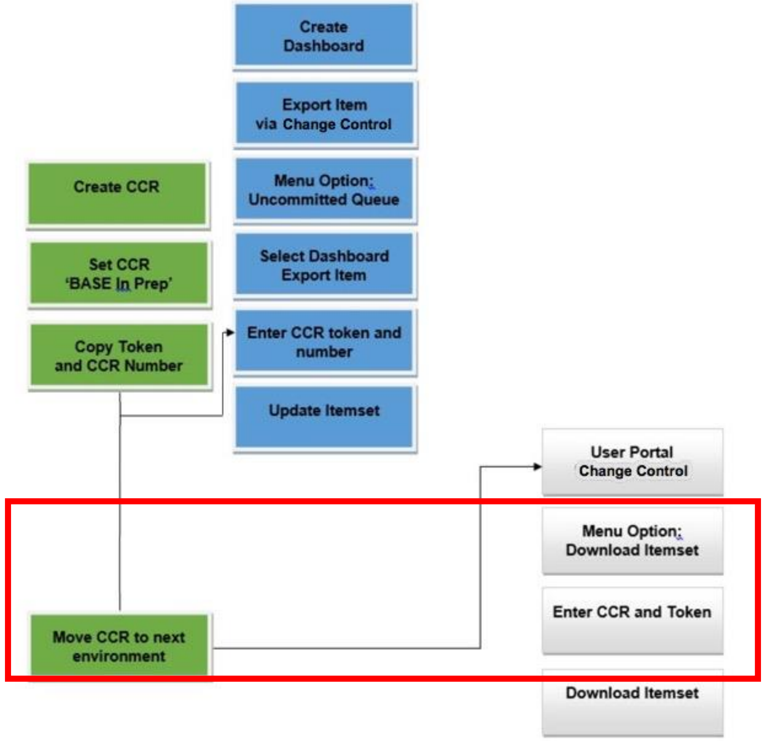
Upload ItemSet Results

```
******   ItemSet Upload() Method   ******
Attempting to upload ItemSet to CCR Server for ENXX0032...
ItemSet Upload Summary:
     ItemSet File:        /perforce/ENXXT2015GOODY/ItemSets/Export/ENXX/ENXX0032_BASE_ENXXT2015GOODY_48.xml
     Upload Status:       Success!

******   ItemSet Upload() Method Complete   ******
```


Create Dashboard
Export Item via Change Control
Create CCR
Menu Option: Uncommitted Queue
Set CCR 'BASE In Prep'
Select Dashboard Export Item
Copy Token and CCR Number
Enter CCR token and number
Update Itemset
User Portal Change Control
Menu Option: Download Itemset
Move CCR to next environment
Enter CCR and Token
Download Itemset

TrakCare Analytics – T2018

- Log on to Management Portal for next environment.
- Access DeepSee User Portal page.
- Click on 'Change Control' link.
- From Menu select 'Download Itemset.'



Create Dashboard

Export Item via Change Control

Create CCR

Menu Option: Uncommitted Queue

Set CCR 'BASE In Prep'

Select Dashboard Export Item

Copy Token and CCR Number

Enter CCR token and number

Update Itemset

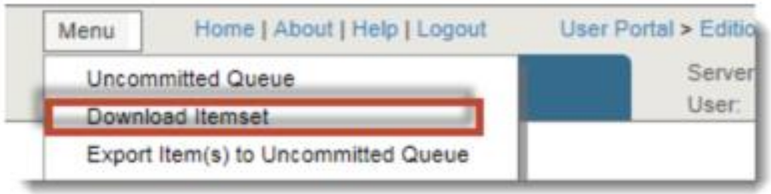
User Portal Change Control

Menu Option: Download Itemset

Enter CCR and Token

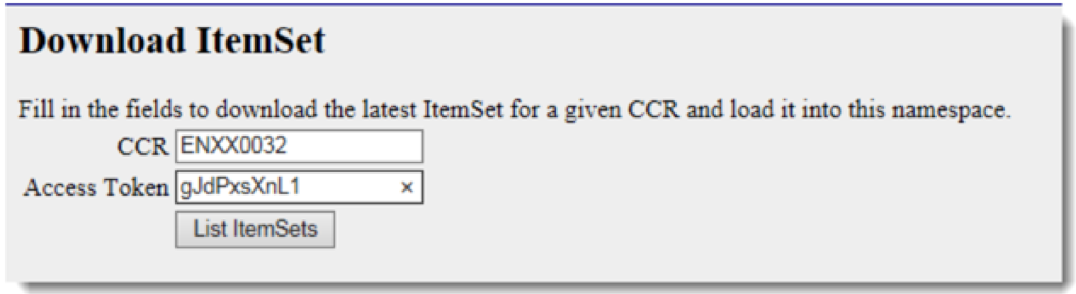
Move CCR to next environment

Download Itemset



TrakCare Analytics – T2018

- Enter CCR number and token and click ’List ItemSets.’

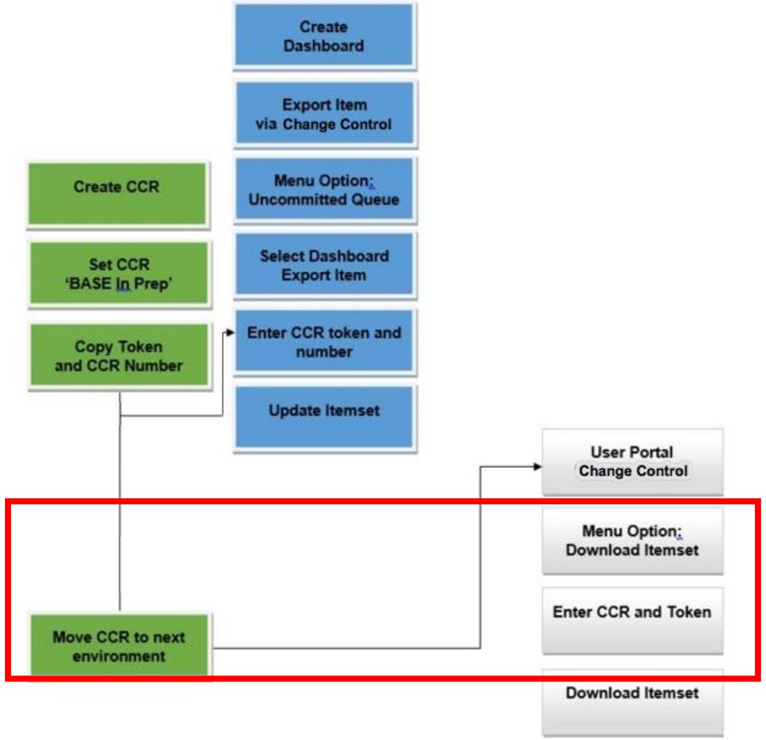


Download ItemSet

Fill in the fields to download the latest ItemSet for a given CCR and load it into this namespace.

CCR ENXX0032

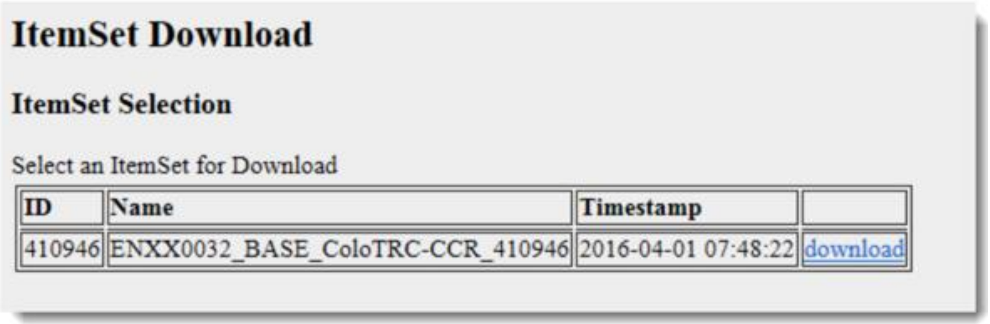
Access Token gJdPxsXnL1



Create Dashboard
Export Item via Change Control
Create CCR
Menu Option: Uncommitted Queue
Set CCR 'BASE In Prep'
Select Dashboard Export Item
Copy Token and CCR Number
Enter CCR token and number
Update Itemset
User Portal Change Control
Menu Option: Download Itemset
Enter CCR and Token
Move CCR to next environment
Download Itemset

TrakCare Analytics – T2018

- Click link to download ItemSet.



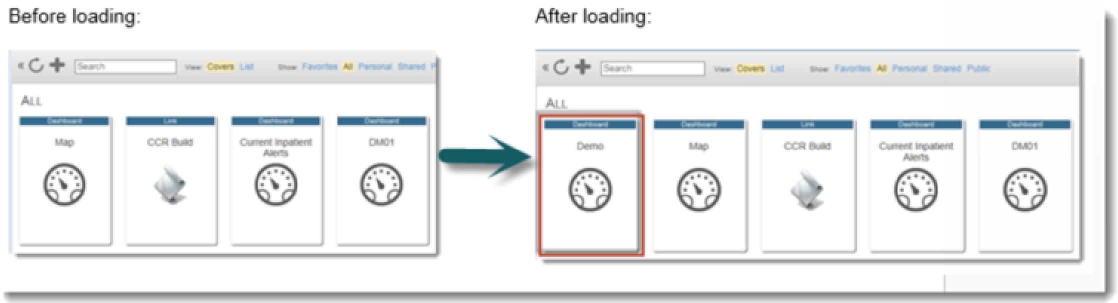
ItemSet Download

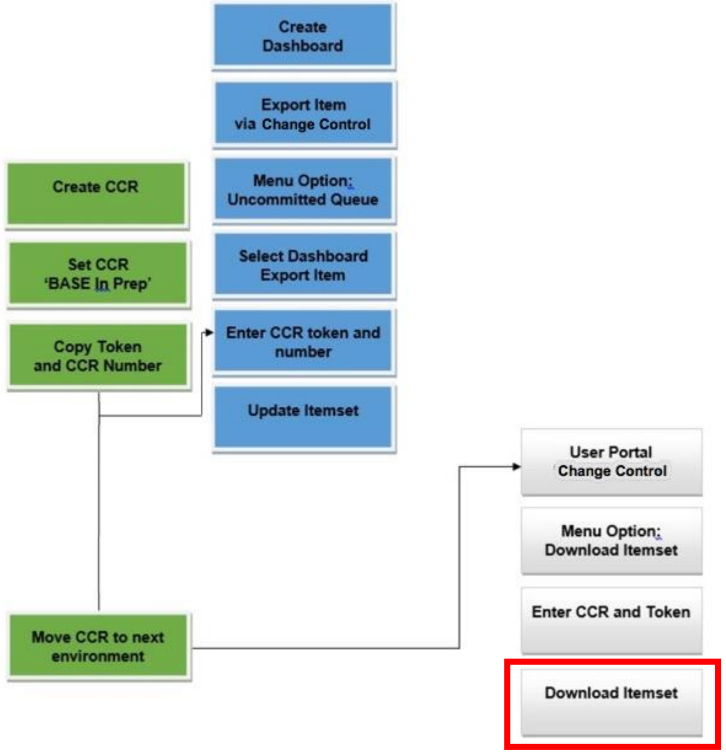
ItemSet Selection

Select an ItemSet for Download

ID	Name	Timestamp	
410946	ENXX0032_BASE_ColoTRC-CCR_410946	2016-04-01 07:48:22	download

- Once downloaded changes now in environment.

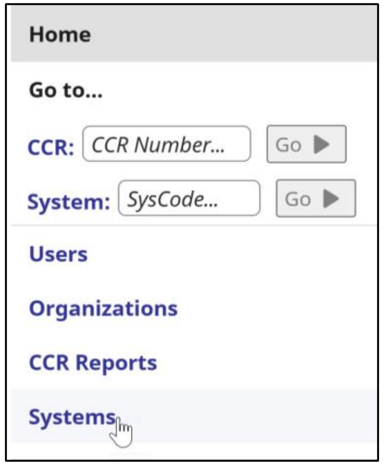


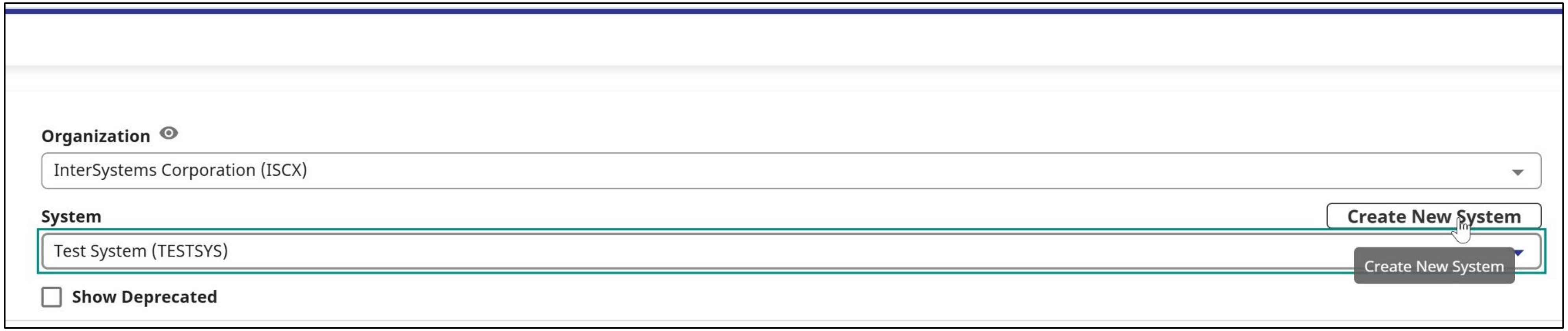

Create Dashboard
Export Item via Change Control
Menu Option: Uncommitted Queue
Select Dashboard Export Item
Enter CCR token and number
Update Itemset
Create CCR
Set CCR 'BASE In Prep'
Copy Token and CCR Number
Move CCR to next environment
User Portal Change Control
Menu Option: Download Itemset
Enter CCR and Token
Download Itemset

Part 10: Configure TrakCare Change Control

CCR Setup: Create CCR System

- Go to main menu > Systems.
- Choose Organization.
- Click Create New System button.

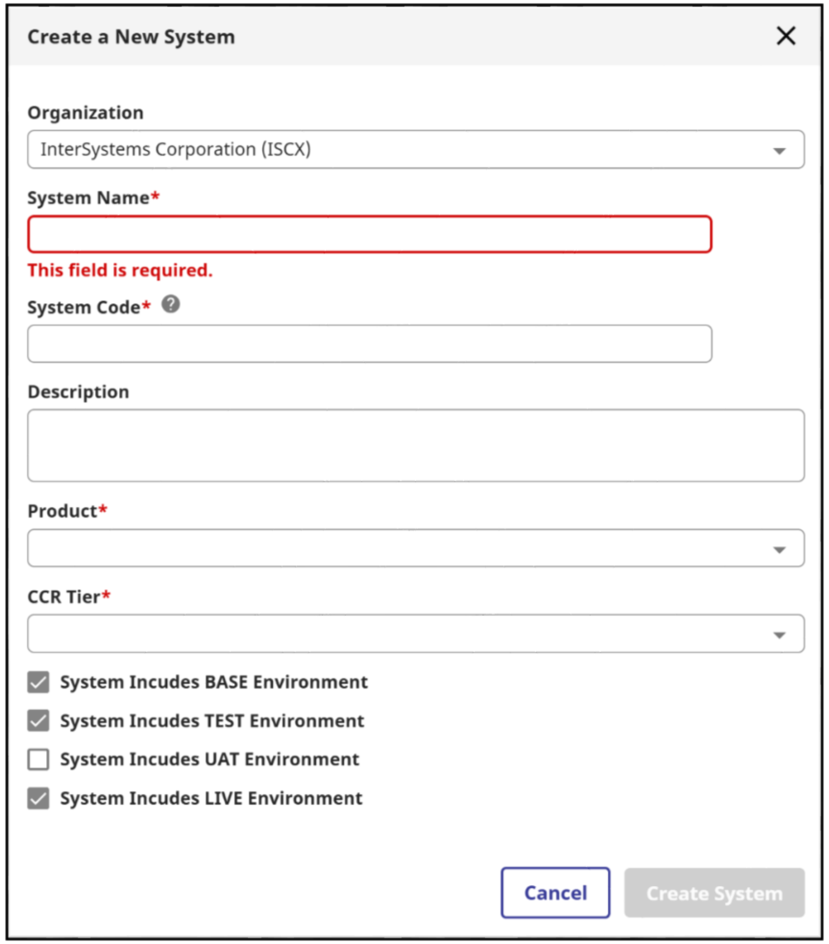



Organization
InterSystems Corporation (ISCX)

System
Test System (TESTSYS)

CCR Setup continued

- Complete fields:
  - System Name.
  - System Code.
  - Description.
  - Product.
  - CCR Tier.
- Select Environments System includes.
- Click Create System.



Create a New System

Organization

InterSystems Corporation (ISCX)

System Name*

This field is required.

System Code*

Description

Product*

CCR Tier*

☑ System Incudes BASE Environment

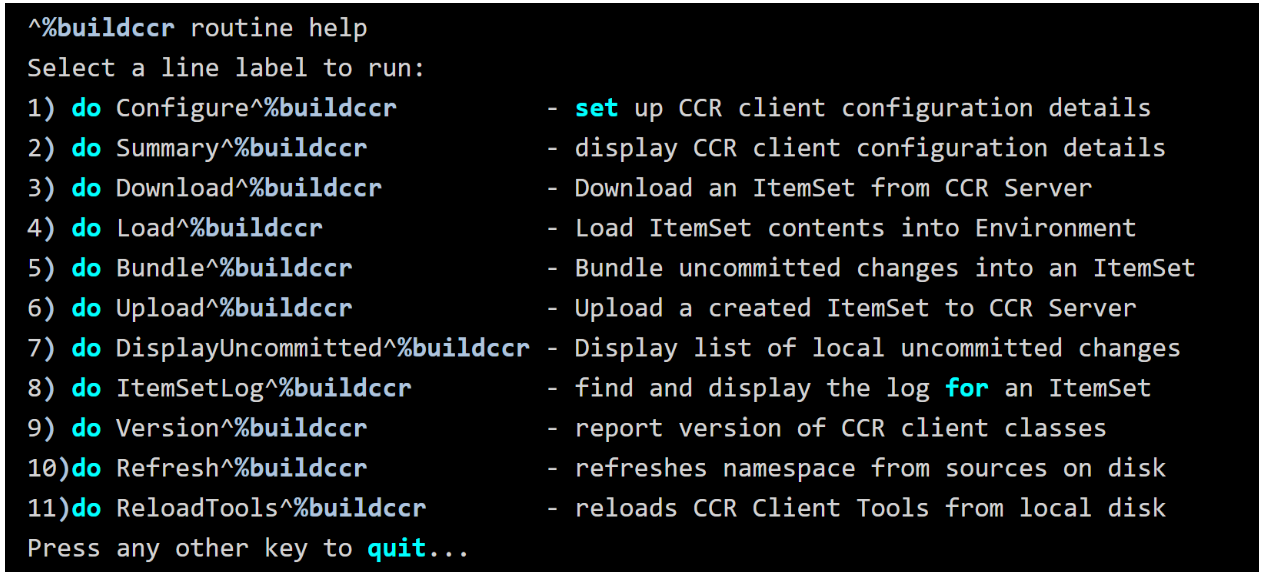
☑ System Incudes TEST Environment

□ System Incudes UAT Environment

☑ System Incudes LIVE Environment

Configuring CCR Client Tools

- Open Terminal session or Putty on target instance.
- Switch to namespace to be configured.
- Enter `do Configure^%buildccr`.

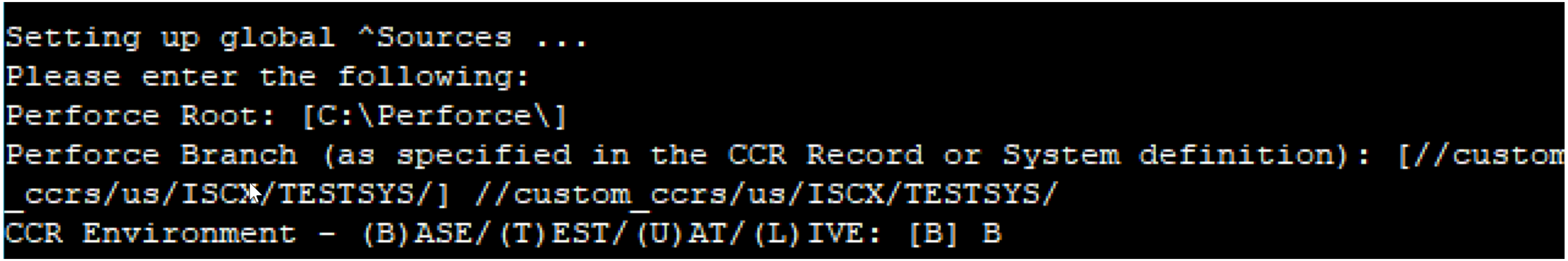


```
^%buildccr routine help
Select a line label to run:
1) do Configure^%buildccr         - set up CCR client configuration details
2) do Summary^%buildccr           - display CCR client configuration details
3) do Download^%buildccr          - Download an ItemSet from CCR Server
4) do Load^%buildccr              - Load ItemSet contents into Environment
5) do Bundle^%buildccr            - Bundle uncommitted changes into an ItemSet
6) do Upload^%buildccr            - Upload a created ItemSet to CCR Server
7) do DisplayUncommitted^%buildccr - Display list of local uncommitted changes
8) do ItemSetLog^%buildccr        - find and display the log for an ItemSet
9) do Version^%buildccr           - report version of CCR client classes
10)do Refresh^%buildccr           - refreshes namespace from sources on disk
11)do ReloadTools^%buildccr       - reloads CCR Client Tools from local disk
Press any other key to quit...
```

Configure^%buildccr prompts

- Perforce Root.
  - Location on file system where source workspace structures will be built.
    - Example: C:\Source.
- Perforce Branch.
  - Copy paste from Perforce Details pane of CCR documenting configuration process.
    - Example: //custom_ccrs/us/ISCU/TESTSYS/.
- CCR environment.
    - Type character corresponding to environment type:
    - B for BASE / T for TEST / U for UAT / L for LIVE.

Configure^%buildccr prompts



```
Setting up global ^Sources ...
Please enter the following:
Perforce Root: [C:\Perforce\]
Perforce Branch (as specified in the CCR Record or System definition): [//custom
_ccrs/us/ISCX/TESTSYS/] //custom_ccrs/us/ISCX/TESTSYS/
CCR Environment - (B)ASE/(T)EST/(U)AT/(L)IVE: [B] B
```

Prompts for Non-BASE Environments

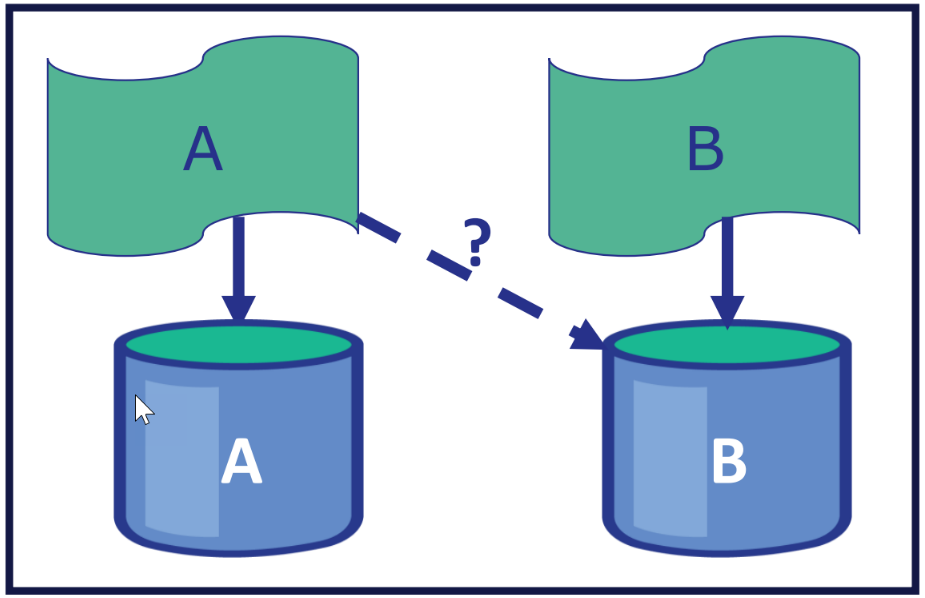
- Lock environment.
  - Disables check out and add to source control functionality to prevent editing any item in source control.
  - **Always respond yes.**
- If respond yes to lock, prompts whether to Admin Lock.
  - Admin lock removes information in UI on how to unlock environment.
    - Most users should not even know how to unlock environment.
    - Unlocking environment causes missing changes because Perforce no longer source of truth.
  - Best practice: yes.

Configure^%buildccr Prompts (cont.)

- Add or Change any CSP mappings.
  - Usually respond no.
- Automatically tag with Perforce version.
  - Adds SrcVer parameter when saving class definition if it does not exist.
  - Indicates branch and revision of class retrieved from Perforce.
  - Best Practice: yes.

Configure^%buildccr Prompts (cont.)

- Ensure mapped items cannot be changed in other namespaces.
- Treat items mapped from other databases as read-only in Studio.
  - Also applies to Management Portal pages with hooks.
  - Yes = can only edit items in default database(s) of namespace.
  - Best practice:
    - Tier 1 = yes.
    - Tier 2 = probably no.
- Use separate Systems for namespace A and B.



Configure^%buildccr Prompts (cont.)

- Keep history of changes.
  - Determines whether to maintain or delete metadata for changes.
    - Username, time stamp, filename, and more.
    - Metadata stored in %Studio_SourceControl.Change table.
    - Does not maintain copies of changes items or what was changed; only metadata.
  - Yes = set committed field for that change to true, with timestamp.
  - No = committed changes deleted from change table.
  - Best practice: yes because can be useful in debugging.
    - Consider setting up task to purge %Studio_SourceControl.Change table.

Configure^%buildccr Prompts (cont.)

- Will this namespace ever communicate directly with the ISC Perforce server?
  - (Y)es if want connected mode.
  - **(N)o if want disconnected mode.**
  - Remember:
    - All customers work in disconnected mode.

Configure^%buildccr prompts (cont.)

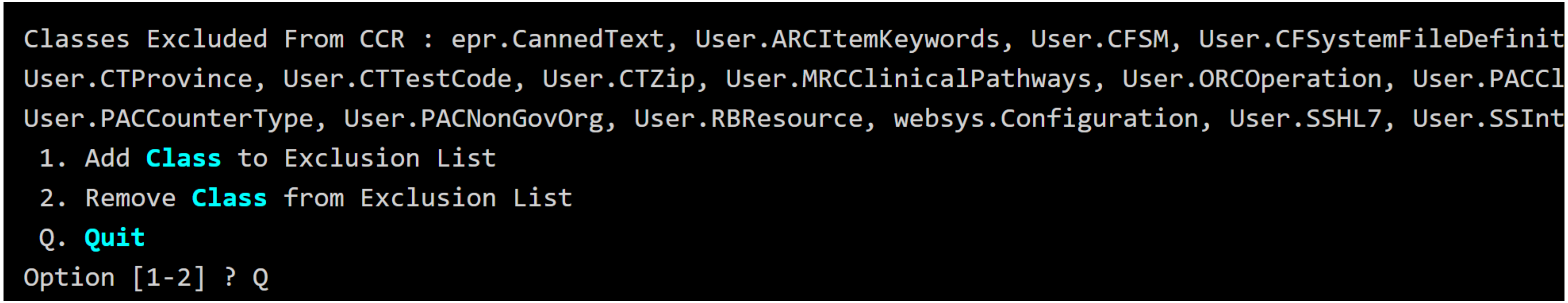
- Will you be importing/exporting ItemSets from this namespace?
  - Always respond Yes.
  - Creates /itemsetsourcelink CSP Application for use in CCR Transport if it does not exist.
  - Creates (if necessary) and initializes ItemSetClient SSL configuration.
- Configuration summary displayed.
  - Copy-paste to Testing Steps Taken in XXX field.

Result of Configure^%buildccr

- Environment configured to use CCR.
- Within Studio, when create new item, will be prompted to add to source control.
- After baselining, existing items must be checked out in BASE for editing.
- In non-BASE environments, only able to edit items in source control through CCR process.
- Studio will automatically import and export items added to source control in the proper format and into the proper directory structure.

Tier 2 CCR Exclusion List

- Check the CCR Exclusion List.
- Within terminal run:
  - Do ^SSADMIN
    - Select option 5 (Change Control).
    - Select option 3 (Manage Exclusion List).
    - If this is blank then get the standard set of exclusion items for each environment from TrakCare Exclusion List.



```
Classes Excluded From CCR : epr.CannedText, User.ARCItemKeywords, User.CFSM, User.CFSystemFileDefinit
User.CTProvince, User.CTTestCode, User.CTZip, User.MRCClinicalPathways, User.ORCOperation, User.PACCl
User.PACCounterType, User.PACNonGovOrg, User.RBResource, websys.Configuration, User.SSHL7, User.SSInt
 1. Add Class to Exclusion List
 2. Remove Class from Exclusion List
 Q. Quit
Option [1-2] ? Q
```

Summary

- What are the key points for this module?