



INTERSYSTEMS LEARNING SERVICES

InterSystems Change Control



InterSystems Change Control

COPYRIGHT NOTICE

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

InterSystems is a registered trademark of InterSystems Corporation.



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

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

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

This document contains trade secret and confidential information which is the property of InterSystems Corporation, One Memorial Drive, Cambridge, MA 02142, or its affiliates, and is furnished for the sole purpose of the operation and maintenance of the products of InterSystems Corporation. No part of this publication is to be used for any other purpose, and this publication is not to be reproduced, copied, disclosed, transmitted, stored in a retrieval system or translated into any human or computer language, in any form, by any means, in whole or in part, without the express prior written consent of InterSystems Corporation.

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

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

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

For Support questions about any InterSystems products, contact:

InterSystems WorldWide Response Center

Telephone: +1-617-621-0700

Tel: +44 (0) 844 854 2917

Email: support@InterSystems.com

ICC 510: CCR Tier 1 – Usage Basics



Objectives

- Demonstrate the CCR process to develop and progress a CCR Tier 1 (Source-Controlled) change through a CCR managed System.
- Describe the programmatic tools available to tailor code to CCR environments.
- Explain the differences between submitting changes from a Connected vs a Disconnected BASE environment.
- Progress Tier 1 CCRs through systems with secondary environments.



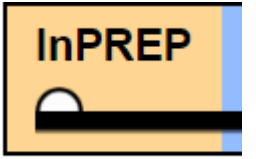
Review: Client Tools

- Most functionality discussed comes from Client Tools.
 - Library of code shared by all namespaces in instance.
- Client Tools improved several times per year.
- If your functionality or screens differ from slides, most likely need to update client tools.
 - Low-risk change.
- To update:
 - Choose 1 system to always maintain Client Tools.
 - Click update Client Tools on system details page.
 - Follow implementation and testing plan of generated CCR.



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.



Quiz: In_PREP for Tier 1

Question:

When planning work during In_PREP for a Tier 1 CCR, a user should 'check out' any files from BASE which they think they will change in order to ensure that others know the scope of the work they intend to do. True or False?

Answer:

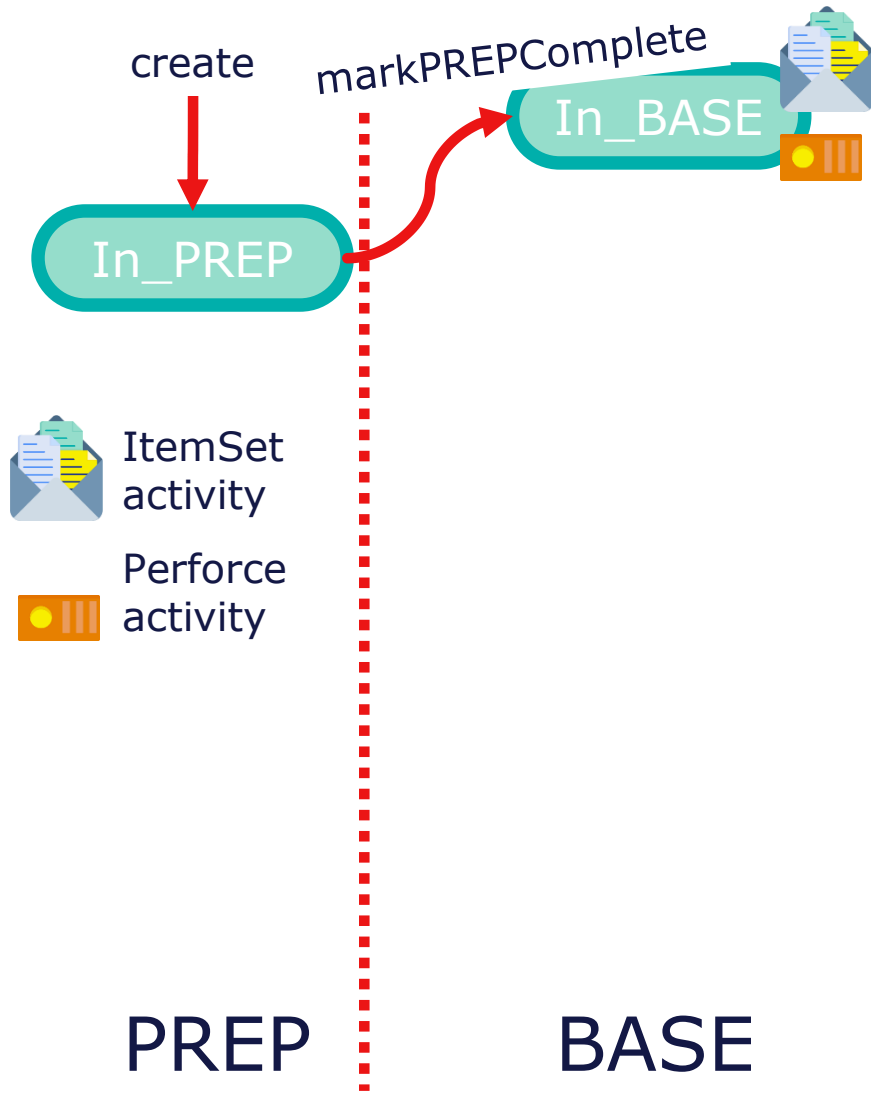
False. Nothing in BASE should be touched while a CCR is still in the PREP phase.



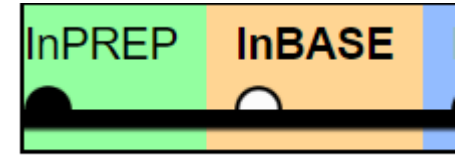
Part 2: CCR BASE Phase: Making Changes



Progressing a Tier 1 CCR...



Tier 1 Considerations: In_BASE

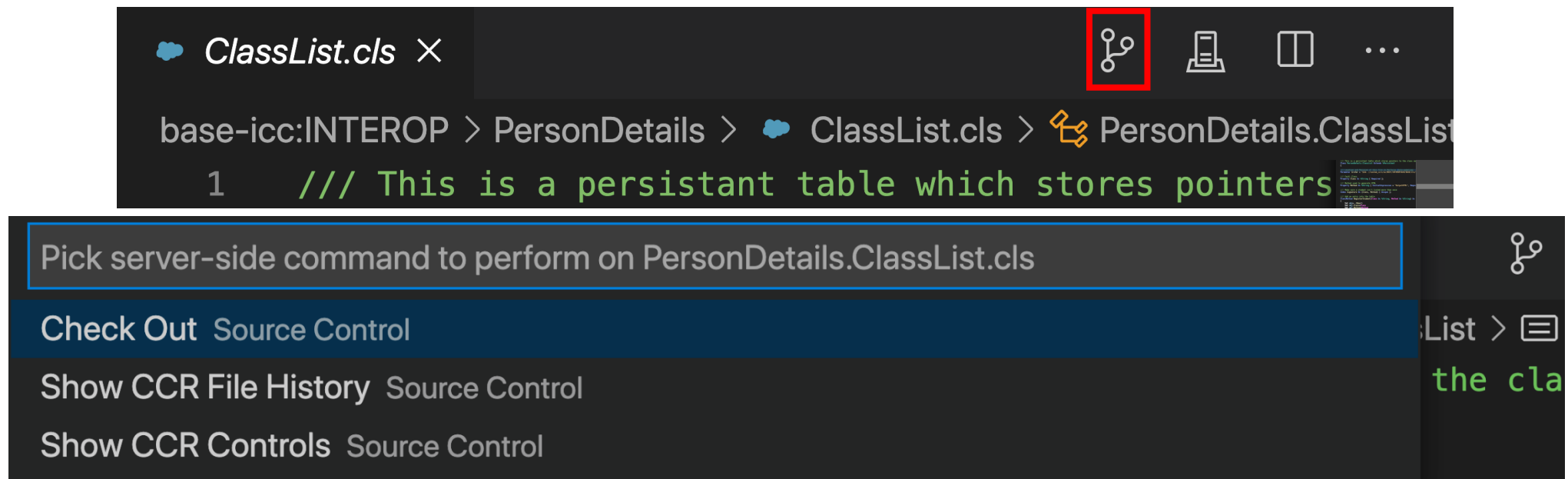


- Work performed via IDE and/or the Management Portal.
- In order to progress, record must have either:
 - Changelist.
 - Uncommitted ItemSet.
- Implementation Plan and Backout Plan have default values
 - Only change if manual steps needed.



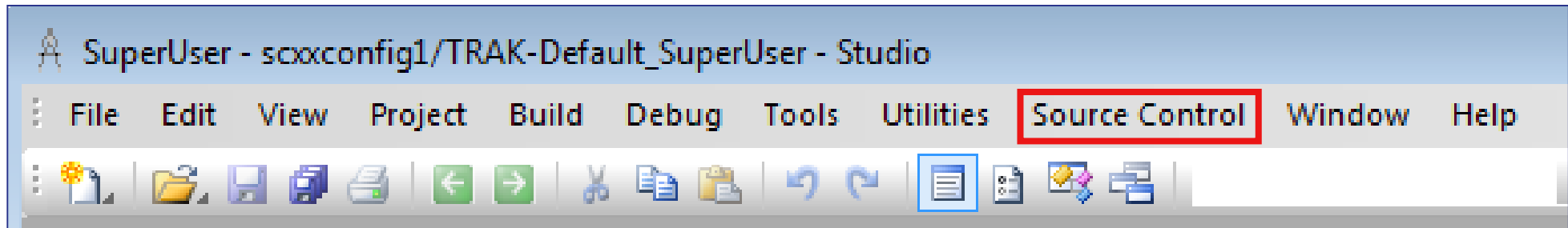
Using VS Code - ObjectScript with CCR

- Use server-side mode to connect to BASE.
 - Can download workspace definition from System Details page in CCR.
- Source Control button appears when viewing items in CCR configured namespace.

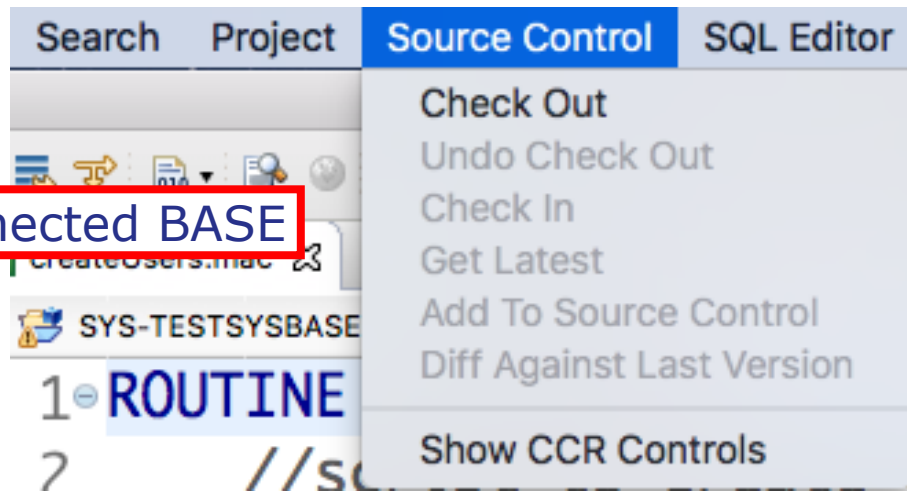


Using Studio with CCR

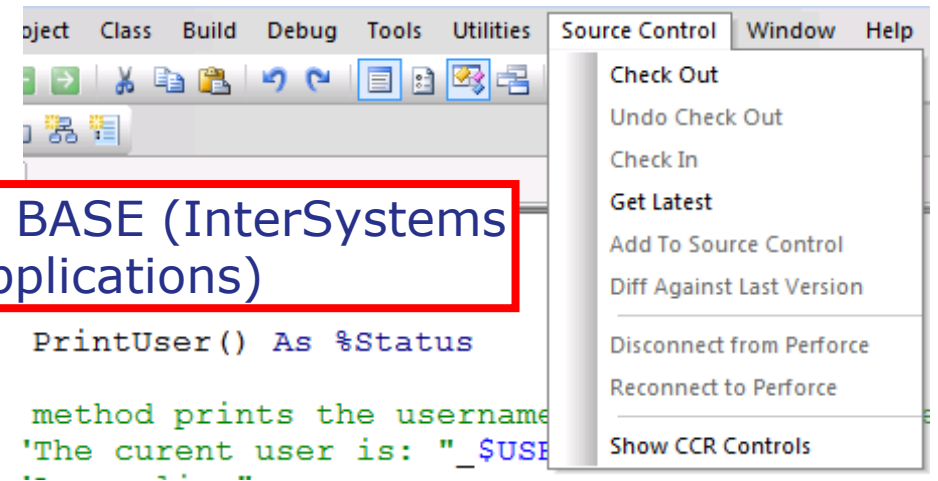
- Source Control menu available when connected to CCR configured namespace.



Disconnected BASE



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.



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.



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.



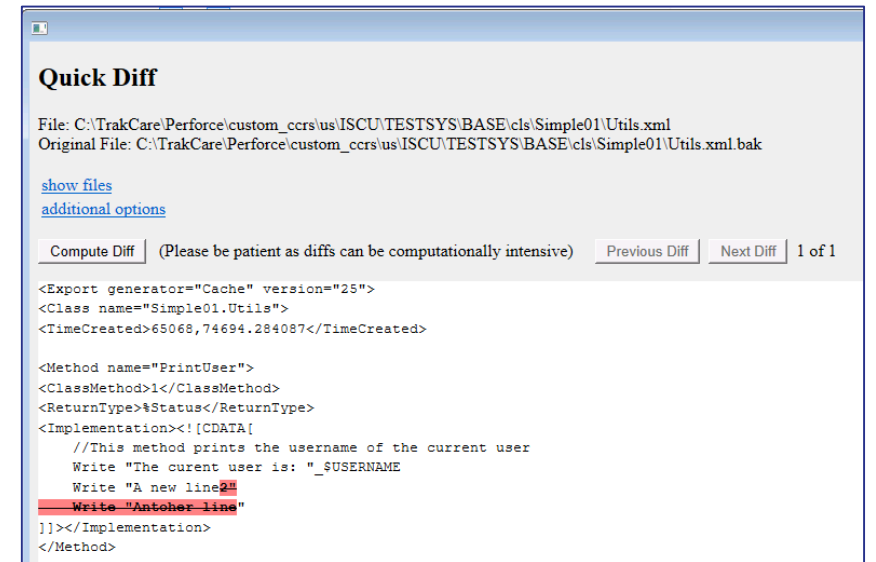
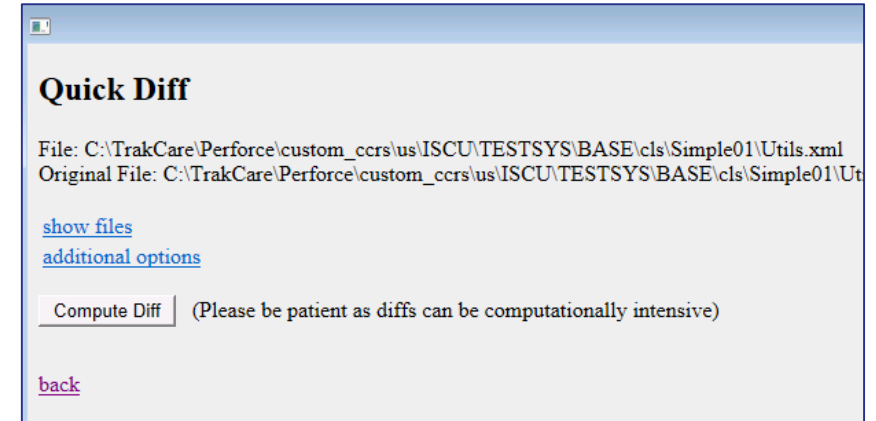
Identify Changes Made to Checked Out File

- 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.
- Client tools before 2019-Sep-20, click Compute Diff.
 - Current client tools automatically compute.



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: 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`.

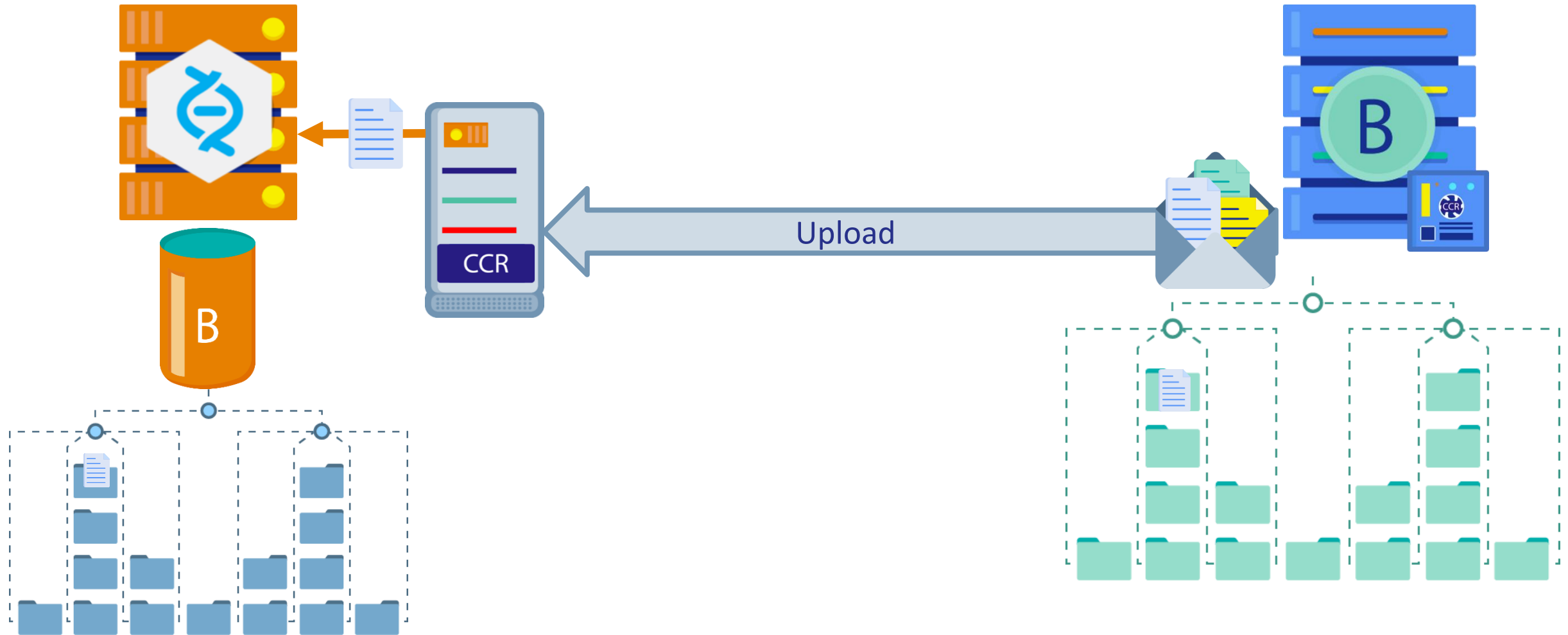
Answer: A



Part 4: CCR BASE Phase: Submitting Changes



Review: Sending Items from **Disconnected** BASE



Bundle Screen

- From IDE, Source Control > Show CCR Controls > ItemSet Bundle.
- Filters have no effect on what is bundled.
- Diff link discussed on previous slide.
- Select what to bundle.
- Access Token and CCR ID from CCR.
- Confirm CCR Details.
 - Validates CCR state and System.
- Click Bundle and Upload.

Filter Uncommitted Changes by:

Source

Changed By [me]

CCR

Uncommitted Queue:

<input checked="" type="checkbox"/>	Last Changed	Source	CCR	Changed By	Action	Name	Internal Name	
<input checked="" type="checkbox"/>	2019-12-16 14:13:19	studio		SuperUser	edit	Simple01.Utils.CLS	Simple01.Utils.CLS	diff

Select Mine (SuperUser)

Access Token ✓

CCR ✓

CCR Details

CCR	ISCU0020
Title	ICC450 CCR1: Implement Simple01.Utils.PrintUser()
Owner	student01
State	In_BASE
SystemCode	TESTSYS
SiteCode	ISCU

Path: C:\TrakCare\Perforce\custom_ccrs\us\ISCU\TESTSYS\BASE\



Review: Sending Items from **Connected** BASE



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.

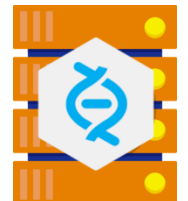
Perforce Checkin
Select files to submit for Perforce user: 'bspead'

<input checked="" type="checkbox"/>	edit: //custom_ccrs/us/ISCX/CCR/BASE/cspapp/ccr/edit.csp	diff
-------------------------------------	--	----------------------

Description (required):
ISCX12943 - Provide JS Alerts for blocked transitions

Perforce Job (optional):

Submit



Connected: CCR Client → Perforce

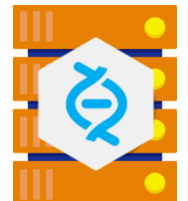
- To submit changes from a Connected BASE:
 5. Copy and paste <CCR ID> - <Title> from CCR title bar to Description. Include additional details if desired.
 6. Copy CCR ID and paste into Job field (required).
 7. Click "Submit."

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):



Verify Submitted Changes


- After submit changes, always verify correct items listed in CCR.
 - Under Performe 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](#)

Itemset Details

Submitted Changes (1)

Create Itemset




Perforce Integration




Perforce Backout

[Changelist 818](#)

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


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

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

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



Bundle and Upload Validation (Client tools 2019-Dec-12)

- 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.

Uncommitted Queue:

<input checked="" type="checkbox"/>	Last Changed	Source	CCR	Changed By	Action	Name	Internal Name	
<input checked="" type="checkbox"/>	2019-12-16 14:13:19	studio		SuperUser	edit	Simple01.Utils.CLS	Simple01.Utils.CLS	diff

Select Mine

(SuperUser)

Access Token

x

CCR

✓

Bundle and Upload Changes

Path:

C:\TrakCare\Perforce\custom_ccrs\us\ISCU\TESTSYS\BASE\



Bundle and Upload: Invalid State

- Valid states:
 - In_BASE.
 - BASE_Pending_Peer_Review.
- Solution:
 - If current state In_PREP perform markPREPComplete transition.
 - If 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 appropriately.

Uncommitted Queue:

<input checked="" type="checkbox"/>	Last Changed	Source	CCR	Changed By	Action	Name	Internal Name	
<input checked="" type="checkbox"/>	2019-12-16 14:13:19	studio		SuperUser	edit	Simple01.Util.CLS	Simple01.Util.CLS	diff

Select Mine (SuperUser)

Access Token

kayUdNrA4r

✓

CCR

ISCU0144

✗

ERROR #5001: Upload not allowed, record is in state In_PREP

Bundle and Upload Changes

CCR Details

CCR	ISCU0144
Title	Update output format of Simple01.Util.PrintUser()
Owner	student01
State	In_PREP
SystemCode	TESTSYS
SiteCode	ISCU

Path: C:\TrakCare\Perforce\custom_ccrs\us\ISCU\TESTSYS\BASE\



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.

Uncommitted Queue:

<input checked="" type="checkbox"/>	Last Changed	Source	CCR	Changed By	Action	Name	Internal Name	
<input checked="" type="checkbox"/>	2019-12-16 14:13:19	studio		SuperUser	edit	Simple01.Utils.CLS	Simple01.Utils.CLS	diff

Select Mine (SuperUser)

Access Token ✓

CCR ✗

ERROR #5001: Record is not valid for the given system

Bundle and Upload Changes

CCR Details

CCR	ISCU0143
Title	Update output format of Simple01.Utils.PrintUser()
Owner	student01
State	In_BASE
SystemCode	CONFIG
SiteCode	ISCU

Path: C:\TrakCare\Perforce\custom_ccrs\us\ISCU\TESTSYS\BASE\



Part 5: Moving to TEST Environment



Quiz: Automatic Integration

Question:

The passPeerReview will kick off the automatic integration to move the diff chunks into the next branch and create an ItemSet. True or False?

Answer:

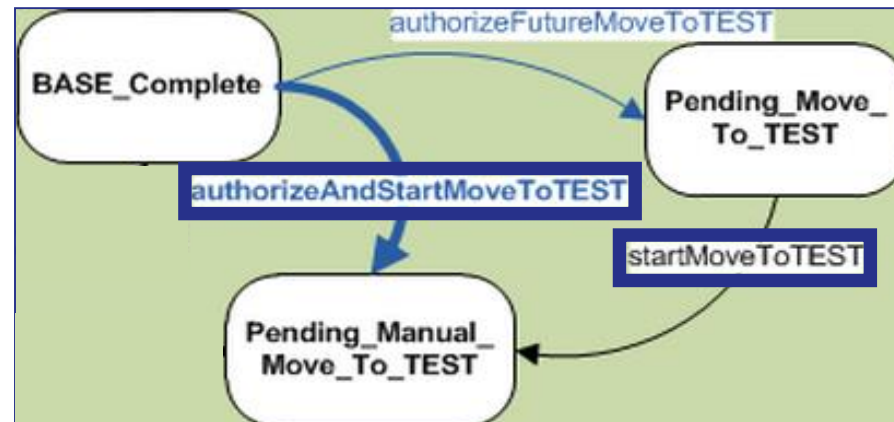
False. passPeerReview will start an “Integration Preview” (if the peer reviewer is a Perforce User), but it will not submit the integration nor create an ItemSet. Automatic integration takes place when the CCR owner performs a transition including the phrase “startMove”.



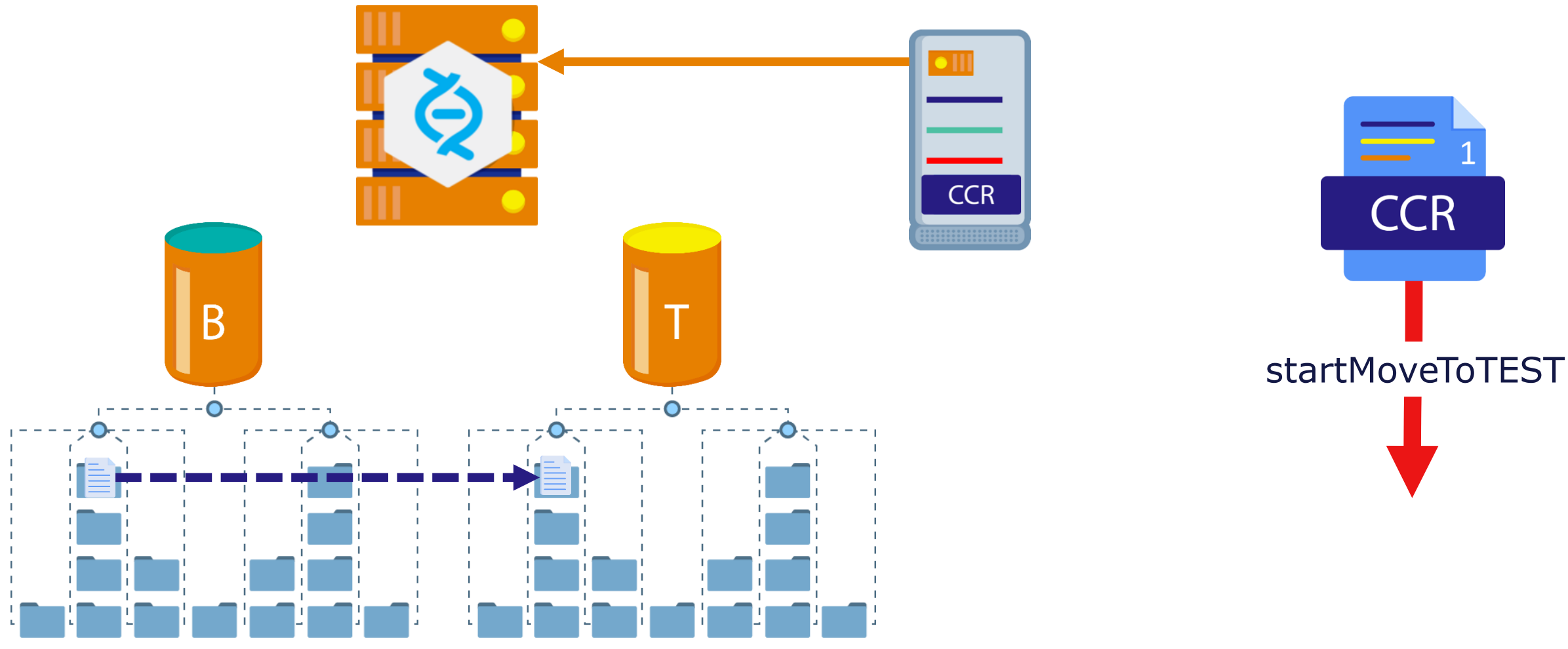
(authorizeAnd)StartMoveToTEST



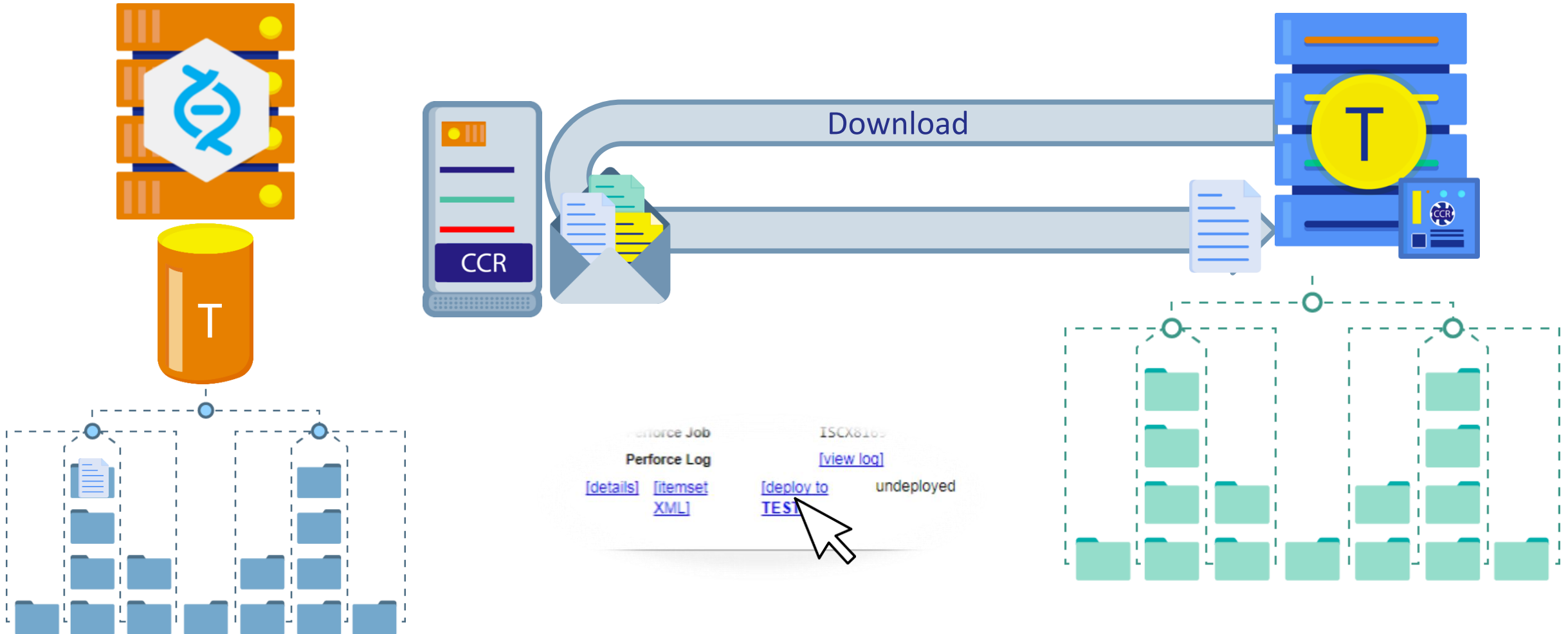
- Only perform authorizeAndStartMoveToTEST or startMoveToTEST when ready to move change.
- Immediately deploy ItemSet.
 - Delaying or abandoning ItemSet deployment will cause problems later.



Review: Transport Walkthrough - Integrating from BASE



Review: Transport Walkthrough - Sending 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** ? Generate Token
Transport Log [View](#) [Download](#)

Itemset Details **Submitted Changes (2)** **Create Itemset** **Perforce Integration** **Perforce Backout**

TEST ☐ **Show All**

ISCX24072_TEST_ColoCCRLive_1167676.xml Created 2023-02-14 20:12:08		deploy to TEST
---	--	--------------------------------



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
Access Token

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
Access Token

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



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 6: Test, Review, Deploy, Repeat



Progress CCR to LIVE

- Perform 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 Perform Details upon successful integration.



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 ICC510-1 through ICC510-6



Part 8: Additional Information



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.



Summary

- What are the key points for this module?

