INTERSYSTEMS LEARNING SERVICES

InterSystems Change Control

InterSystems
Creative data technology

InterSystems Change Control

COPYRIGHT NOTICE

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

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.

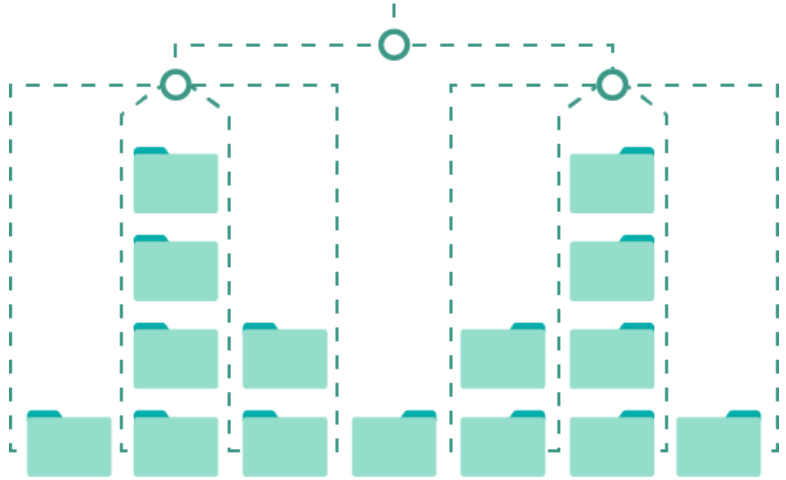
- ItemSet: File that combines one or more Items along with metadata for transporting Items.
  - Think of it like an envelope for sending mail.
  - Only used for transport purposes, not persisting change.

Source Workspace

- Directory in filesystem of each environment.
- Items exported as xml files to source workspace for transport to CCR server.

Database of BASE instance

File system of BASE machine

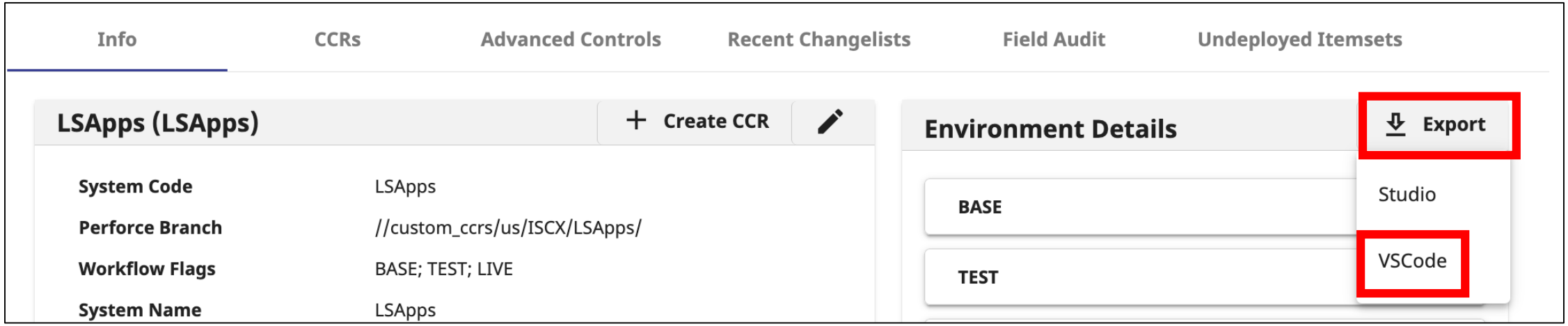


Connected vs Disconnected Mode

- 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.

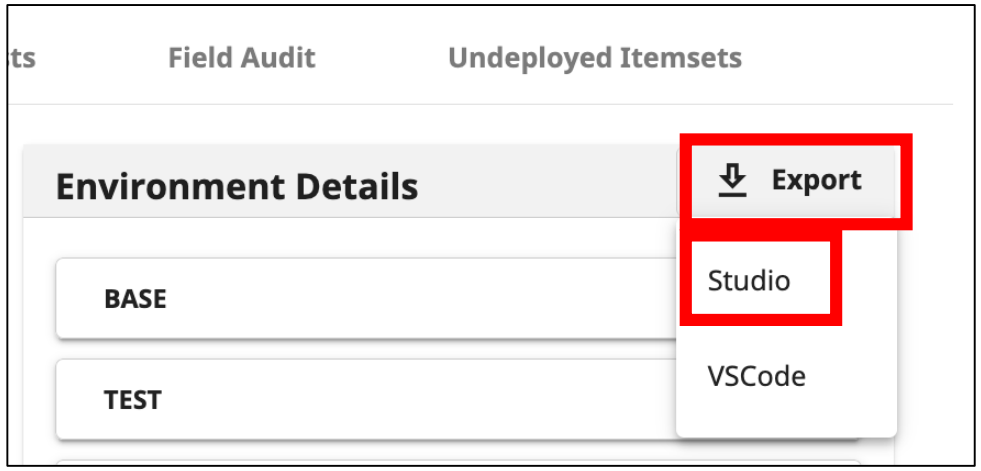
Download VS Code Workspace

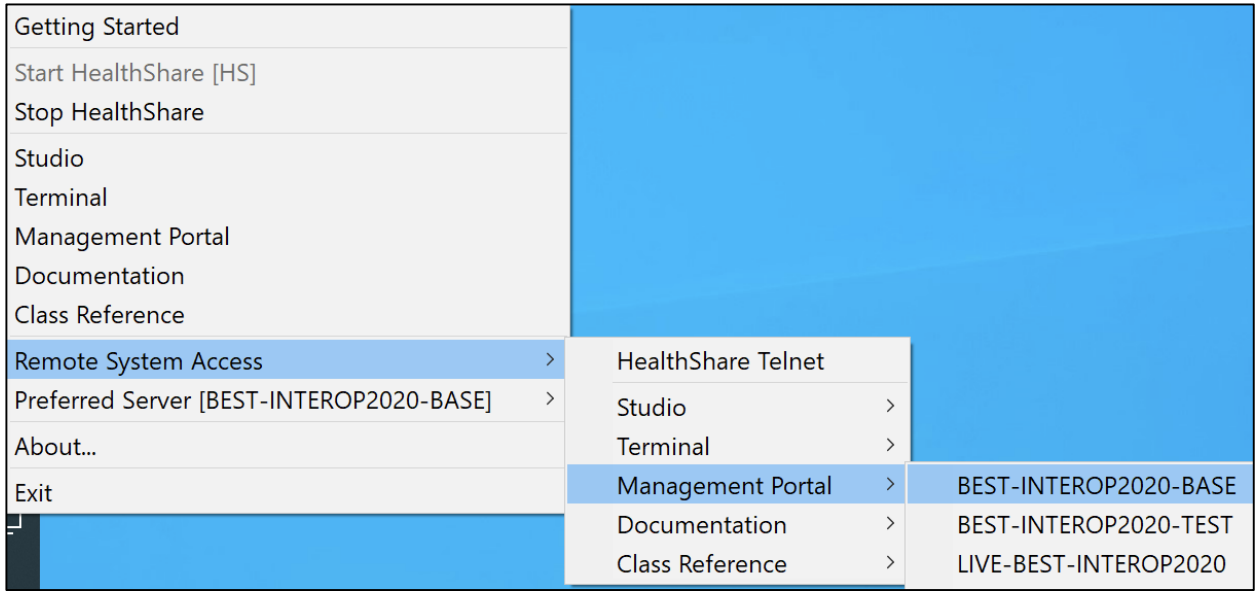
- Can download VS Code Workspace from Systems page.
  - Contains connection details for environments defined in System.
- Menu > Systems > Choose System > Export > VSCode.
  - Double click file to open in VS Code.



Windows: Configure Launcher

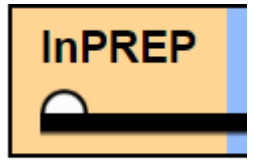
- Launcher provides remote system access to environments.
- Menu > Systems > Choose System > Export > Studio.
  - Run downloaded registry file.
    - Administrator credentials required.





Part 1: CCR PREP Phase

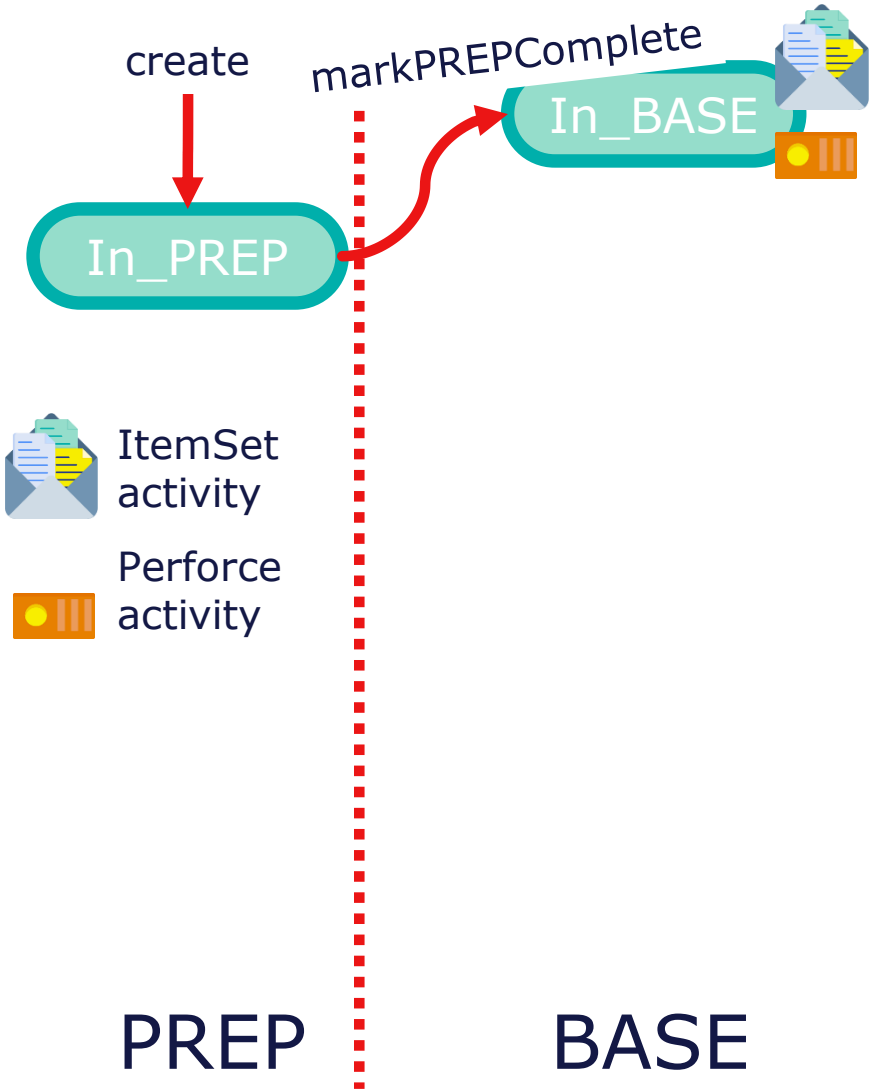
Tier 1 Considerations: In_PREP



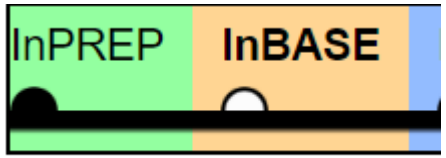
- Confirm Tier field set to “1 – Source Control.”
- Modified Items: Usually matches list of submitted changes.
- Perform markPREPComplete transition before checking out files.
  - Only make changes while in In_BASE state.

Part 2: CCR BASE Phase: Making Changes

Progressing a Tier 1 CCR…



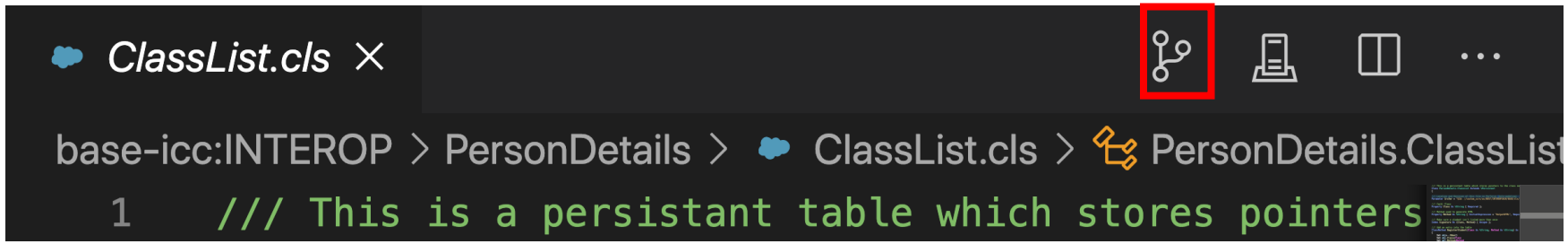
Tier 1 Considerations: In_BASE

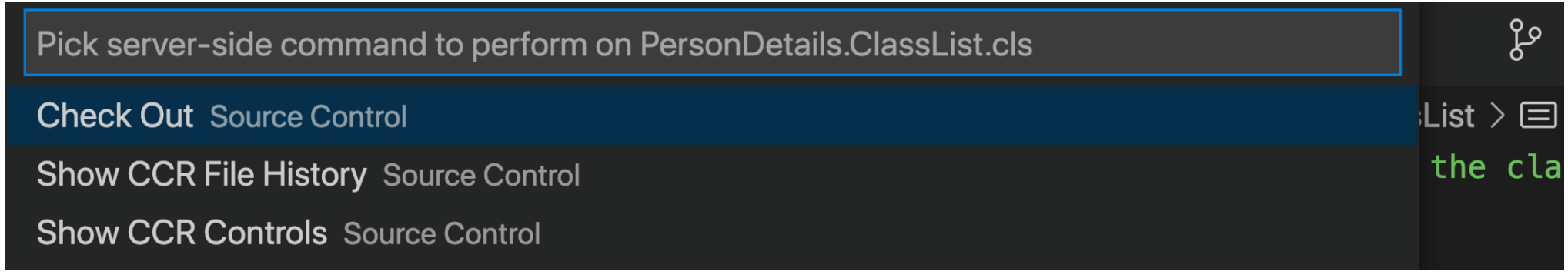


- Work performed via IDE and/or the Management Portal.
- In order to progress, record must have either:
  - Changelist.
  - Uncommitted ItemSet.
- Implementation Plan and Backout Plan have default values
  - Only change if manual steps needed.

Using VS Code - ObjectScript with CCR

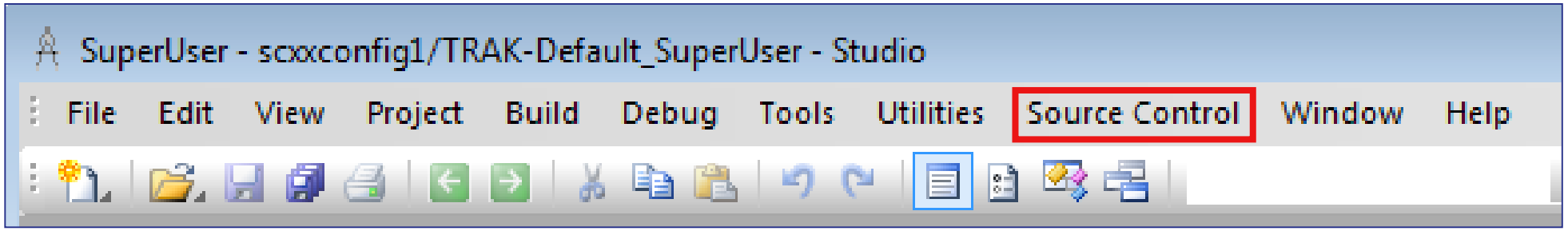
- Use server-side mode to connect to BASE.
  - Can download workspace definition from System Details page in CCR.
- Source Control button appears when viewing any item.

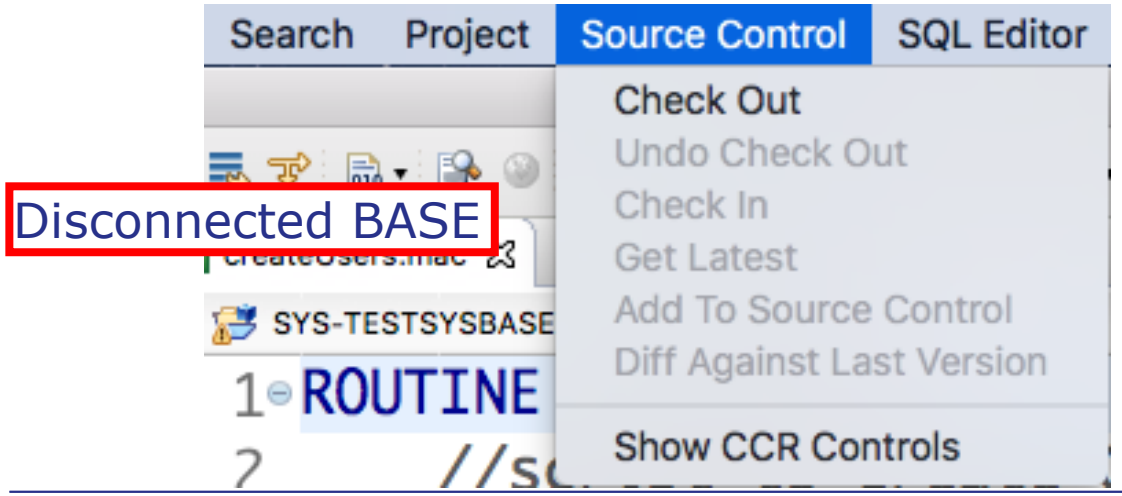




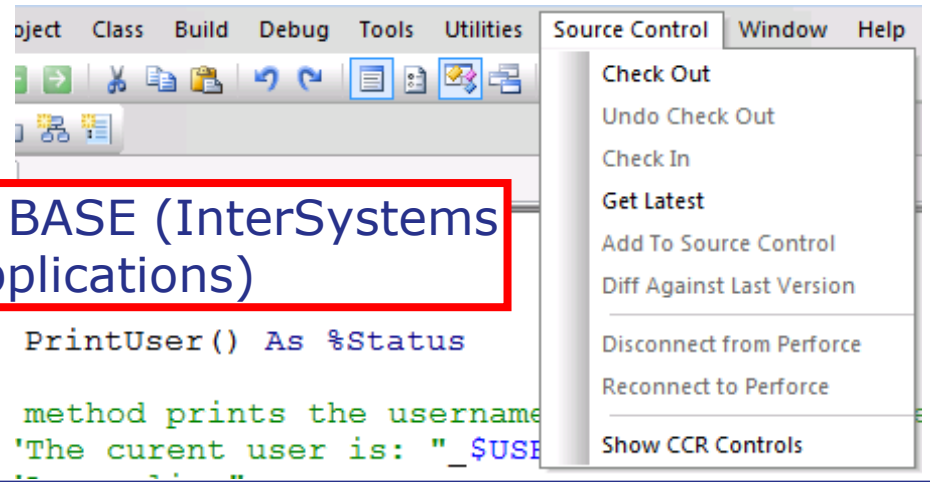
Using Studio with CCR

- 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.

Edit Existing Item

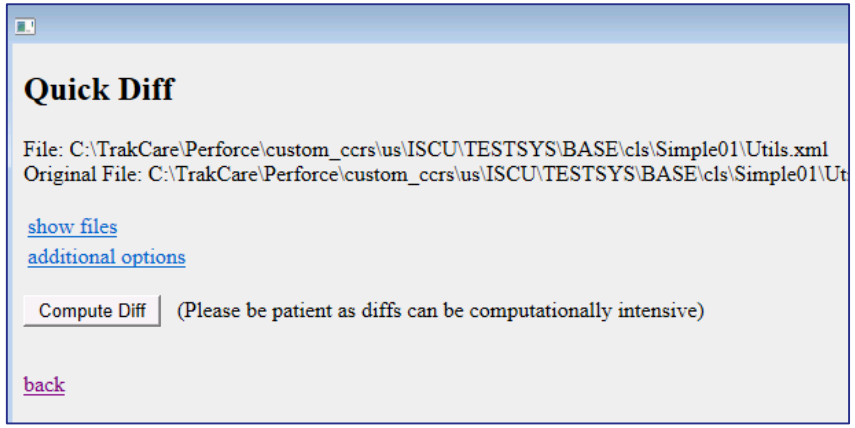
- Cannot checkout item checked-out to another username.
- 2 ways to checkout:
  - Start making change.
    - Popup message prompts to confirm checkout.
  - Open item, then use checkout menu option.
- Check out action:
  - Guarantees only your user can make changes.
  - Backup file created to enable revert.
  - For connected BASE, checks out item in Perforce as well.
    - InterSystems internal applications.

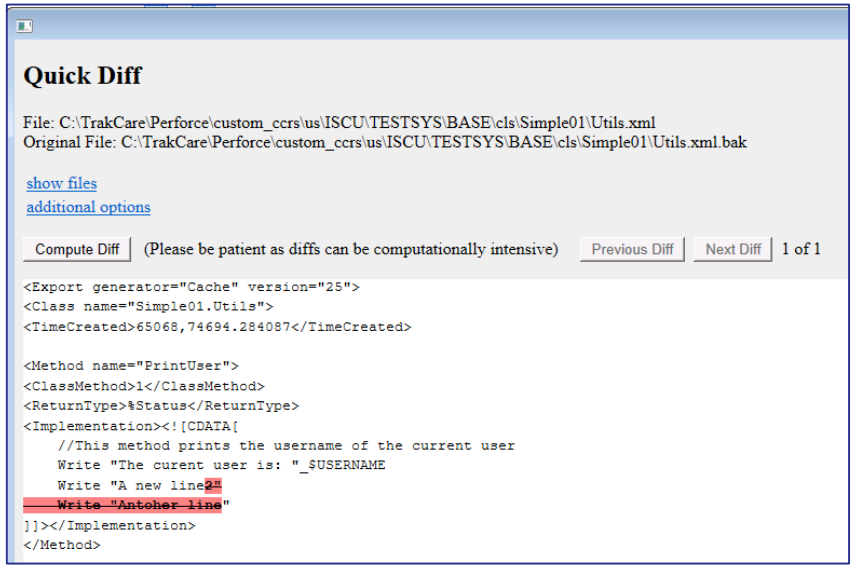
Identify Changes Made to Checked Out File

- 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.





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.)

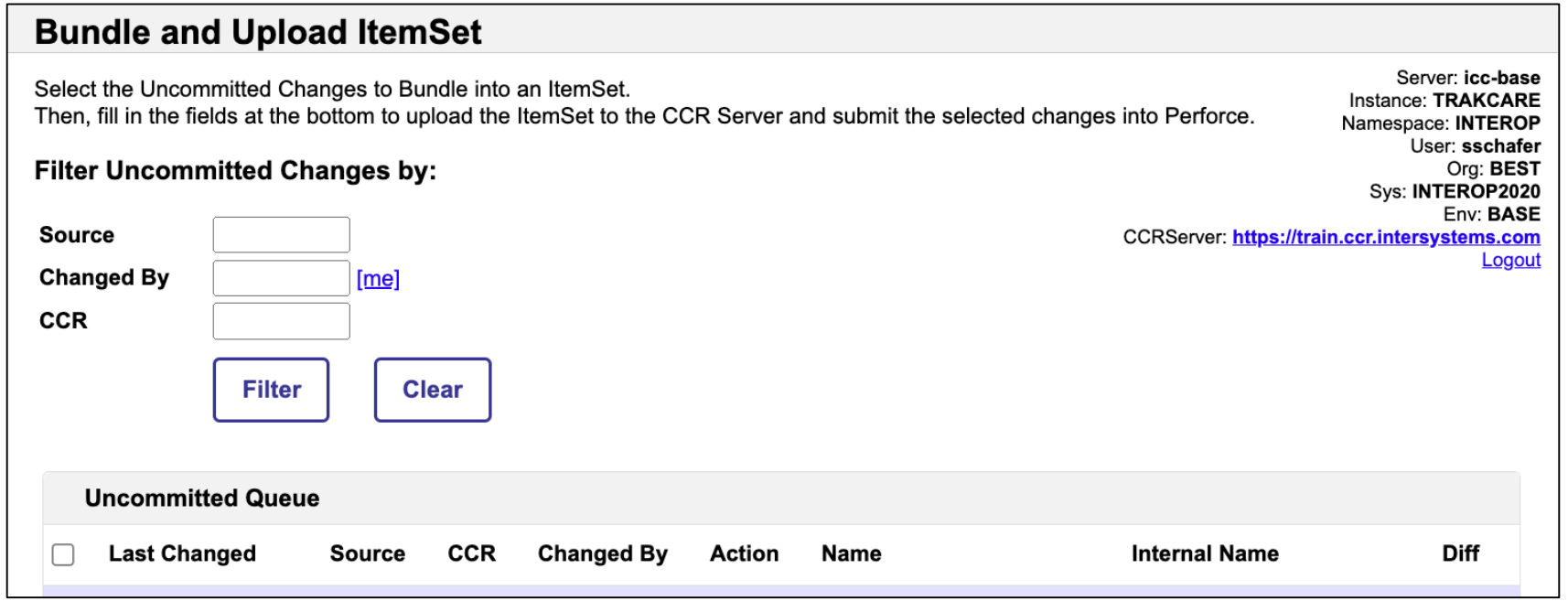
Answer:

A. Only John Smith.

Part 3: CCR BASE Phase: Submitting Changes

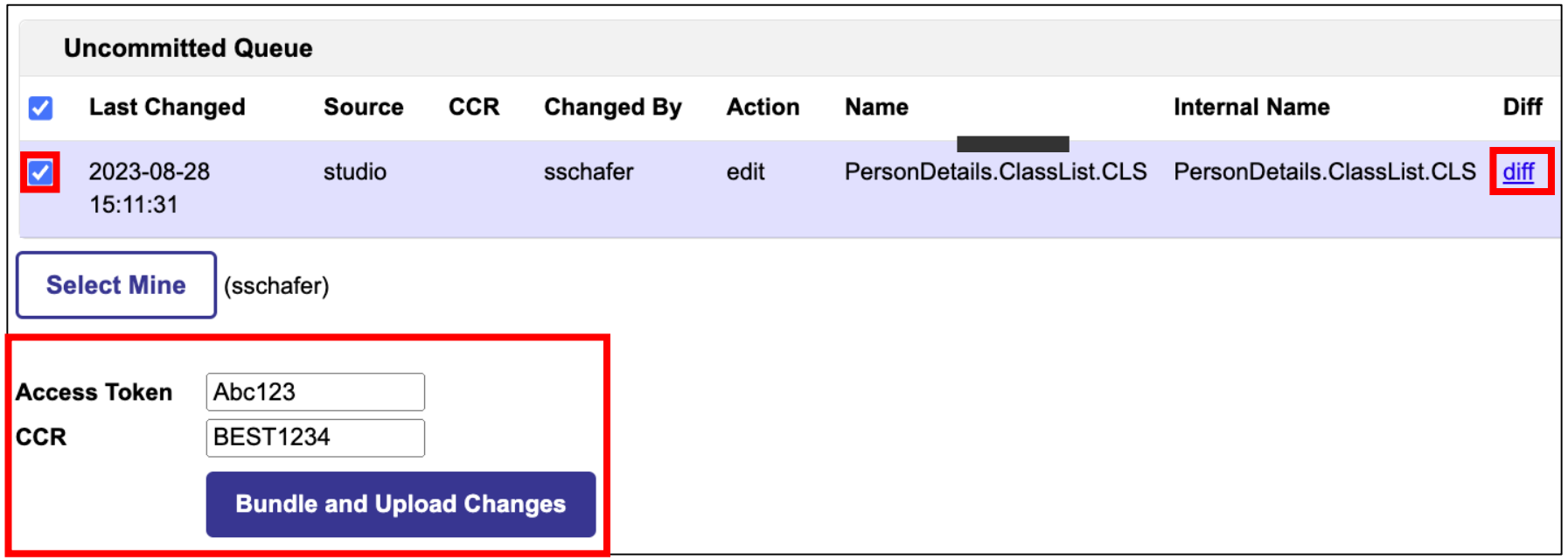
Bundle Screen

- 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.



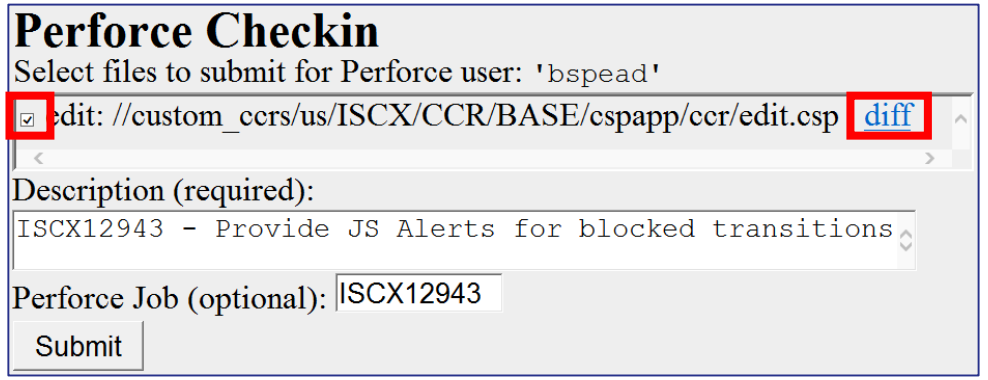
Bundle Screen (cont.)

- Use diff link to diff changes.
- Select which items to bundle.
- Copy Access Token and CCR ID from CCR.
- Click Bundle and Upload Changes.



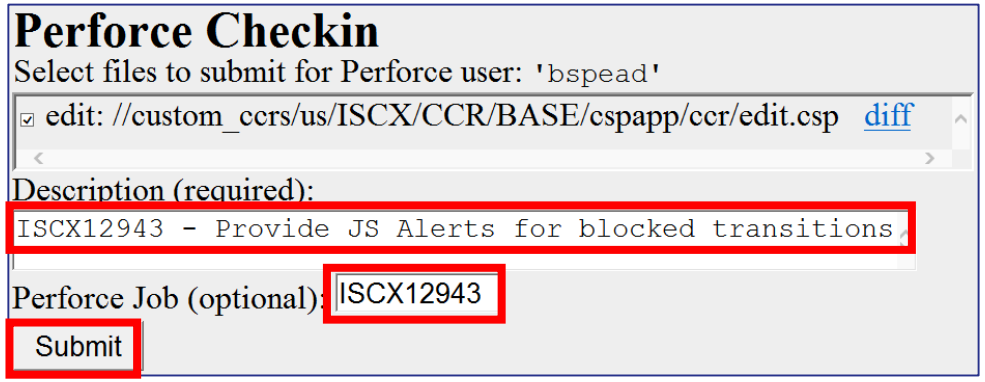
Connected: CCR Client → Perforce

- 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.



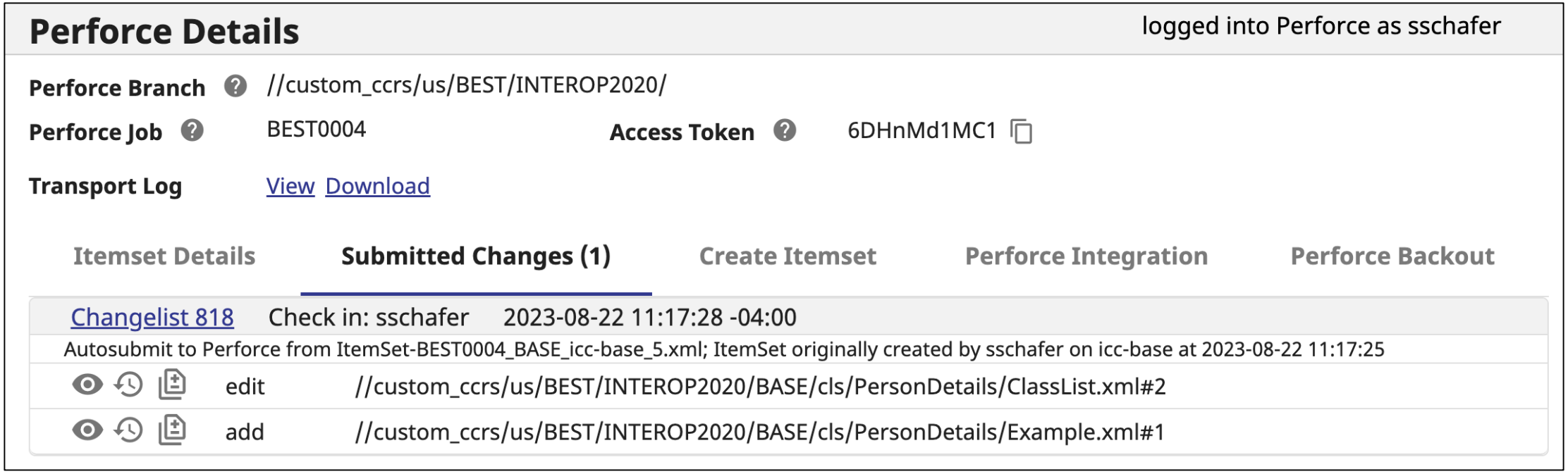
Connected: CCR Client → Perforce

- 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."



Verify Submitted Changes

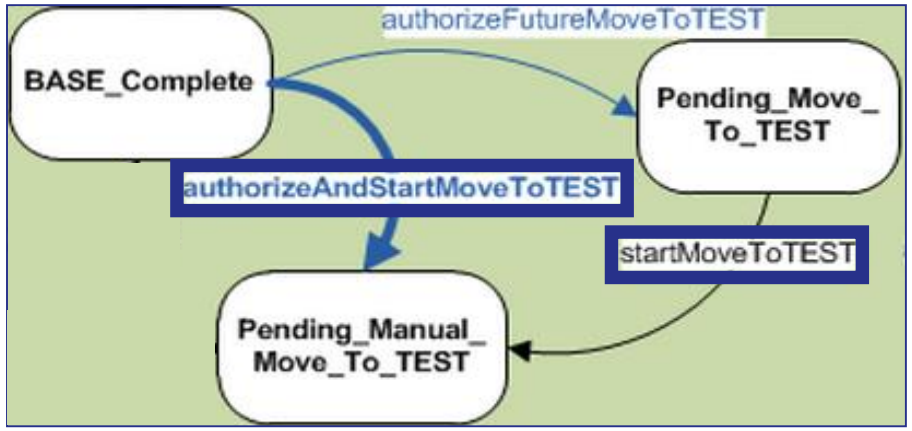
- After submit changes, always verify correct items listed in CCR.
  - Under Perforce Details of CCR, click Submitted Changes tab.



Part 4: Moving to TEST Environment

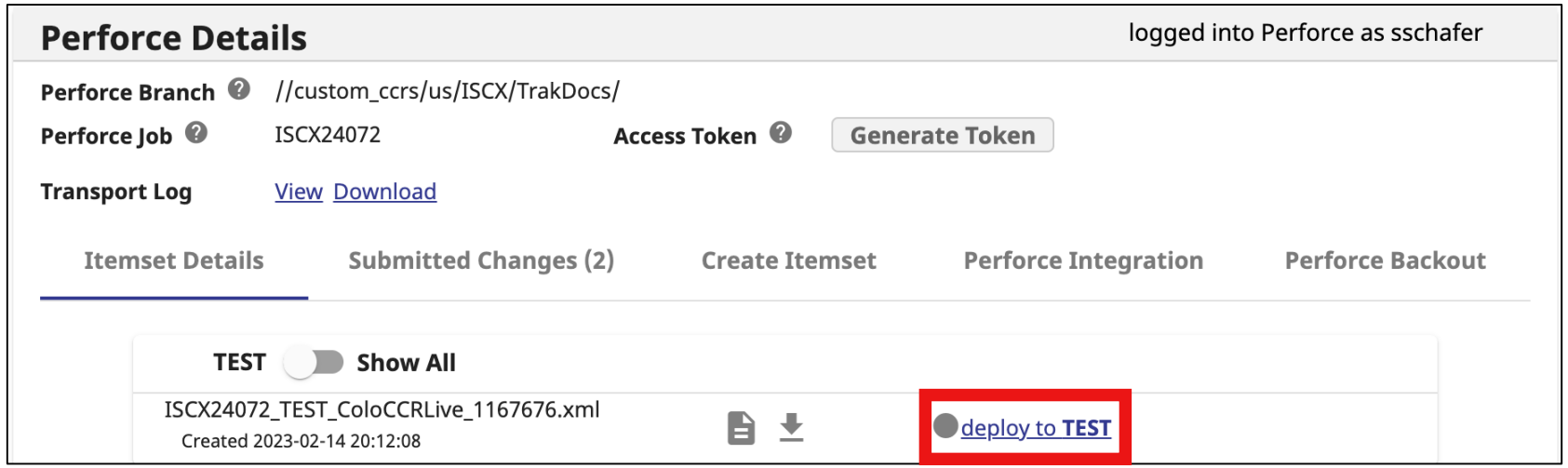
(authorizeAnd)StartMoveToTEST

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



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.



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

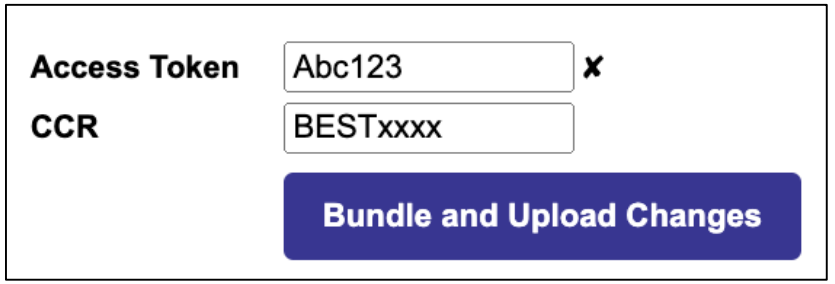
Part 7: Additional Information

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.

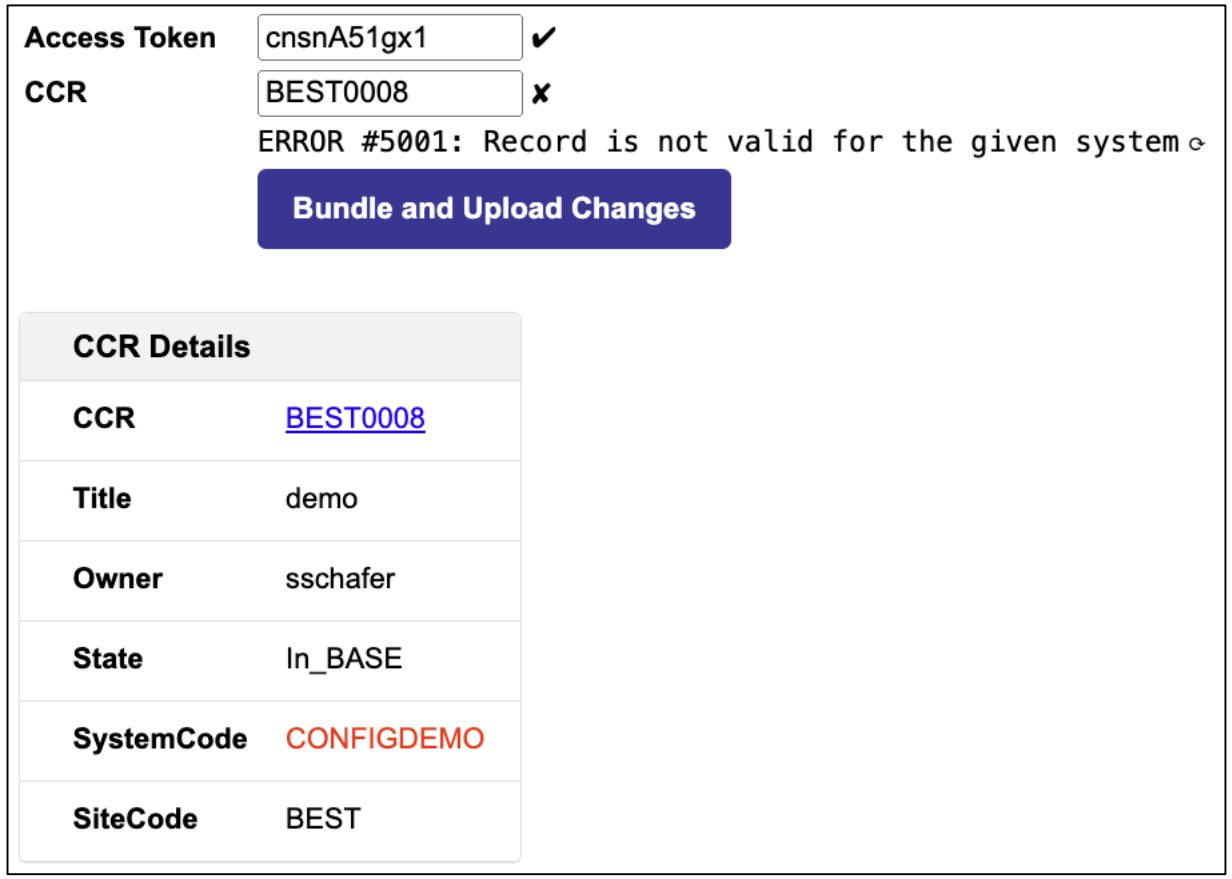


Bundle and Upload: Invalid State

- 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.

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.



Tier 1 Transport Load To Namespace Flow

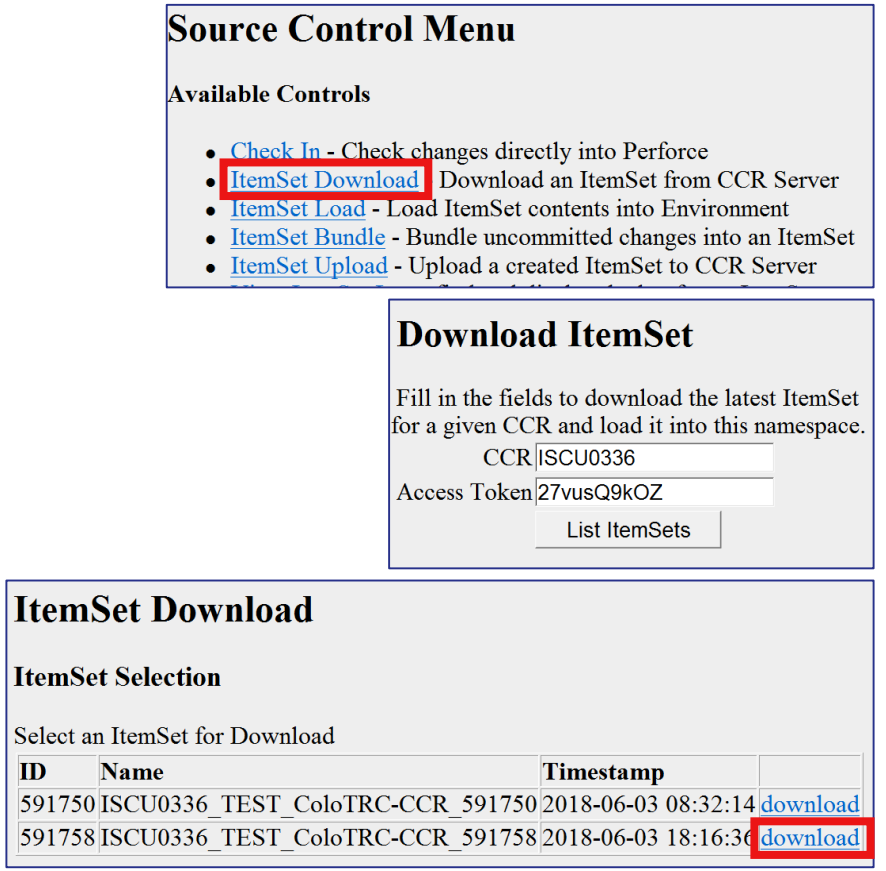
- After Tier 1 items are loaded to the workspace, a number of additional things occur.
  1. If a CCR Event Handler is configured, the ItemSetAfterLoadToOS() method is executed.
  2. Any items in Tier 1 SubFolders are automatically loaded into the namespace:
     - /cls (class files).
     - /cspapp (web 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.

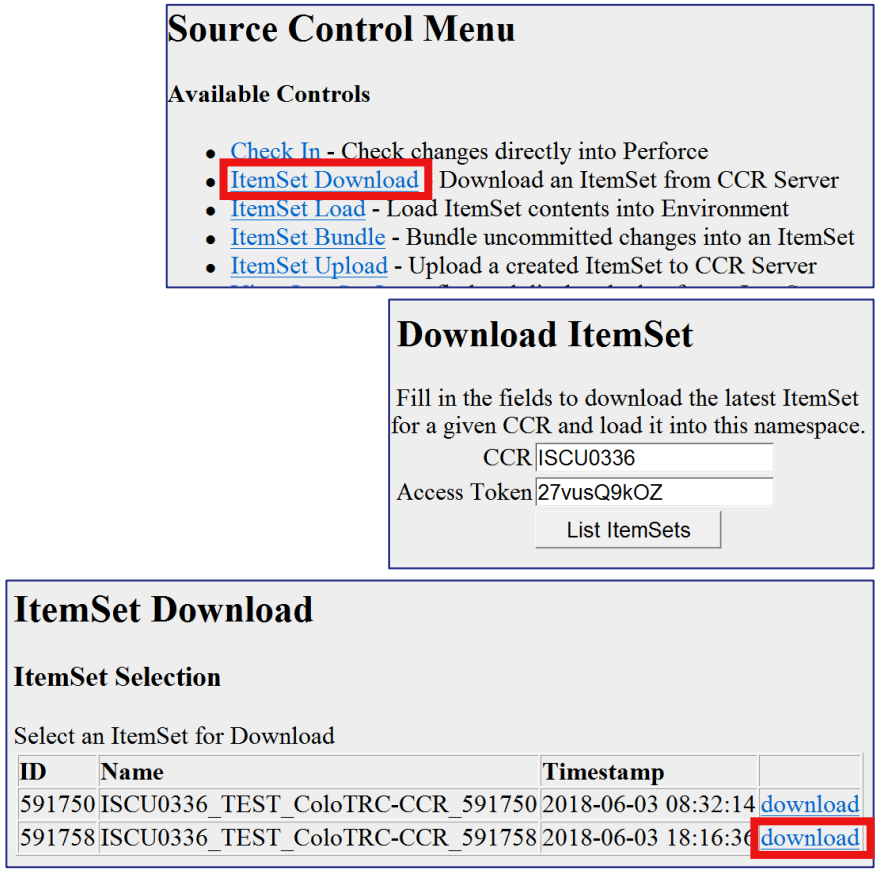
Downloading an ItemSet via IDE

- 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!



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.



Summary

- What are the key points for this module?