INTERSYSTEMS LEARNING SERVICES

InterSystems Change Control

InterSystems
Creative data technology

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 Congress Street, Boston, MA 02114, 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 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.)

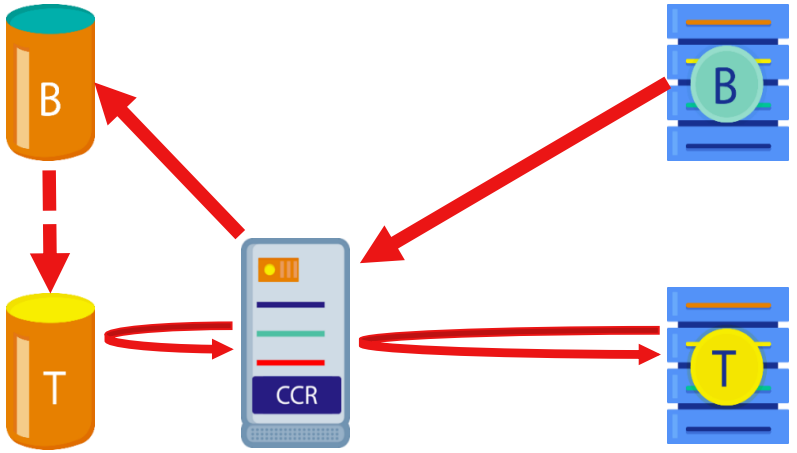
- CCR client configuration steps:
  - Create a CCR to document responses to configuration prompts.
  - Run Configure^%buildccr routine in BASE.
  - Progress CCR record and run Configure^%buildccr in for each environment in each phase.

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

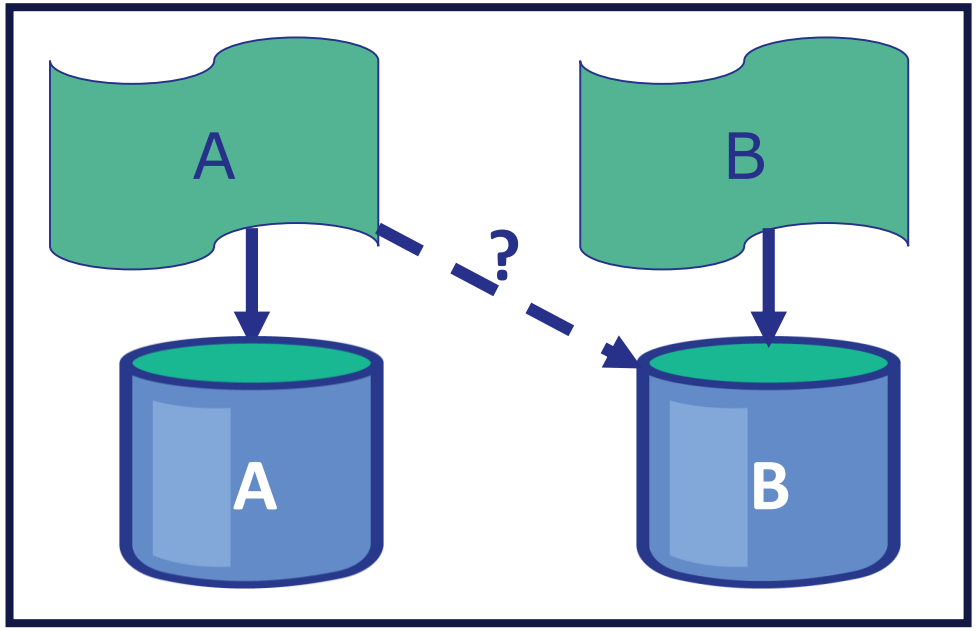
- Lock environment.
  - Disables check out and add to source control functionality to prevent editing any item in source control.
  - **Always respond yes.**
- If respond yes to lock, prompts whether to Admin Lock.
  - Admin lock removes information in UI on how to unlock environment.
    - Most users 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

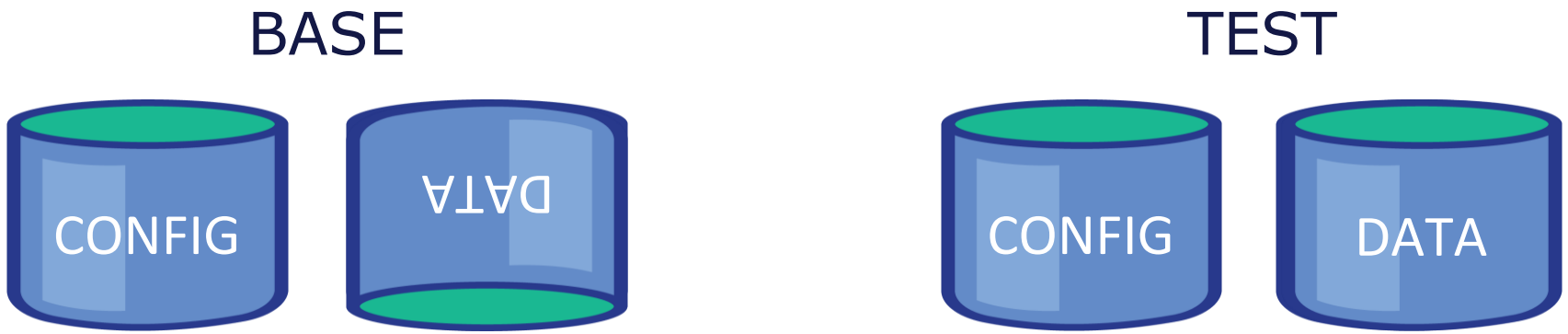
- CCR client configuration globals control behaviour for namespace.
- Some sites prefer to have configuration globals in separate database.
  - Facilitates data refresh on testing environments.

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

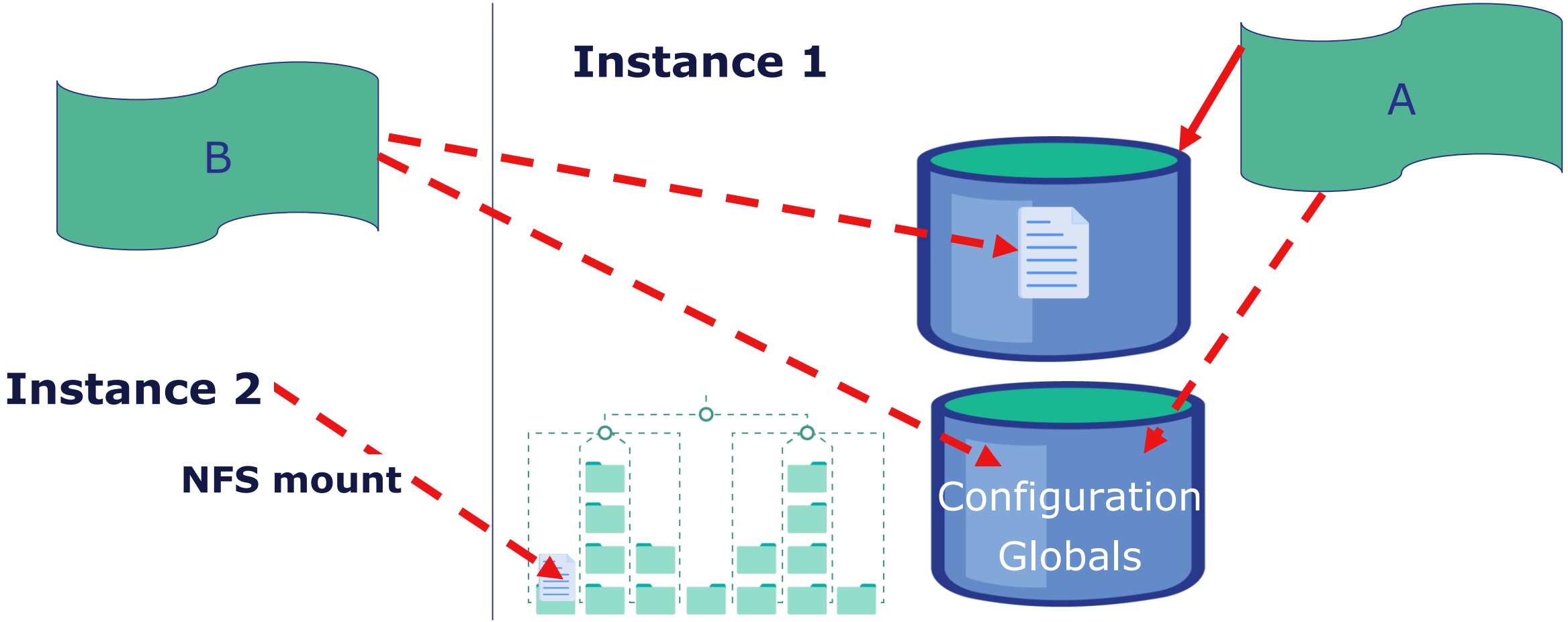
Complexity of CCR for ECP Deployments

- 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.

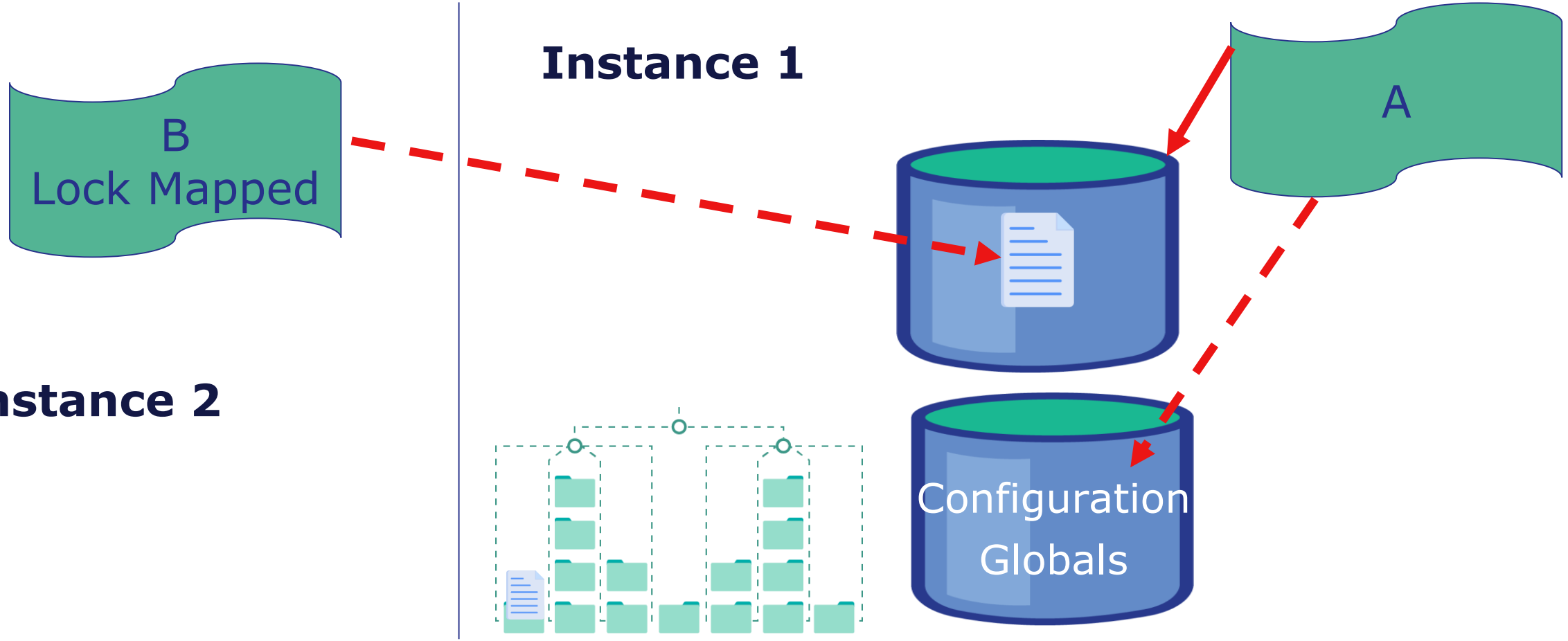
Complexity of CCR for ECP Deployments (cont.)

- 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?