



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 520: CCR Tier 1 - Baselining Source Code



# Overview

- Baseline Overview.
- Simple Baseline.
- Tier 1 Baseline tools.
- Re-baselining.



# Objective

- Explain how to manually run simple Tier 1 baseline.
- Explain which tools can be used to perform a Tier 1 baseline of an environment.
- Demonstrate ability to use baseline export tools.



# Part 1: Baselineing Overview



# Review: What is Baselineing?

- 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 Baselineing 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.



# Baselining Process Notes

- InterSystems internal task.
  - Requires access to Perforce using p4 client.
- **Always** create a CCR to document process.
- Consider connecting via Remote Desktop to Cambridge to improve efficiency.
- Choose between doing steps manually (aka simple baseline) or using autobaseline scripts.
  - 2 scripts:
    - Export and upload to ftp site.
    - Download from ftp site and submit to Perforce.



# Quiz: Purpose of Baselineing

## Question:

Which of the following is true about baselineing? Select all that apply.

- A. Rollback not possible without baselineing?
- 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: Simple Baseline



# Simple Baseline

- Process for performing baseline tasks manually.
- Performed when can't use automation scripts or automation is overkill.
- %Studio.SourceControl.ISC class definition exists in all installations.



# How To: Perform Baseline Export

## 1. Run Export method.

```
Set status =
```

```
##class(%Studio.SourceControl.ISC).BaselineExport().
```

## 2. Verify status equals 1.

```
Write status.
```



# BaselineExport() Prompts

- Use current export directory?
  - Recommend answering no.
  - No read only files may exist in export directory.
- Include items mapped from other databases?
  - Usually answer no.
    - Control mapped databases with their default namespace in its own System.
- Change files to read-only?
  - Yes.



# Reviewing Exported Contents

- By default, exports:
  - Classes.
  - Routines.
  - Include Files.
  - Ensemble Rules.
  - Ensemble Lookup Tables.
  - Ensemble HL7 Schemas.
  - DeepSee Folder Items.
  - CSP Applications.
  - Projects.



# Reviewing Exported Contents (cont.)

- Verify includes all source.
  - May need to manually export additional items not exported by default.
    - Particularly for HealthShare (see ICC530).
- Remove items that should not be in Perforce.
  - Generated classes or routines.
  - Product items which were exported in error.
  - Report to Support so they can fix bug.



# Transporting Baseline Content

- Compress output.
  - Typically, small enough to not pose challenge.
- Move to machine with:
  - p4v installed.
  - Access to InterSystems internal network.



# Submitting Baseline to Perforce

- Get latest revision of BASE branch for System.
- Open Perforce workspace location on local file System.
- Copy and paste all files from export into subfolders.
- Use Add action and submit changelist.



# Systems with More Than 1 Environment

- Populate all branches used by system from BASE branch.
  - Use copy files action.
- Clone BASE VM to create other environments.
  - Move source workspace contents to appropriate branch in file system.
    - Change /BASE to /TEST, /UAT or /LIVE.
  - Use `Configure^%buildccr` to lock environment and change environment type.





# Systems with More Than 1 Environment (cont.)

- Once branch(es) in Perforce have been populated, bring files back to populate the source workspace in that environment.
  - This ensures any classes or other files using RCS keywords have correct \$Id\$ values in the workspace.



# Simple 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.



# Part 3: AutoBaseline Script



# 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 FTP 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.



# AutoBaseline Script

- Automates some steps in simple baseline, still must submit files to Perforce.
- Scripts maintained in Perforce.
  - `//custom_ccrs/_common/tools/AutoBaselineScripts/client/cls/Custom/Utility/Baseline/BaselineHelper.xml`.
  - `//custom_ccrs/_common/tools/AutoBaselineScripts/client/cls/Custom/Utility/Baseline/Menu.xml`.
    - TrakCare only.
  - `//custom_ccrs/_common/tools/AutoBaselineScripts/client/inc/Custom_Utility_Baseline_Main_inc.xml`.



# Steps in AutoBaseline Script

1. Export XML files to temp location.
2. Compress files.
  - CodeRef\_Org\_System\_CCRBaseline.tar.gz.
3. Create configuration file.
  - CodeRef\_Org\_System\_CCRBaseline.ini.
4. Upload compressed file and configuration file to InterSystems FTP 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 BASE.
    - Custom\_Utility\_Baseline\_Main\_inc.xml.
  4. Import BaselineHelper into BASE.
    - Custom.Utility.Baseline.BaselineHelper.xml.
      - Must import after Custom\_Utility\_Baseline\_Main\_inc.xml.
- 



# How To: Run AutoBaseline Script (cont.)

## 5. Run BaselineClasses().

```
Do ##class(Custom.Utility.Baseline.BaselineHelper).BaselineClasses(ExportRef,  
Packages) .
```

- ExportRef = tag to prepend on file name.
- Packages = (Optional) Comma delimited string of packages to export.
  - Omit to export all non-system packages.
- File names will be ExportRef\_Org\_System\_CCRBaseline.





# BaselineHelper Script Prompts

- Test FTP process?
  - Recommend Y.
- Environments in System?
  - If previously run, asks whether System changed.

```
USER>set status = ##class(Custom.Utility.Baseline.BaselineHelper).BaselineClasses("MyExport")

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█
```



# BaselineHelper Prompts (cont.)

- Will prompt similar questions as BaselineExport() prompts.
- Supply 1 letter answers (Y or N).
- Export classes only?
  - Best practice: N.
- Change files to ReadOnly?
  - Best practice: Y.
- Include mapped items?
  - Best practice: N.



# Configuration Ini File

- Created by script.
- Contains information about System.

```
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: Baseline Script

## Question:

Which of the following is true about the AutoBaseline script?  
Select all that apply.

- A. Automates processes that were previously done and can still be carried out manually.
- B. Reduces risk of human error in the baselining process.
- C. Creates configuration file to hold CCR baseline information.
- D. Reduces time to complete baselining process.



# Quiz: Baseline Script (cont.)

Answer:

All of the above.



# Part 4: Perforce Baseline Batch File Script



# Perforce Baseline Batch File Script

- Automates process after running AutoBaseline Script.
- Requires identical perforce username and SSO username.
  - Contact [ccrdev@intersystems.com](mailto:ccrdev@intersystems.com) for help.
- 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 FTP 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.
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. Get latest version of script from Perforce.

- Must have connected to Perforce on this machine before running script.
- Always use current version of script!

## 2. Create local filesystem location for temporary Perforce workspace.

- Directory, but not workspace, must exist before running script.
- Script will delete all files in this directory at end of script.
- On Perforce VMs, use D:\ drive.

## 3. 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.



# Quiz: Baseline Batch script

## Question:

Which of the following is true about the configuration ini file?  
Select all that apply.

- 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.



# Quiz: Baseline Batch script (cont.)

Answer:

All of the above.





# Part 5: Refresh Environment



# Refresh Environment

- Need to update environments with all exported code.
  - May not be necessary if cloned environment.
  - Always necessary after re-baseline.
- Copy Perforce branch to source workspace of environment.
- 3 ways to run refresh:
  - Menu option of ^%buildccr.
  - Do `Refresh^%buildccr`
  - CCR Controls menu from Studio Source Control menu.



# CCR Event Handler Callbacks

- Can write pre and post refresh callbacks in CCR Event Handler.
- Use for custom load logic for additional configuration.
- See ICC420 for how to write CCR Event Handler.
  - Use RefreshPrepare() and RefreshFinalize() methods.



# Part 6: 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.



# How To: Re-Baseline (cont.)

5. Choose just the edits > New Changelist > Reconcile.
6. Repeat steps 4 and 5 for Adds, adding to a new changelist.
7. Submit changelists.
  - Don't forget to attach CCR number as a job.



# How To: Re-Baseline (cont.)

## 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.





# Summary

- What are the key points for this module?

