



Welcome...

Working In ClearCase At Department of Education Modernization Partner Engagement

April 24, 2002

- 35
- 36
- 37
- 38
- 39
- 40
- 41
- 42
- 43
- 44
- 45
- 46
- 47
- 48
- 49
- 50
- 51
- 52
- 53
- 54
- 55
- 56
- 57
- 58
- 59
- 60
- 61
- 62
- 63
- 64
- 65
- 66
- 67
- 68
- 69
- 70



Course Goals

- Understand the customized configuration management process.
- Identify the initial baseline that will be managed under ClearCase.
- Be able to update files in the initial ClearCase baseline from the shared development area.
- Understand how a release to the Test Server is performed.
- Demonstrate how to make changes to a unit managed under ClearCase.
- Learn how to merge changes from production fixes into your ClearCase view.
- Understand how a production build is done using ClearCase.



Module 1

Your Configuration Management Process

- 35
- 36
- 37
- 38
- 39
- 40
- 41
- 42
- 43
- 44
- 45
- 46
- 47
- 48
- 49
- 50
- 51
- 52
- 53
- 54
- 55
- 56
- 57
- 58
- 59
- 60
- 61
- 62
- 63
- 64
- 65
- 66
- 67
- 68
- 69
- 70



Process Summary

Three areas of the process:

- ◆ Initial “cut-over” to ClearCase where repository is filled with most current baseline of code
- ◆ Day-to-day working environment where all changes to files are managed within ClearCase
- ◆ Production builds and maintenance of production code



Initial “Cut-over”

- ◆ CMA updates ClearCase to contain the most recent baseline of code
 - ◆ Baseline refers to the latest “golden” (trusted) build
- ◆ Developers update ClearCase with new versions of files that have changed since the “golden” build
 - ◆ Only recently changed files are updated



Day-to-Day Work

- DBAs perform builds to Test Server
 - Files are identified by README files and can be checked against ClearCase reports
- Developers fix defects in Test or make other changes needed for next build
 - Checkout files from ClearCase before changing, checkin files when complete
- Repeat above steps on a weekly basis



Production Build

- DBAs perform builds to staging area
 - Files are identified by README files, checked by ClearCase reports
- Developers make production fixes and test in a separate ClearCase view
 - Fixes made on \prodsup branch
- DBAs include prodsup files in builds to test and production servers as necessary
 - ClearCase reports identify files that have been fixed



Module 2

Performing the Initial "Cut-Over"

- 35
- 36
- 37
- 38
- 39
- 40
- 41
- 42
- 43
- 44
- 45
- 46
- 47
- 48
- 49
- 50
- 51
- 52
- 53
- 54
- 55
- 56
- 57
- 58
- 59
- 60
- 61
- 62
- 63
- 64
- 65
- 66
- 67
- 68
- 69
- 70

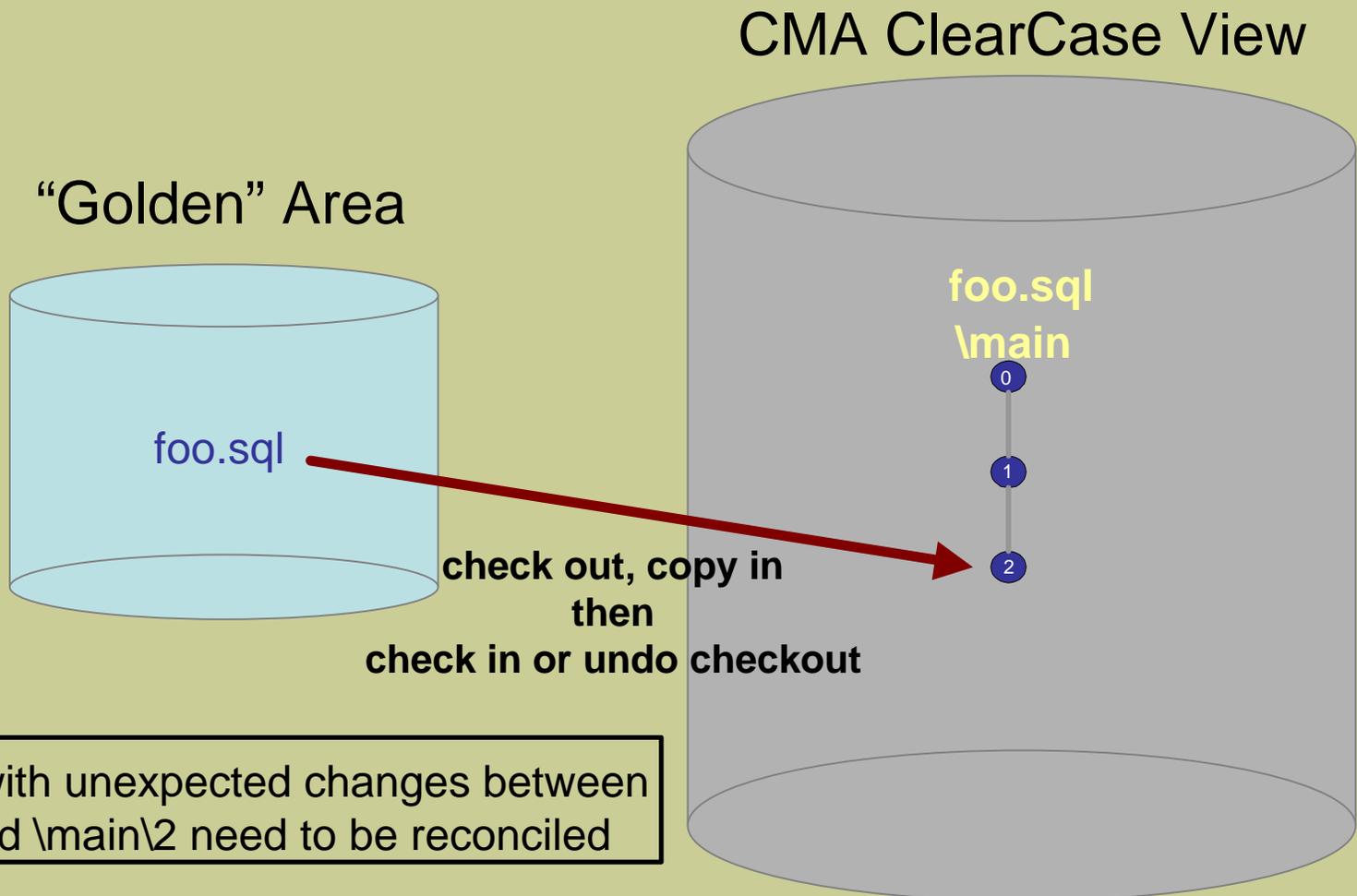


Step 1: CMA Updates ClearCase

- The current contents of the \FMS VOB are updated from the latest “golden” release.
 - “Golden” indicates a known working version
- CMA will add new files, and update existing files using Checkout-Update-Checkin
- Once complete, developers now must ensure ClearCase keeps the latest version of all code (see Step 2)



Step 1 Process: Done by CMA





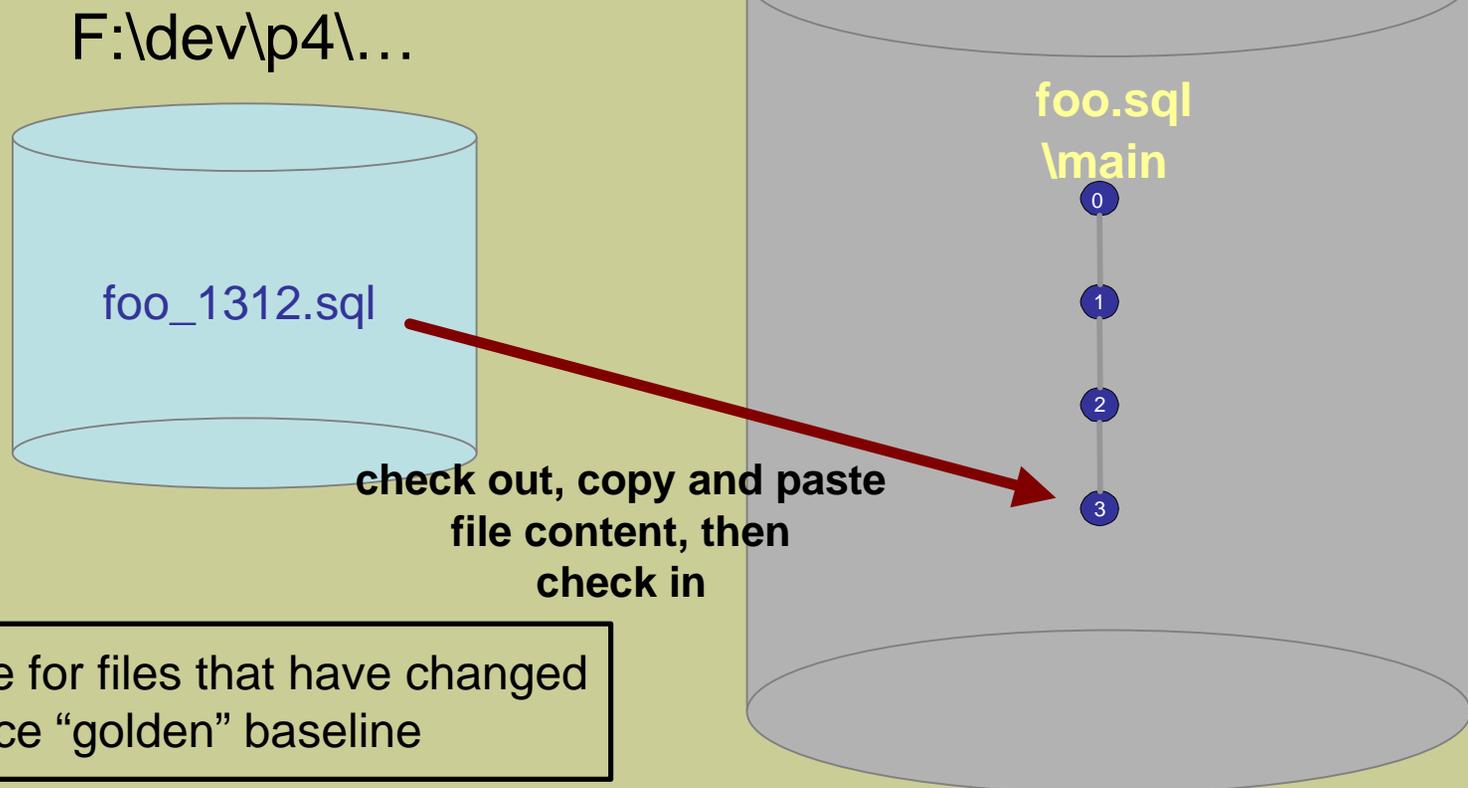
Step 2: Developers Update Changed Files

- Any files which have changed in the common development area (F: drive) are updated in ClearCase by the developers
- This is done by a checkout, copy the newer version of the file, then checkin
- Only files which have changes since the “golden” baseline must be updated
- This should correspond to the same file list given to the DBAs for a build currently
- Place external version # in comments



Step 2 Process: Done by Developers

Developer P4 ClearCase View



* Only done for files that have changed since "golden" baseline



Refresher: What is a View?

- Access data in VOBs using rules called configuration specifications (config specs)
- Manage different development streams with unique configurations by using different views
- Enable parallel development
- Provide access to all versioned objects from standard development tools
- Use the conventions of native operating systems
- Provides a workspace for a single developer or a group of developers



Refresher: What do Views do?

- Displays versions
- Using a config spec, selects versions of VOB directories and files
- Provides a workspace for a user or group
- Enables developers to work in parallel

View directory

Version of a directory element

Enter a VOB directory pathname to browse:

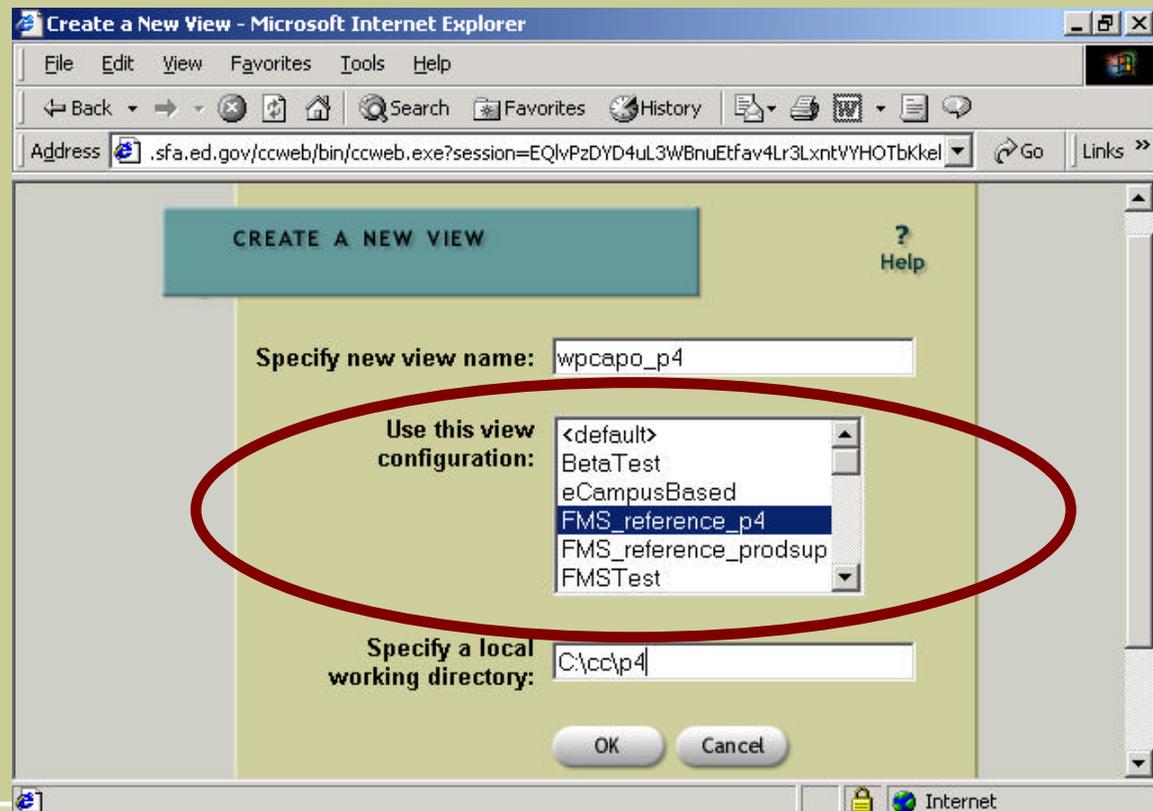
	Name	Size	Modified	Version
<input checked="" type="checkbox"/>	ipp_ap_to_gaps_pkg_b_dev3324_noncont.sql	73.99KB	Feb 19 14:30	/main/1
<input type="checkbox"/>	ipp_gl_to_gaps_pkg_b_dev3324_noncont.sql	68.94KB	Feb 19 14:30	/main/1
<input type="checkbox"/>	ipp_in_gaps_to_fms_pkg_b_noncont_v329.sql	256.14KB	Feb 19 14:30	/main/1
<input checked="" type="checkbox"/>	ipp_in_insert_script_v322.sql	52.17KB	Feb 19 14:30	/main/1

View: wpcapo_p4 Working directory: C:/cc/p4



Refresher: Creating a P4 View

- During view creation, ClearCase Web will allow you to base your view off an existing reference view.





Module 3

Day-to-Day Work in ClearCase

35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70



Step 3: DBAs Populate Test Area

- When ClearCase has been updated with all changes for a build, development lead collects list of files/versions for README
- The CMA applies a label to the correct version of each file in the README
- The label (ex. INT_042402) will identify this INT build by the date of the build
- Any time there is a need to recover or view this build at a later date, use a config spec that requests all files with this label



Step 3: DBAs Populate Test Area (cont)

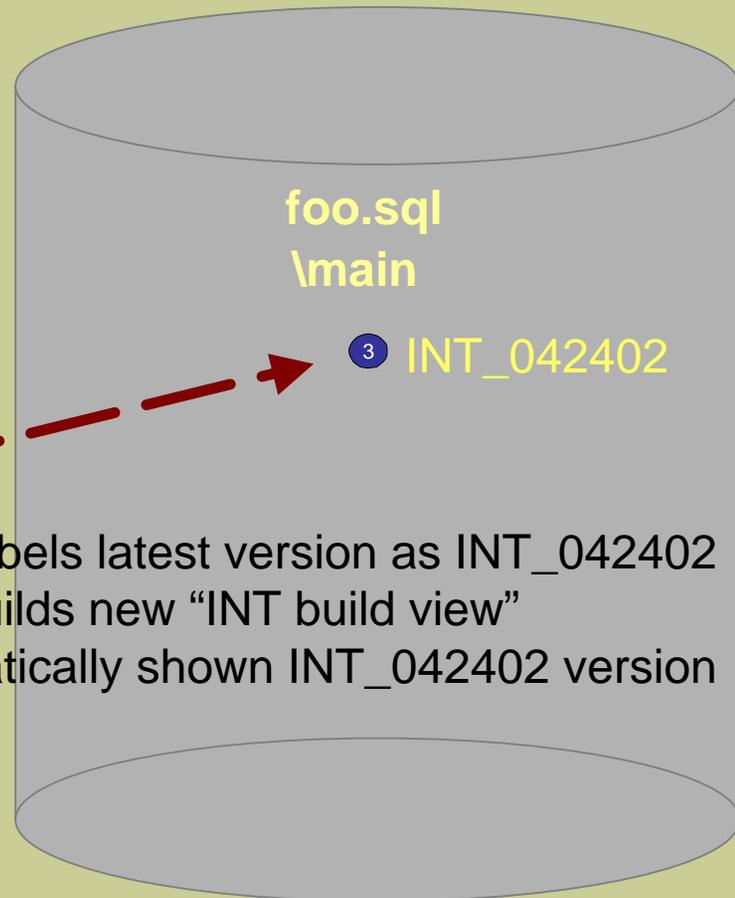
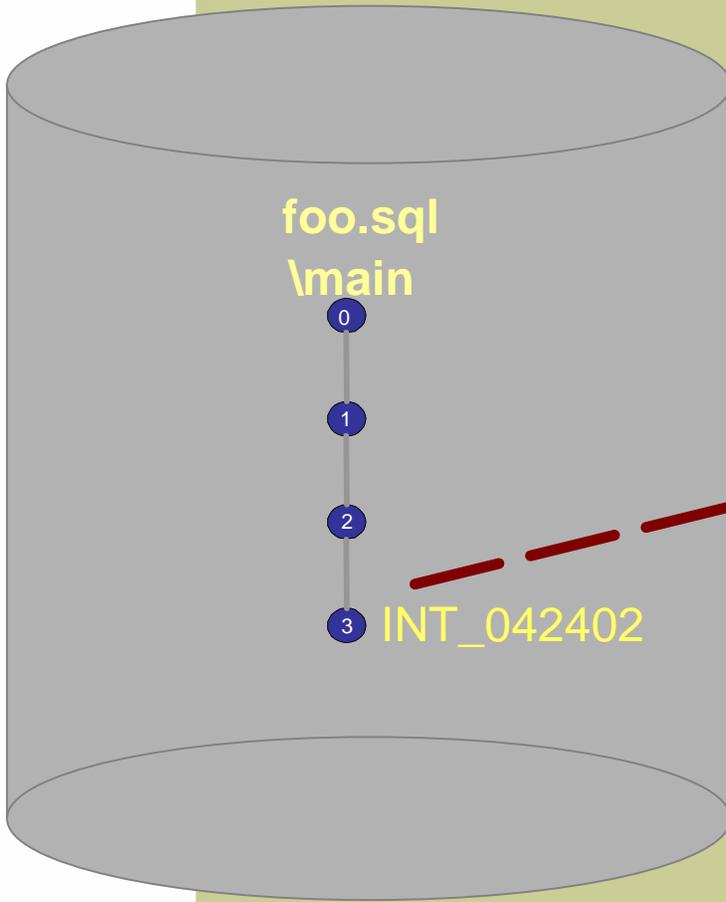
- With the build labelled, the DBAs create a view which requests all files labelled INT_042402
 - Done using a config spec as seen later
- This automatically contains only the files needed for this build
- Only files from this view are copied from ClearCase to the Test area; no need to rename files



Step 3 Process: Done by CMA/DBAs

CMA P4 ClearCase View

DBA INT Build ClearCase View

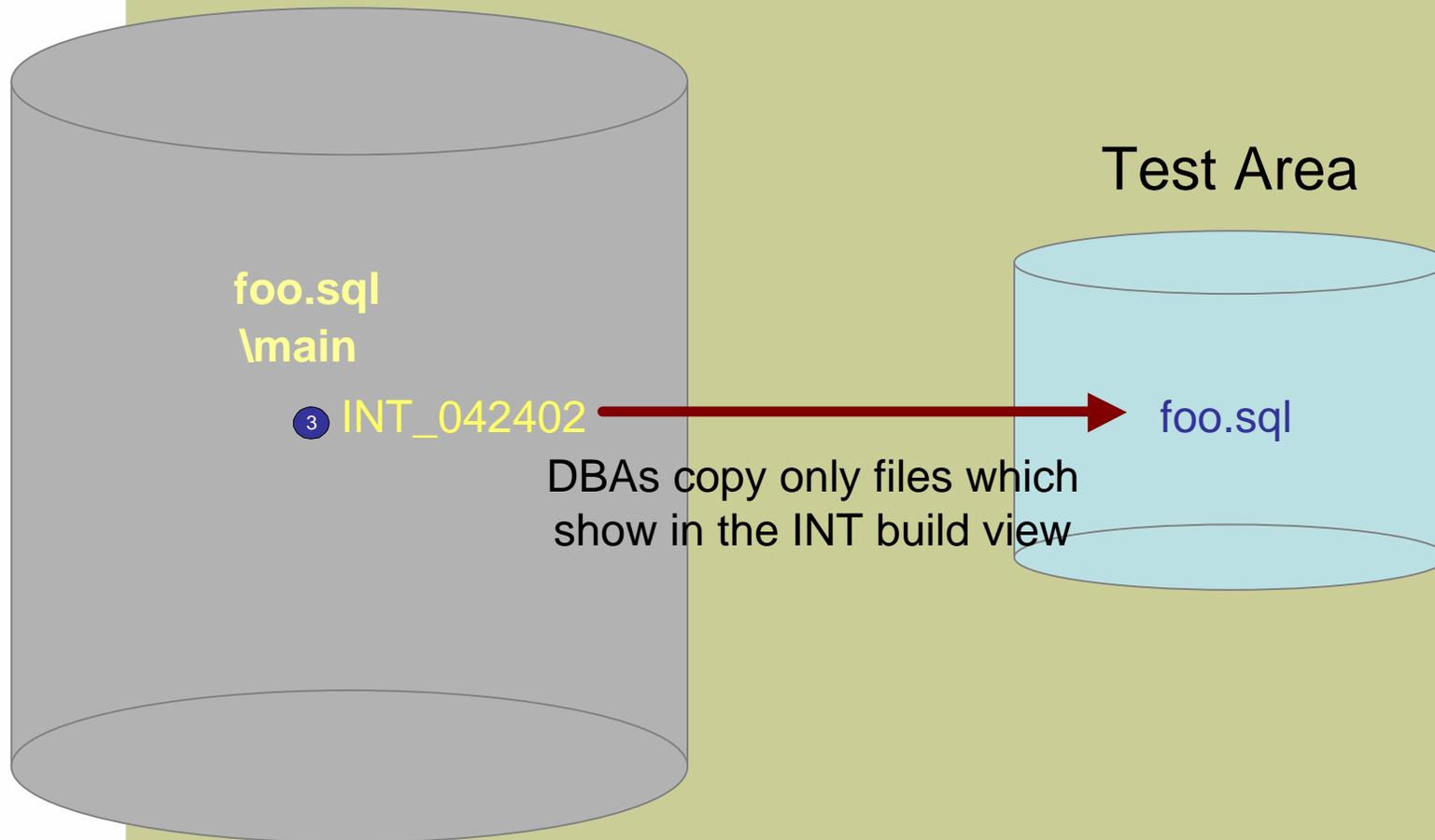


CMA labels latest version as INT_042402
DBA builds new "INT build view"
Automatically shown INT_042402 version



Step 3 Process (cont): Done by CMA/DBAs

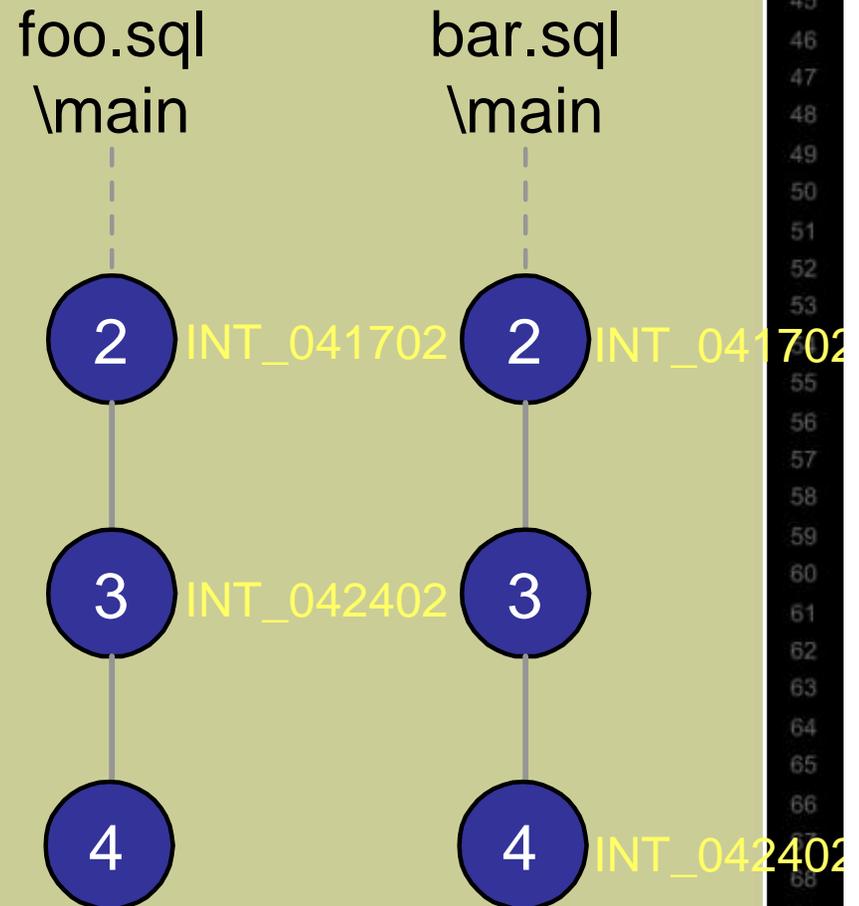
DBA INT Build ClearCase View





Labels in ClearCase

- A label in ClearCase is a method of identifying a set of versions which together create a baseline such as a build
- Not all elements must have a label, not all versions of an element must have a label
- Usually, the same label appears only once per element





Labels in Config Specs

- A label can be used in the config spec to select versions which carry that label:

Rule 1: `element * CHECKEDOUT`

Rule 2: `element * INT_042402`

Rule 3: `element * \main\LATEST`

- Rules #2 and #3 are interpreted as:
 - If there is a version of this element labeled INT_042402, select that version.
 - Otherwise, select the latest version on the main branch.



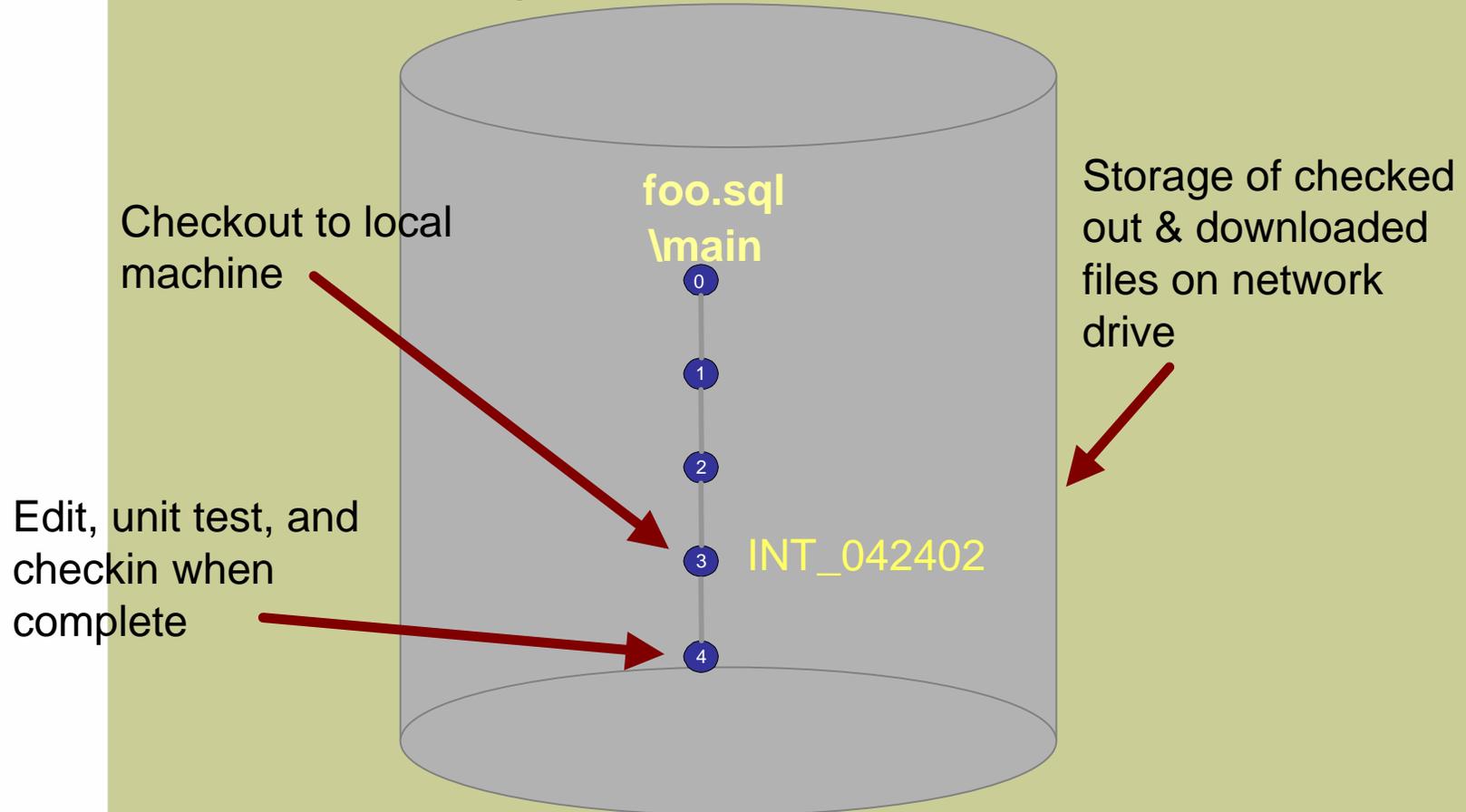
Step 4: Developers Make Fixes

- Developers can now change individual files managed by ClearCase in order to
 - Fix problems found in test
 - Develop towards next build
- Checkout and download files to a network drive as needed to support development
- Checkout, modify, unit test, and checkin files directly from ClearCase



Step 4 Process: Done by Developers

Developer P4 ClearCase View





Things To Watch For

- If work is going on for the same file in both P4 development (\main) and production support (\main\prodsup), both developers must coordinate to ensure that changes are not overwritten and merged when ready
- CMA can tell you when there is both P4 and prodsup work going on or you can use History & Properties ClearCase commands
- If manual merges are done, mention version numbers in checkin comments



Module 5

Production Builds and Support

- 35
- 36
- 37
- 38
- 39
- 40
- 41
- 42
- 43
- 44
- 45
- 46
- 47
- 48
- 49
- 50
- 51
- 52
- 53
- 54
- 55
- 56
- 57
- 58
- 59
- 60
- 61
- 62
- 63
- 64
- 65
- 66
- 67
- 68
- 69
- 70



Step 5: DBAs Populate Staging Area

- When ClearCase has been updated with all changes for the production build, development lead collects list of files/versions for Prod README
- The CMA applies a label to the correct version of each file in the README
- The label (ex. PROD_042402) will identify this PROD build by the date of the build



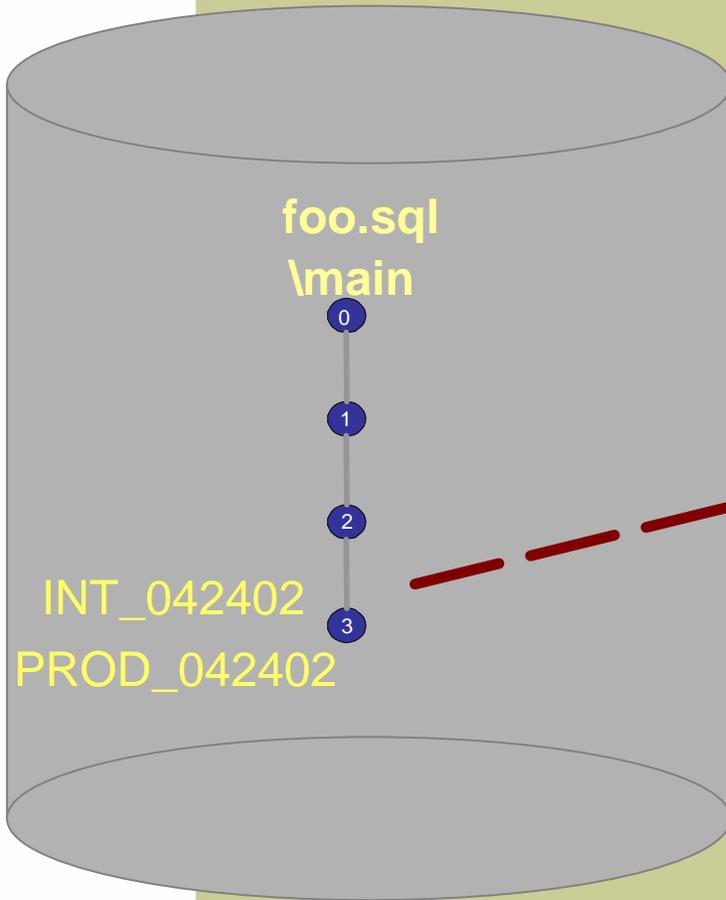
Step 5: DBAs Populate Staging Area (cont)

- With the build labelled, the DBAs create a view which requests all files labelled PROD_042402
 - Done using a config spec as seen later
- This automatically contains only the files needed for this build
- Only files from this view are copied from ClearCase to the Staging area; no need to rename files
- VDC copies from Staging to PROD area

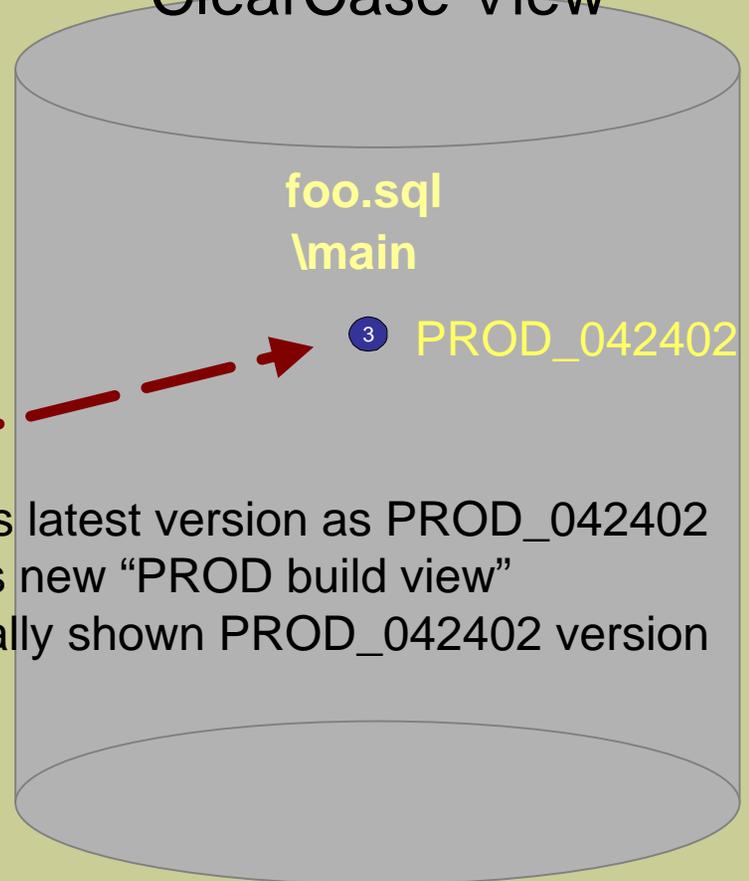


Step 5 Process: Done by CMA/DBAs

CMA P4 ClearCase View



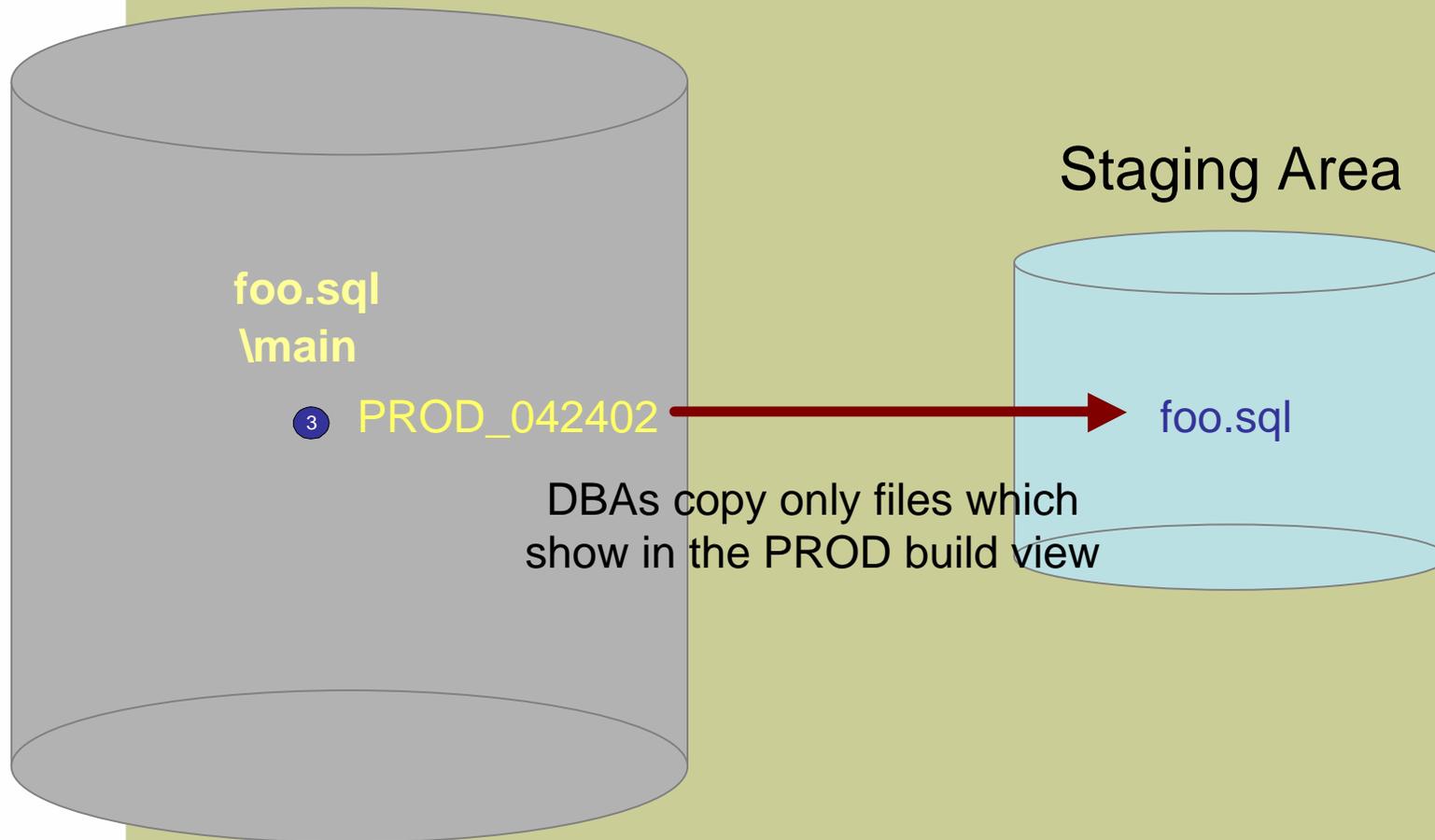
DBA PROD Build
ClearCase View





Step 5 Process (cont): Done by CMA/DBAs

DBA PROD Build ClearCase View





Things To Watch For

- CMA can check to be sure all versions labeled PROD_XXXXXX also have an INT_XXXXXX label
 - This shows they have been previously built to the Test Area
- CMA can also check for INT tested files that have never been promoted to PROD



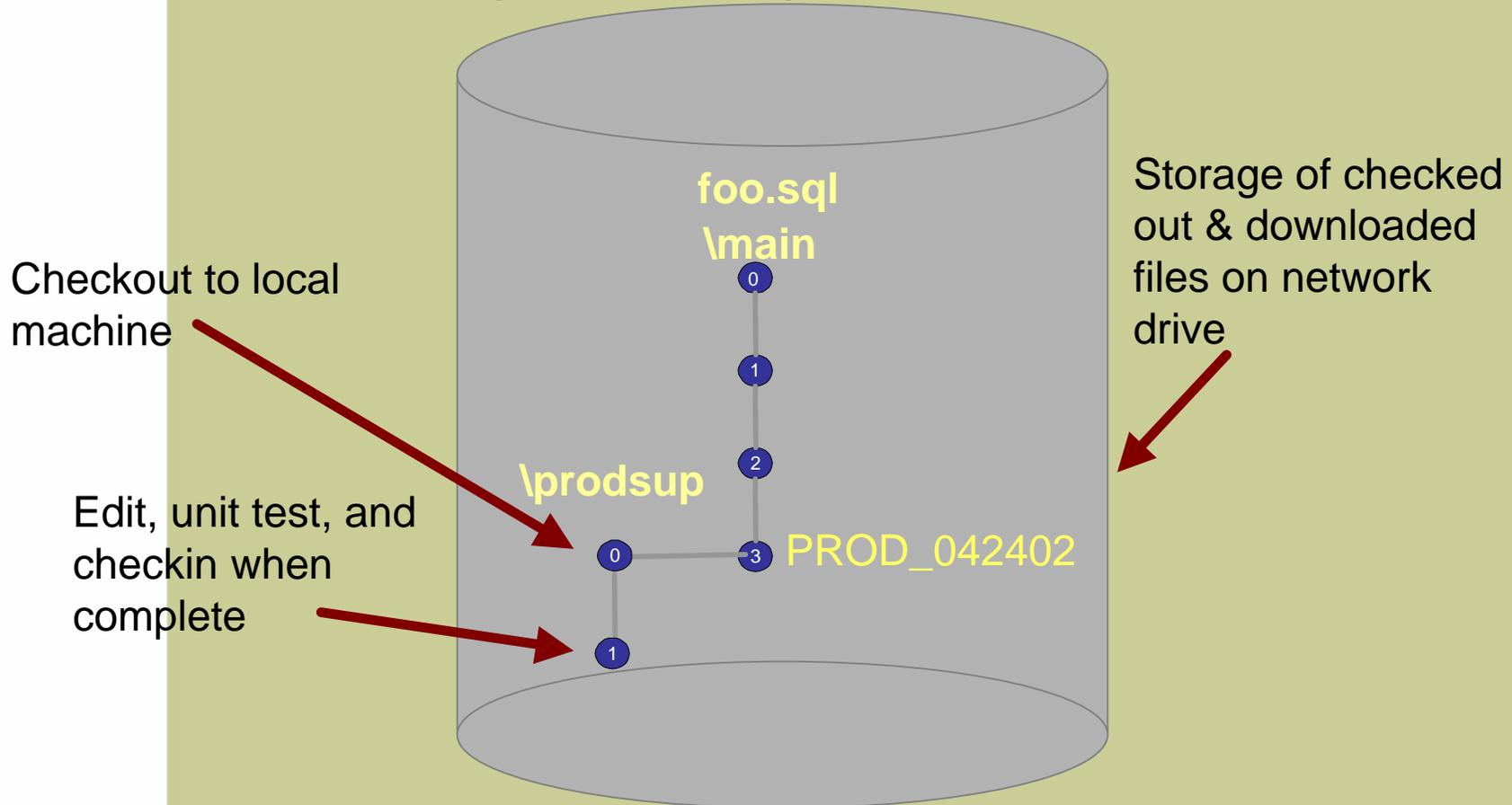
Step 6: Fixes to Production Builds

- Developers can make changes to production builds using the prodsup branch
 - Allows parallel development on fixes to production and coding towards next P4 release
- Checkout and download files to a network drive as needed to support development
- Checkout, modify, unit test, and checkin files directly from ClearCase



Step 6 Process: Done by Developers

Developer ProdSup ClearCase View





Things To Watch For

- Be sure to use a prodsup view... this will automatically make the prodsup branch
- Make sure you branch off the latest PROD_XXXXXX label
- If there is an existing \prodsup branch from a previous PROD_XXXXXX build for this file, the CMA will be able to use the merge tool to ensure proper branching



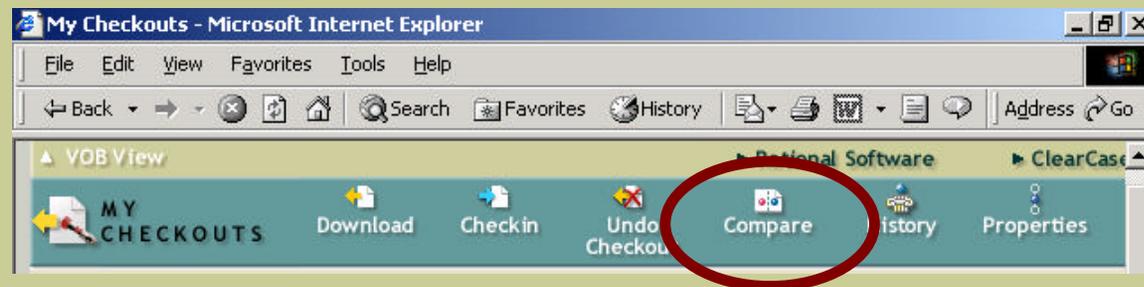
Things To Watch For (cont)

- Prodsup and P4 developers need to communicate to make sure changes from each branch are merged properly
- Developers can merge manually (be sure to update comments) or work with CMA to perform automated merge



Refresher: Directory Compare

- The Compare tool enables you to view, side-by-side, the list of elements appearing in two different versions of the same directory element.
- Can also be used to compare contents between any two versions of any two objects





Refresher: Compare Text Files

- The Compare command displays a textual comparison of the files.

A screenshot of a Microsoft Internet Explorer browser window titled "Compare Versions". The address bar shows a URL from "www.casetool.sfa.ed.gov". The main content area displays a comparison of two SQL files. On the left, "Version 1:" and "Version 2:" are listed. The text of the files is shown in a text area, with a "Compare" button below. The comparison output shows the file paths and a difference: "----[after 0 inserted 1]----" and "> #added line from prodsup view". The status bar at the bottom indicates the working directory is "C:/cc/prodsup".

```
Version 1: /sql/ippg_gl_to_gaps_pkg_b_dev3324_noncont.sql@@/main/1
Version 2: /sql/ippg_gl_to_gaps_pkg_b_dev3324_noncont.sql@@/main/2

Compare

<<< file 1: /FMS/fmsSource/Common/APPL TOP/ippg/Oracle
Version
#/install/sql/ippg_gl to gaps pkg b dev3324 noncont.sql@@/main/1
>>> file 2: /FMS/fmsSource/Common/APPL TOP/ippg/Oracle
Version
#/install/sql/ippg_gl to gaps pkg b dev3324 noncont.sql@@/main/2

----[after 0 inserted 1]----
> #added line from prodsup view
```



CM Process Summary

- Developers are responsible for:
 - Updating files changed since “golden” baseline at cut-over
 - Checking in and out files for P4 development changes
 - Checking in and out files for prodsup changes
 - Reporting files and versions for INT and PROD builds to dev lead
 - Ensuring P4 and prodsup changes are merged together



CM Process Summary (cont)

- Dev leads are responsible for:
 - Creating README files for INT and PROD views with file & version information
- DBAs are responsible for:
 - Verifying INT and PROD builds contain correct files
 - Copying files from INT & PROD build views into the Test or Staging Areas



Where To Get Help

- Coming soon: A checklist with step-by-step procedures to follow for developers
- Ongoing support from CMA
- Process will be continually refined as we see what works, what does not
- Chris available today, Thursday specifically to work on CM process and tool concerns