

Agile Development Using the IBM Rational ClearCase Remote Client (CCRC)

Sujeet Mishra IBM Rational Software sujmishr@in.ibm.com



CRMA43

© 2009 IBM Corporation



Agenda

- Software Configuration Management (SCM) issues important to agile teams
- ClearCase object topologies and practices that support agile development
- ClearCase Remote Client (CCRC) features that support agile development
- Questions/Discussion





Agenda

- What this presentation *is not* about
 - Agile methodologies like scrum, xp, etc.
 - How to connect build management tools to ClearCase
 - How to kick off automated tests following a build



IBM

SCM Issues important to agile teams

Rapid parallel development

- Isolation
- Response to changing requirements
- Geographically distributed development
- Integration
 - Continuous integration
 - Conflict resolution
 - Atomic commit
 - Continuous consistent builds

Yes, all these, but primarily agile teams want to work faster and more efficiently





Respond to changing requirements

- Ability to set aside in-progress changes
 - New high priority defect
 - Lowered priority of new feature
 - Need to be able to easily restore in-progress changes that are set-aside
- Some SCM systems have inbuilt support
- If not
 - A system that makes it easy to do parallel development





Agile Globally Distributed Development

- Additional challenges
 - Planning
 - Communication
 - Architecture
 - ▶ SCM





Continuous integration – definition

Continuous Integration is a set of software development practices, behaviors and principles for automating the integration and inspection of software continuously, so that engineers can detect and fix problems early and validate software quality.







Continuous integration – micro integration

- File level integration
 - Avoid code divergence
 - Find integration issues early
 - Integrating changes within an active development branch
- Deliver stable code on the active development branch
 - Application builds
 - Manual or automated tests succeed
 - No regressions





Continuous integration – macro integration

Component level integration

- Published APIs
- Integration between teams working on larger components
- Pick up new staged derived objects no merging
- Merge in new version of libraries and full build
- Integrating changes between active development branches
- Integration Is for multiple active development branches
- Not likely to be as frequent as micro integration





Atomic commit

- Transactional support for creating a logical grouping of new versions
 - Beneficial for builds that are triggered by the commit
 - Generally irrelevant for nightly builds
 - Beneficial for keeping an active development branch stable
- Compensate if atomic commit is unavailable
 - Build at designated times
 - Use time rules
 - Avoid commits, deliveries and merges close to build time





Continuous build

- Developer workspace build vs. Active development branch build vs. Integration branch build
- Triggered builds
- Scheduled builds
- Nightly builds
- Build tools
 - Rational BuildForge automation



Agile using ClearCase

- ClearCase is a great solution for agile in a large organization
 - Simple, flexible for the agile team
 - Scales to support enterprise complexity
 - Careful not to over use rich feature set Keep it simple
 - Add complexity above the active development branch to maintain agility
 - Or not at all
- Agile topologies and practice
 - Base ClearCase
 - **UCM**





Base ClearCase

- **Isolation** with checkouts, views and branches
- Branches represent promotion levels
 - Developers share a branch for active development
 - Active development branch merged with stabilization branch for release
 - Optionally, branches for features lasting more than a day
 - Optionally, additional branches for testing, intermediate integration

Integration

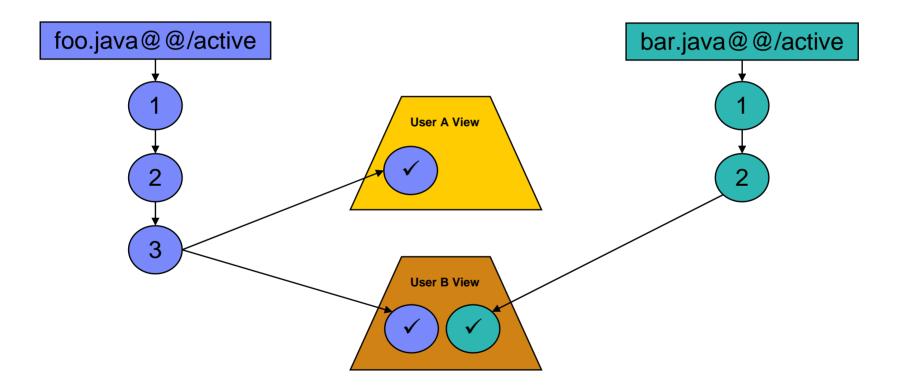
- Checkout and Checkin
- View update
- Findmerge
- New Pending Changes view

Build Consistency

- Timed builds
- Config Spec time rule
- Label



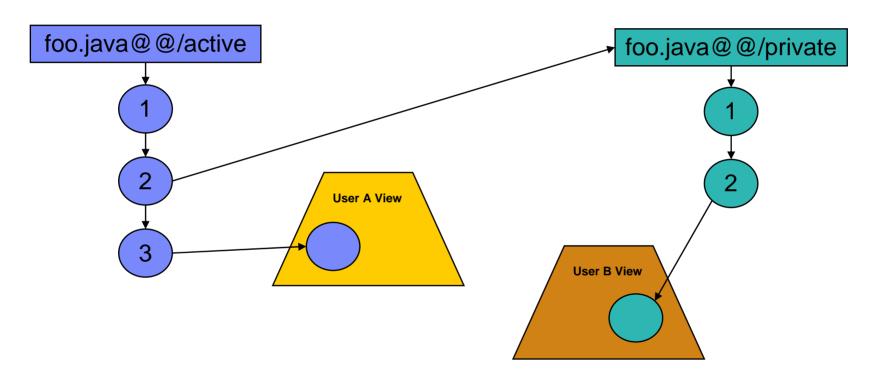
Base ClearCase: View-based isolation







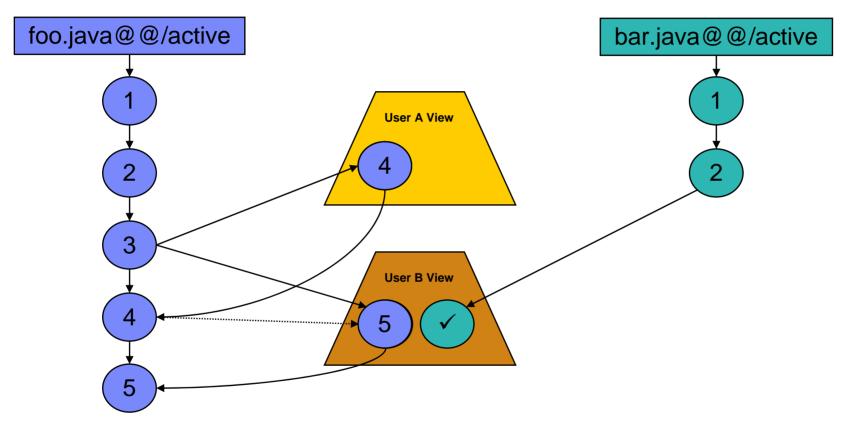
Base ClearCase: Branch-based isolation







Base ClearCase: Integration on the same branch



Checkin makes your changes visible to others

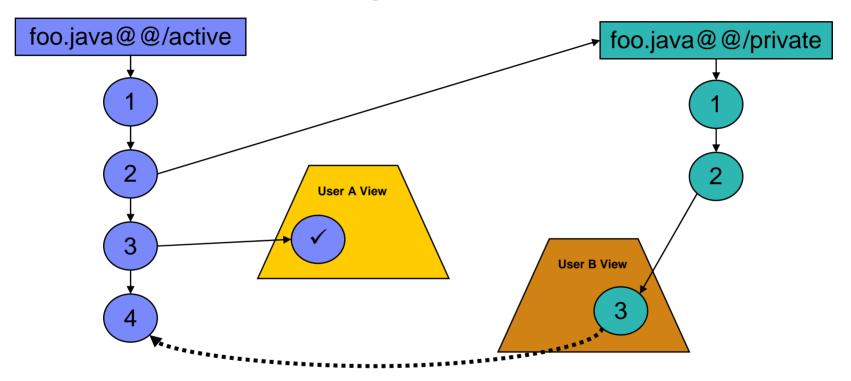
CRMA43

.

View update allows you to see other user's changes



Base ClearCase: Integration between branches

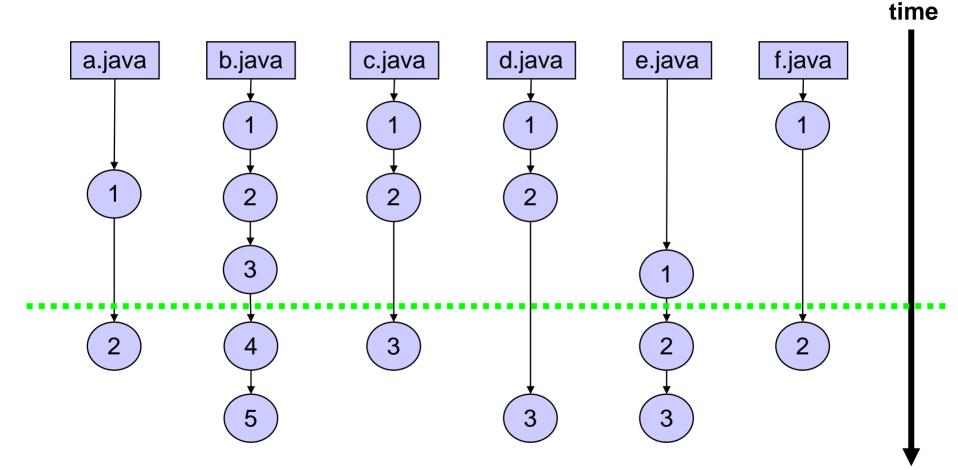


Findmerge allows you to make your changes visible on the active branch

Findmerge also allows you to get the latest changes from the active branch



Build Consistency – Time Rule







ClearCase Remote Client (CCRC)

- ClearCase over a Wide Area (high latency) Network
 - Pockets of remote users where MultiSite would not be cost effective
 - Working at home
 - Working disconnected, traveling
- Minimizes communication between CCRC client and server
 - CCRC server is the ClearCase "client"
- May be used in a LAN environment
 - Supports Web views only not Snapshot or Dynamic views
- Integrated with Visual Studio
- CCRC CLI



Recursive Add to Source Control

Add to Source Control Dialog

	E Add to Source Control						
	Add selected file(s) to source control						
	There are 4 artifacts selected to be added to source control. Show Det				Show <u>D</u> etails		
\leq	☑ Include descendant artifacts of the selected directories						
<	Exclude artifacte named: *3 Apply Filter						
	The selected <u>r</u> esource(s):						
	Artifact	Comment	Checkout	Path			
	🖃 🔽 🚰 New Folder		false	howie_rsdc_cvob/Project Foo 2/New Folder			
	🔽 📓 New File 2		false	howie_rsdc_cvob/Project Foo 2/New Folder/N			
	🔽 📑 New File 1		false	howie_rsdc_cvob/Project Foo 2/New Folder/N			
	🔽 📓 New Ne 4		false	howie_rsdc_cvob/Project Foo 2/New Folder/N	New File 4		
	Check All Check No	one		Select All	Select None		
	0			ОК	Cancel		

- Recursive add to source control
- Filtering

Common Dialogs

Checkout Dialog (Simple)



Checkout Dialog (Customized)

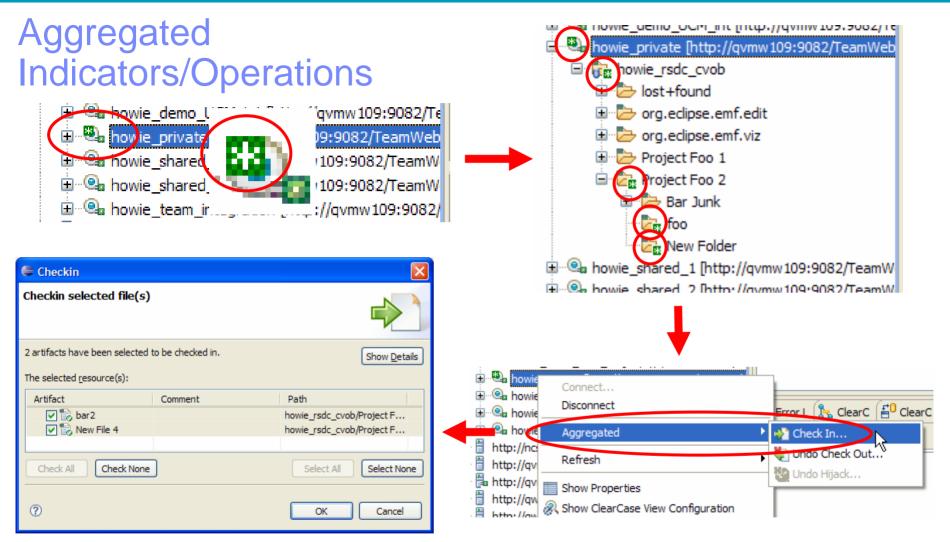
€	Checkout
C	Checkout selected file(s)
	2 artifacts have been selected to be checked out Show Details
-	• Reserved Checkout • Prefer reserved, unreserved if necessary • Unreserved Checkout • OK • Cancel

Checkout Dialog (Detailed)

🖨 Checkout 🛛 🔀								
Checkout selected file(s)								
2 artifacts have been selected t	Hide <u>D</u> etails							
Include descendant artifacts of the selected directories Checkout descendant files only, do not checkout descendant directories								
Include artifacts named: 💟	*,*		Apply Filter					
The selected <u>r</u> esource(s):								
Artifact	Comment	Path						
V Foo1.txt V Foo2.txt		howie_rsdc_cvob/Project Foo 2/Foo1.txt howie_rsdc_cvob/Project Foo 2/Foo2.txt						
Check All Check None]	Select All	Select None					
<u>C</u> omment:								
O Prefer reserved, unreserved if necessary								
Keep the hijack content								
0		ОК	Cancel					

- Simple/Customized/Detailed
- Last use remembered
- Easy to change with preferences





Aggregated change indicator and operations

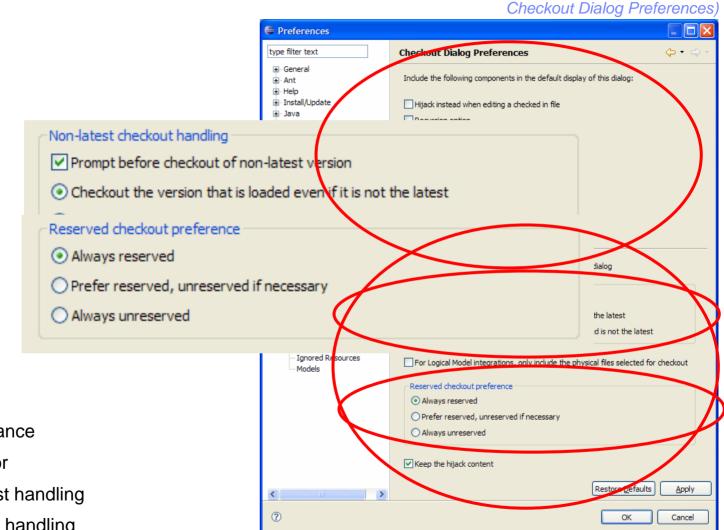
CRMA43

Checkin, Undo Checkout, Undo Hijack

. .



Preferences



- Customize appearance
- Customize behavior

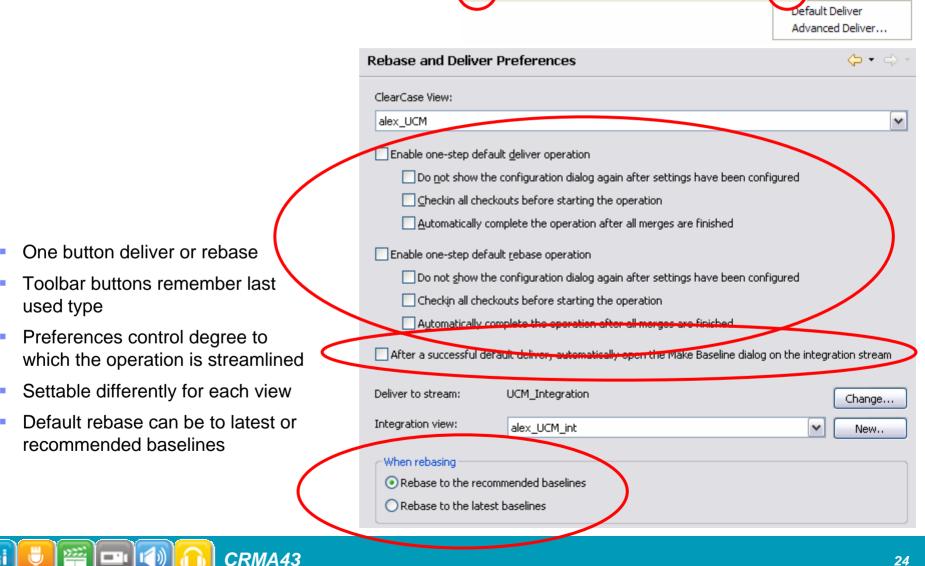
1 H

- Checkout non-latest handling
- Checkout reserved handling
- Other behaviors: hijacks, logical resources



¥

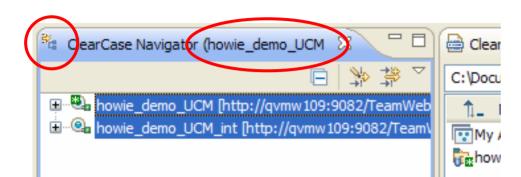
Streamlined Deliver/Rebase



fix the fribberjabber

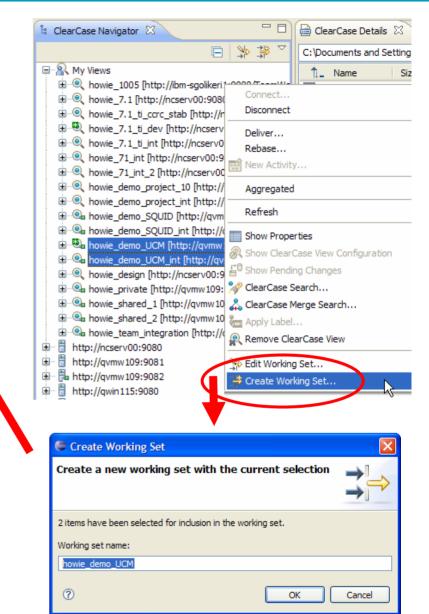


Working Sets



CRMA43

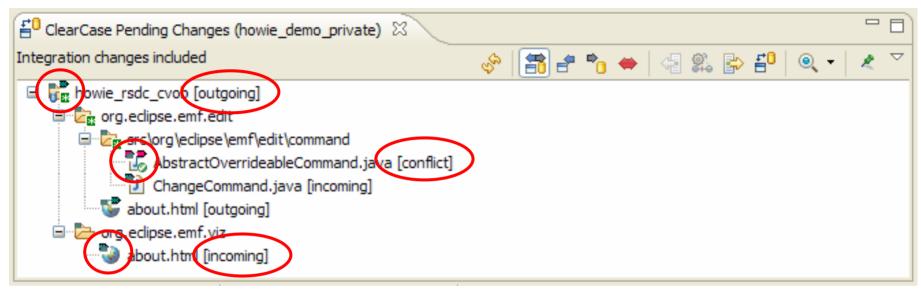
- Save a named, persistent set of objects
- Avoid repetitive navigation to reach those objects
- Combine objects from different parts of the hierarchy into one working set
- Edit working sets
- Export/Import working sets to share with colleagues





Pending Changes View – Base ClearCase

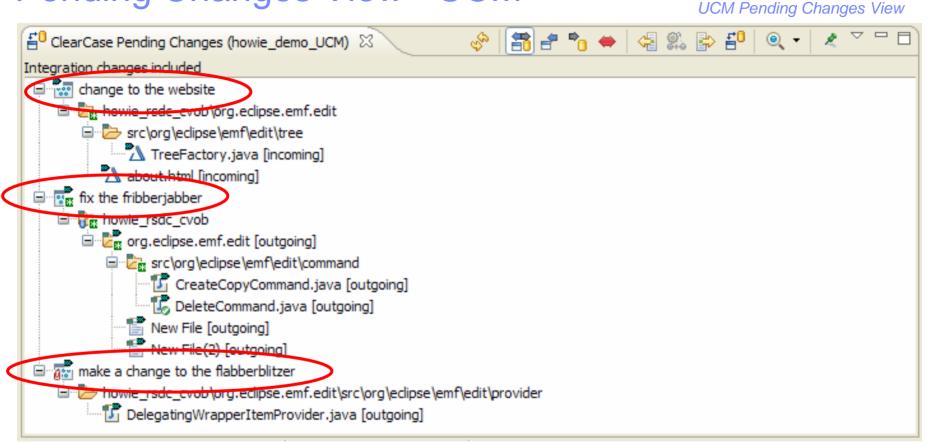
Base ClearCase Pending Changes View



- Combines many operations to display differences between developer's view and the integration view
- Single view for comparison and integration
- Supports all ClearCase operations
- Base CC and UCM, shared branch/stream, private branch, stream



Pending Changes View - UCM

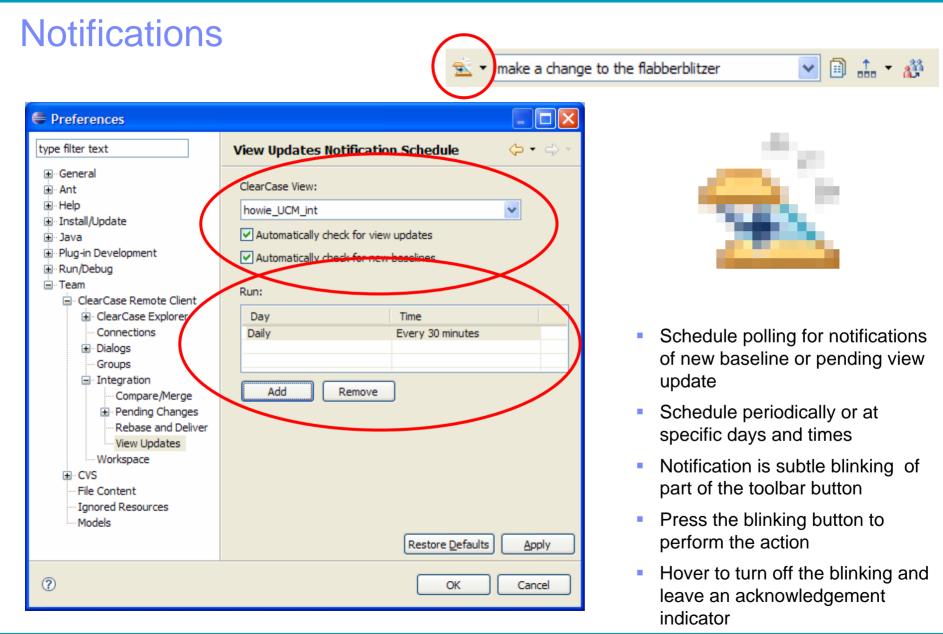


- Combines many operations to display differences between developer's view and the integration view
- Single view for comparison and integration
- Supports all ClearCase operations
- Base CC and UCM, shared branch/stream, private branch, stream

CRMA43

.





222

= . (

Pending Changes Scheduled Refresh

🖨 Preferences							
type filter text	Pending Changes Schedu	ule 🔶 - 🗇 -					
General G	ClearCase View: howie_demo_UCM Note: Pending Changes have to be configured before you can add a schedu Edit Configuration Image: Automatically collect pending changes for this view Day Time Daily Every hour Add Remove						
 Wiew Updates Workspace CVS File Content Ignored Resources Models 		Restore <u>D</u> efaults <u>Apply</u> OK Cancel					

CRMA43

Depending on the content, Pending Changes can be expensive to calculate

Provide a scheduled refresh for pending changes so information is available when needed

Improve Pending Changes to integrate local changes without requiring a refresh













© Copyright IBM Corporation 2009. All rights reserved. The information contained in these materials is provided for informational purposes only, and is provided AS IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, these materials. Nothing contained in these materials is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software. References in these materials to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. Product release dates and/or capabilities referenced in these materials may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way. IBM, the IBM logo, Rational, the Rational logo, Telelogic, the Telelogic logo, and other IBM products and services are trademarks of the International Business Machines Corporation, in the United States, other countries or both. Other company, product, or service names may be trademarks or service marks of others.

