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

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

Rational. software

CRMA43

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

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

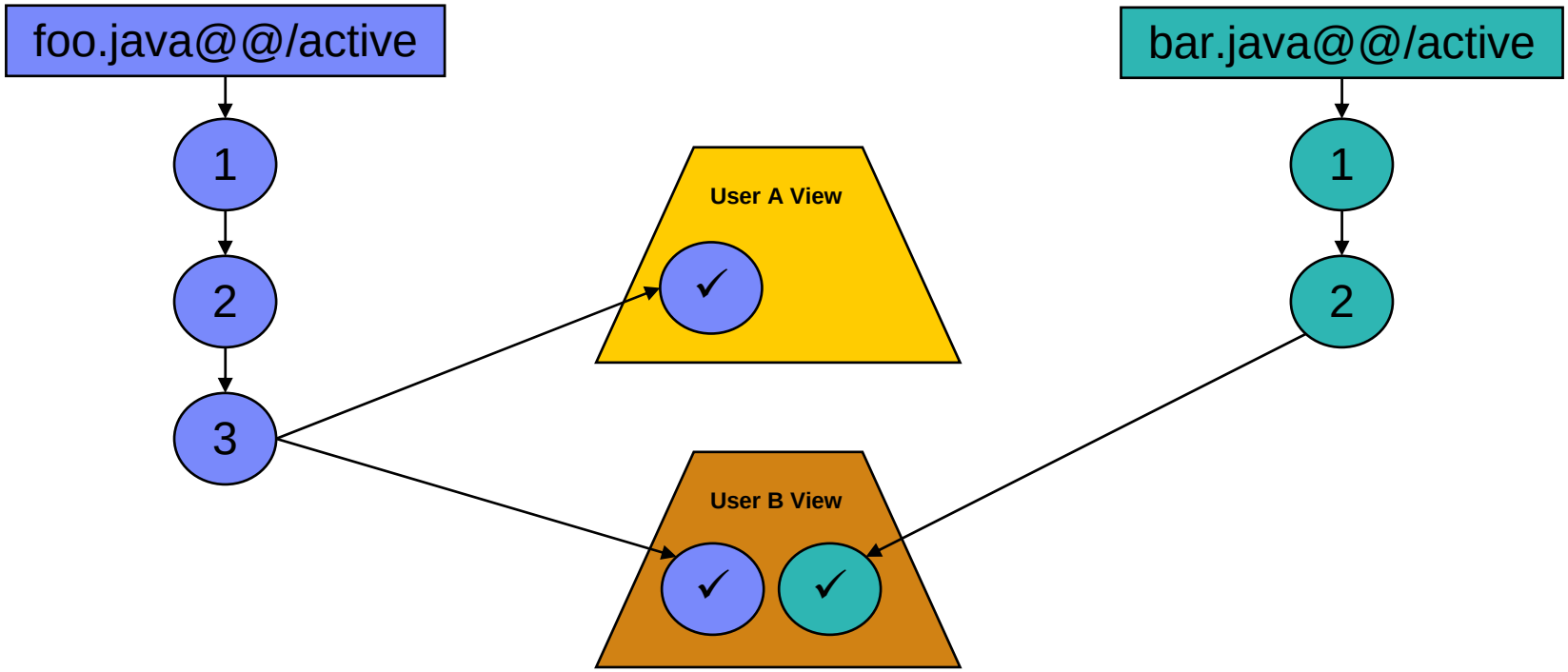
Agile using ClearCase

- 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

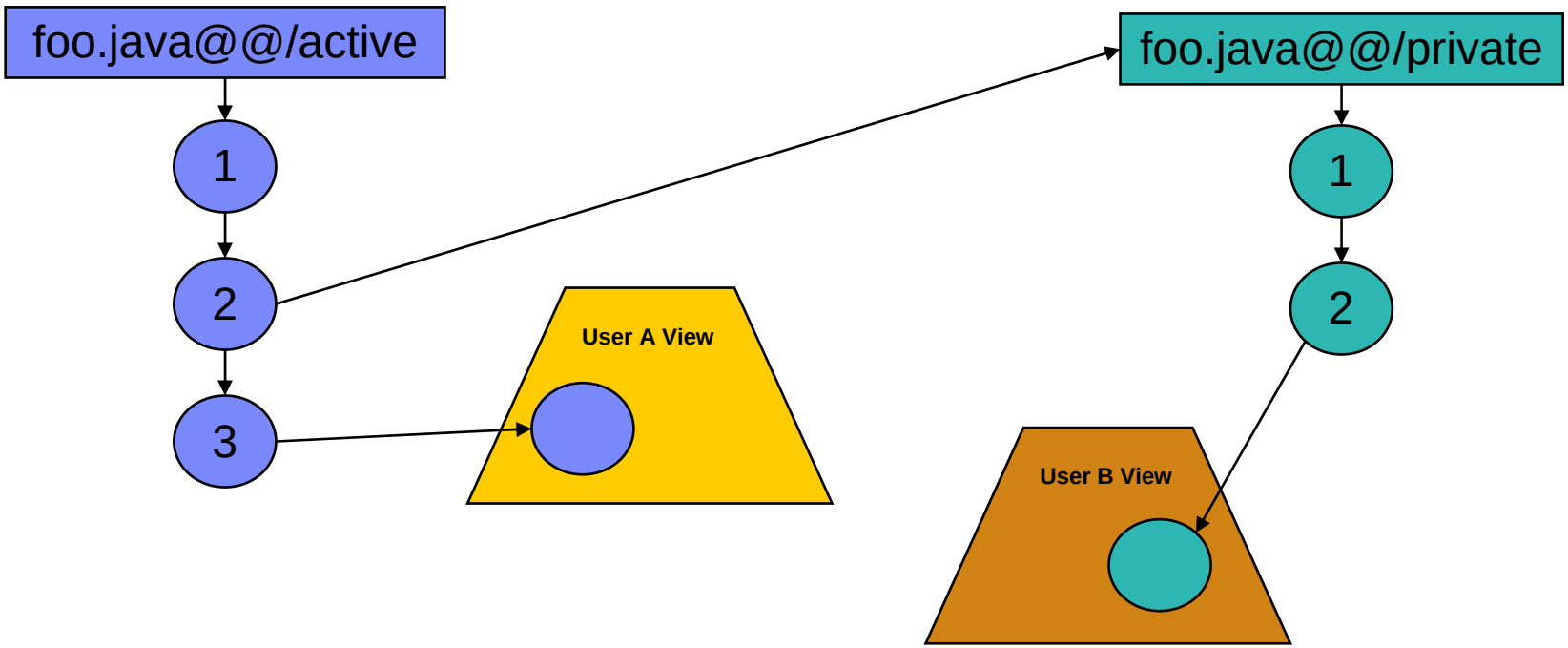
Base ClearCase

- **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

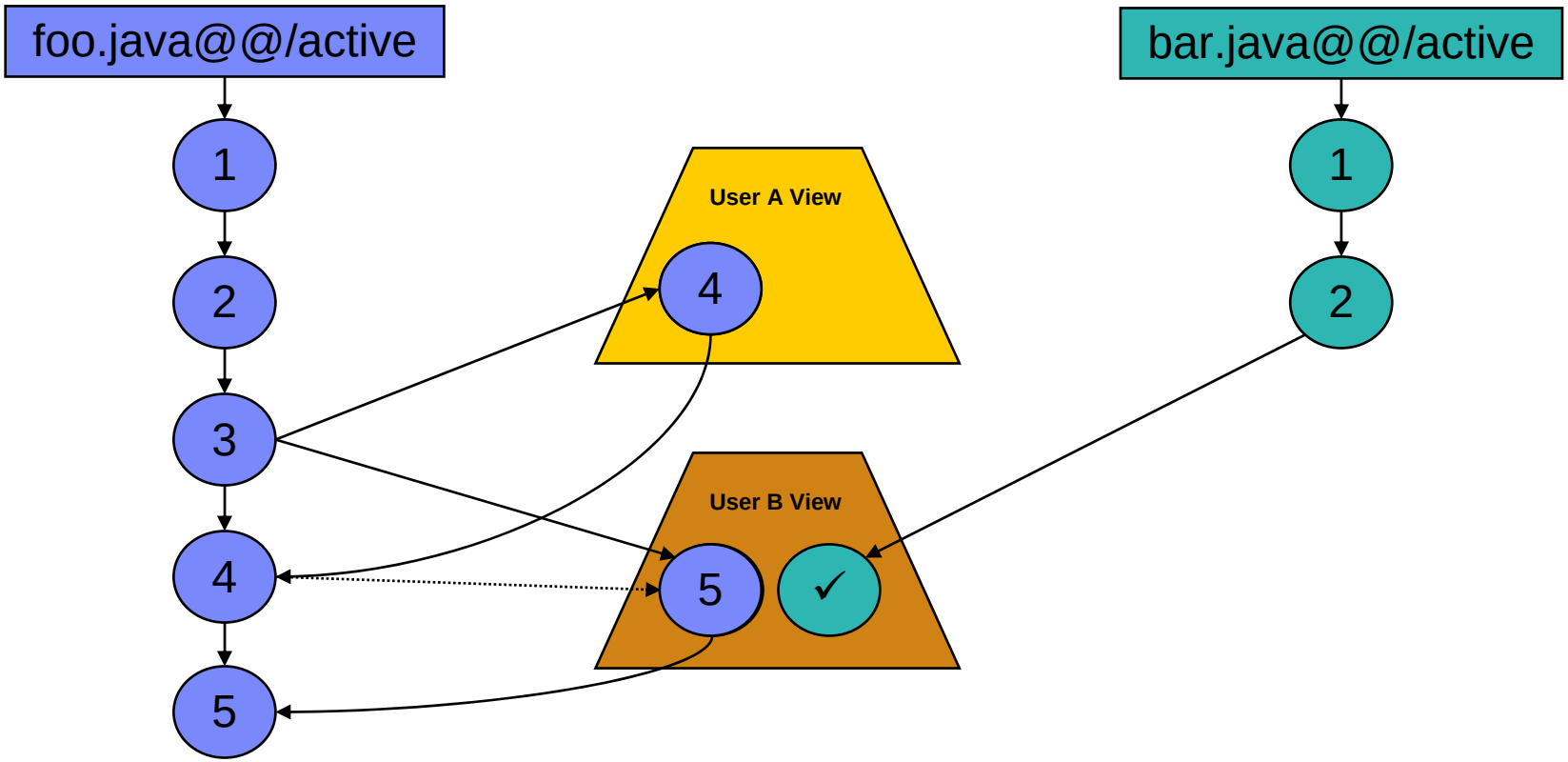
Base ClearCase: View-based isolation



Base ClearCase: Branch-based isolation

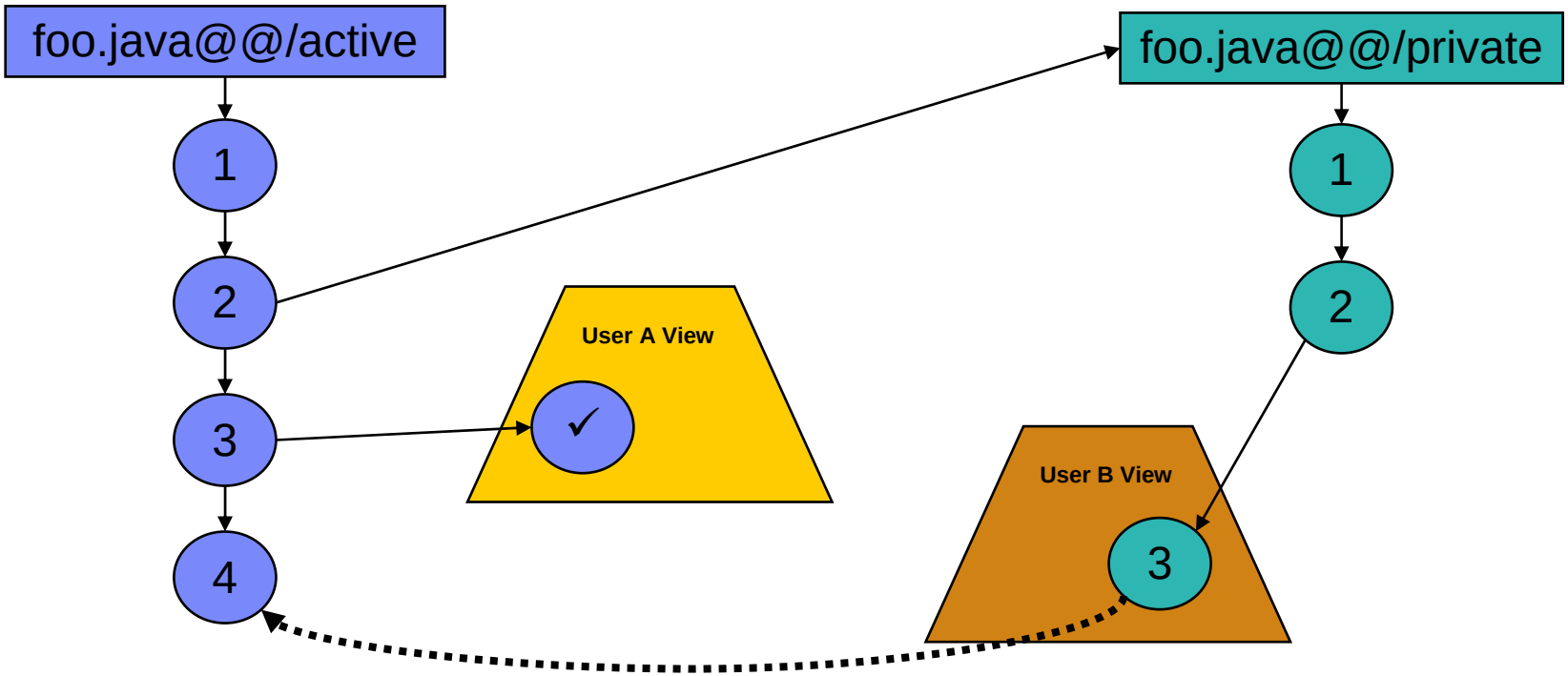


Base ClearCase: Integration on the same branch



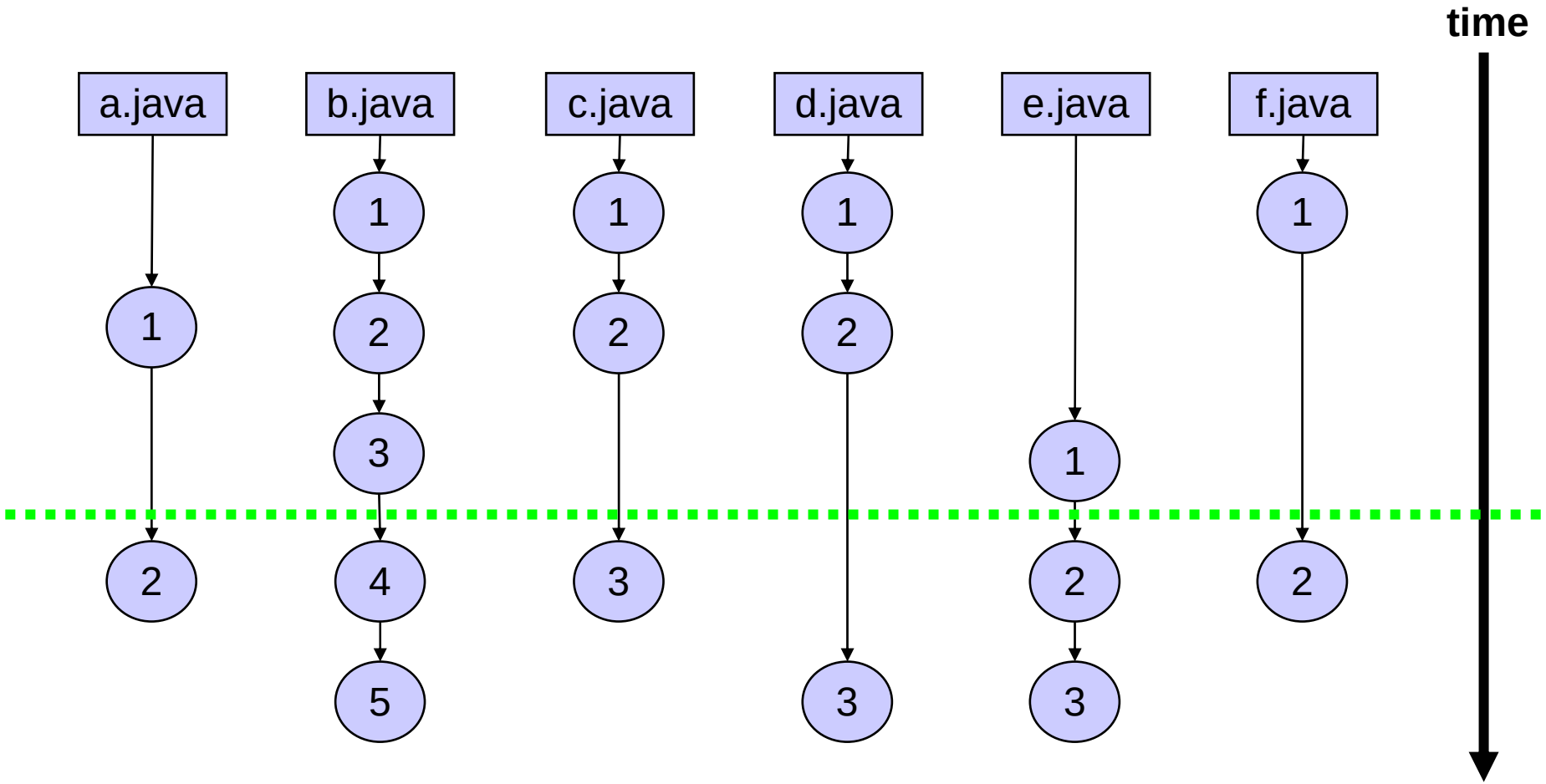
- Checkin makes your changes visible to others
- View update allows you to see other user's changes

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

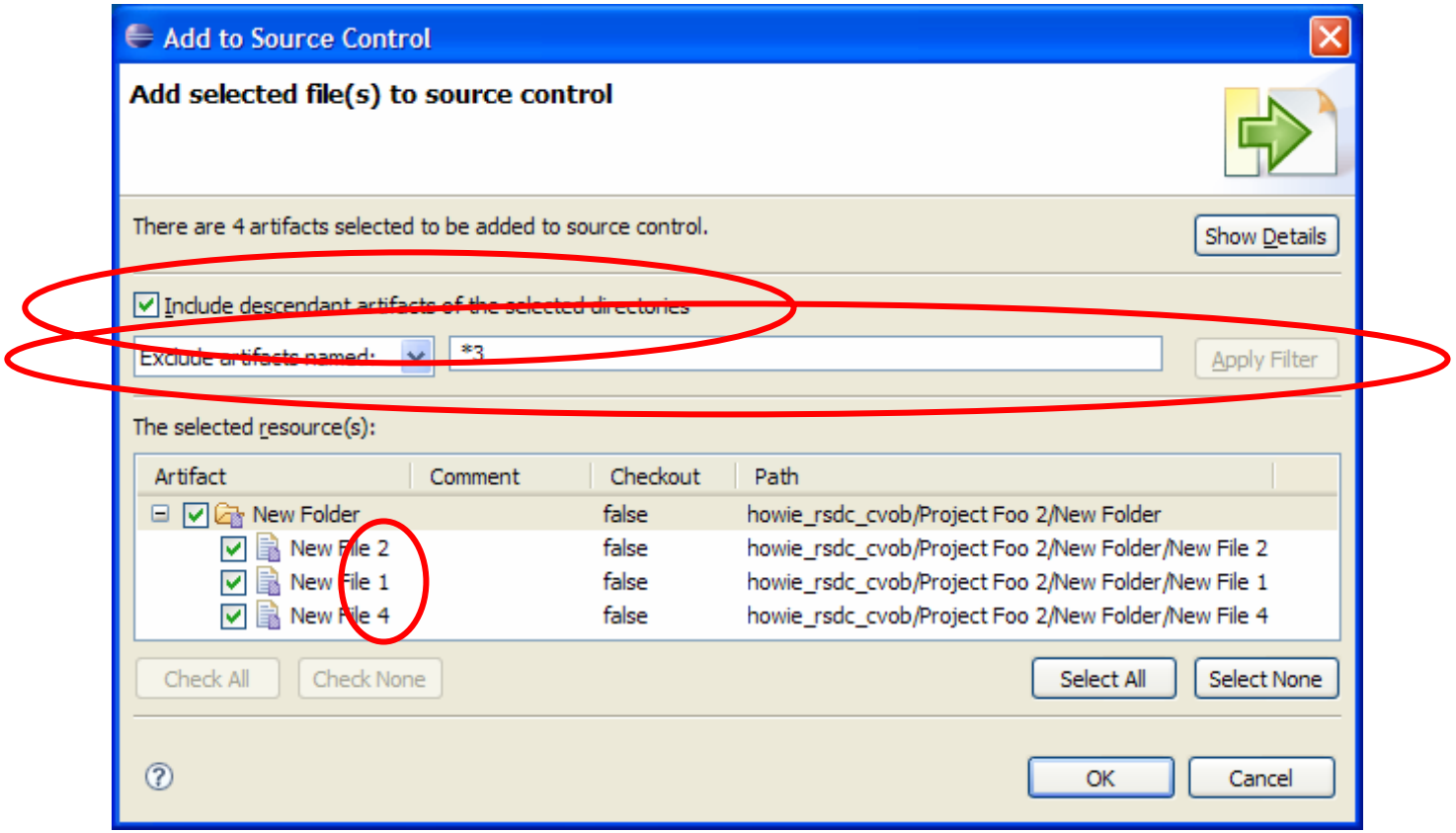


ClearCase Remote Client (CCRC)

- ClearCase over a **W**ide **A**rea (high latency) **N**etwork
  - 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

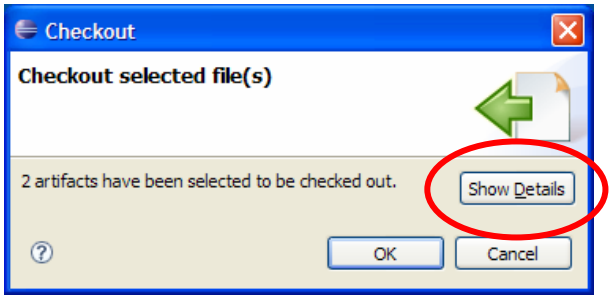
Add to Source Control Dialog



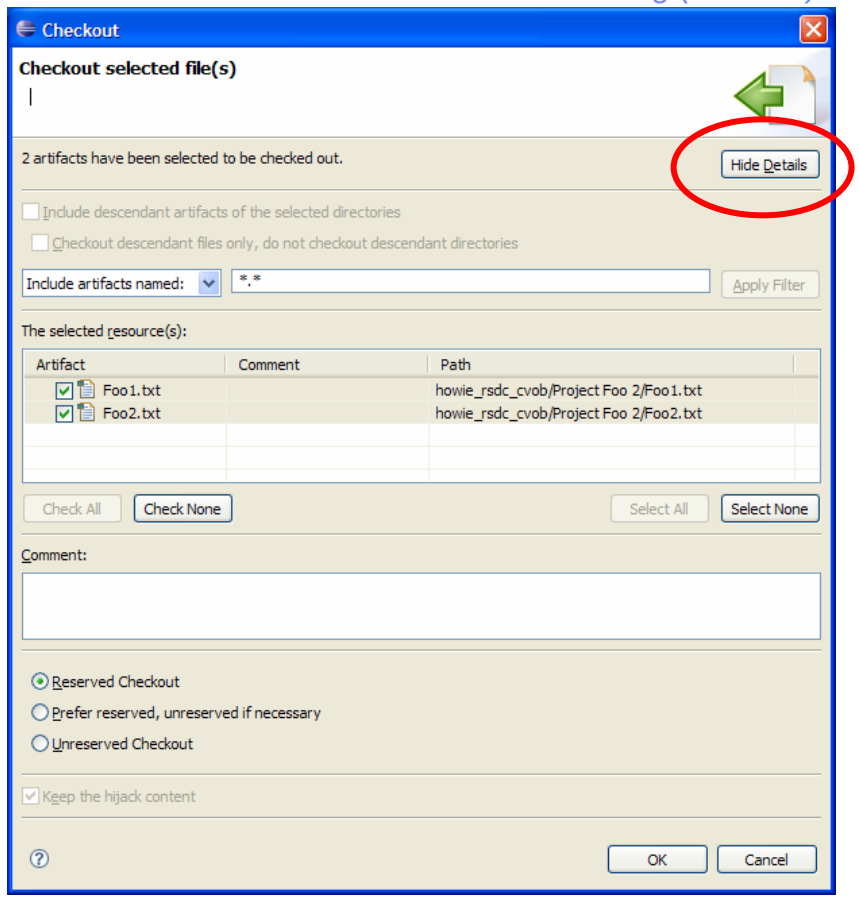
- Recursive add to source control
- Filtering

Common Dialogs

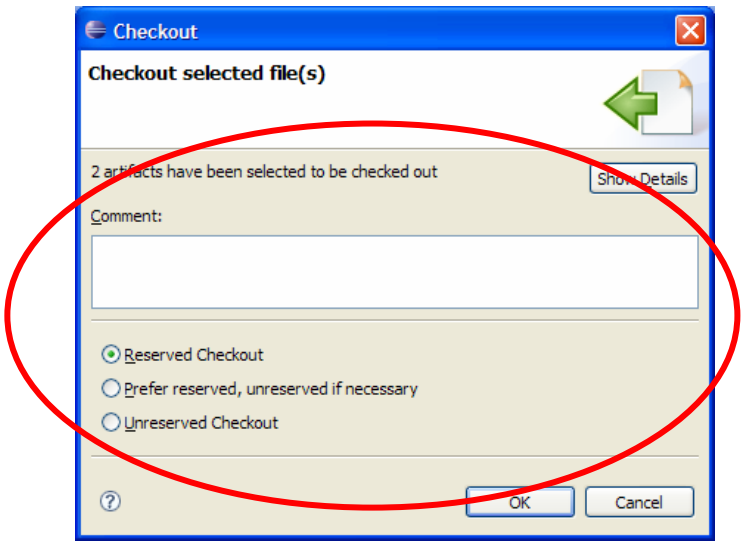
Checkout Dialog (Simple)



Checkout Dialog (Detailed)

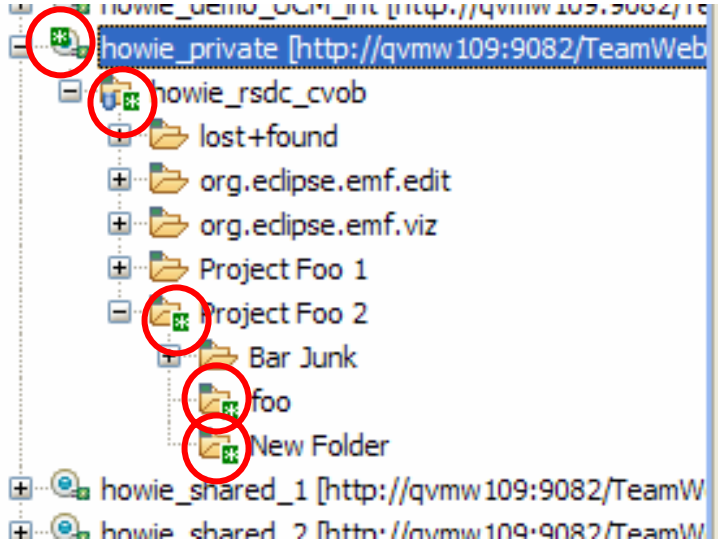


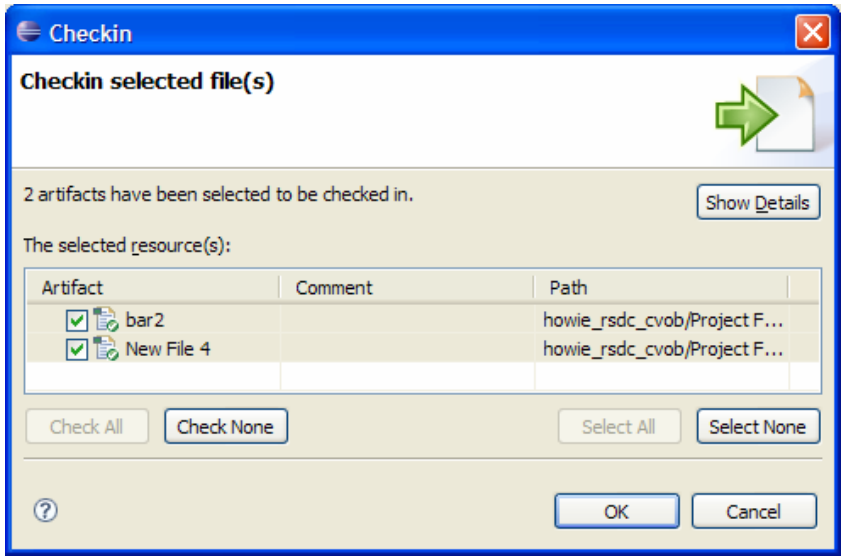
Checkout Dialog (Customized)

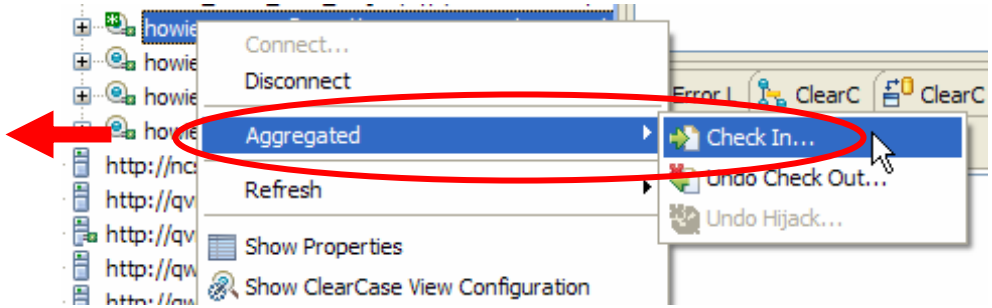


- Simple/Customized/Detailed
- Last use remembered
- Easy to change with preferences

Aggregated Indicators/Operations



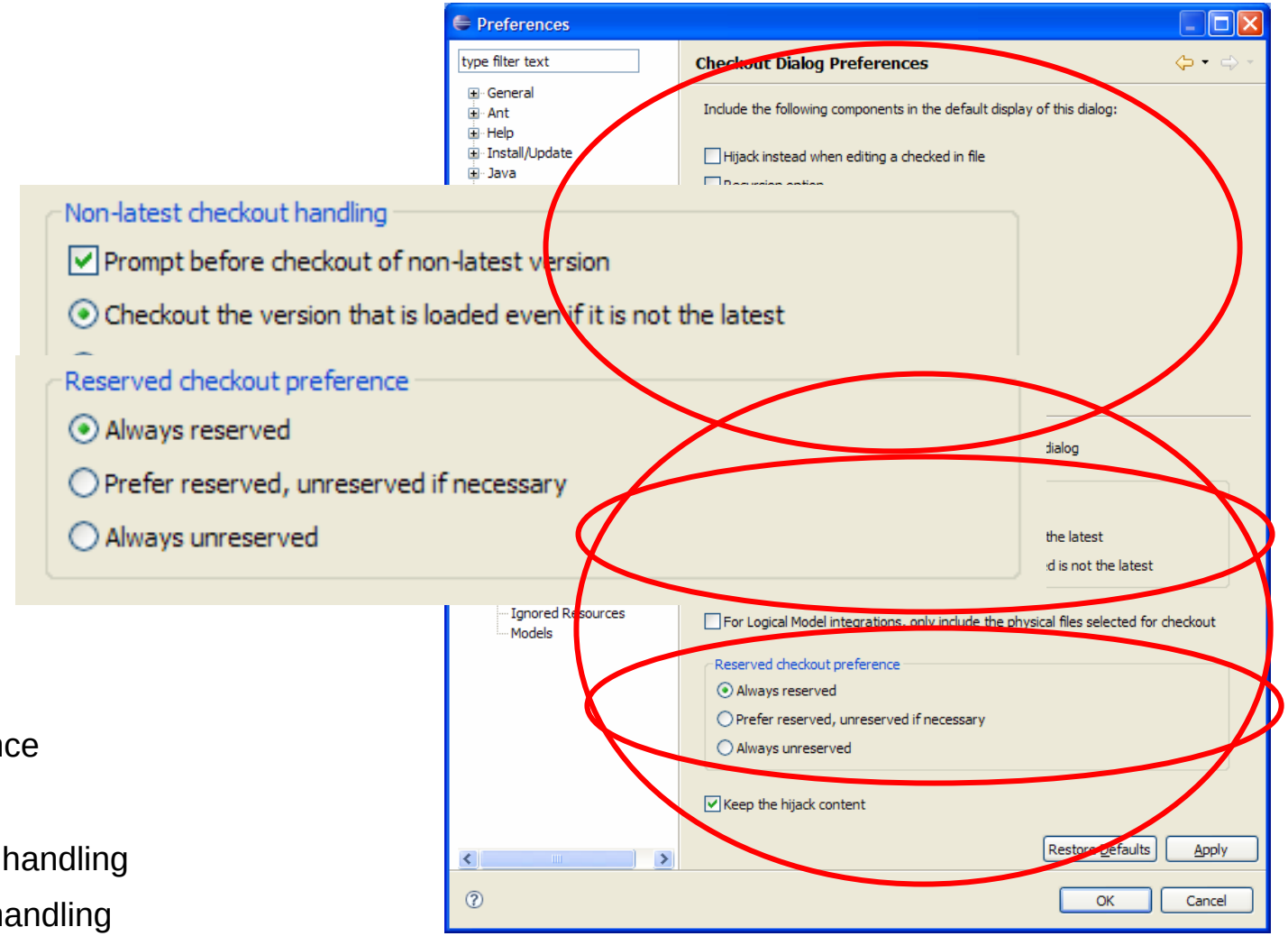




- Aggregated change indicator and operations
- Checkin, Undo Checkout, Undo Hijack

Preferences

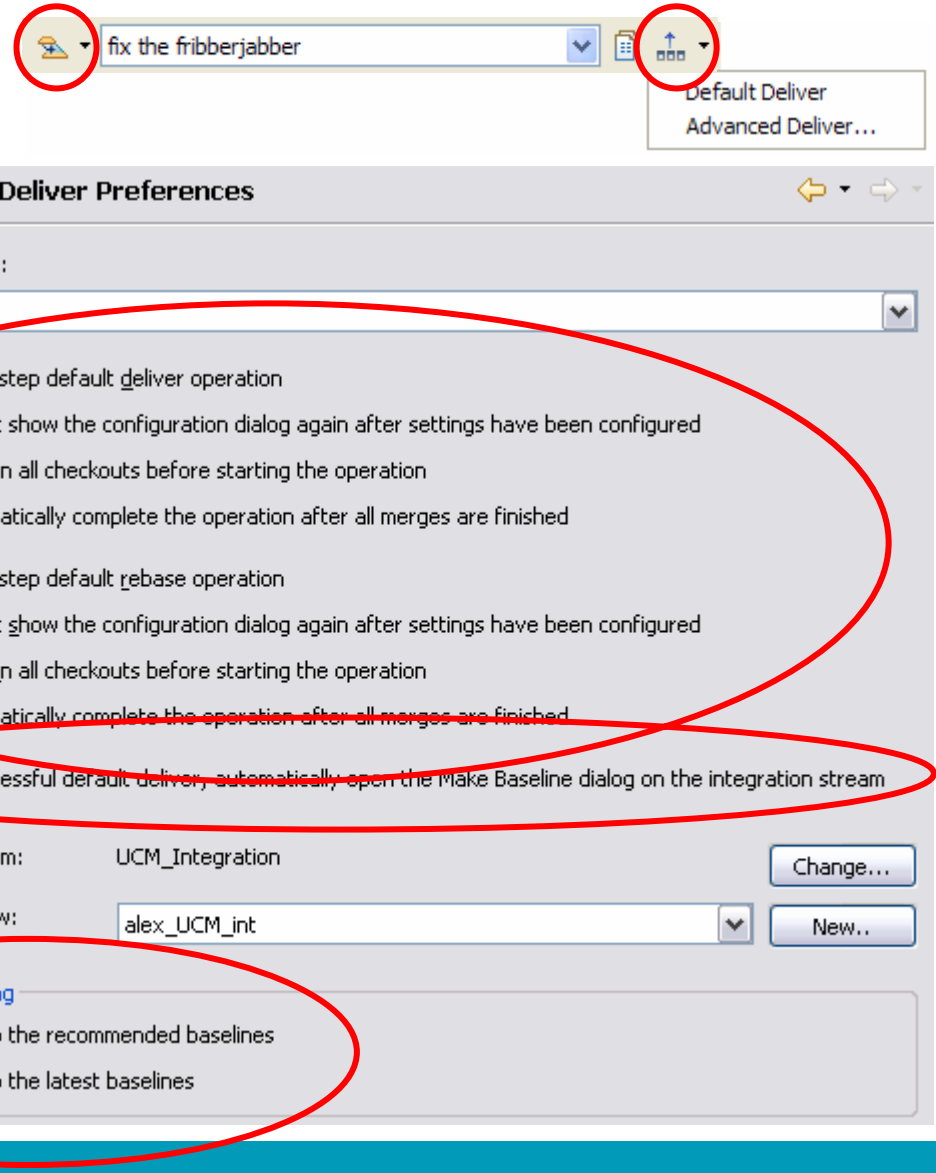
Checkout Dialog Preferences)



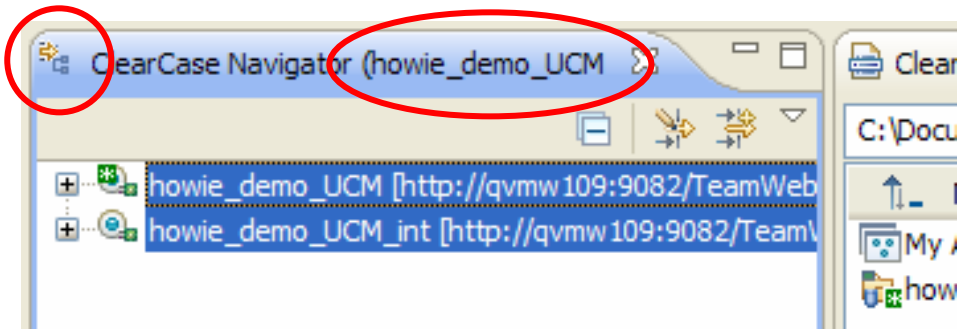
- Customize appearance
- Customize behavior
- Checkout non-latest handling
- Checkout reserved handling
- Other behaviors: hijacks, logical resources

Streamlined Deliver/Rebase

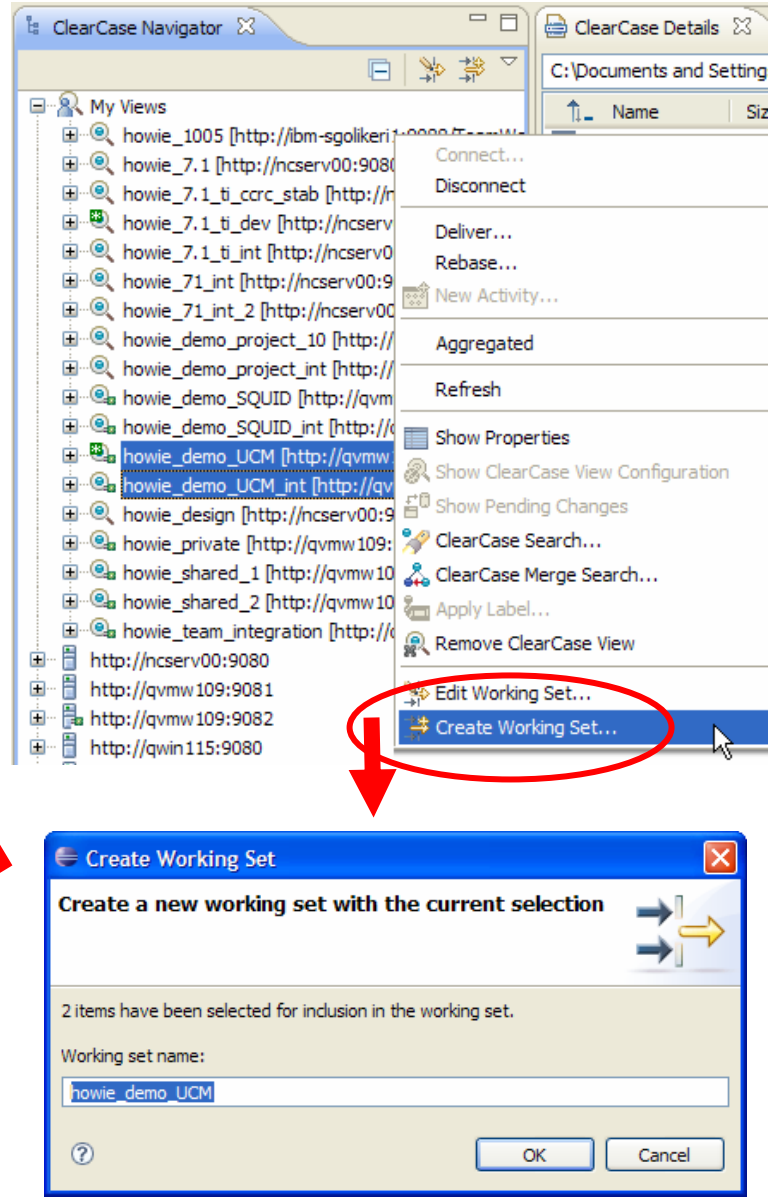
- One button deliver or rebase
- Toolbar buttons remember last used type
- Preferences control degree to which the operation is streamlined
- Settable differently for each view
- Default rebase can be to latest or recommended baselines



Working Sets

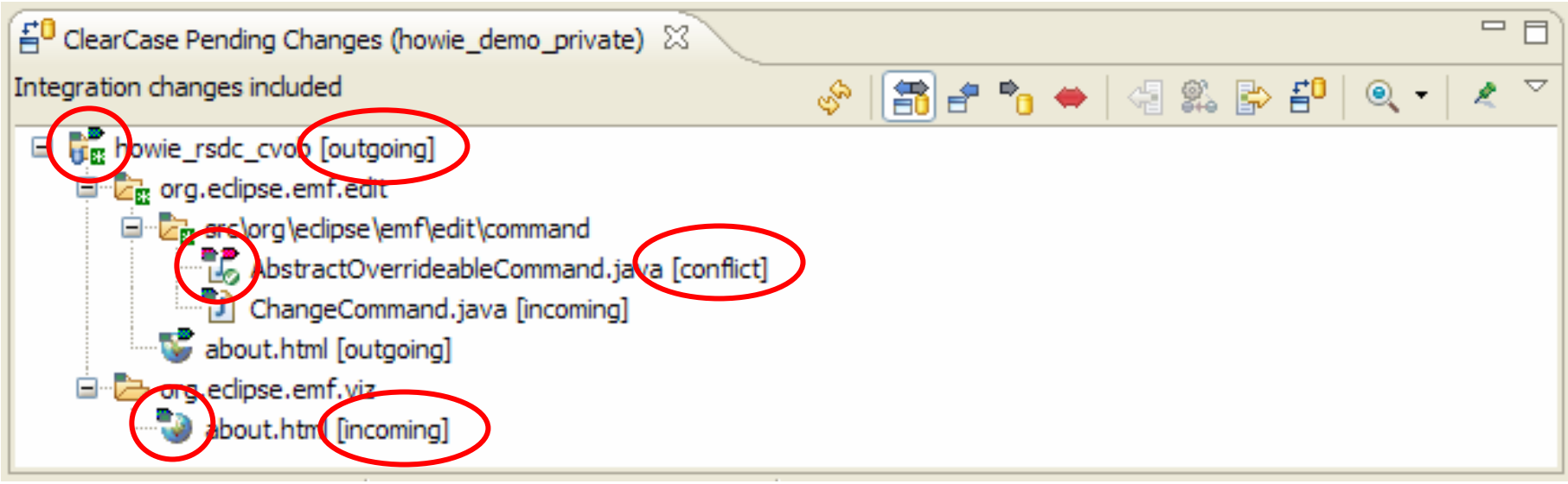


- Save a named, persistent set of objects
- Avoid repetitive navigation to reach those objects
- Combine objects from different parts of the hierarchy into one working set
- Edit working sets
- Export/Import working sets to share with colleagues



Pending Changes View – Base ClearCase

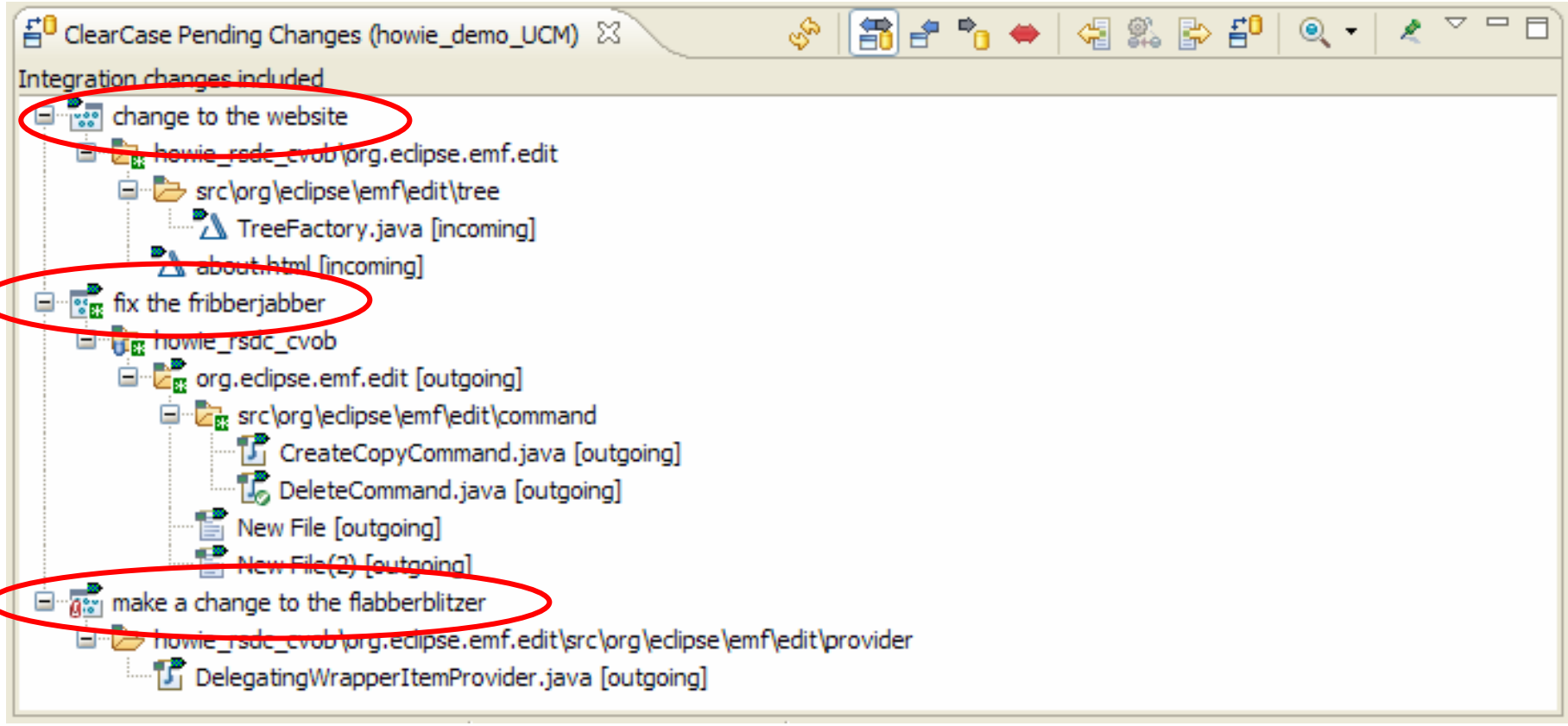
Base ClearCase Pending Changes View



- 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

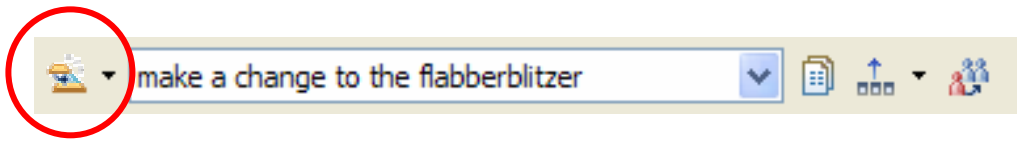
Pending Changes View - UCM

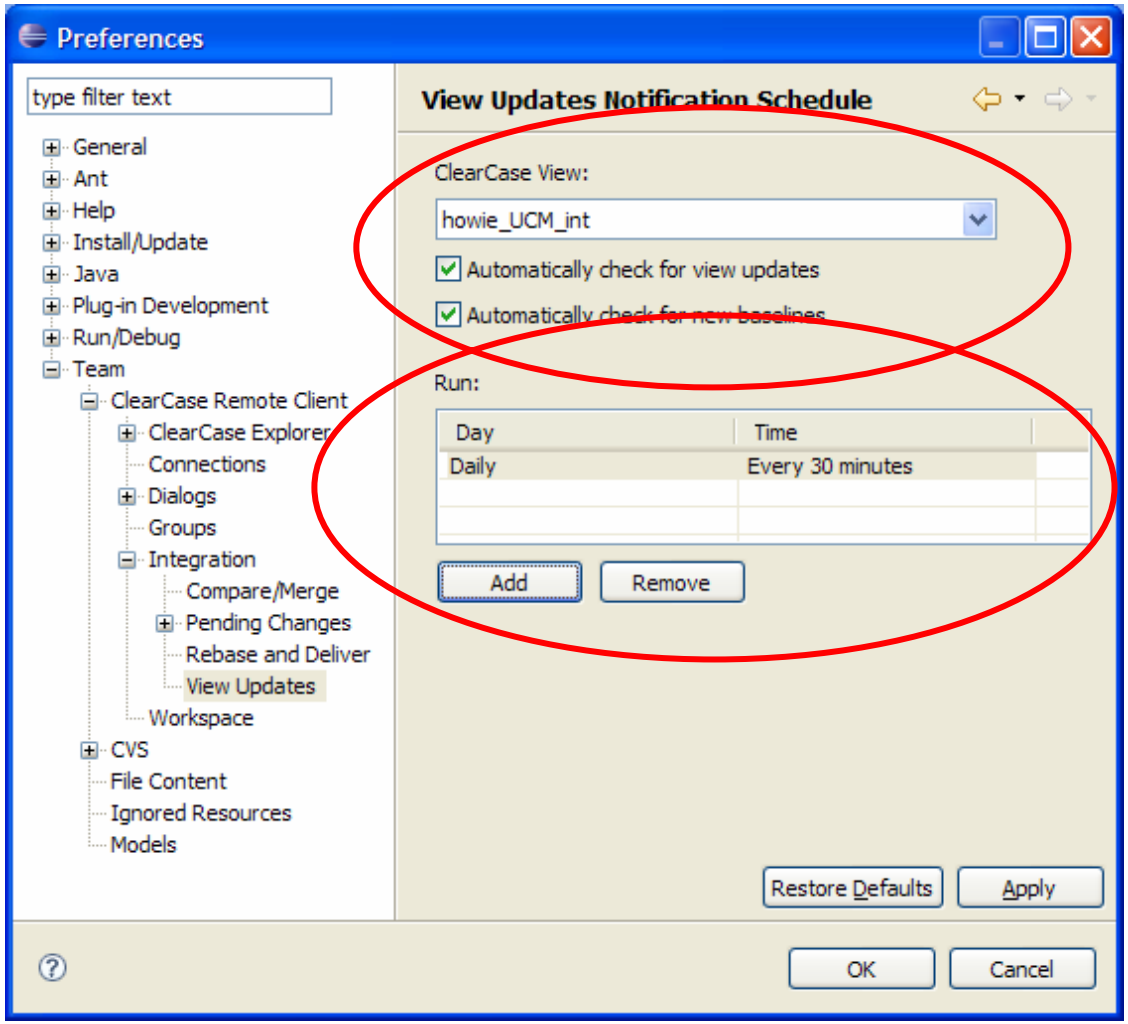
UCM Pending Changes View



- 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

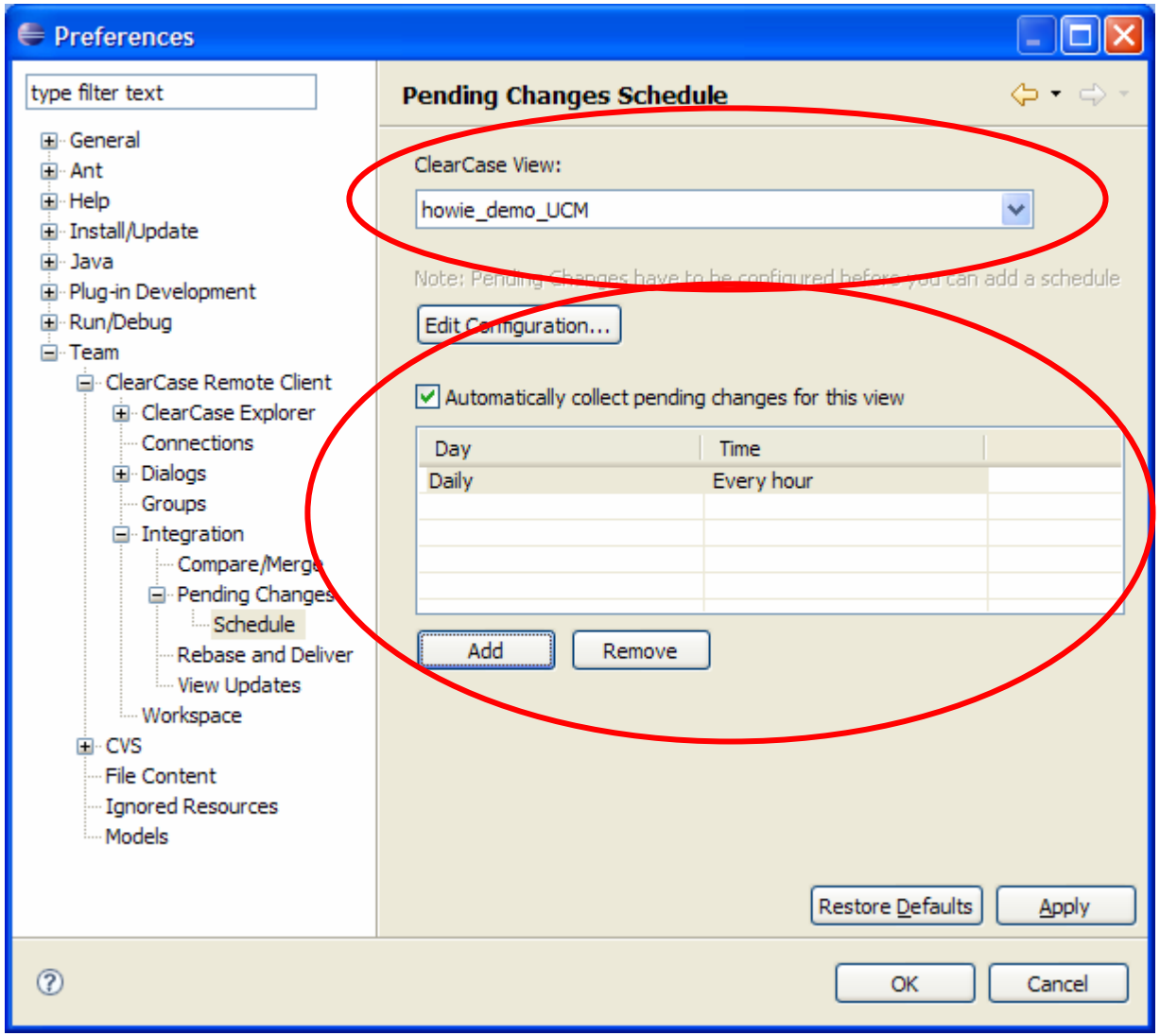
Notifications





- Schedule polling for notifications of new baseline or pending view update
- Schedule periodically or at specific days and times
- Notification is subtle blinking of part of the toolbar button
- Press the blinking button to perform the action
- Hover to turn off the blinking and leave an acknowledgement indicator

Pending Changes Scheduled Refresh



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


Questions