This FAQ covers all of the basics in using Rational ClearQuest and Rational ClearQuest together in your development project. Need to know what functionality you risk when foregoing one product? How about the ins and outs of file checkouts? Check here for answers to some of your UCM questions.

This FAQ was originally published in December, 2000

This FAQ covers all of the basics in using IBM® Rational® ClearQuest® and IBM® Rational® ClearQuest® together in your development project. Need to know what functionality you risk when foregoing one product? How about the ins and outs of file checkouts? Check here for answers to some of your UCM questions. IBM Rational ClearCase and IBM Rational ClearQuest FAQ

1. **What is Unified Change Management?**
   Rational's Unified Change Management (UCM) is a key component of the Rational Unified Process®, a comprehensive framework for delivering best software practices. Both Rational Suite® and IBM® Rational® ClearCase® support Unified Change Management. The foundation or enabling-technology of UCM is Rational ClearCase and Rational ClearQuest, Rational's integrated change management solution. The Rational UCM solution delivers an unprecedented level of change automation across the entire development lifecycle that strengthens team collaborative efforts.
   UCM unifies cross-functional teams through an easy-to-adopt out-of-the-box process that is optimized for team acceptance across all roles. Working with ClearCase and ClearQuest, UCM automatically sets-up each individual's personal workspace with a personalized to-do-list. So, regardless of whether you're an analyst, tester, developer or release engineer, you will have a clear understanding of any project's scope, priorities and all of the activities associated with any project.
   UCM automatically keeps all team members informed of true project status and workload. Every team member is on the same page at all times. Project managers can organize project status, assess workload and estimate completion dates.
   With UCM, integrators and release engineers can assure project quality. Systems can be built consistently through software-reuse and build management. Build-related challenges are greatly minimized through the use of ClearCase UCM components and baselines.

2. **If you use ClearQuest with ClearCase, could enhancements be performed differently than problem fixes within ClearCase? In other words, would ClearQuest help**
differentiate among the types of development activities being performed within ClearCase?
Yes. When using ClearCase and ClearQuest together, the lightweight activities stored in ClearCase are no longer user-visible. Only the ClearQuest entities are used, even if you pull up the property sheet from a ClearCase dialog. Basically, you can UCM-enable any entity type from the ClearQuest designer. One of the big benefits of using ClearQuest in a UCM environment is defining different types that have different states and collect different data. A good example is defects vs. enhancements.

3. What UCM functionality do you have in ClearCase 4.0 without ClearQuest?
ClearQuest gives you scheduled activities, assignment, state, user defined fields and forms, roles and security. ClearCase gives you all new objects and UCM infrastructure, projects, components, baselines, change sets, streams, etc.

4. When a developer delivers changes to an integration stream (branch), are mechanisms in place for merging these changes to MAIN? Can MAIN be utilized in a developer's existing environment, and will a manual merge from data on the integration stream (branch) be required?
There is nothing special about MAIN, other than it is the only branch type that exists in a new VOB, and that most people try to converge code to the MAIN branch over time. For smaller and/or new UCM customers, the integration stream is the main line. What's more, UCM does not take over main primarily to allow existing ClearCase work to proceed. Merges to the MAIN branch in UCM 1 are performed using base ClearCase findmerge options outside the context of UCM.

5. Can customers "undo" change sets?
There are various levels of undo. In UCM 1, you can decide not to deliver a change During delivery, and before checking into the integration stream, you can cancel a delivery. Once a change has been incorporated into a baseline, it cannot be removed via UCM in the first release. However, you can create a new activity called "Remove change set X", and use subtractive merge to create new versions that can remove the old versions. You would then deliver/baseline this change.

6. Can I deliver with files checked out?
Checkouts during delivery are allowable via a UCM project policy. A project leader can set this policy based upon whether or not this behavior is desired.

7. Can I rebase with files checked out?
Checkouts are not allowed during rebase (so we could easily support the cancel operation for rebase). If checkouts were allowed, UCM would need to store a copy of all checked out files (as merging may be required if some files are affected by the rebase). This functionality could be provided in a future release, but for this release of UCM you cannot have any checkouts in your development stream during rebase.

8. Is there an issue with too many baselines? What if I have many rejected baselines?
There should be no issue with this, just like there are no issues with many labels or attributes in base ClearCase

9. Is it possible for two different developers to work on the same activity?
It is only possible for two developers to work on the same activity if they are working in the same stream.

10. Is there any way to "unjoin" or secede from a project?
There is no "unjoin" operation. Users can remove their views and their development stream(s).

11. **When more than one baseline is at or above the recommended level, which one is recommended by a rebase operation?**
The rebase operation will recommend the most recent baseline that meets the project's recommended policy requirements.

12. **Is it possible to rebase to a single activity, or must I always rebase to a baseline?**
The rebase operation will always rebase to a baseline.

13. **Does UCM provide an automated way to move an activity from one stream to another?**
No. ClearCase 4.0 does not provide automated features for selective activity movement between streams. Moving activities between streams can be accomplished manually with additive merge.

14. **Can I associate more than one activity with a single change?**
No. Any change that is made on a UCM stream is associated with a single activity. If using ClearQuest-enabled UCM, it may be possible to create relationships (i.e. parent/child) to establish multiple references to a single activity's change set, however the change set itself is associated with only one activity.

15. **Is there a correlation between views and streams?**
Yes. A stream may have 0..n views attached to it. In the "join project" wizard, a development stream and development view is created. These are named the same unless the user overrides the name of either stream or view. Additionally, each developer will have at least one integration view that is attached to the project's integration stream.

16. **How does UCM change set information get stored/displayed in ClearQuest?**
All UCM change set information is stored in ClearCase, even if the project is CQ-enabled. ClearQuest can display the change set information by referencing the activity through a ClearCase view. The ClearCase/ClearQuest integration will try to find an available view to use if a view context is not available. In some instances, users may not see change set information in ClearQuest if no suitable ClearCase view can be located.

© Copyright IBM Corporation 2004
Trademarks
(www.ibm.com/developerworks/ibm/trademarks/)