



Whitepaper: Combining IBM Rational Change Management with Subversion

Version 1.0

Author Gerry Tombs / Clearvision CM

The content of this document is owned by Clearvision CM.
Disclosure, reproduction or distribution is not permitted without the prior written authority of Clearvision.



Table of Contents

1.	Overview	3
1.1.	The History of Software Change and Configuration Management ...	3
1.2.	The Rise of Open Source SCM.....	3
2.	The SCCM Challenge.....	5
2.1.	Introduction.....	5
2.2.	Achieving the Ideal SCCM Solution.....	5
2.3.	Achieving a One Vendor SCCM Solution	6
2.4.	Vendor Solutions In Today's SCCM Market	6
2.5.	Challenges for SCCM Policy Enforcers.....	7
2.6.	Recent Market Trends.....	7
3.	Migrating Your Change Management Solution.....	9
3.1.	Focusing Investment in the Right Area	9
4.	The Emergence Of Subversion	11
4.1.	A Brief History of Subversion	11
4.2.	Benefits of Subversion	11
4.3.	Disadvantages of Subversion	12
5.	Combining Vendor/Subversion Solutions	13
6.	Integrated Change Management: Rational ClearQuest and Subversion	14
6.1.	Integrating Rational ClearQuest and Subversion	14
6.2.	How does CQ2SVN work?	14
6.3.	CQ2SVN Screen Shots	15
7.	About Clearvision.....	17
7.1.	Document Version History	17

1. OVERVIEW

1.1. *The History of Software Change and Configuration Management*

Software Change and Configuration Management (SCCM) considers the full software development life-cycle from the initial emergence of changes (requirements, customer enhancements, defects) through implementation, build, testing and eventually the formation and post-event management of a software release. Software Configuration Management (SCM) focuses purely on the process of tracking and controlling changes in software (and is therefore a component of SCCM).

A variety of SCCM standards are in use by the main software providers however, the foundations stones for effective software development are clear and understood by all. Multiple products exist to enable effective implementation and day-to-day running of an end to end SCCM strategy and process, the bad news is that today (despite many commercially available and free open source solutions) many companies continue to use spreadsheets and paper based systems.

The basic principles of source code management were originally documented in the early 1960's by Professor Leon Pressor however, the first serious source code management solutions were not available until the 1970's. Early solutions were created to help individuals manage their personal workspace for small quantities of files, modern tools can handle 1000's of users in multiple locations with many more multiples of files and complexity.

One of the largest changes in the commercial market was the launch of ClearCase in 1992 by Atria Software (who were subsequently purchased by Rational Software and ultimately by IBM). ClearCase was fundamentally different with its Dynamic View approach and eventually led IBM to develop (and purchase) multiple tools to manage the end to end software development process such as Rational ClearQuest, Rational RequisitePro, Rational Build Forge etc. and to create one of the more popular SCM methodologies, the Rational Unified Process (RUP).

1.2. *The Rise of Open Source SCM*

This paper discusses relatively recent changes to the SCCM marketplace, in particular the rise and success of the open source SCM product Subversion.

The intended audience for this paper are SCCM decision makers/architects who define and enforce process and policy within their organisation. SCCM product budget holders are also likely to find points of interest.

In the current economy every organisation is closely examining costs and looking for possible reductions. At the same time developers want to use easier and leaner change and configuration management solutions with less over-bearing process, restriction



and control. Subversion seems to meet the SCM goals of both parties: no license cost, simple to use, quick to learn, inexpensive to administer and maintain.

Subversion does have some disadvantages which are described later in this document but more importantly it is only half the story, the other key component of an effective software development process is change management. Regardless of which SCM strategy is adopted one of the most valuable and in many cases straight forward integrations is between change management and version control tools. Recording a change set within a change management tool enables or simplifies:

- Continuous integration – one standard process.
- Full audit, traceability and detailed history of change.
- Improved communication and visibility throughout the organisation (including communication to customers).
- Reduction in duplication of effort e.g. the same defect recorded multiple times.
- Automated builds.
- Automated and detailed release notes.
- Improved testing i.e. each requirement can be specifically tested upon release.
- Compliance to standards such as ISO and SOX etc.
- Management and balance of workload across individual developers.
- Improved reporting and management information.

Although not trivial, it is generally considered simpler to migrate software and data into new configuration management tools such as Subversion than to migrate change records and processes into a new change management tool.

The success of open source software is, without a doubt, impacting traditional vendor solutions. The smart providers such as IBM embrace open source solutions to extend their products and provide flexible and fully functional development environments to help meet the needs and requirements of software development organisations.

2. THE SCCM CHALLENGE

2.1. Introduction

The software development industry is clear that change and configuration management tools add value, improve software quality and reduce ongoing costs, particularly as development teams grow larger, span multiple locations and develop increasingly complex products. For effective software development it is no longer enough for these tools to stand in isolation, they need to be integrated, flexible and adaptable.

Of all the tools used within the software development process, the simplest of integrations, generally revealing more benefit than any other (greatest return on investment) is between change management and configuration management.

2.2. Achieving the Ideal SCCM Solution

The aims and key requirements for any SCCM system will vary depending upon the role and tasks of the individual user:

From a developer perspective the ideal SCCM solution must:

- Be unobtrusive and easy to use.
- Allow them to work against one or more assigned activities.
- Give them the ability to have more than one active work item at any one time BUT not have each work item interfere with each other.
- Associate the resolution for one change record with other change records which are also resolved by the same fix.
- Make their code available for testing quickly and easily.
- Record requirements - clearly understood (XP / Agile) or defined (RUP).
- Manage merging i.e. only integrate the files which actually need to be merged.
- Work from a “to do” list of items assigned specifically to each developer (also allowing the manager to prioritise work items).

From a Development Team Lead/Project Manager perspective the ideal SCCM solution must:

- Easily receive, evaluate and manage change requests.
- Easily assign work items or split into smaller work items as appropriate for the specifics of the team OR the target delivery i.e. a single work request might need to be broken down into smaller work items because of team capabilities, time constraints or flexibility for staggered delivery of the original work item.
- See which work items are complete verses incomplete.
- Easily assign work items for testing and inclusion to a build.
- Provide visibility at various levels including status of project, release, components and individual changes. Includes management reports such as change record distribution across team members, average bug fix time etc.

- Provide visibility of individual developers workload and progress.

From a Business Analyst/Manager perspective the ideal SCCM solution must:

- Allow them to request/register new requirements.
- Run reports against people, costs, project status, etc.
- Keep their customers updated with the progress of new releases.

2.3. Achieving a One Vendor SCCM Solution

All software development organisations want an SCCM solution which is scalable, flexible, adaptable, all encompassing and ideally an integrated suite of tools (covering requirements definition through to testing and deployment) from a single vendor that will slot neatly into existing development polices and processes (at the lowest possible cost). Primarily the driver is to avoid making and maintaining bespoke integrations but rather implement a reliable and tested solution which will not break integrations as new version upgrades are installed.

There is a general perception which says ‘it would be best if we could purchase our complete development environment from a single vendor’ but the reality is this happens rarely. The reasons are obvious, every product influencer has their favourite vendors and products, legacy tools become deeply rooted into organisations making them hard to migrate from and over time new tools enter the market using modern technologies with functionality which out-performs last months “tool of choice”. Regardless of how you look at it, the reality for most organisations is a mixture of software development products.

2.4. Vendor Solutions In Today’s SCCM Market

In the software development market today there exists the following types of vendor:

1. Single flagship product with potential integrations to other aspects of the SCCM suite.
2. Multiple point tools from one vendor which claim to integrate seamlessly.
3. One product vendor solutions for end to end SCCM.
4. Single licensed, all encompassing Application Lifecycle Management (ALM) solutions.
5. Hosted solutions on remote servers (software as a service - SaaS).

There are so many choices, any decision can be overwhelming and difficult with each one having their own advantages/disadvantages. If starting a development team today the latter options sound like the obvious choice, however the problem with both options 4 and 5 is that one solution does not fit all. Inevitably many organisations will recognise that an alternative product fulfils part of their overall solution better, that they are paying for parts of the solution they will never use or that the solution is not flexible enough for their preferred development techniques (xp, agile, iterative). The truth about “out of the box” ALM solutions is that all customers are different, the more functionality becomes built into an ALM system the harder it becomes to adapt and customise that solution for the needs and processes of each customer. The best



ALM solutions are lightweight and allow the customer to adapt or integrate them with other products.

No one would claim a different product/vendor should be used for every software development project but we must openly acknowledge that a utopia is difficult to find. There are a vast selection of excellent point tools available, the challenge is around tighter, cleaner integration. The ideal world would be to have one common interface regardless of the underlying SCM products.

2.5. Challenges for SCCM Policy Enforcers

The past 2 years have seen a shift in attitude by all parties involved in software development. Historically the consensus was to rigidly guard all source code and strictly govern how it is managed, more recent software development techniques suggest that the tighter the shackles the more difficult it becomes to develop products on time, within budget and to a high quality level.

As pressure to get products to market increases developers inevitably spend more time developing/testing and resist time spent following what they consider to be bureaucratic process, using complex tools and creating a solid, integrated SCM structure. The open nature of Subversion allows developers to download and start using it to effectively circumnavigate company policy without having to seek approval or authorisation. Traditional SCCM policy enforcers want to keep a clear process for good reason but find themselves powerless to intervene as developers force the issue. In fact they are often better to embrace this change rather than fighting against it.

Tools such as Subversion address the minimum business and development requirements, with the most basic and essential features. Why not allow developers to use faster, more efficient tools if integrated with a company approved solution which provides all the back up, security and audit requirements needed to meet with compliance?

2.6. Recent Market Trends

Cost has always been a major factor in the uptake of commercially available SCCM products, many companies are now moving to the simpler “use and learn” tools for many aspects of an SCCM suite, especially Open Source SCM solutions such as Subversion or Mercurial.

IBM Rational has been at the front of the development tools market for years with their integrated suite of products. Traditionally they have not created integrations beyond their own suite of tools however more recently they have acknowledged there is a wider market opportunity, especially with tools from the Open Source arena.

In summary, many organisations recognise they no longer require overly complex configuration management solutions so long as they place more emphasis on the



change management solution, placing tools such as Rational ClearQuest at the heart of development.

3. MIGRATING YOUR CHANGE MANAGEMENT SOLUTION

Central to everyone involved in the software development lifecycle is the change management process, for those organisations without a change management solution the benefits of implementing one are clear and compelling.

For organisations who already have a change management product, nearly every aspect of a business is likely to have involvement with the software change management process, furthermore processes will have been built up and refined over many years. Moving from their existing change management solution to a new product is a significant undertaking, not to be tackled lightly. The question is not if you should change but rather if it is right to change in light of the potential benefits vs. cost of migration.

Companies considering migrating from their change management solution must contemplate:

- Losing or breaking reports based on historic data or statistical analysis (often essential for companies with SLA contracts).
- The vast majority of commercial change management tools use databases in the backend, however recreating a schema and behaviours can be challenging.
- Exposing information which should remain private/confidential.
- Migrating scripted customised behaviours and existing product integrations.
- Existing integrations and customer portals.
- Educating all users in a short time frame while managing disruption to the business.
- Impact on remote users and customers.

Companies must ask themselves if switching their change management product for an alternative solution is a sensible decision, and not just automatically migrate to a new change management tool because their version control product is changing. If nothing else any company considering an SCCM solution migration should migrate in manageable stages i.e. version control first, change management after the initial move has been proved successful and stable.

The key question is “how do you choose and implement an SCCM solution which meets such a wide ranging collection of requirements?”. The answer is to create a set of processes which incorporate best practice and are easily portable regardless of the technology and software products around them.

3.1. Focusing Investment in the Right Area

Traditionally the heart of any SCCM solution was configuration management (version control), as a minimum any configuration management solution must:

- Version and manage files and directories while tracking and maintaining history.
- Provide audit capability.
- Manage baselines and releases.



- Allow for parallel development and manage merging from branches.
- User access (easily manage data security and user access).
- Must be dependable and scalable.
- Integrate into the development environment (IDE's).
- Give light application lifecycle management.
- Support for multiple platform and cross platforms.
- Automate manual or error-prone tasks.

The majority of stand alone configuration management solutions today provide this functionality as standard, it is generally accepted that investment in change management gives a better return than configuration management because a wider audience make use of the information available from change management systems (requirements, design, defects, enhancements, build information, release notes etc.). Change management systems quickly become the focal point for software development, controlling and managing the flow of information throughout the organisation.

Many organisations are looking for a simple approach to ALM, something which can manage small isolated activities of work and control their integration with other activities. The essential requirement for recording, retrieving and reporting on development tasks are change management tools such as Rational ClearQuest, Jira, Trac, Bugzilla etc.

The change management tool inevitably becomes the focal point for all information. Ensuring change sets are automatically recorded within the change management tool is invaluable:

- Should a record require rework, the original change set file list is available for inspection.
- Release notes (down to the level of specific files changes) can be easily extracted.
- The change management tool can be used to drive a fully automated continuous integration and merge process. Team leader or build masters select from the list of completed activities, those required in the next build. The change management tool reads the change set list and performs an automated merge on the recorded change set.

4. THE EMERGENCE OF SUBVERSION

4.1. A Brief History of Subversion

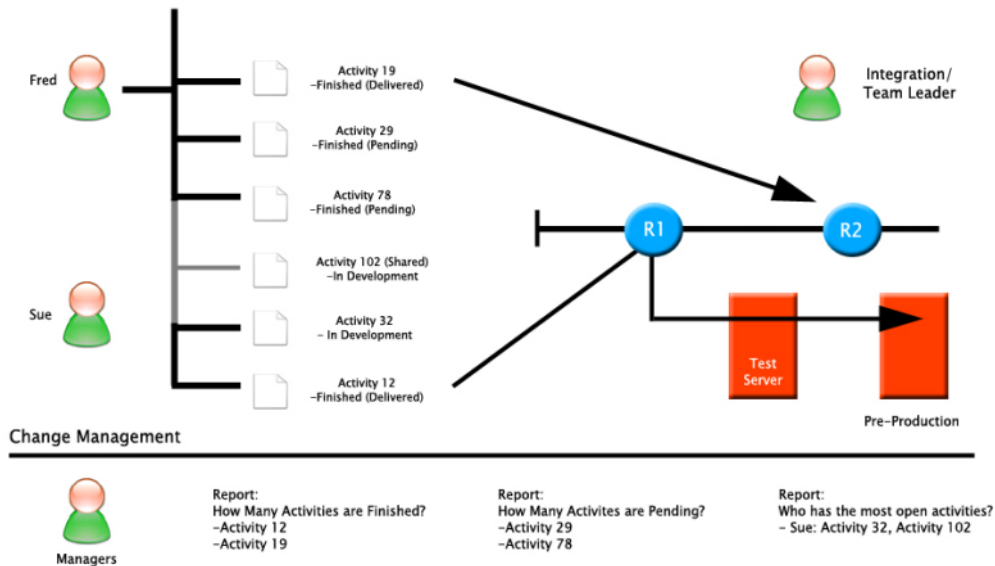
In early 2000, Karl Fogel and Jim Blandy initiated a new project to correct what were considered imperfections with CVS. Initially the community was small but grew quickly once the simple objectives of Subversion became public knowledge.

They were not attempting to write a heavy weight configuration management tool but something which resembled CVS with improved performance, also addressing the software development challenges of today.

4.2. Benefits of Subversion

- **Cost:** Under the Debian Free Software guidelines, Subversion is free to download, modify and redistribute.
- **Reliability:** It's no exaggeration to say one of the great advantages of open source is that everything works reliably and is simple to download and use, of course there are defects but they are quickly resolved with pre-release testing from the community being intensive. The most reliable statistics indicate that subversion is growing at a rate of more than 200,000 downloads per month, with over 2.5 million Subversion users.
- **Flexibility:** The architecture of Subversion has been carefully and cleverly designed with regard to performance, usability and scalability. Furthermore it is built with internet technology and distributed development in mind unlike many of the older tools available.
- **Ease of use:** Those with experience of CVS will understand SVN quickly. The third party graphical interfaces that integrate with Subversion (Tortoise, Eclipse, NetBeans etc.) effectively make Subversion disappear into the background. Subversion is also relatively simple to deploy and maintain.
- **Openness:** Information is visible and open to scrutiny. This helps partner companies and users plan new releases or evaluate how a recently submitted enhancement is progressing.

Subversion is suited for developers who want to work in a secure, agile way but include a minimal amount of control at the file version level. It is not just technically minded developers who need to register file changes for a project but also less technically aware business users who need to manage non-code related artefacts. Installing a heavy configuration management tool can be off putting for even technically minded people, Subversion however is intuitive to use and install.



The diagram above illustrates developers changing files based upon activities of work, each activity is isolated and self contained. The activities are integrated one at a time to an integration stream after which they are built and deployed to appropriate test environments. Managers are not interested in file versions, only the finished artefacts or more importantly what has not been finished and might impact a deadline so that they can accurately report back to the key decision makers and customers.

4.3. Disadvantages of Subversion

Subversion has many advantages but no solution meets all of the ideals, listed below are a few of the disadvantages of Subversion:

- Cost to migrate data to a new SCM tool. It is not always possible to capture the full history trail from the original tool, it should be noted that this is not unique to Subversion.
- It is open source, no single vendor exists to provide support and guarantee the product will continue to be enhanced and developed. Clearvision offer a full range of Subversion related services including support, training and additional tools.
- Subversion is still relatively young and therefore lacks some of the key functionalities of long term SCM solutions. Development of Subversion has a clear and open plan, inevitably the community will implement the missing pieces of the jigsaw.



5. COMBINING VENDOR/SUBVERSION SOLUTIONS

The impact of Subversion on traditional vendors has been significant, whether due to cost, architecture or functionality Subversion is and will continue to be a successful SCM product.

The development IDE's such as Eclipse, JBuilder, NetBeans, IBM Rational Application Developer (RAD), IBM Jazz and independent ALM solutions such as CodeBeamer from Intland are masking the need to interact with the underlying version control tool, in Q2 2008 Clearvision will release a light weight Rational Unified Change Management (UCM) style layer for Subversion. Prior to these third party interfaces vendors such as IBM, Serena, Perforce and Accurev provide user interfaces to improve usability of the underlying change and configuration management tool.

Companies who have historically used a combined solution of Rational ClearQuest and Rational ClearCase and are now introducing Subversion to harness the individual strengths of both products and create a hybrid solution (Rational ClearCase combined with Subversion) can continue to use Rational ClearQuest as their total change management solution (by using the ClearQuest Subversion integration from Clearvision). Anyone considering a change in tools must balance the impact of transitioning to a new product versus continued usage and extension of an existing tool.

Every tool has an "end of life", the question is whether each part of your existing solution remains cost effective and functionally suitable for today's environment (also considering the investment in time and effort used to tailor/implement, roll out, train staff and maintain the original solution).

6. INTEGRATED CHANGE MANAGEMENT: RATIONAL CLEARQUEST AND SUBVERSION

Rational ClearQuest is an industry-leading change management solution which provides flexible change tracking, process automation, reporting and lifecycle traceability for better visibility and control of the software and systems development lifecycle. Integration with software configuration management provides the ability to link development activities with file changes, recording the version of the file(s) changed within Rational ClearQuest and the activity information within the configuration management tool.

Once a change management tool has visibility over which files have been changed for any activity it is relatively simple to perform complex tasks such as delivering just changed files into a main development line. Having a development process which enables an integrator to pick and choose which items of functionality should be integrated with or backed out from a release gives incredible flexibility however, without the initial record of the change set it would be extremely difficult to achieve.

It is generally accepted that development should not take place unless a request for change has been raised. Change requests are generally considered to fall into one of three types Defects, Enhancements and New Functionality.

Even if development is in a preliminary state and considered to be exploratory, work should be registered against a Change Record, worst case is that the change record starts off as vague or high level, at least it is being tracked and documented.

Developers generally understand the benefits of a change management system if the process is simple and does not interfere with or slow their progress.

6.1. Integrating Rational ClearQuest and Subversion

Traditionally Rational ClearQuest has only integrated fully with Rational ClearCase, with the introduction of [CQ2SVN](#) (a Clearvision product) Rational ClearQuest now fully integrates with Subversion allowing Subversion users to benefit from a fully functional and integrated change management solution.

Rational ClearQuest, whilst mature, is recognised as an industry leader in the change management market. Subversion whilst relatively new, suits many projects that are seeking a development environment with reduced overheads and is thus especially suited to the agile way of working.

6.2. How does CQ2SVN work?

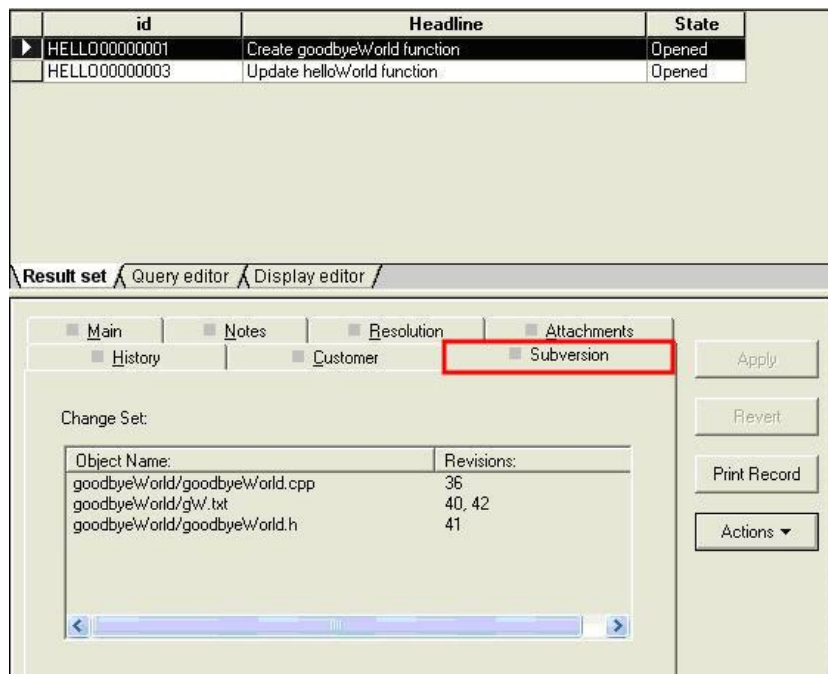
For those organisations presently using Rational ClearCase and Subversion (whether integrated or as separate SCCM products), the CQ2SVN interface works seamlessly

along side the existing Rational ClearCase/Rational ClearQuest integration (whether CC Base and/or UCM).

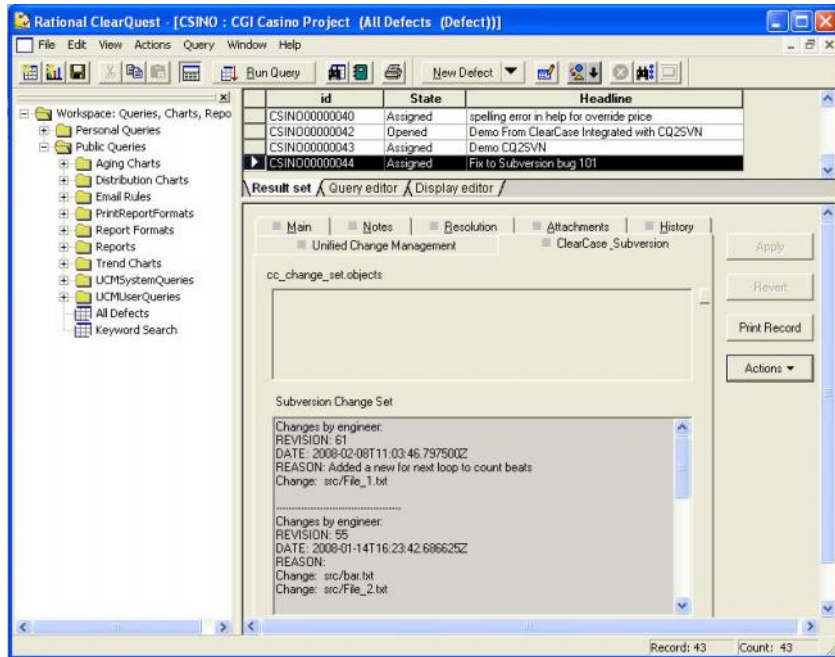
The Subversion to Rational ClearQuest integration is simple in its approach:

- Subversion user commits a set of files.
- CQ2SVN traps the commit and queries the Rational ClearQuest repository for a filtered list of records (those assigned to the developer).
- The list of records is presented to the user on the client machine.
- The user selects one or more records they wish to associate with the activity.
- CQ2SVN then associates the check-out to an activity by writing the change set of files against the Rational ClearQuest record and simultaneously writing the Rational ClearQuest record ID against the Subversion revision.

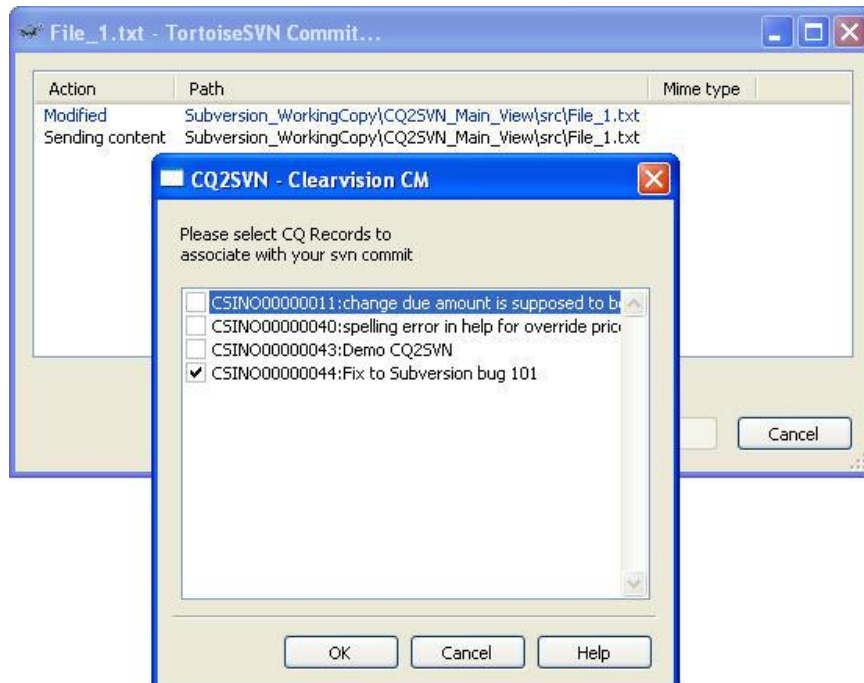
6.3. CQ2SVN Screen Shots



CQ2SVN: Additional Subversion Tab in Rational ClearQuest Records



Automatically Showing the Developers Subversion Check-In Comments Within Rational ClearQuest



Window Allowing the Subversion Committer to Select the Appropriate Activity



7. ABOUT CLEARVISION

[Clearvision](#) was established in 1997 and is the leading independent change and configuration management consultancy. The first UK IBM Rational partner, we are partnered with the world leaders in configuration and change management including IBM Rational, WANdisco, Intland, Polarion, CollabNet and Atlassian. These relationships have been built over many years and allow us to sell the latest products while also giving us access to resources and technical assistance.

Originally founded in the UK, Clearvision now has offices throughout Europe and the USA and regularly works with clients in offices throughout the world. The Clearvision customer base includes global top 100 companies such as CitiGroup, HSBC, Barclays, Ericsson, Sony and many more.

In recent years Clearvision have helped customers combine the power and reliability of IBM Rational change and configuration management solutions with the flexibility and low-cost of open source solutions.

Clearvision are market leaders in:

- CM Training (both [Web Based](#) and conventional classroom training).
- CM consulting, strategy, process definition and product recommendation.
- Product integrations e.g. CQ2SVN and Jira2SVN.

CQ2SVN is an IBM Ready for Rational certified product, see the IBM Rational website for more details:

<http://www-304.ibm.com/jct09002c/gsdod/solutiondetails.do?solution=33614&expand=true&lc=en>

7.1. Document Version History

Date	Version	Author/Approved	Comments
26 th Feb 2008	0.9	Gerald Tombs	Waiting approval
11 th March 2008	1.0	Neil Butler	Approved