



IBM Rational Software Conference 2009
As Real as It Gets!



Heterogeneous Development using Rational® Team Concert™ Eclipse and Rational® Team Concert™ For Microsoft® Visual Studio™ IDE

Sreerupa Sen, sreerupa.sen@in.ibm.com

Architect/Developer, RTC for Visual Studio

Rational. software

Why do we care about heterogeneity?

- The Jazz initiative is composed of three elements:
 - ▶ An architecture for lifecycle integration
 - ▶ A portfolio of products designed to put the team first
 - ▶ A community of stakeholders

Contents

- What is the Rational® Team Concert™ For Microsoft® Visual Studio™ IDE?
- A quick tour
- Self-hosting: **Real teams, Real insights**
- Homogeneity in heterogeneity
 - ▶ Any resemblance to the RTC Eclipse Client is purely intentional ☺

Contents

- What is the Rational® Team Concert™ For Microsoft® Visual Studio™ IDE?
- A quick tour
- Self-hosting: **Real teams, Real insights**
- Homogeneity in heterogeneity
 - ▶ Any resemblance to the RTC Eclipse Client is purely intentional ☺

What is RTC for Microsoft Visual Studio™ IDE?

Team Collaboration for .NET Developers

- A Team Collaboration environment for .NET developers.
- Provide source control, work item tracking, build, process & collaboration services
- leverage and extend the Jazz Platform



A Visual Studio Client Integration

A First Class Integration with Microsoft Visual Studio
to enhance developer productivity



Innovative Software Engineering

Heterogeneity

- A Common Team Collaboration Server infrastructure based on Jazz.
- A Visual Studio Client for .NET developers
- An Eclipse Client for Java developers
- A Web Client for contributors
- Powered by Jazz



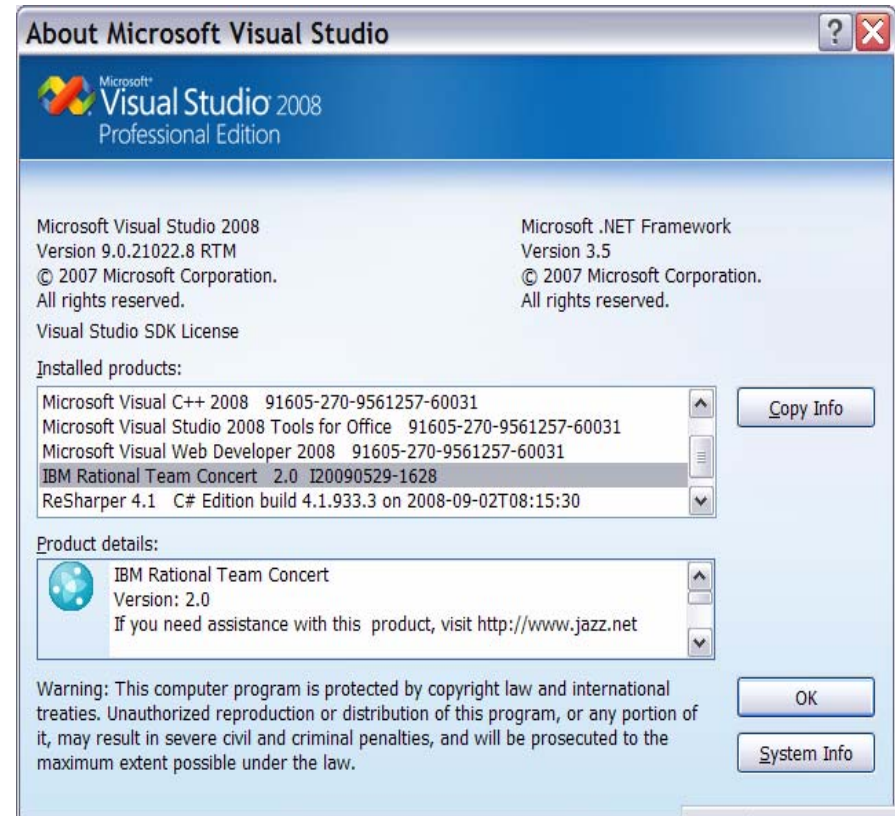
Vision

Brings the Jazz vision to .NET developers :
The value and experience that future Rational products can bring to software and systems delivery teams



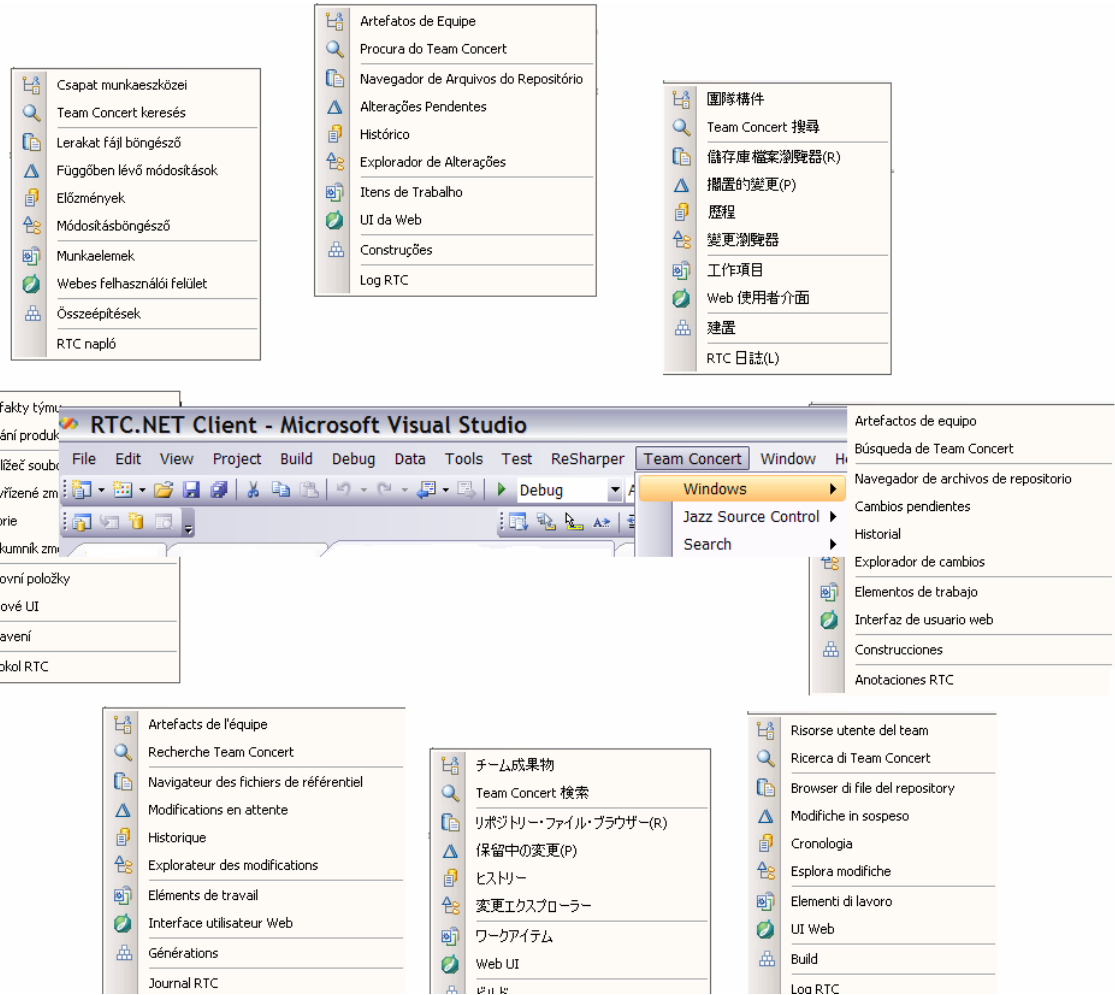
RTC for Microsoft® Visual Studio™ IDE

- **Natively implemented in Visual Studio**
 - ▶ **A first class integration into Visual Studio**
- An extension of the Visual Studio IDE
 - ▶ Integrates into Solution Explorer, Options pages, Toolbars, Menus, Tool Windows
 - ▶ Integrates into Visual Studio as a source control provider
 - ▶ Integrates into the Error List View
 - ▶ Integrated Help
- Supports Visual Studio 2005 and 2008
- Developed using WPF, the latest Microsoft UI technology
- **The first Jazz client written in C#**
- Follows Visual Studio international settings



Internationalization

- Available in all group 1 languages.



Installs and Downloads

- Installable via IBM Installation Manager
- Downloadable from jazz.net
 - ➔ <https://jazz.net/downloads/rational-team-concert/milestones/2.0.0.1M1>
 - ➔ 2.0, 1.0.1.1, integration builds, betas...

Focus

- Caters to teams developing in .NET
- Currently provides
 - ▶ SCM integration with VS.NET artifacts
 - ▶ Work Items integration
 - ▶ Build Integration
- **Emphasis on Developer use cases**
- **Supports heterogeneous teams in conjunction with RTC Eclipse**

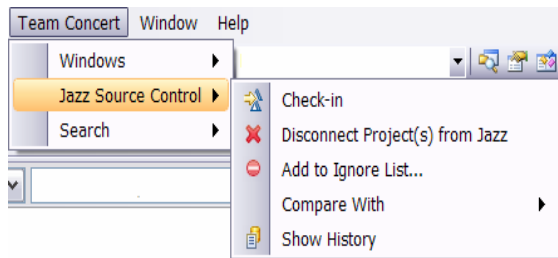
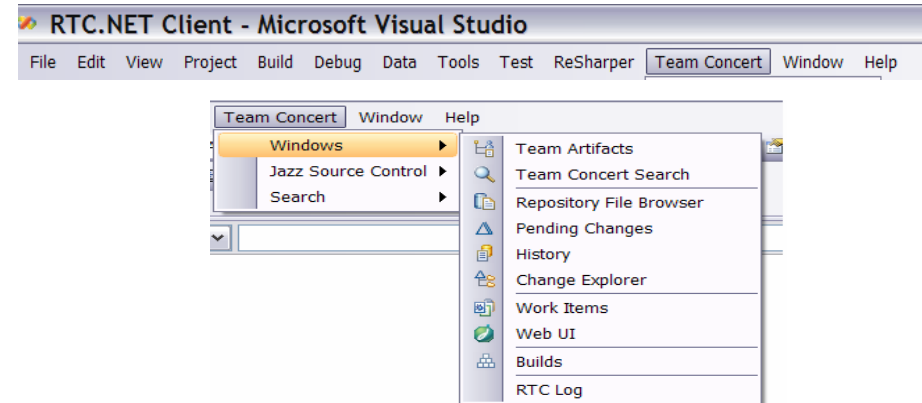
Contents

- What is the Rational® Team Concert™ For Microsoft® Visual Studio™ IDE?
- A quick tour
- Self-hosting: **Real teams, Real insights**
- Homogeneity in heterogeneity
 - ▶ Any resemblance to the RTC Eclipse Client is purely intentional ☺

RTC Visual Studio™ at a glance

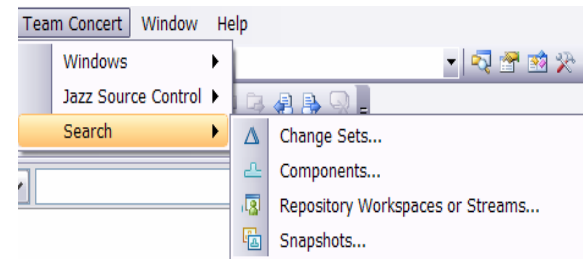
Visual Studio IDE extended with Jazz/RTC capabilities for

- ▶ Artifacts management
 - Repository connections, Project areas, streams, workspaces, components
- ▶ Work items management
- ▶ Build management



- ▶ Repository search
 - Change sets, components, streams and workspaces, snapshots

- ▶ Source control management



RTC Visual Studio™ at a glance

The screenshot displays the Microsoft Visual Studio IDE for an experimental hive. The main window is titled 'ClassLibrary1 - Microsoft Visual Studio - Experimental Hive'. The interface is divided into several panes:

- Code Editor:** Shows the source code for 'Class1.cs' with the following content:

```

namespace ClassLibrary1
{
    public class Foo
    {
        public int count;
    }
}

```
- Solution Explorer:** Displays the project structure for 'ClassLibrary1' (1 project), including folders for Properties, AssemblyInfo.cs, References, and Class1.cs. A context menu is open over Class1.cs with options: Open, Check-in, Disconnect Project(s) from Jazz, Add to Ignore List..., and Compare With.
- Team Artifacts:** Shows a tree view of artifacts including Repository Connections, Builds, Source Control, and Work Items.
- Team Concert Search - Snapshots:** A table listing snapshots with columns for Date Created and Name.

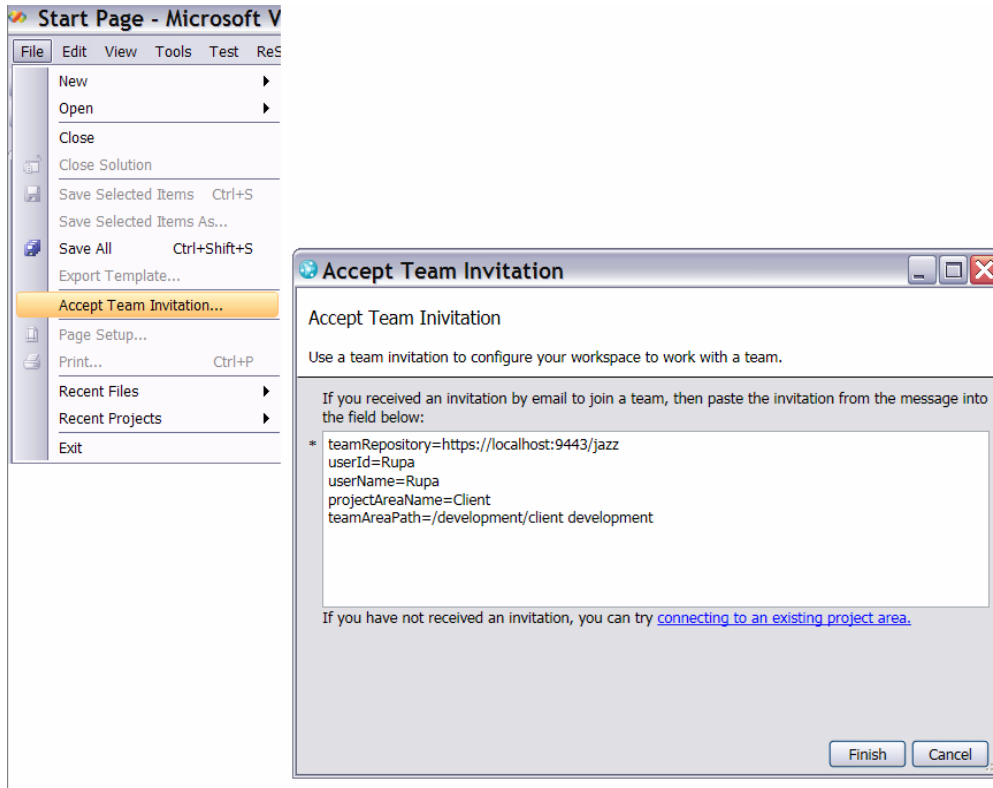
Date Created	Name
5/14/2009 8:39 PM	BuildDefForClientUITeam_20090514-2039
5/13/2009 2:27 PM	TestAreaFor20RC1 Team build_20090513-1427
5/13/2009 2:46 PM	TestAreaFor20RC1 Team build_20090513-1445
5/13/2009 2:47 PM	TestAreaFor20RC1 Team build_20090513-1447
5/13/2009 2:48 PM	TestAreaFor20RC1 Team build_20090513-1448
5/13/2009 3:09 PM	TestAreaFor20RC1 Team build_20090513-1509
5/13/2009 3:10 PM	TestAreaFor20RC1 Team build_20090513-1510
- Change Explorer:** Shows a comparison of 'TestAreaFor20RC1 Team build_20090513-1427' with other builds, listing changes like 'Only In TestAreaFor20RC1 Team build_20090514-2041' and 'Kishore - 22: New ClassLibProject - Share projects'.
- Builds:** A table showing build status and progress for various builds.

Build Status	Build State	Label	Progress
✓	Build	20090513-1455	Completed
✓	Build	20090513-1509	Completed
✓	Build	20090513-1509	Completed
✗	Build	20090513-1445	Completed
✗	Build	20090513-1427	Completed
✓	Build	20090513-1448	Completed
- Pending Changes:** Shows 1 unresolved local change and 1 outgoing change set.
- Test Workspace 3:** A tree view showing 'Unresolved' and 'Outgoing' changes, with a specific change '10: Share code with Jazz Source Control - Share projects'.
- Work Items:** Shows 'Found 7 work items - Open assigned to me'.
- History:** Shows a list of SCM entries for 'Build Workspace' with comments like '<No Comment>', '27: ctc changes', and 'Move folder 'Folder1' from component 'TestAreaFor20RC1'.
- Repository Files:** Shows the file structure for 'Build Workspace'.



Getting initiated is a breeze...

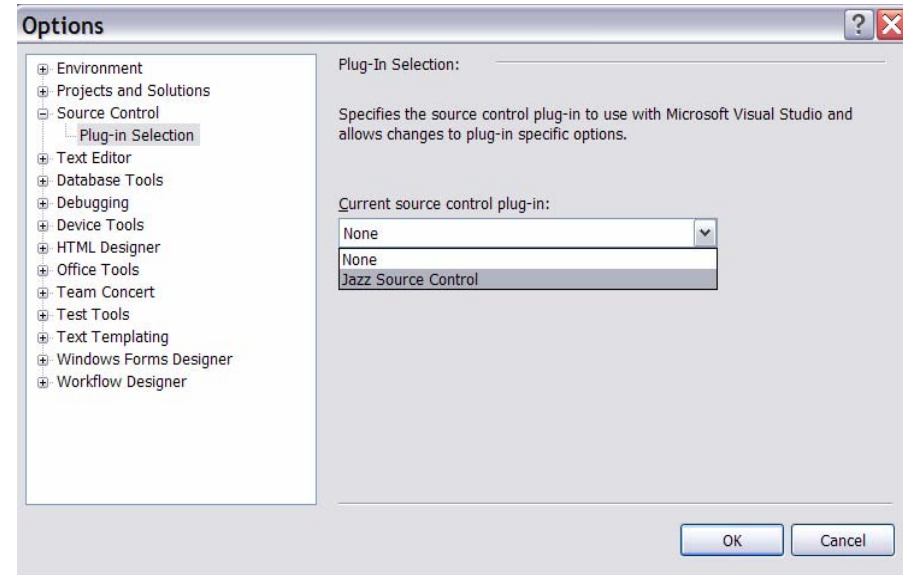
When your team lead adds you to a team, you get an email.



All you need to do is to paste the contents of the mail into the Accept Team Invitation dialog box.

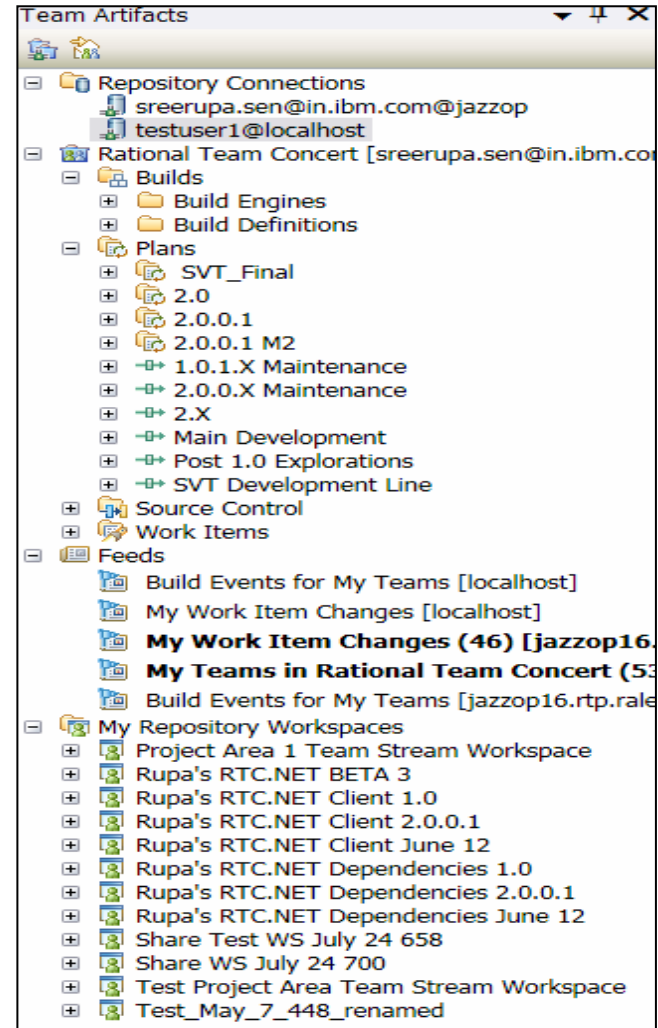
Activating Jazz Source Control

- Once you choose Jazz as your source control provider
 - ▶ A Source Control node will appear under each project area that you've connected to
 - ▶ Expanding Source Control will show you your team's streams and components
 - ▶ My Repository Workspaces will be populated with any existing repository workspaces that you may have.



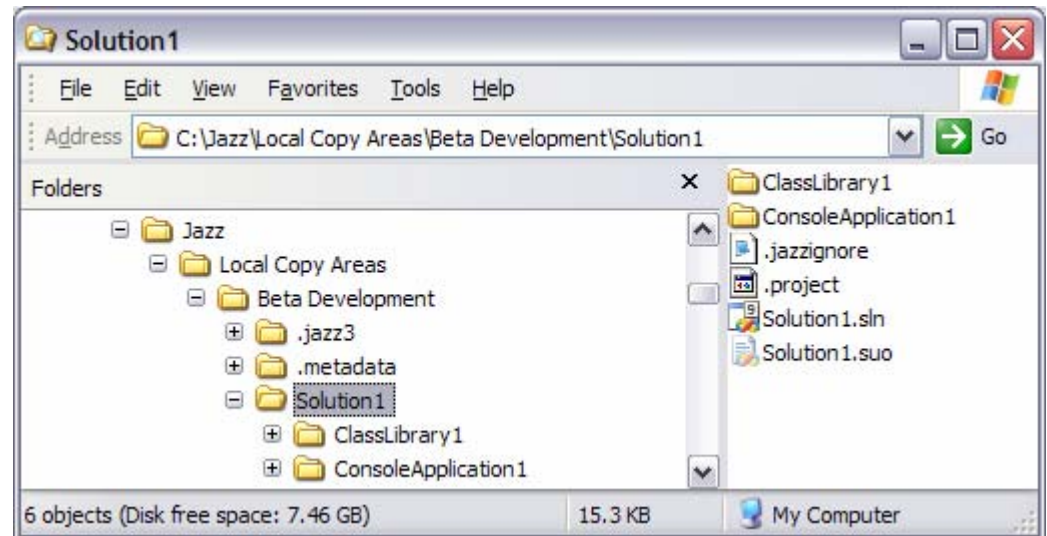
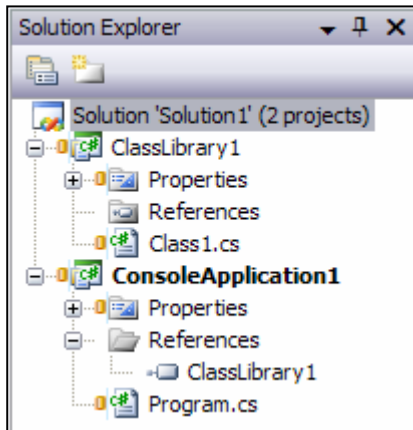
Exploring your Repository : Team Artifacts

- The Team Artifacts Tool Window is your window into the jazz repository
- Use the Team Artifacts Window to
 - ▶ Connect to Project Areas
 - ▶ Create and manage
 - Streams
 - Components
 - Workspaces
 - Snapshots
 - Work Items
 - Queries
 - Build Requests
 - Plans
 - Feeds



Sharing a solution

- To start checking in your sources to the repository, you need to share your solution
- A solution is shared into a Jazz component
 - ▶ The Microsoft recommended way is to have the physical file layout follow the solution structure
 - The solution file sits in a root folder
 - The projects are in sub-folders



Organizing your source code

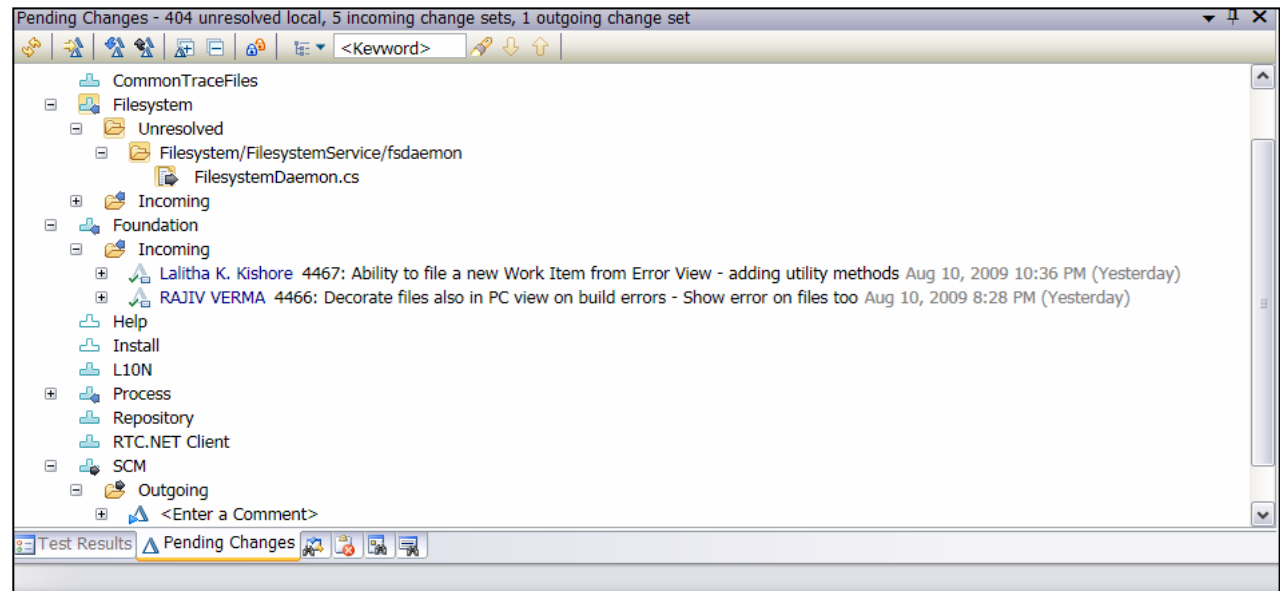
- We support and recommend the best practices outlined by Microsoft team development literature
 - ▶ <http://msdn.microsoft.com/en-us/library/ms998208.aspx>
- Use a single jazz component/solution in simple cases
 - ▶ Does not scale well
 - ▶ Project dependencies result in frequent rebuilds of multiple projects event for minor changes
- Use a partitioned solution approach for scalable component based development

Multi-component development with Partitioned Solutions

- Each component corresponds to a solution.
 - ▶ A project may be included in multiple solutions
 - ▶ A project is created in it's "home" solution
 - ▶ And referenced by dependent components/projects in other solutions, via project references.
- Separate solution files let you work in smaller sub-systems fairly independently.
 - ▶ Each sub-system corresponds to a jazz component.
 - Scalable
 - ▶ Rebuilds less frequently
- Having project references ensures that you always have the latest sources for components you depend on.
- You can have a master solution for system-wide builds.
- **We followed the partitioned solution approach for the RTC Client for Microsoft Visual Studio.**

Keeping in sync: Pending Changes

- Your window to your team's source code changes
- Lets you check in, deliver, suspend, resume, discard, reverse change sets
 - ▶ Resolve merge conflicts
 - ▶ Create snapshots and baselines
 - ▶ Lock/unlock source files
 - ▶ Filter change sets



Managing your Tasks and Defects: Work Items

- Create/run queries to find out what's assigned to you.
- Create new work items and queries.
- Update/Bulk-edit existing work items
- Have active discussions with the rest of your team.

The screenshot displays the IBM Rational software interface. At the top right, the 'Team Artifacts' tree shows a project structure including 'Project Area 1', 'RTC.NET', 'Builds', 'Source Control', 'Work Items', 'My Queries', 'Shared Queries', and 'Predefined'. Below this, the 'Work Items' table lists 75 work items, with columns for Type, Id, Summary, Owned By, Status, Priority, Severity, and Modified Date. A 'Create Work Item' dialog box is open in the foreground, showing the following details:

Create Work Item

Fill in the work item details:

Project Area: Rational Team Concert

Type: Defect

Summary: [Empty]

Description: [Empty]

Found In: 1011iFix1

Severity: Normal

Filed Against: Unassigned

Tags: [Empty]

Owned By: Unassigned

Planned For: Unassigned

Buttons: OK, Cancel

Building your projects/solutions

- Visual Studio developer can access Jazz Builds from within the IDE
- Developers can
 - ▶ Request Builds
 - ▶ Request Personal Builds
 - ▶ Monitor Build Activity
 - ▶ See Details of Individual Build and download associated artifacts
 - ▶ Create a repository workspace or stream from a given build

The screenshot displays the Rational Team Concert web interface. At the top, a 'Builds - 'integration.rtcnet' - Found '161' Builds' window shows a table of build records:

Build Status	Build State	Label	Progress	Start Time	Duration	Build	Completion Time	Build Engine
83% (Run...		I2009060...		5:29 AM (1...		integratio...	5:29 AM (1...	RTC.NET_...
✓		I20090602-1...	Completed	6/2/2009 8:...	3 hours, 42...	integration.r...	6/2/2009 11...	RTC.NET_Cli...
✗		I20090529-1...	Completed	5/29/2009 6...	2 hours, 59...	integration.r...	5/29/2009 9...	RTC.NET_Cli...
✗		I20090602-0...	Completed	6/1/2009 7:...	23 minutes,...	integration.r...	6/1/2009 7:...	RTC.NET_Cli...
✗		I20090531-0...	Completed	5/30/2009 7...	1 hour, 39...	integration.r...	5/30/2009 8...	RTC.NET_Cli...
✗		I20090529-1...	Completed	5/29/2009 2...	1 hour, 40...	integration.r...	5/29/2009 4...	RTC.NET_Cli...

The main interface shows the 'Build integration.rtcnet I20090602-1834' details page. The status is 'Successful' and 'Completed'. Key information includes:

- Status: Successful
- State: Completed
- Start time: Jun 2, 2009 8:59 a.m.
- Completed: Jun 2, 2009 11:59 a.m.
- Duration: 3 hours, 42 seconds

General Information:

- Requested by: Lalitha K Kishore
- Build Definition: integration.rtcnet
- Build Engine: RTC.NET_Client_Build_Engine
- Build History: 161 builds
- Tags: None

Staying Current with Feeds

- Create feeds for
 - work items, queries
 - builds, build definitions
 - team events
- A consolidated view of events in your team area/project area
- Feeds refresh automatically
- You can customize feeds by filtering

The screenshot displays the RTC.NET Client interface within Microsoft Visual Studio. The main window is titled "My Work Item Changes" and shows a list of work item updates. The list is divided into "Yesterday" (6 items found) and "Last Week" (40 items found). The "Yesterday" section includes items such as "Ability to file a new Work Item from Error View (4467)", "Sharing a project with a different user does not actually show the new user. (2814)", "Solution explorer context menus should prompt for login if not already logged in (1939)", "Adopt RepositoryConnectionUtils.ValidateRepositoryConnection for Show History menu in Solution E ... (4465)", and "Locking resources from solution explorer while not having the corresponding repository connectio ... (4435)". The "Last Week" section includes items like "New 'Approval' request: 'Review' (3772)", "Get the mandatory display fields for work item creation dialog, from the repository (4402)", "TVT test cases for RTC VS Client 2.0.0.1 (4453)", "The cumulative state of the 'Approval' request 'TVT Test case written' is now 'Approved' (4453)", and "Fix foundation preference tests and preference framework usage (4464)".

On the right side, there is a "Team Artifacts" pane showing a tree view of repository connections, builds, plans, and source control streams. The "Feeds" section is expanded, showing various feed options like "Build Events for My Teams", "My Work Item Changes", and "My Teams in Rational Team Concert".

At the bottom, there is a "Pending Changes" pane showing 404 unresolved local changes, 5 incoming change sets, and 1 outgoing change set. The "FileSystem" pane shows the project structure, including "FilesystemService/fsdaemon" and "FilesystemDaemon.cs".

Agile Planning

- Visual Studio users can now
 - Access project plans from within the IDE
 - Create/update plans

The screenshot shows the IBM Rational Team Concert interface within Microsoft Visual Studio. The main window displays a list of tasks with columns for priority, status, and estimated time. The right-hand pane shows the Team Artifacts view with a hierarchical tree structure of project artifacts.

Task Description	Priority	Status	Estimated Time
Story item to track Feed support related enhancements	High	Open	4246
Unable to Items under My Teams from Feeds	Medium	Open	4147
[RFT] Share menu is enabled on previously shared RFT projects after reopening the solution	Medium	Open	3282
[Intel] Sharing the solution does not share nested projects	Medium	Open	3276
Unloading / Reloading projects should be covered by progress monitor	Medium	Open	2752
Add to ignore can make project inconsistent	Low	Open	3395
Add a new folder into a project, project file modification doesn't show up in PC	Low	Open	3280
Possible bug in ResilientPath.cs IsPathContained ?	Unassigned	Open	2034
Need test coverage for progress/sync view notification	High	Open	1467
Dilemma handling not proper for project file reload	Medium	Open	2500
Missing icon of the change set in the Check-in to change set context menu	Medium	Open	1765
Pressing Refresh on PC after logout - does not attempt to login the user	Medium	Open	1525
Add preference support for to show workspaces previously tracked only (not loaded) in next session	Low	Open	4381
Implement backup support introduced in 2.0	Low	Open	3796
FilesystemRestClient API never throw AuthenticationRequiredException	Low	Open	3869
Logged-Off Repo connection shows connected project areas	Low	Open	3383
Web UI window is blank when I select Team Concert > Windows > Web UI	Unassigned	Open	4318

Contents

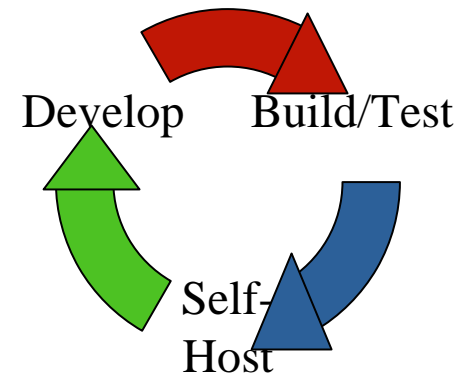
- What is the Rational® Team Concert™ For Microsoft® Visual Studio™ IDE?
- A quick tour
- Self-hosting: **Real teams, Real insights**
- Homogeneity in heterogeneity
 - ▶ Any resemblance to the RTC Eclipse Client is purely intentional 😊

Self-hosting in RTC for Microsoft® Visual Studio™ IDE

- We started with
 - ▶ A plain vanilla Visual Studio IDE for development and
 - ▶ The RTC Eclipse Client for our Jazz Needs
- As we integrated SCM, Work Items, Builds into Visual Studio we went to the Eclipse Client less and less
- We've been self-hosting in the RTC client in Visual Studio since October 2008
- **Real teams, Real Insights, Real Results!**

Self-hosting (contd.)

- At the completion of each iteration, we self-host.
- Our continuous and integration builds are Jazz builds
- We've integrated our builds with Microsoft's unit testing framework
 - ▶ Each build runs through all the unit tests
- We follow the develop-build-test-selfhost-develop cycle



Real teams, Real Insights

- As we developed using the RTC Client for Visual Studio, we added features
 - ▶ Based on our self-hosting experiences
 - ▶ Improving the user experience for the Visual Studio developer

When we faced the music...

- Visual studio specific processing for SCM operations
 - ▶ Warnings for incomplete change set check-ins, to avoid build breaks
 - For example if you're checking in a new file, but forgot the associated project
 - ▶ Automatic reloading of multiple project files while accepting changes from other users
 - ▶ Filtering in Pending Changes: searching for specific files in your change sets or the change sets of your team
 - ▶ Project can be part of multiple solutions, but physically associated with one.
 - We let you share it from any location and also detect and warn you about it.

And...

- Background execution of long repository operations
 - ▶ enabling you to continue your work in Visual Studio in the meantime
- Support for code refactoring
 - ▶ Renaming a file preserves its history in the repository
 - ▶ Moving file(s) within a project preserves history as well
 - ▶ Moving top level folders across components preserves history
- Support for compare/merge
 - ▶ Automatic merge in most cases
 - ▶ UI-aided conflict resolutions
- Integrating Pending Changes with the Error List



Contents

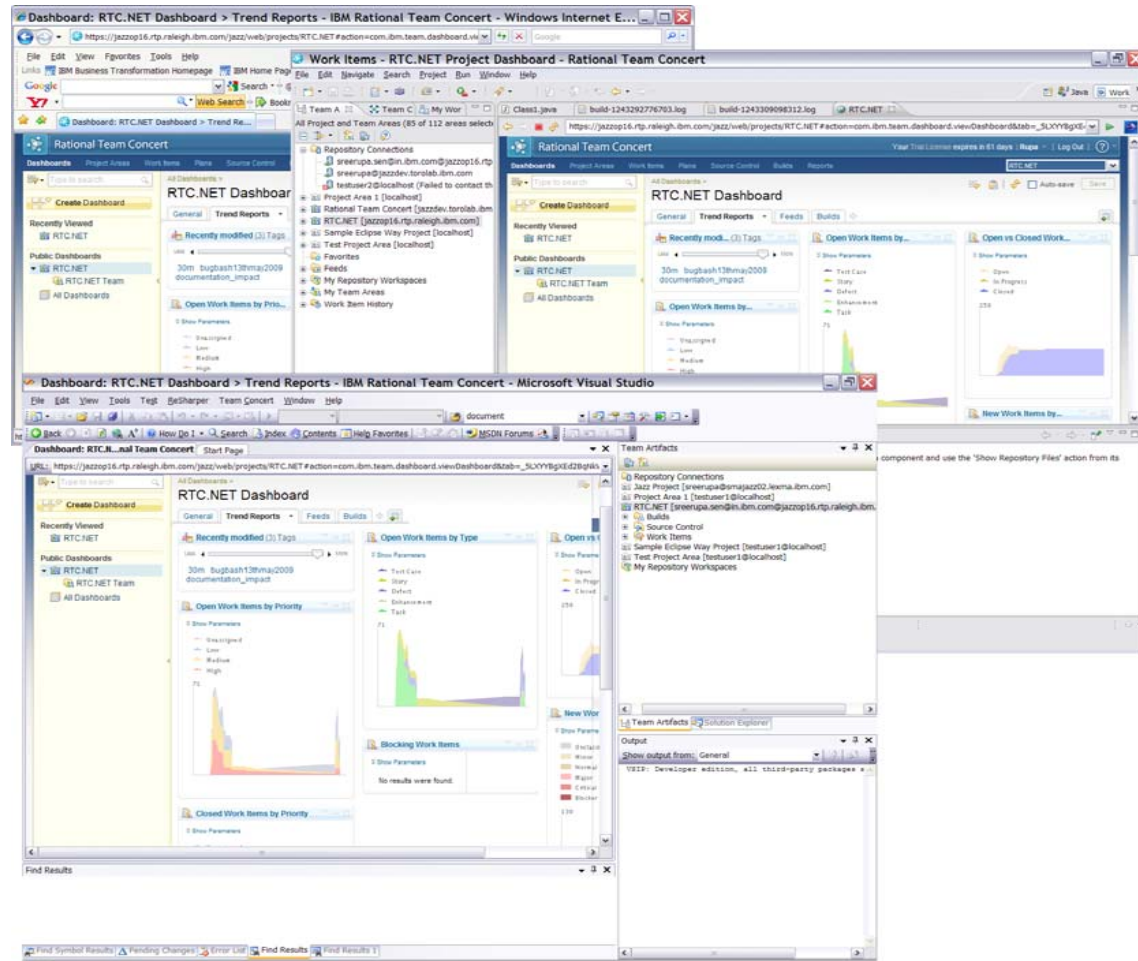
- What is the Rational® Team Concert™ For Microsoft® Visual Studio™ IDE?
- A quick tour
- Self-hosting: **Real teams, Real insights**
- Homogeneity in heterogeneity
 - ▶ Any resemblance to the RTC Eclipse Client is purely intentional ☺

Teams not restricted to

- Single Physical location
- Single Operating System
- Single Platform
- Single IDE
- Single Programming language

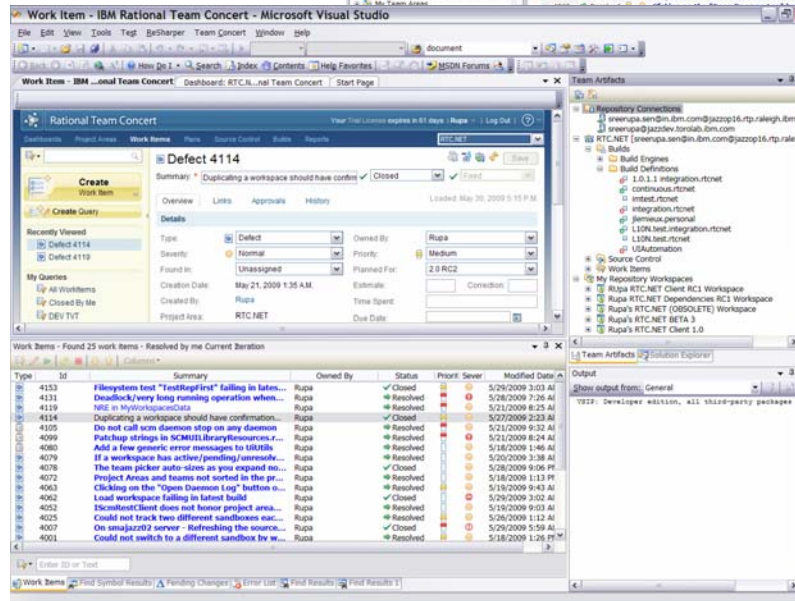
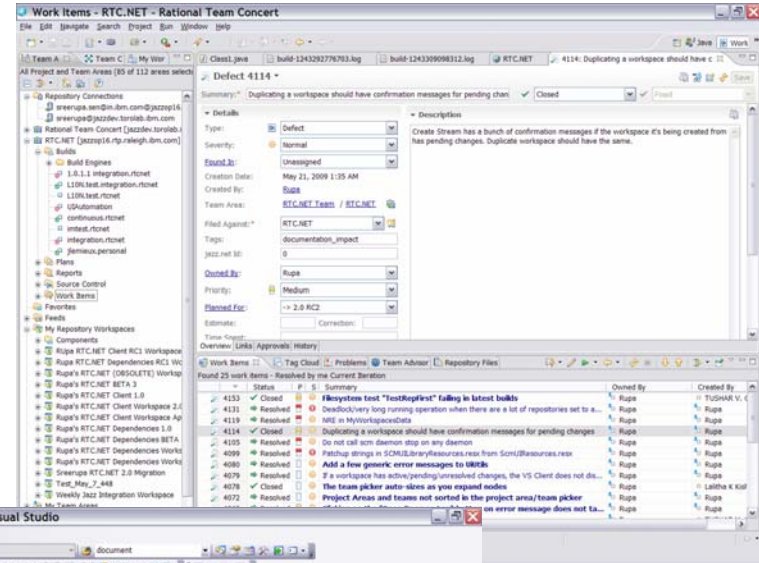
Homogeneity in Heterogeneous Teams

- One repository for all your artifacts
- Which means you share
 - ▶ Same Project and Team
 - ▶ Same Dashboard



Homogeneity in Heterogeneous Teams

- ▶ Same Process
- ▶ Same Builds Infrastructure
- ▶ Same Work Items and related information
 - Discussions
 - Builds
 - Approvals
 - Attached change sets



Our commitment

- People will work in the area of technology best suited to the requirements...

... rather than being restricted by their infrastructure.

- RTC clients will preserve the flavor of your favorite IDE
 - ▶ They are non-intrusive.
 - ▶ They're natively implemented for Eclipse and for Microsoft Visual Studio.
 - ▶ They are first class integrations into the IDEs that they extend.
- Barring environment specific needs, the over all look and feel will be homogeneous across RTC clients in different IDEs
 - ▶ Should you switch from one IDE to the other, you will not get lost

Take Our Survey!

- First Impressions of RTC for Visual Studio
 - ▶ <https://www.ibm.com/survey/oid/wsb.dll/s/ag246>

The demo scenario

- Echo Server
 - ▶ Server is in Java
 - ▶ Client is in C#

- Like any project
 - ▶ Defects move across components
 - ▶ There are lots of discussions between team members
 - ▶ The same process holds: same approval rules, same delivery rules, same permissions...

DEMO

Questions



Thank You

© 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.



BACKUP

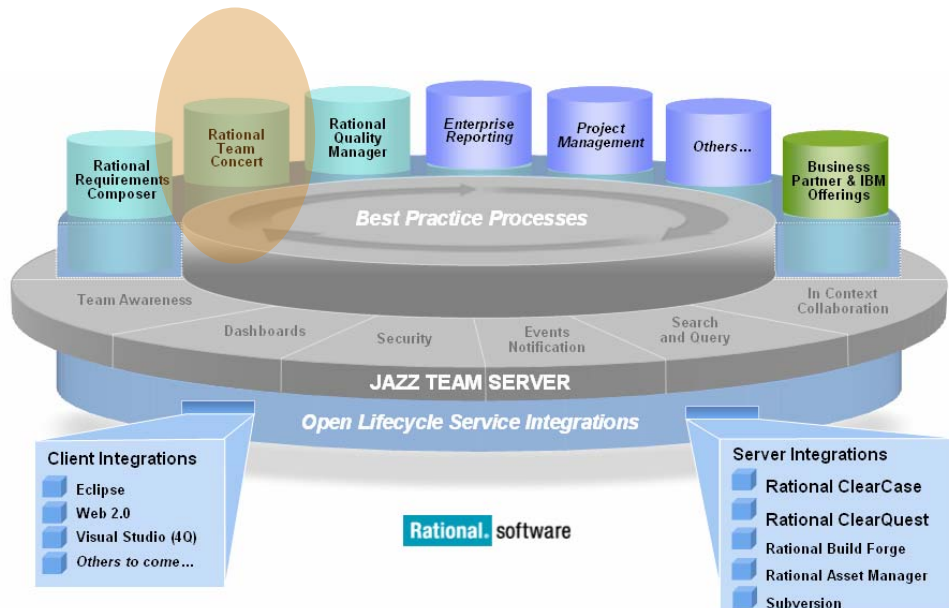
Jazz and Rational Team Concert

Jazz is a technology platform



Jazz is the foundation of the collaborative application lifecycle management (C/ALM) system

Jazz is a scalable, extensible team-collaboration platform that integrates tasks across the software lifecycle. The platform also provides useful building blocks and frameworks that facilitate the development of new products and tools. This unprecedented approach is unique within the industry and has significant advantages.



Rational Team Concert the first Jazz based product

- A collaborative software development environment
- Rational Team Concert enables developers to collaborate together using

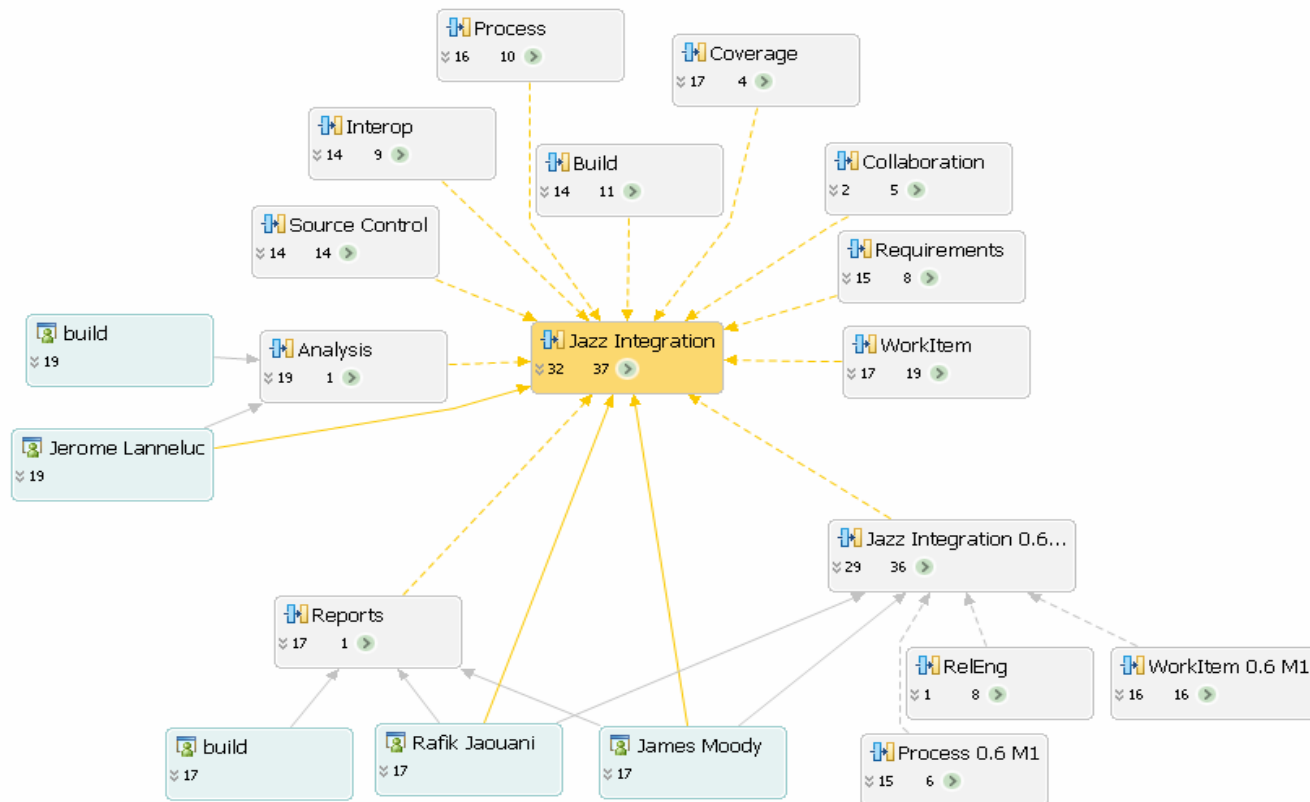
- ✓ **Integrated Planning**
- ✓ **Source Control**
- ✓ **Work Items**
- ✓ **Build**
- ✓ **Dashboards**
- ✓ **Reports**
- ✓ **Process Support**



- Native Clients in Eclipse and Visual Studio .NET.
- Web UI support based on Web2.0

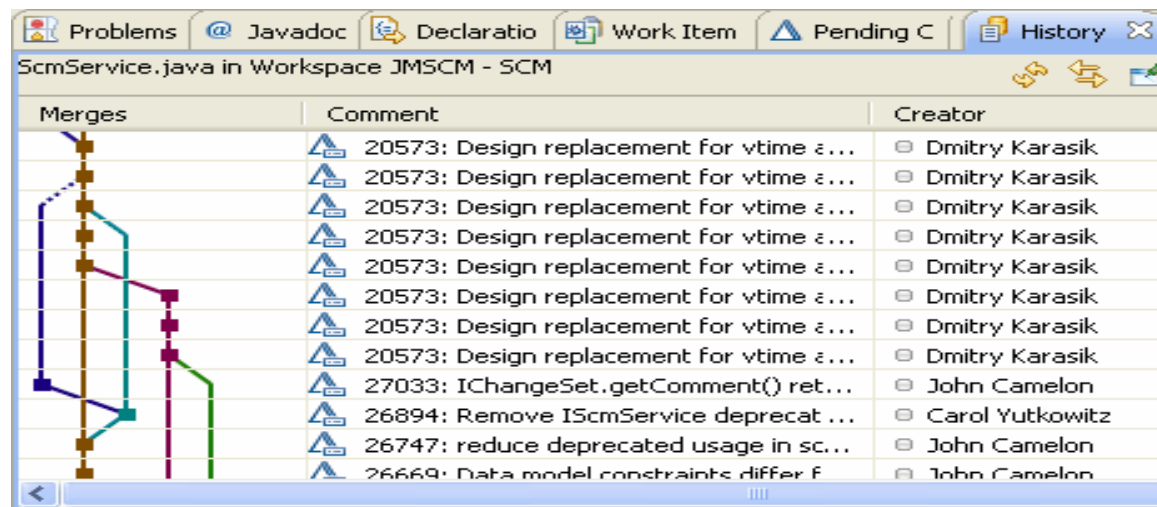
Teams of teams

- This simple model allows scaling to teams of teams.
- Provides isolation when needed at both the individual and team level.



Summary

- The Jazz SCM is entirely change set based. There is no file level branching or tagging.
- Focuses on user level simplification with the added power of parallel development.
- Conscious effort to never use the word branch in the UI. People are afraid of branches, but want isolation and the features they provide



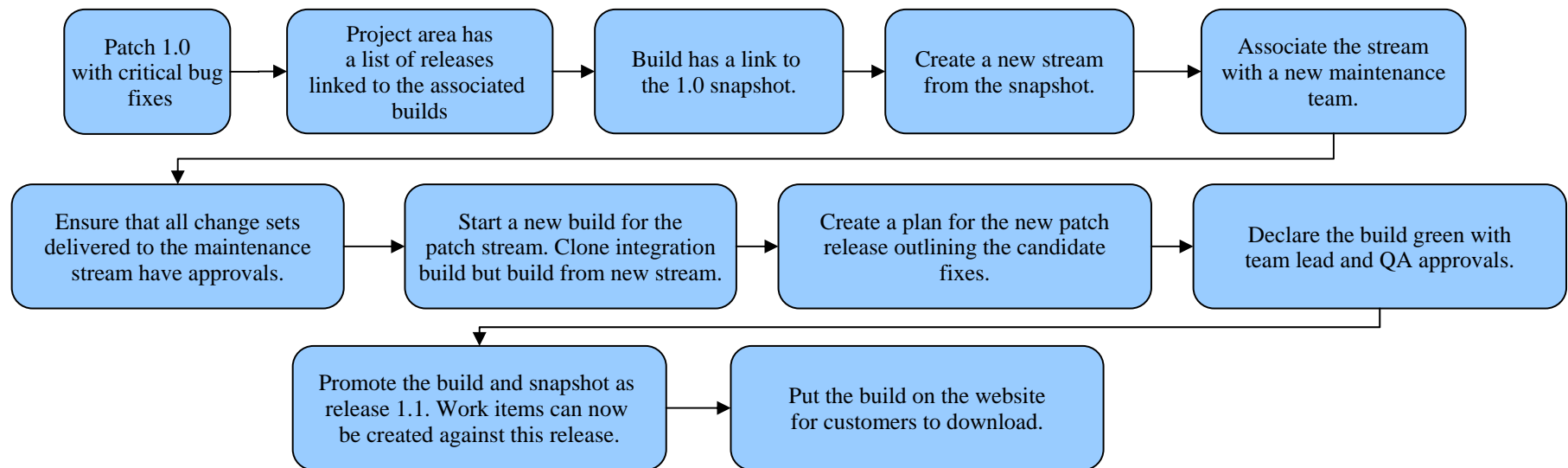
Parallel Development

- The model supports several levels of isolation
 - ▶ **Repository workspaces** – Provides constant isolation. You don't have to make your changes visible to the team just to backup or use the repository features.
 - ▶ **Suspend and Resume** – Provides task level isolation for personal work.
 - ▶ **Work Item links** – Provides light weight task level isolation for personal or team work. Work on a feature, attach to a work item and discard from your workspace. You or someone else continues the work by accepting the change sets back into their repository workspace.
 - ▶ **Streams** – Provides team isolation.
 - ▶ **TimeLines** – Provides process isolation.

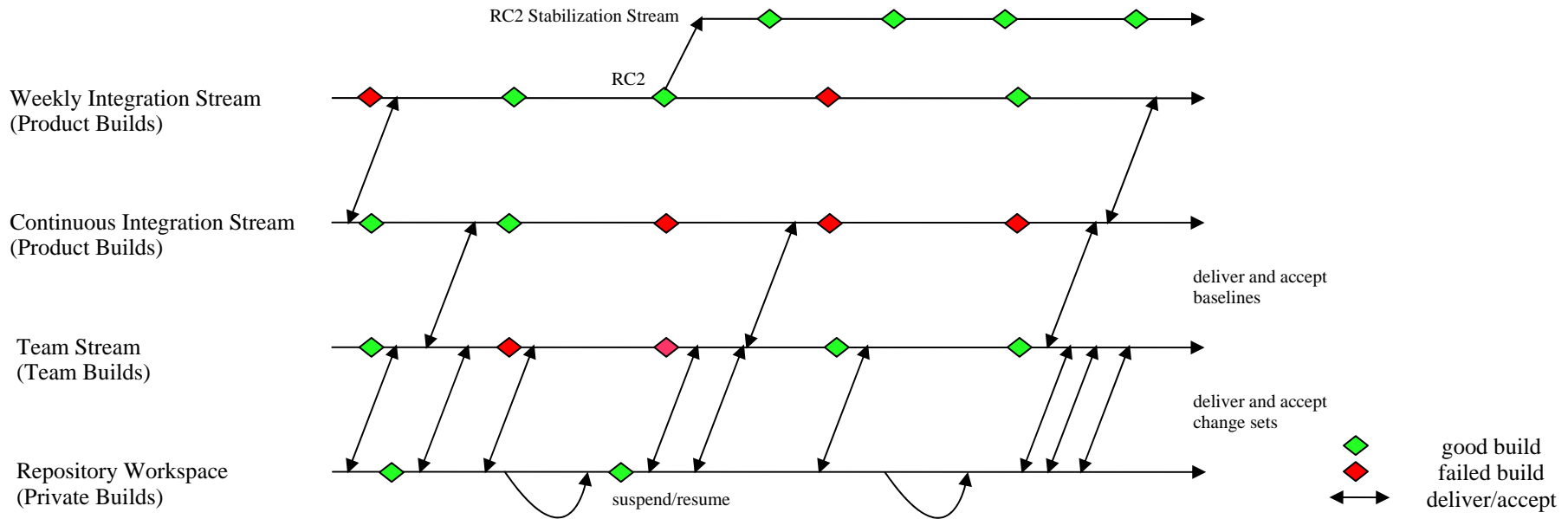
Isolate *work* not *people*. Encourage transparent planning and flexible isolation.

It's not just about SCM

- There is much more to product level parallel development than patching the code.
- Team level productivity and turn around time for delivering fixes is impacted by the tooling of the entire process.
- Consider the number of tasks involved in patching a release that aren't related to SCM, but are tooled in Rational Team Concert.



How the Jazz team works



- Each build references a snapshot of the artifacts that were built. Reproducing a build or patching a build is as simple as creating a new stream from the snapshot. There is no planning needed to tag or baseline, builds take care of that for us.
- Developers deal mainly with accepting and delivering to their team stream. Flowing changes to and from the integration streams is the responsibility of one person on each team.

Coming Soon: Feeds

Feeds - Team Concert - Microsoft Visual Studio

File Edit View Tools VMware Test ReSharper Team Concert Window Help

Process: Thread: Stack Frame:

Feeds - Team Concert Start Page

My Teams in Rational Team Concert VS Client

Showing 53 of 53 Items | [Expand All](#)

Search Text

Scope

- All
- Owner
- Creator
- Subscriber

Show

- Approvals
- Work Items
- Build
- Source Control Streams
- Source Control Changes
- News
- Team Process Changes
- Others

Exclude

- Read News
- News Older one Day
- Modified by Me

Yesterday (18 items found)

- [4] Ability to file a new Work Item from Error View (4467) 10:39 PM
- Lalitha K. Kishore delivered 3 change sets to RTC.NET Client 2.0.0.1 Maintenance 10:37 PM
- [2] Decorate files also in PC view on build errors (4466) 8:32 PM
- RAJIV VERMA delivered 2 change sets to RTC.NET Client 2.0.0.1 Maintenance 8:28 PM
- Without charset / encoding for .resx file (.Net resource) (3614) 3:32 PM

Show All Items...

Last Week (35 items found)

- SREERUPA SEN delivered '3772: Redesign the FSDaemon/Proxy framework' to RTC.NET Client 2.0.0.1 Maintenance Sunday, August 09, 2009 11:43 PM
- Get the mandatory display fields for work item creation dialog, from the repository (4402) Sunday, August 09, 2009 10:47 PM
- [8] TVT test cases for RTC VS Client 2.0.0.1 (4453) Sunday, August 09, 2009 10:39 PM
- Saikarthik Natesan delivered '4402: Get the mandatory display fields for work item creation dialog, from the repository' to RTC.NET Client 2.0.0.1 Maintenance Sunday, August 09, 2009 10:07 PM
- PRABODH MAIRH delivered '4250: User should get a Color picker to customize the text color on feeds view' to RTC.NET Client 2.0.0.1 Maintenance Sunday, August 09, 2009 1:44 PM

Show All Items...

Team Artifacts

- Repository Connections
 - krishna.kishore@in.ibm.com@jazzop16.rtp.raleigh.ibm.com
 - Rational Team Concert [krishna.kishore@in.ibm.com@jazzop16.rtp.ra...
 - Builds
 - Plans
 - SVT_Final
 - 2.0
 - 2.0.0.1
 - 2.0.0.1 M2
 - +0+ 1.0.1.X Maintenance
 - +0+ 2.0.0.X Maintenance
 - +0+ 2.X
 - +0+ Main Development
 - +0+ Post 1.0 Explorations
 - +0+ SVT Development Line
 - Source Control
 - Work Items
 - Feeds
 - Build Events for My Teams [jazzop16.rtp.raleigh.ibm.com]
 - My Teams in Rational Team Concert (53) [jazzop16.rtp.ra...
 - My Teams in Rational Team Concert VS Client (53) [jazzo...
 - My Work Item Changes (78) [jazzop16.rtp.raleigh.ibm.co...
 - My Repository Workspaces

Team Artifacts Solution Explorer

start Lalitha K Kis... 2 Window... Jazz-RTC4... System Def... Feeds - Tea... untitled.JPG... Desktop EN 96% 10:48 AM