Why Customization?

- RTC provides excellent Out-of-the-Box experience:
  - Minimal setup for most cases
  - No customization required
- However: RTC is highly customizable – Why?
  - Different organizations have varying requirements for project governance
  - Different teams want to adapt the tool to the process that works best for them
    - Important for agile teams: continuously adapt and improve

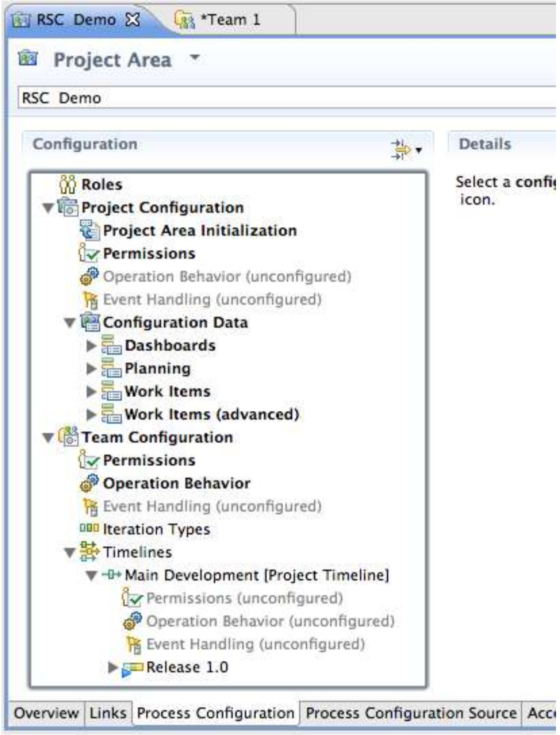
Customizations: Scope

- RTC supports customizations for these components:
  - Process (Jazz Foundation)
  - Work Items
  - Planning
  - Dashboards
  - SCM
  - Build

How is it done?

- **Project Area**:
  - Roles
  - Project configurations
    - Initialization
    - Role based *Operations*
    - Configurations
  - Team configurations
    - Role based *Operations*
    - Timeline/Role based *Operations*

- **Team Area** (optional):
  - Roles
  - Role based *Operations*
  - Timeline/Role based *Operations*


RSC Demo | *Team 1

Project Area

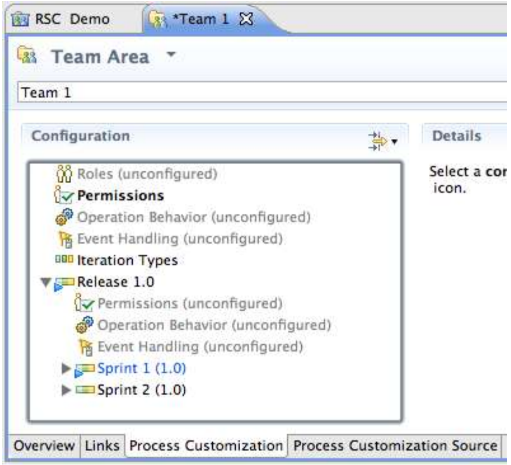
RSC Demo

Configuration | Details

- Roles
- Project Configuration
  - Project Area Initialization
  - Permissions
  - Operation Behavior (unconfigured)
  - Event Handling (unconfigured)
  - Configuration Data
    - Dashboards
    - Planning
    - Work Items
    - Work Items (advanced)
- Team Configuration
  - Permissions
  - Operation Behavior
  - Event Handling (unconfigured)
  - Iteration Types
  - Timelines
    - Main Development [Project Timeline]
      - Permissions (unconfigured)
      - Operation Behavior (unconfigured)
      - Event Handling (unconfigured)
      - Release 1.0

Select a confi
icon.

Overview | Links | Process Configuration | Process Configuration Source | Acc


RSC Demo | *Team 1

Team Area

Team 1

Configuration | Details

- Roles (unconfigured)
- Permissions
- Operation Behavior (unconfigured)
- Event Handling (unconfigured)
- Iteration Types
- Release 1.0
  - Permissions (unconfigured)
  - Operation Behavior (unconfigured)
  - Event Handling (unconfigured)
  - Sprint 1 (1.0)
  - Sprint 2 (1.0)

Select a cor
icon.

Overview | Links | Process Customization | Process Customization Source

Process Customization: Operations

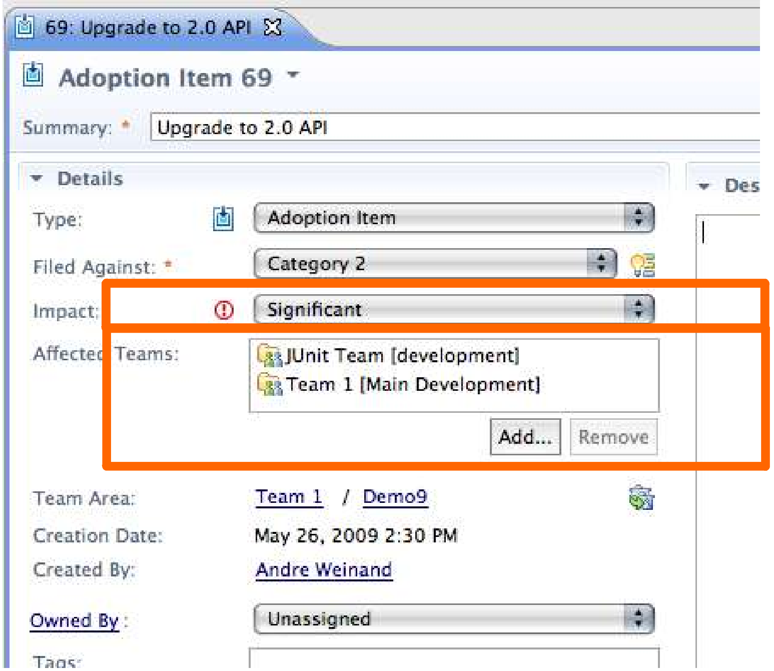
- **Example:**
  - "Save Project Dashboard": project level
  - "Save Work Item": team level

- **Operation Permissions**:
  - Based on finer grained "actions":
    - ***Create*** Work Item
    - ***Modify*** Work Item
    - ***Modify Attribute "Priority"*** of Work Item

- **Operation Behavior**:
  - *Preconditions*, e.g. "Required Properties"
  - *Follow-up actions*, e.g. "Create Initial Work Items"

Work Item Types & Custom Attributes

- **Work Item Type** defines:
  - Display name, Icon
  - Built-in attributes: e.g. *Summary*, *Description*, *Owner*, *Creator*, …
  - Optional custom attributes
  - A state transition matrix (aka *workflow*)



69: Upgrade to 2.0 API

Adoption Item 69

Summary: * Upgrade to 2.0 API

Details

Type: Adoption Item

Filed Against: * Category 2

Impact: Significant

Affected Teams: JUnit Team [development]
Team 1 [Main Development]

Team Area: Team 1 / Demo9

Creation Date: May 26, 2009 2:30 PM

Created By: Andre Weinand

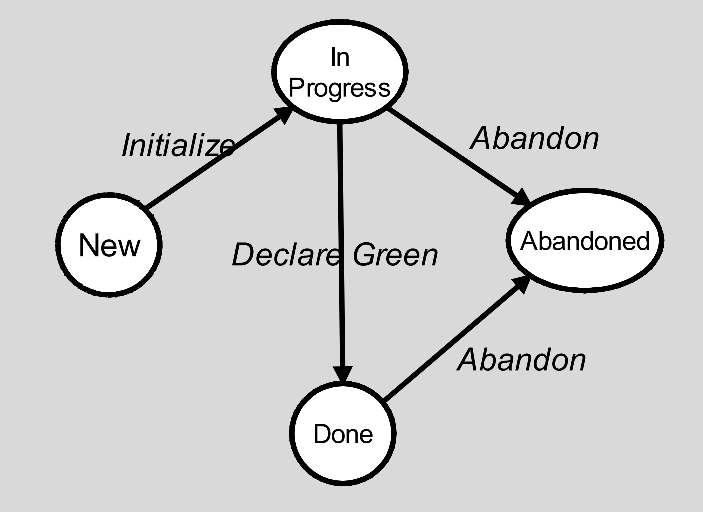
Owned By : Unassigned

Tags:

Work Item Workflows & Workflow Bindings

- A **Workflow** consist of:
  - States
  - *Actions*
  - Resolutions (optional)

- **Workflow Bindings**
  - bind workflow to work item type
  - workflows can be shared

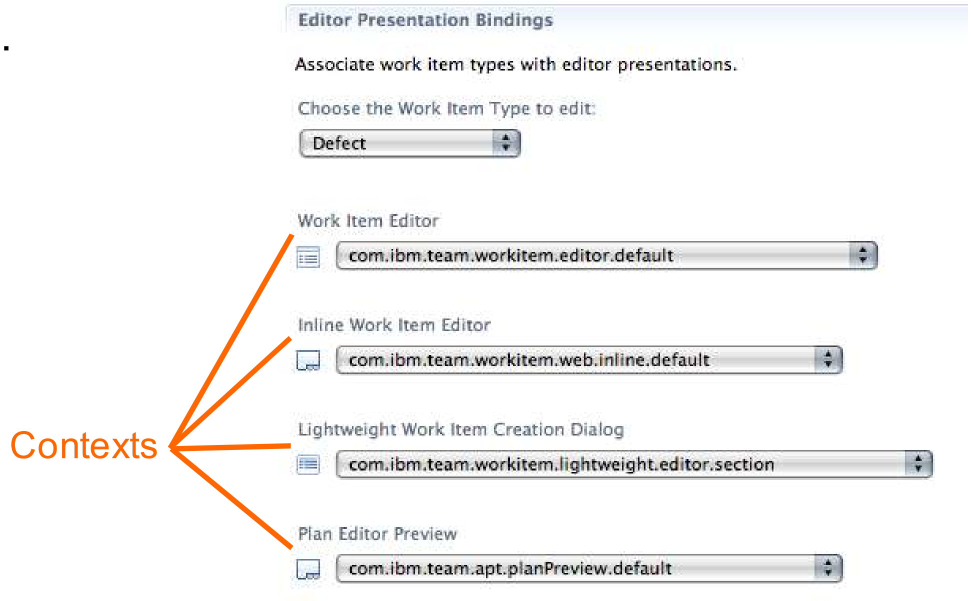


“Build Tracking” Workflow

Customizing the UI – Editor & Attribute Presentations

- **Editor Presentations**
  - Define how a work item is presented based on:
    - Tabs, Sections, Attribute presentations
    - Small number of predefined layouts (no free-form layout yet)
  - Can be used in various contexts:
    - Editor, inline editors, dialogs, hovers, …

- **Editor Presentation Bindings**
  - Bind presentation to work item type
  - Presentations can be shared


Editor Presentation Bindings

Associate work item types with editor presentations.

Choose the Work Item Type to edit:

Defect

Work Item Editor

com.ibm.team.workitem.editor.default

Inline Work Item Editor

com.ibm.team.workitem.web.inline.default

Lightweight Work Item Creation Dialog

com.ibm.team.workitem.lightweight.editor.section

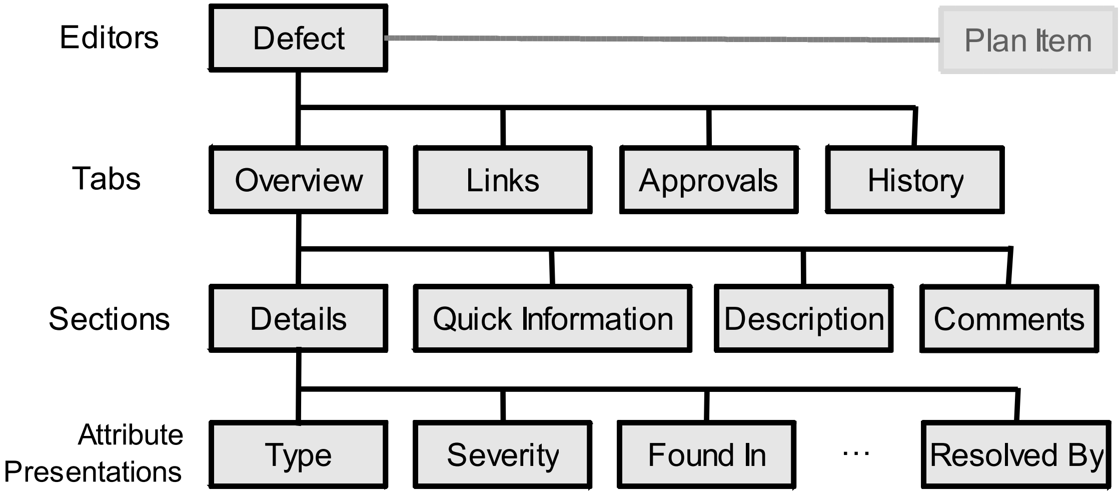
Plan Editor Preview

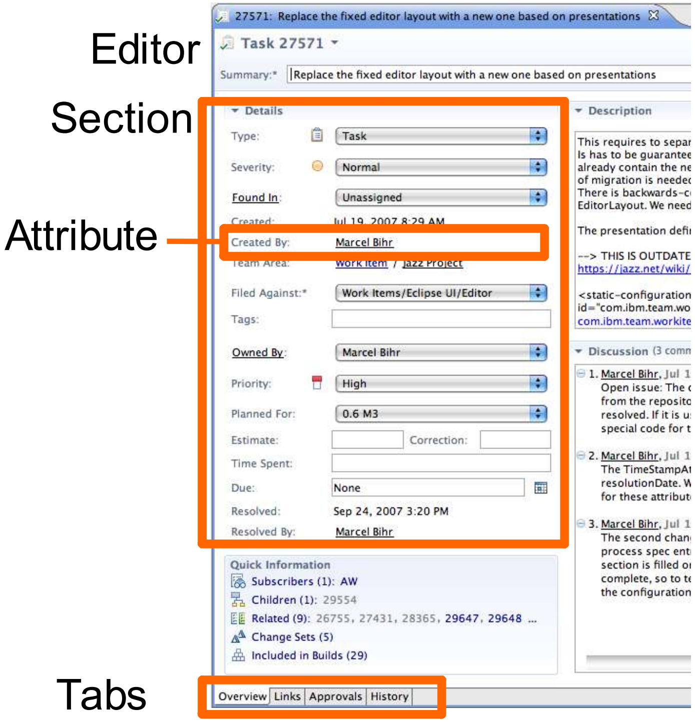
com.ibm.team.apt.planPreview.default

Contexts

Editor Presentation

- Editor’s UI element hierarchy:





27571: Replace the fixed editor layout with a new one based on presentations

Task 27571

Summary:* Replace the fixed editor layout with a new one based on presentations

Details

Type: Task
Severity: Normal
Found In: Unassigned
Created: Jul 19, 2007 8:29 AM
Created By: Marcel Bihr
Team Area: Work Item / Jazz Project
Filed Against:* Work Items/Eclipse UI/Editor
Tags:
Owned By: Marcel Bihr
Priority: High
Planned For: 0.6 M3
Estimate: Correction:
Time Spent:
Due: None
Resolved: Sep 24, 2007 3:20 PM
Resolved By: Marcel Bihr

Quick Information
Subscribers (1): AW
Children (1): 29554
Related (9): 26755, 27431, 28365, 29647, 29648 ...
Change Sets (5)
Included in Builds (29)

Description

This requires to separ
Is has to be guarantee
already contain the ne
of migration is needec
There is backwards-c
EditorLayout. We need

The presentation defi

--> THIS IS OUTDATE
https://jazz.net/wiki/

<static-configuration
id="com.ibm.team.wo
com.ibm.team.workite

Discussion (3 comm

1. Marcel Bihr, Jul 1
Open issue: The c
from the reposito
resolved. If it is u
special code for t

2. Marcel Bihr, Jul 1
The TimeStampAt
resolutionDate. W
for these attribut

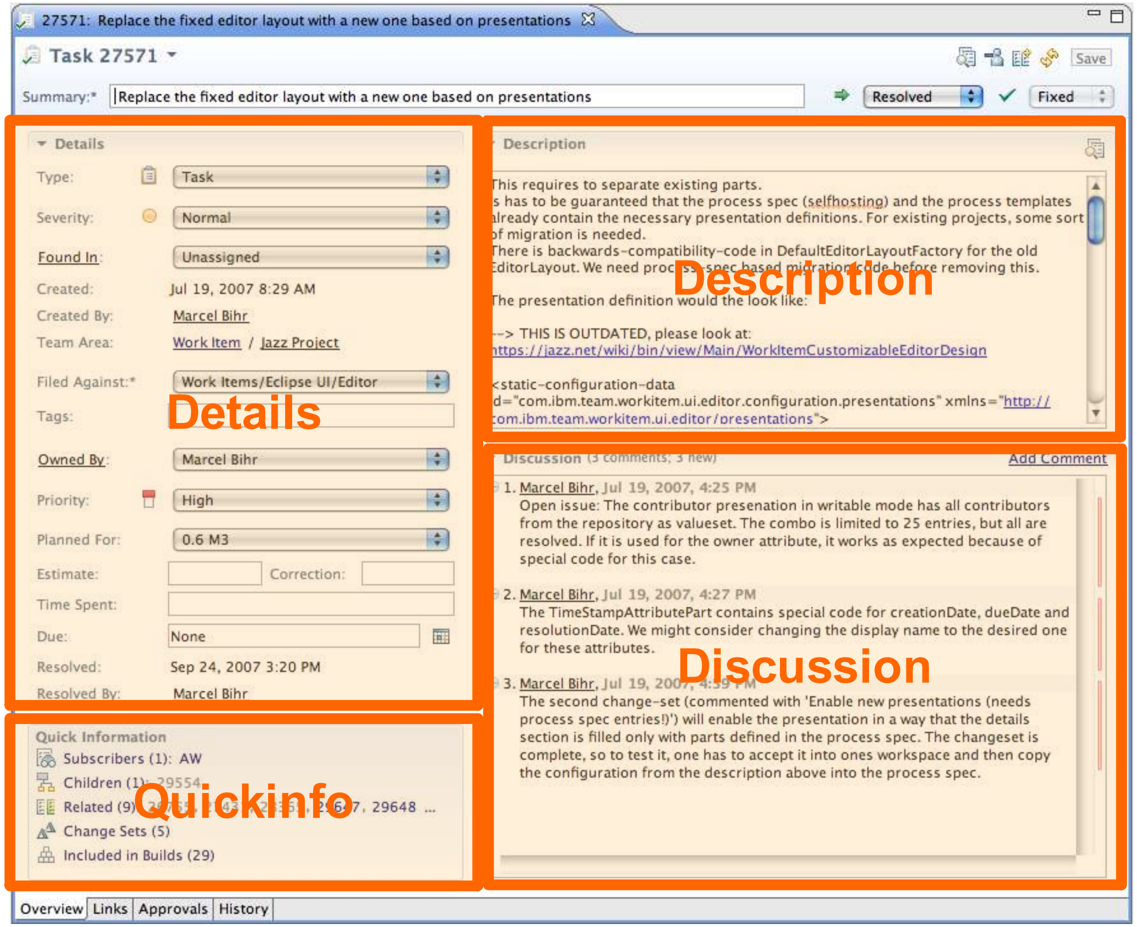
3. Marcel Bihr, Jul 1
The second chan
process spec ent
section is filled o
complete, so to te
the configuration

Overview | Links | Approvals | History

- RTC 2.0 supports structural editing
  - children can be added/removed to/from container

Editor Presentation Layout

- Layout limitations:
  - Tabs have predefined ‘section slots’
  - Sections can be added to slot but not arbitrarily positioned
- Web UI uses same slots
  - but different layout
- Similar limitation for Sections
  - Attributes can be added but not arbitrarily positioned



27571: Replace the fixed editor layout with a new one based on presentations

Task 27571

Summary:* Replace the fixed editor layout with a new one based on presentations

Resolved | Fixed | Save

Details

Type: Task

Severity: Normal

Found In: Unassigned

Created: Jul 19, 2007 8:29 AM

Created By: Marcel Bihr

Team Area: Work Item / Jazz Project

Filed Against:* Work Items/Eclipse UI/Editor

Tags:

Owned By: Marcel Bihr

Priority: High

Planned For: 0.6 M3

Estimate: Correction:

Time Spent:

Due: None

Resolved: Sep 24, 2007 3:20 PM

Resolved By: Marcel Bihr

Description

This requires to separate existing parts.
s has to be guaranteed that the process spec (selfhosting) and the process templates
already contain the necessary presentation definitions. For existing projects, some sort
of migration is needed.
There is backwards-compatibility-code in DefaultEditorLayoutFactory for the old
EditorLayout. We need process-spec based migration code before removing this.

The presentation definition would the look like:

--> THIS IS OUTDATED, please look at:
https://jazz.net/wiki/bin/view/Main/WorkItemCustomizableEditorDesign

<static-configuration-data
d="com.ibm.team.workitem.ui.editor.configuration.presentations" xmlns="http://
com.ibm.team.workitem.ui.editor/presentations">

Discussion (3 comments; 3 new) Add Comment

1. Marcel Bihr, Jul 19, 2007, 4:25 PM
Open issue: The contributor presenation in writable mode has all contributors from the repository as valueset. The combo is limited to 25 entries, but all are resolved. If it is used for the owner attribute, it works as expected because of special code for this case.

2. Marcel Bihr, Jul 19, 2007, 4:27 PM
The TimeStampAttributePart contains special code for creationDate, dueDate and resolutionDate. We might consider changing the display name to the desired one for these attributes.

3. Marcel Bihr, Jul 19, 2007, 4:59 PM
The second change-set (commented with 'Enable new presentations (needs process spec entries!)') will enable the presentation in a way that the details section is filled only with parts defined in the process spec. The changeset is complete, so to test it, one has to accept it into ones workspace and then copy the configuration from the description above into the process spec.

Quick Information

Subscribers (1): AW

Children (1): 29554

Related (9): [illegible], 29647, 29648 ...

Change Sets (5)

Included in Builds (29)

Details

Description

Discussion

Quickinfo

Process Customization: Best Practices

- Process can be easily changed at any time
  - But there are some constraints (see below)
- At end of iteration reflect on how to improve process
  - e.g. collect improvements in “Process Improvement” work items
- After enough reflection/discussions customize existing process instance
  - Start with simple customizations
    - e.g. add a new work item type, add query to find all items of this type
  - Plan how to avoid corruption of existing data
    - e.g. add only, do not change IDs,
      do not remove process definitions that are still used
- After thorough testing create a process template of your customizations that other projects can use

What’s new in RTC 2.0

- More Custom Attribute Types:
  - Tags
  - Item References:
    - User, Category, Iteration, Deliverable, Work Item, Team Area, Project Area
  - Multi Valued Item References:
    - User, Category, Iteration, Deliverable, Work Item, Team Area, Project Area

- Required Attributes scoped to Workflow States
  - Example: require owner on close/fixed

What's new in RTC 2.0 (contd.)

- Improved Editor Presentations:
  - Configurable Header in Work Item Editor
  - Editor Presentations now used in different contexts (Web & Eclipse UI):
    - Work Item Editor
    - Inline Editor
    - Work Item Creation Dialog
    - Rich Hovers
    - Preview
- Conditional Visibility for Attribute Presentations
  - Attribute value based, e.g. *hideIfEmpty*
  - Work Item state based, e.g. *hideIfInState "resolved"*

What’s new in RTC 2.0.0.1

- Initial support for e-Signatures:
  - Add an attribute of type 'E-Signature' to the desired Work Item type.
  - Configure the editor presentation to show the E-Signature attribute in the Approvals section.
  - Enable the 'E-Signature' precondition for the 'Save Work Item' operation and configure it for the desired Approval and Work Item type.

What’s planned for RTC 2.0.0.2

- e-Signatures:
  - Support for Workflow state transitions (“*Actions*”)
- “Require Approvals” Precondition:
  - enforces a matching Approval for a specific state
- Dependent Attributes:
  - Values, value sets, and default values of an attribute are dependent on other attributes