News in RSA-RTE 10.0

updated for sprint 2016.41

Mattias Mohlin/Anders Ek, October 2016

Overview of Improvements (1/5)

- **Now based on Eclipse Mars (4.5.2)**
- **New installation scheme**
  - RSARTE is now installed as a P2 update site directly on-top of Eclipse
  - The update site includes EGit 4.4.1
- **Compare/Merge**
  - Improved grouping of dependent changes
  - Better change descriptions
  - Possible to suppress (i.e. automatically resolve) conflicts that only affect presentation
  - New Delta Tree Configuration options for filtering out non-code changes and to hide/show the description text field
  - The useful code editor area is increased to show more code by removing meaningless header for code snippets
  - Better support for navigation from changes

Overview of Improvements (2/5)

- **Project Explorer**
  - More compact and logical representation of guards
  - Improvements for Multiplicity in Project Explorer
  - Default Value Visualization in Project Explorer
  - Keyword Visualization for Activities in Project Explorer
- **Properties View**
  - Sorting order of Signature column is kept when switching context
  - The Documentation tab is now the default tab for comments
- **Editing**
  - Change of type no longer clears default value
  - Support for more C++ like syntax for creating operations
  - Increased Undo history size for model editing
- **Code View**
  - New preference for ensuring that Code View tabs always appear in fixed order
  - Hiding code snippets that are irrelevant for the selected element

Overview of Improvements (3/5)

- **Refactoring**
  - Refactorings now allow to set the active TC when running the refactoring
  - When deleting a TC it’s now possible to avoid modifying referencing TCs
- **Navigation**
  - Navigation into sub state machine diagram improved
  - Navigate into TOP states when double-clicking state machine in Project Explorer
  - Properties and Code view now updates when selecting search result items for Find commands (no longer necessary to run a navigation command)
- **TC Editor**
  - Visual indication of the different kinds of values in a TC
  - Support for navigating to inherited values
  - Support for deleting values

Overview of Improvements (4/5)

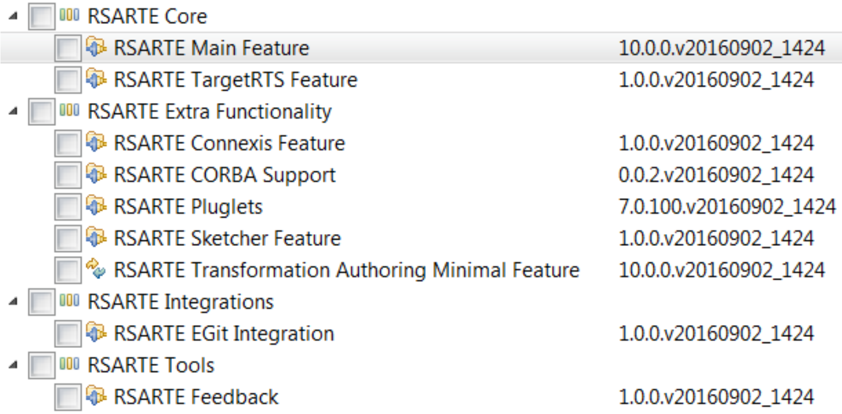
- **Build**
  - Model Compiler – a new command-line tool for building RSARTE models
  - Support for template parameters in ”Detect Source Dependencies Automatically”
  - Missing source elements now added to TC where Sources list is defined
  - Removed checks for minimum port cardinality
- **C++ Code Generator**
  - Support for state entry operations at class/capsule level
  - Removed Lint Warning for Equality Operator
- **Diagrams**
  - Incremental textual search is now supported in Browse diagrams
  - Easier to navigate by double-clicking on symbols in composite structure and activity diagrams
  - Transition tooltip in a state chart diagram now tells the events that trigger the transition

Overview of Improvements (5/5)

- **Documentation**
  - Context sensitive help available for the TC editor
  - Welcome page with links to RSARTE wiki
  - Built-in help improvements
- **New search (experimental)**
  - A first glimpse of the new search capability
  - The focus of current work is to improve performance and the core work flow: Search, explore search results and navigate to found matches.
- **Miscellaneous**
  - Support for ”authorized user” licenses

New Installation Scheme

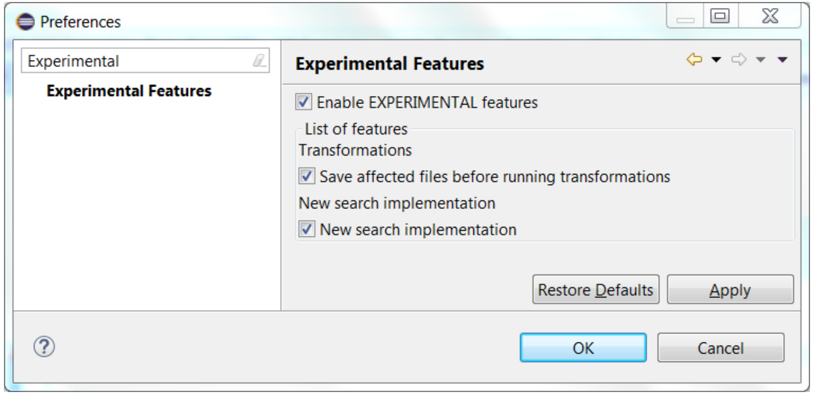
- The Installation Manager is no longer used for installing RSARTE. Instead, RSARTE is now a standard P2 update site.
- This makes it easier to install RSARTE into different kinds of Eclipse distributions (where RSARTE may only be one out of many extensions to standard Eclipse)
- The installation experience has become more streamlined
  - Previously the installation had two phases (base installation using Installation Manager and then patch installation using a P2 update site). Now everything is installed in a single phase through a single P2 update site.
  - Installation units have been refactored to better suit which RSARTE components users typically need to install
  - Oomph can be used for installing RSARTE customized according to the needs of your organization



□ RSARTE Core	
□ RSARTE Main Feature	10.0.0.v20160902_1424
□ RSARTE TargetRTS Feature	1.0.0.v20160902_1424
□ RSARTE Extra Functionality	
□ RSARTE Connexis Feature	1.0.0.v20160902_1424
□ RSARTE CORBA Support	0.0.2.v20160902_1424
□ RSARTE Pluglets	7.0.100.v20160902_1424
□ RSARTE Sketcher Feature	1.0.0.v20160902_1424
□ RSARTE Transformation Authoring Minimal Feature	10.0.0.v20160902_1424
□ RSARTE Integrations	
□ RSARTE EGit Integration	1.0.0.v20160902_1424
□ RSARTE Tools	
□ RSARTE Feedback	1.0.0.v20160902_1424

Consequences of New Installation Scheme

- Less ”branding”. The standard Eclipse splash screen, icon, About dialog etc. are used. The philosophy is that RSARTE is only one out of possibly many extensions used with Eclipse, and therefore should not ”hijack” more than necessary of common Eclipse user interface.
- Experimental features are now enabled using the standard Preferences dialog



Experimental Features

☑ Enable EXPERIMENTAL features

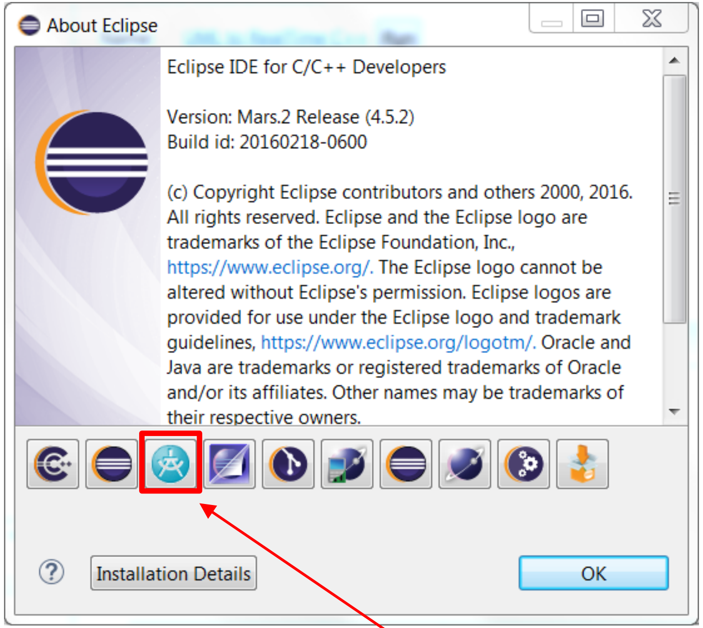
List of features

Transformations

☑ Save affected files before running transformations

New search implementation

☑ New search implementation



Eclipse IDE for C/C++ Developers

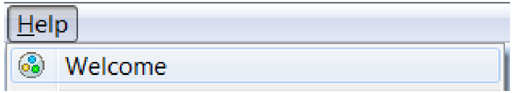
Version: Mars.2 Release (4.5.2)
Build id: 20160218-0600

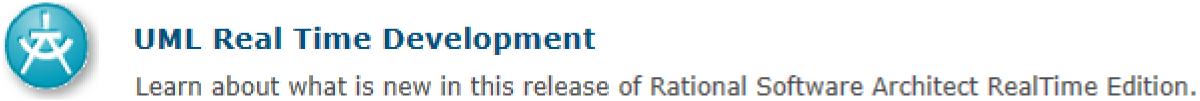
(c) Copyright Eclipse contributors and others 2000, 2016. All rights reserved. Eclipse and the Eclipse logo are trademarks of the Eclipse Foundation, Inc., https://www.eclipse.org/. The Eclipse logo cannot be altered without Eclipse's permission. Eclipse logos are provided for use under the Eclipse logo and trademark guidelines, https://www.eclipse.org/logotm/. Oracle and Java are trademarks or registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

click to view version of RSARTE and components that are currently installed

RSARTE Links on the Welcome Page

- Links to Release Notes on the RSARTE wiki that previously were present in the About dialog can now be found in the Welcome page (in the section ”What’s New”)

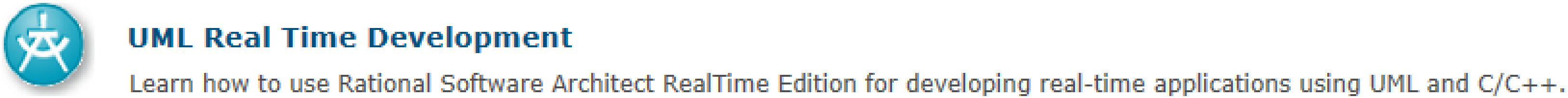




UML Real Time Development

Learn about what is new in this release of Rational Software Architect RealTime Edition.

- There is also a link to the RSARTE wiki (in the section ”Overview”)

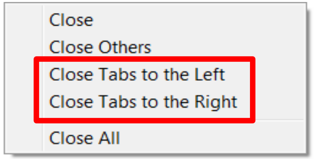


UML Real Time Development

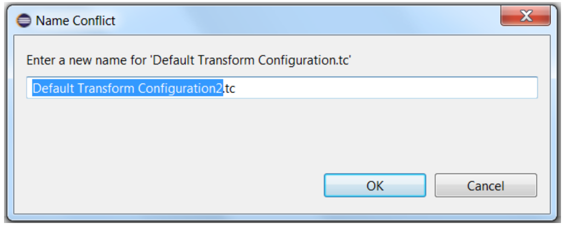
Learn how to use Rational Software Architect RealTime Edition for developing real-time applications using UML and C/C++.

Eclipse 4.5.2 (Mars)

- Performance improvements
  - For example, text search is now 3-4 times faster on multi-core machines
- Easier to manage open editors using the new ”Close Tabs to the Left/Right” commands



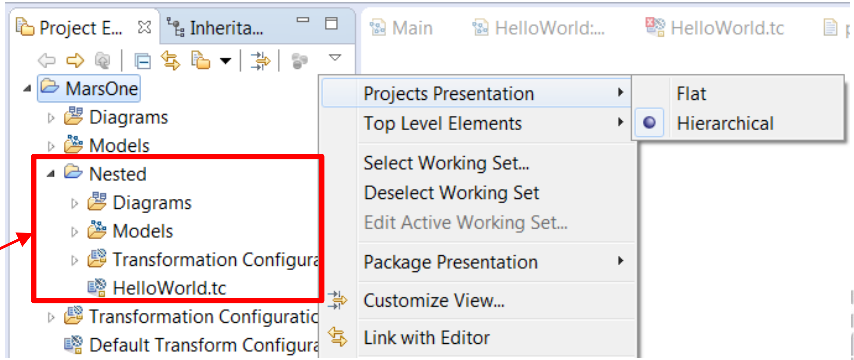
- Improved usability of copying files
  - Better default names suggested (using numbers instead of ”Copy of”)
  - Base file name (without extension) is selected by default


Name Conflict

Enter a new name for 'Default Transform Configuration.tc'

Default Transform Configuration2.tc

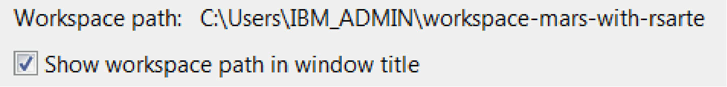
- Support for nested projects in Project Explorer
  - Can help keep the Project Explorer simple



nested project

Eclipse 4.5.2 (Mars)

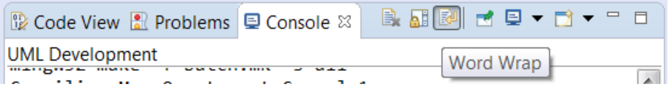
- View workspace location in preferences and window title
  - No longer necessary to use the `–showLocation` command-line argument to achieve this



Workspace path: C:\Users\IBM_ADMIN\workspace-mars-with-rsarte

☑ Show workspace path in window title

- Support for word wrap in the Console view
  - Useful for example to avoid scrolling when looking at long compiler messages

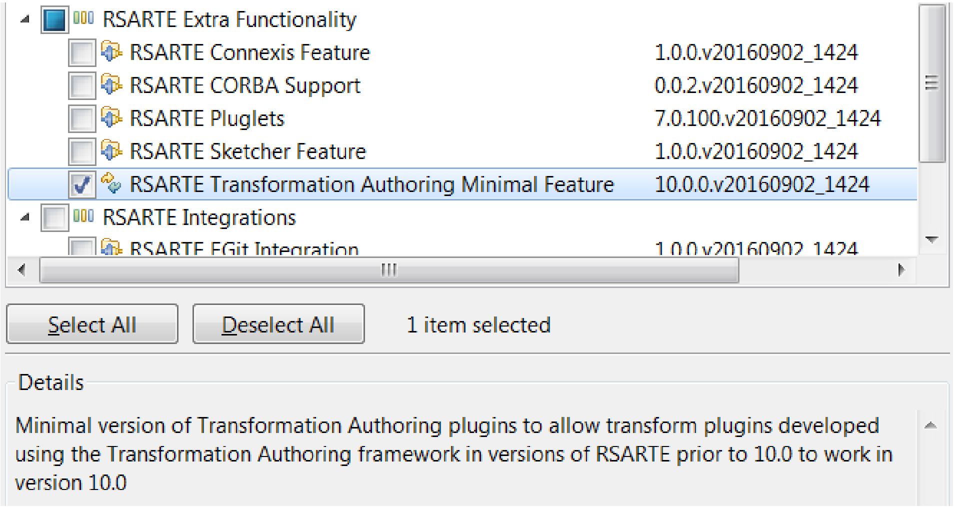


- Major improvements in CDT
  - Version upgraded from 8.1 to 8.8
  - Many new features (in particular nice debug improvements) and performance improvements (for example, the indexer is ~45% faster)
    for details, see https://wiki.eclipse.org/CDT/User/NewIn82, https://wiki.eclipse.org/CDT/User/NewIn83, https://wiki.eclipse.org/CDT/User/NewIn84, https://wiki.eclipse.org/CDT/User/NewIn85, https://wiki.eclipse.org/CDT/User/NewIn86, https://wiki.eclipse.org/CDT/User/NewIn87, https://wiki.eclipse.org/CDT/User/NewIn88

- For more information about Eclipse improvements see
  - News in Eclipse 4.5 (Mars) https://www.eclipse.org/eclipse/news/4.5/platform.php

Transformation Authoring

- Transformation Authoring was a tool in older RSARTE versions for simplifying the process of creating custom transformation plugins
- RSARTE 10 no longer includes this tool. However, it includes a minimal version of the Transformation Authoring framework which allows legacy transform plugins to continue to work.
- This is an optional choice during installation



■ RSARTE Extra Functionality	
□ RSARTE Connexis Feature	1.0.0.v20160902_1424
□ RSARTE CORBA Support	0.0.2.v20160902_1424
□ RSARTE Pluglets	7.0.100.v20160902_1424
□ RSARTE Sketcher Feature	1.0.0.v20160902_1424
☑ RSARTE Transformation Authoring Minimal Feature	10.0.0.v20160902_1424
□ RSARTE Integrations	
□ RSARTE EGit Integration	1.0.0.v20160902_1424

1 item selected

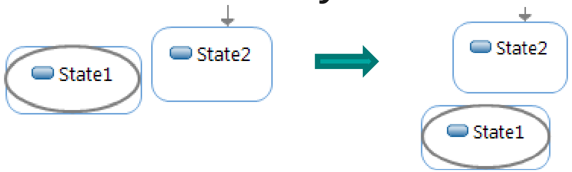
Details

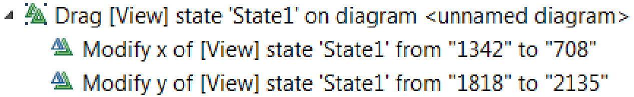
Minimal version of Transformation Authoring plugins to allow transform plugins developed using the Transformation Authoring framework in versions of RSARTE prior to 10.0 to work in version 10.0

Improved Presentation of Compare/Merge Changes

- Several small changes caused by the same model modifications are now grouped together and given a better change description.
  - This reduces the number of changes and conflicts and makes it easier to quickly understand how the model has been modified.
  - If you are interested in the low-level details of the changes, they are still available (under the group)

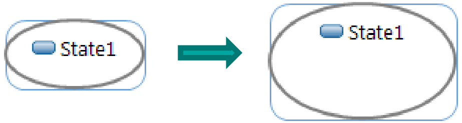
Move a symbol

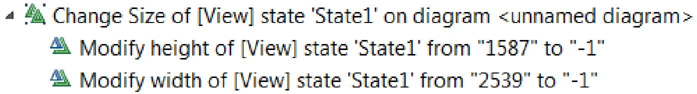




- Drag [View] state 'State1' on diagram <unnamed diagram>
  - Modify x of [View] state 'State1' from "1342" to "708"
  - Modify y of [View] state 'State1' from "1818" to "2135"

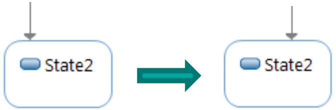
Resize a symbol

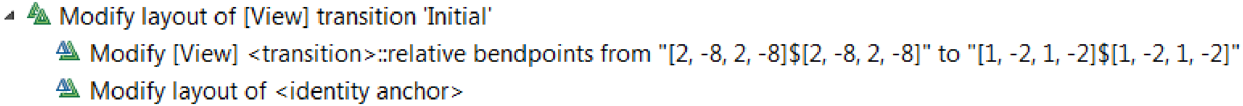




- Change Size of [View] state 'State1' on diagram <unnamed diagram>
  - Modify height of [View] state 'State1' from "1587" to "-1"
  - Modify width of [View] state 'State1' from "2539" to "-1"

Move the end-point of a line along the edge of the connected symbol

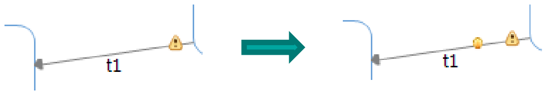


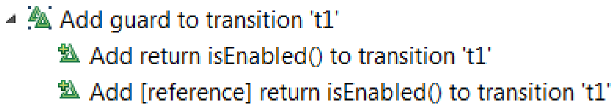


- Modify layout of [View] transition 'Initial'
  - Modify [View] <transition>::relative bendpoints from "[2, -8, 2, -8]$[2, -8, 2, -8]" to "[1, -2, 1, -2]$[1, -2, 1, -2]"
  - Modify layout of <identity anchor>

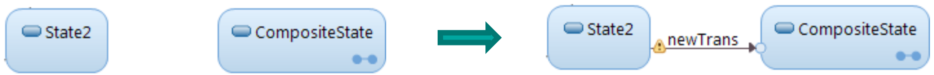
Improved Presentation of Compare/Merge Changes

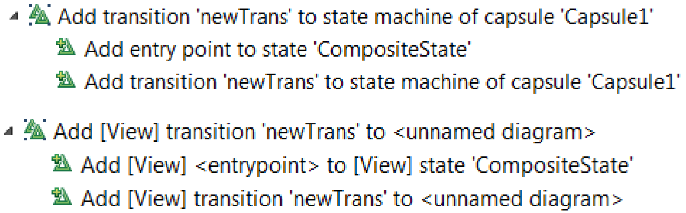
Add a transition guard




- Add guard to transition 't1'
  - Add return isEnabled() to transition 't1'
  - Add [reference] return isEnabled() to transition 't1'

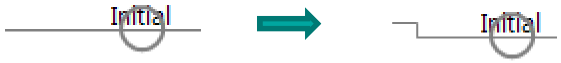
Add transition targeting a composite state

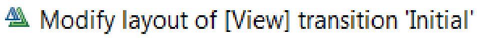



- Add transition 'newTrans' to state machine of capsule 'Capsule1'
  - Add entry point to state 'CompositeState'
  - Add transition 'newTrans' to state machine of capsule 'Capsule1'
- Add [View] transition 'newTrans' to <unnamed diagram>
  - Add [View] <entrypoint> to [View] state 'CompositeState'
  - Add [View] transition 'newTrans' to <unnamed diagram>

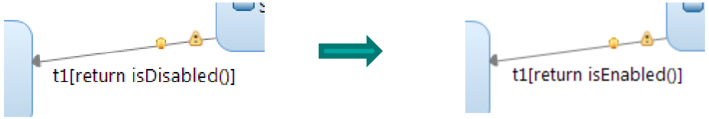
- In some cases no group is necessary, and then only the change description was improved

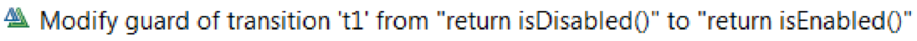
Change routing of a line




Modify layout of [View] transition 'Initial'

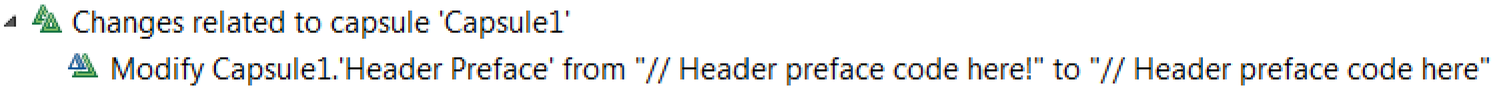
Modify a transition guard




Modify guard of transition 't1' from "return isDisabled()" to "return isEnabled()"

Improved Presentation of Compare/Merge Changes

- Changes in values for applied stereotypes are now shown under the element where the stereotype is applied (if ”Hierarchy Structure” is used in the Delta Tree Configuration)
  - In particular this simplifies the presentation of changes in code snippets that are stored as PropertySets stereotype values
- Also, the change description was improved to be consistent with changing any other property of the element

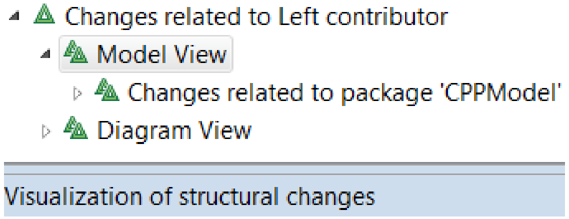


Changes related to capsule 'Capsule1'

Modify Capsule1.'Header Preface' from "// Header preface code here!" to "// Header preface code here"

Change Description Text Field Hidden

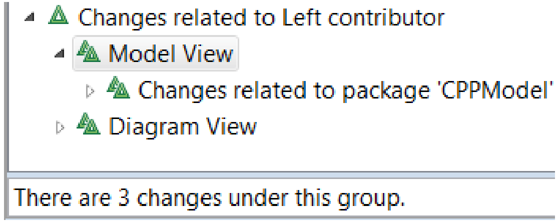
- The text field that shows the detailed description of a change is now by default hidden
  - This saves valuable space in the Compare/Merge editor
- It can be made visible by using the ”Delta Tree Configuration” button menu



Changes related to Left contributor
Model View
Changes related to package 'CPPModel'
Diagram View

Visualization of structural changes

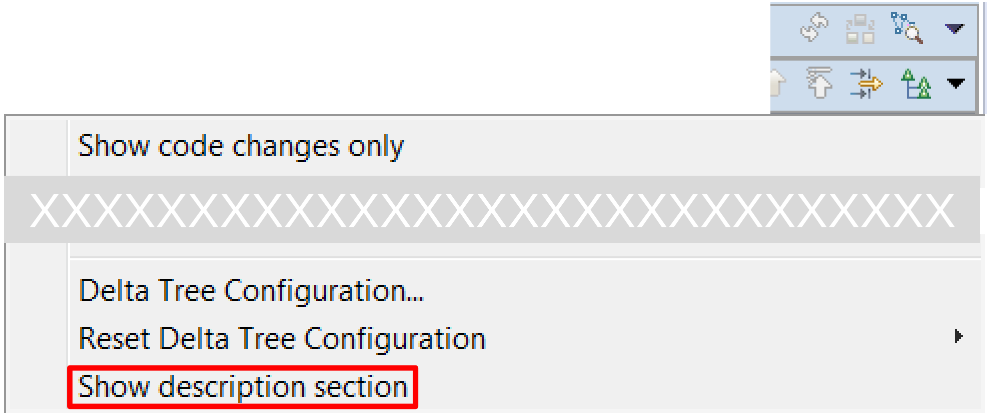
Description field hidden



Changes related to Left contributor
Model View
Changes related to package 'CPPModel'
Diagram View

There are 3 changes under this group.

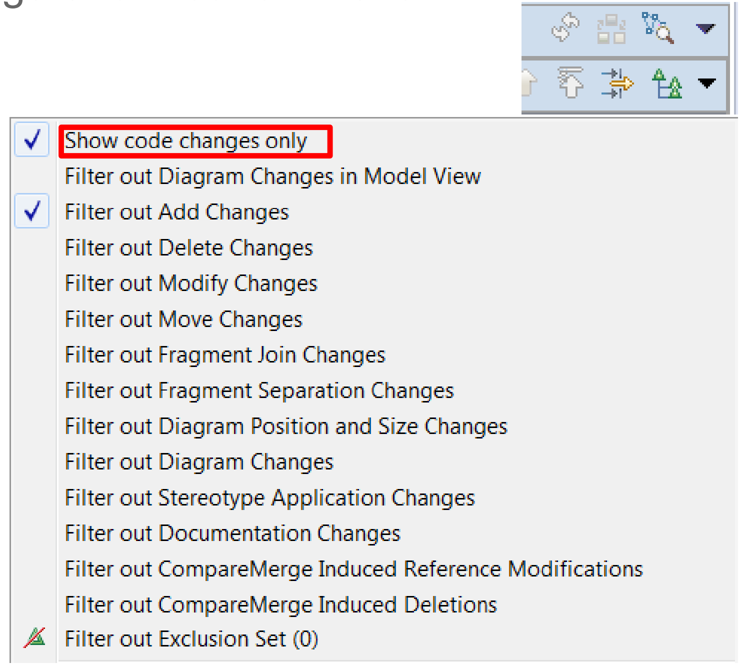
Description field visible



Show code changes only
XXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Delta Tree Configuration...
Reset Delta Tree Configuration
Show description section

Filter Out Non-Code Changes

- A new filter is available which can be used to only show code changes in Compare/Merge
- It’s available in the ”Delta Tree Configuration” button menu



✓ Show code changes only

Filter out Diagram Changes in Model View

✓ Filter out Add Changes

Filter out Delete Changes

Filter out Modify Changes

Filter out Move Changes

Filter out Fragment Join Changes

Filter out Fragment Separation Changes

Filter out Diagram Position and Size Changes

Filter out Diagram Changes

Filter out Stereotype Application Changes

Filter out Documentation Changes

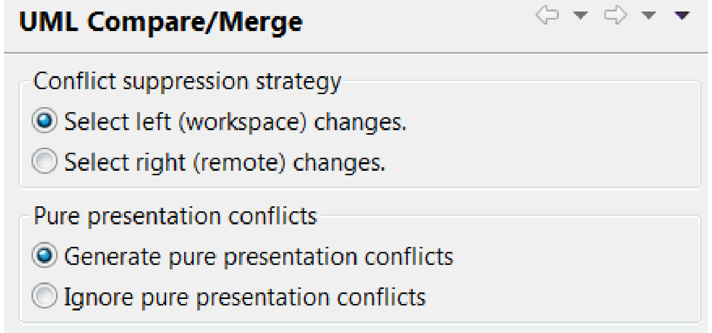
Filter out CompareMerge Induced Reference Modifications

Filter out CompareMerge Induced Deletions

Filter out Exclusion Set (0)

Suppression of Conflicts

- Compare/Merge is now able to suppress some kinds of conflicts, by automatically resolving them. The preference *General – Compare/Patch – Modeling Compare/Merge – UML Compare/Merge – Conflict suppression strategy* controls which version of the model to use when resolving such conflicts automatically.
- In particular it can be useful to suppress pure presentation conflicts (i.e. changes that only affect how a model is presented). This can be controlled using the preference *Pure presentation conflicts*.



UML Compare/Merge

Conflict suppression strategy

● Select left (workspace) changes.

○ Select right (remote) changes.

Pure presentation conflicts

● Generate pure presentation conflicts

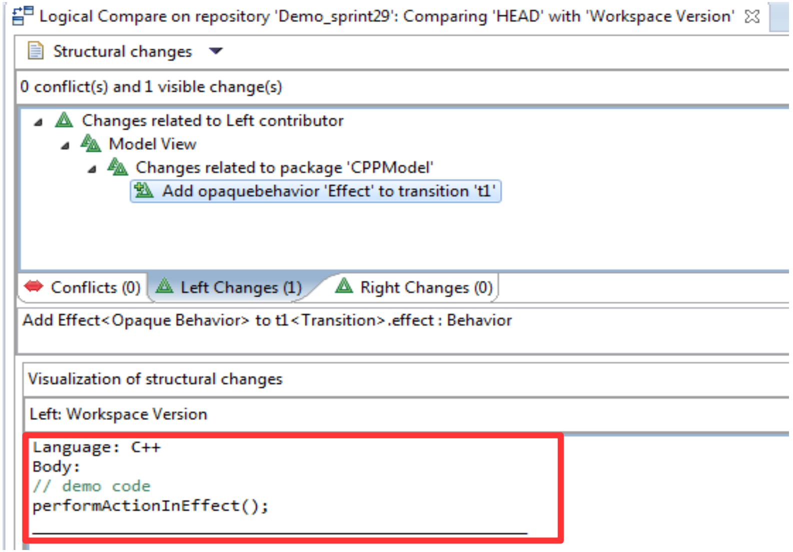
○ Ignore pure presentation conflicts

- This feature is also supported by the command-line Compare/Merge tool by means of new command-line arguments
  - `-CPconflictSuppressionStrategy=left|right`
  - `-CPconflictSuppression=notation=on|off`

Bigger Code Editor Area in Compare/Merge

- The useful code editor area is now bigger in the Compare/Merge editor so it can show more code. This was accomplished by removing the two lines “Language” and “Body” since they did not provide much value.

before



Logical Compare on repository 'Demo_sprint29': Comparing 'HEAD' with 'Workspace Version'

Structural changes

0 conflict(s) and 1 visible change(s)

- Changes related to Left contributor
  - Model View
    - Changes related to package 'CPPModel'
      - Add opaquebehavior 'Effect' to transition 't1'

Conflicts (0) | Left Changes (1) | Right Changes (0)

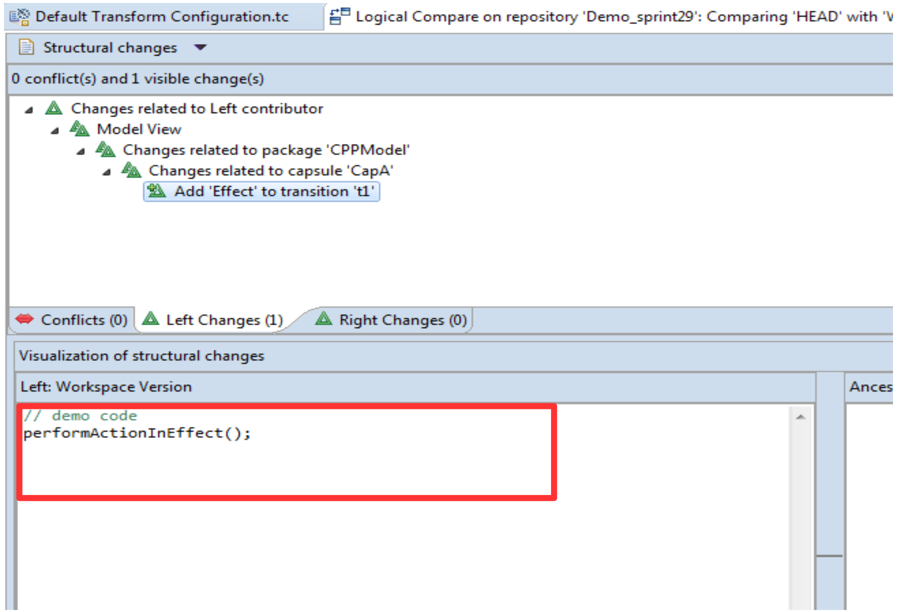
Add Effect<Opaque Behavior> to t1<Transition>.effect : Behavior

Visualization of structural changes

Left: Workspace Version

```
Language: C++
Body:
// demo code
performActionInEffect();
```

now



Default Transform Configuration.tc | Logical Compare on repository 'Demo_sprint29': Comparing 'HEAD' with 'V

Structural changes

0 conflict(s) and 1 visible change(s)

- Changes related to Left contributor
  - Model View
    - Changes related to package 'CPPModel'
      - Changes related to capsule 'CapA'
        - Add 'Effect' to transition 't1'

Conflicts (0) | Left Changes (1) | Right Changes (0)

Visualization of structural changes

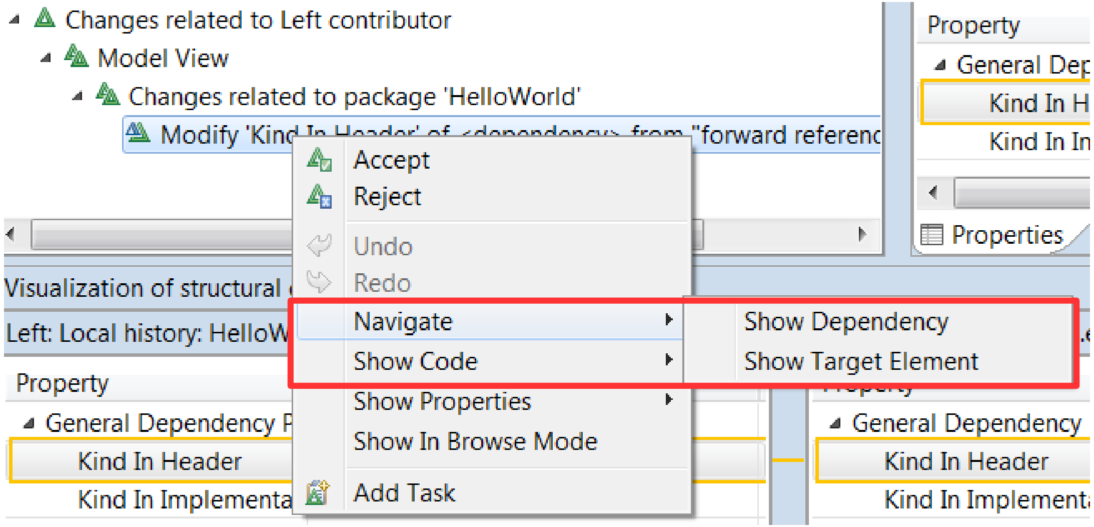
Left: Workspace Version | Ances

```
// demo code
performActionInEffect();
```

Better Support for Navigation from Changes

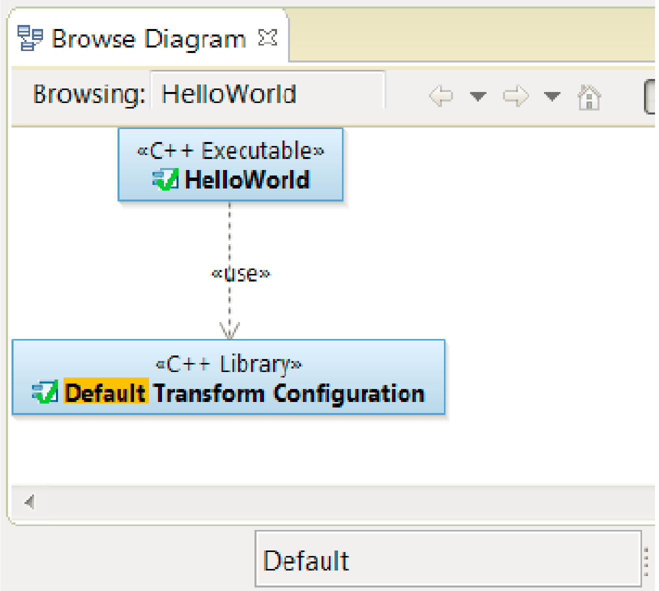
- There is now a Navigate context menu on changes with commands for navigating
  - to the element itself
  - to other related elements (where applicable)

- For example, this allows to navigate to the target element of a dependency that has been changed.



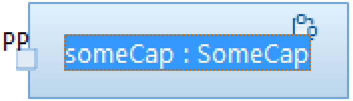
Incremental Textual Search in Browse Diagrams

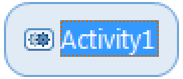
- Browse diagrams now support searching for text using Incremental Find (Ctrl + J)
- This significantly improves the usability of working with big Browse diagrams
- For example, when using a Browse diagram for a TC in order to visualize its prerequisite TCs, Incremental Find is useful in order to find a certain prerequisite TC that is shown on the diagram.



Easier to Navigate by Double-Click in Diagrams

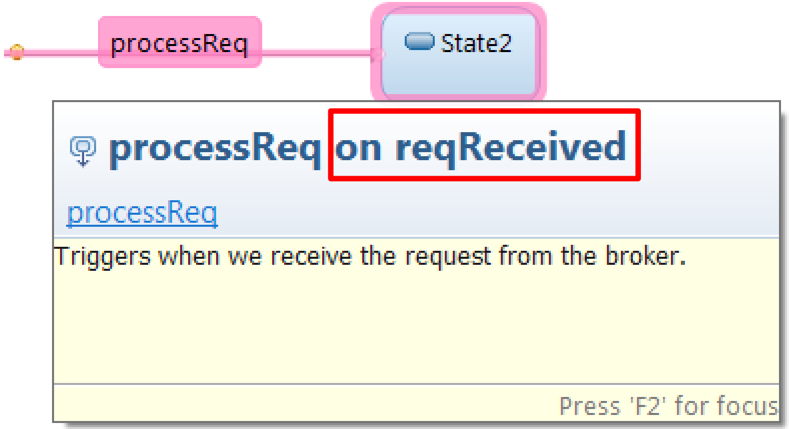
- Composite structure and activity diagrams were improved to work like the state chart editor when it comes to navigation by double-click on symbols
- Now the whole symbol, including the name label, can be double-clicked for navigation
- This makes navigation easier (no longer necessary to be so precise when double-clicking as before)





Enhanced Tooltip for Transitions

- The tooltip for a transition in a state chart diagram now includes information about the events that may trigger the transition
- This makes it easier to understand the behavior of a state machine without navigating to the Project Explorer or Properties view.



processReq on reqReceived

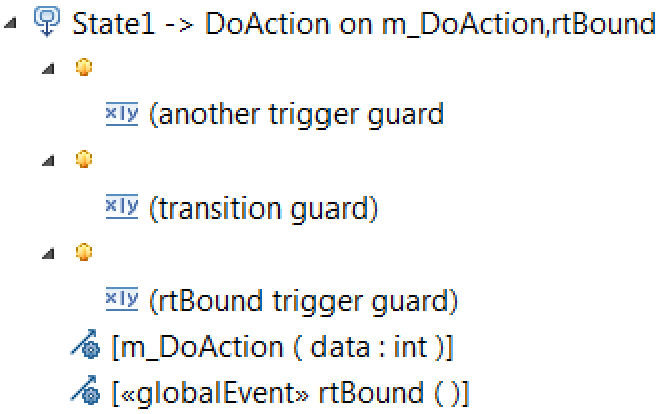
processReq

Triggers when we receive the request from the broker.

Improvements for Guards in Project Explorer

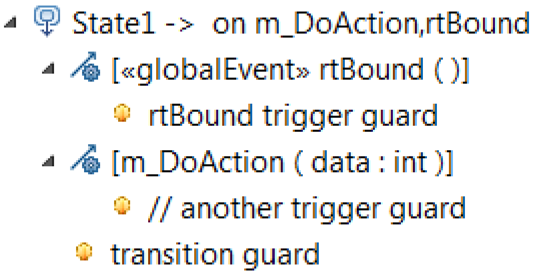
- Nodes representing guards for transitions and triggers now show the first line of code only (avoids performance problem on Linux in case long guard code snippets are used)
- The representation of a guard is now more compact only using one node in the Project Explorer (previously two nodes were used)
- Guards for triggers are now placed below the trigger node to make it more clear where guards belong in case there are multiple triggers with guards for a transition

before



```
State1 -> DoAction on m_DoAction,rtBound
    
        (another trigger guard
    
        (transition guard)
    
        (rtBound trigger guard)
    [m_DoAction ( data : int )]
    [«globalEvent» rtBound ( )]
```

now

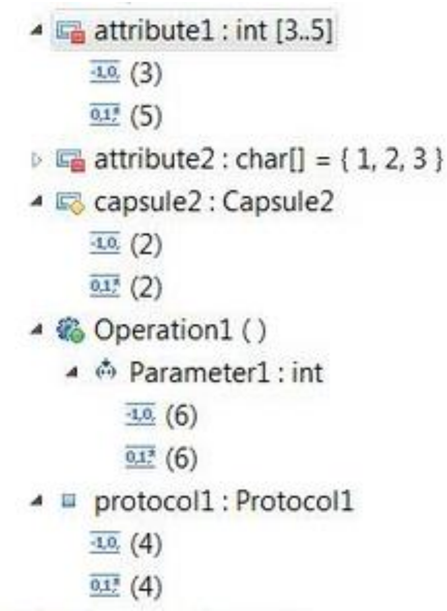


```
State1 ->  on m_DoAction,rtBound
    [«globalEvent» rtBound ( )]
        rtBound trigger guard
    [m_DoAction ( data : int )]
        // another trigger guard
    transition guard
```

Improvements for Multiplicity in Project Explorer

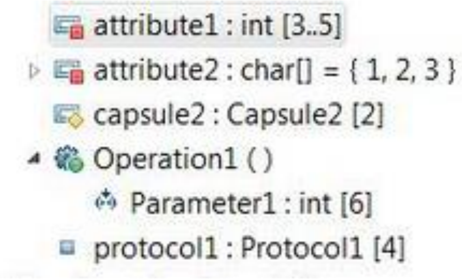
- The multiplicity for attributes, parameters and ports are now shown using inline syntax on the entity itself, and not as before using separate nodes
- This makes the Project Explorer tree more compact and readable

before



```
attribute1 : int [3..5]
    (3)
    (5)
attribute2 : char[] = { 1, 2, 3 }
capsule2 : Capsule2
    (2)
    (2)
Operation1 ( )
    Parameter1 : int
        (6)
        (6)
protocol1 : Protocol1
    (4)
    (4)
```

now

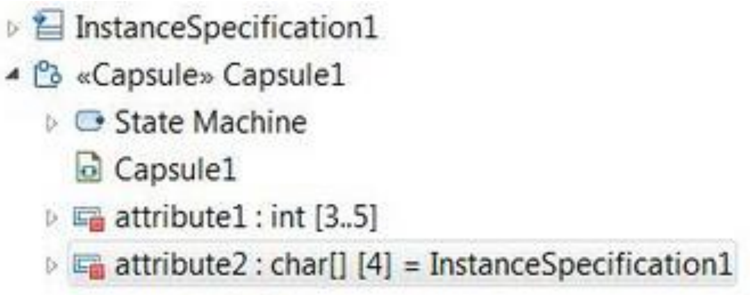


```
attribute1 : int [3..5]
attribute2 : char[] = { 1, 2, 3 }
capsule2 : Capsule2 [2]
Operation1 ( )
    Parameter1 : int [6]
protocol1 : Protocol1 [4]
```

Default Value Visualization in Project Explorer

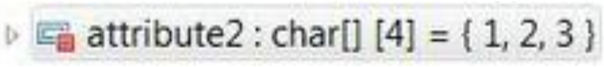
- In previous versions the default value for attributes sometimes showed up as “InstanceSpecifications”
- Now the value will be shown directly instead

before



InstanceSpecification1
«Capsule» Capsule1
State Machine
Capsule1
attribute1 : int [3..5]
attribute2 : char[] [4] = InstanceSpecification1

now

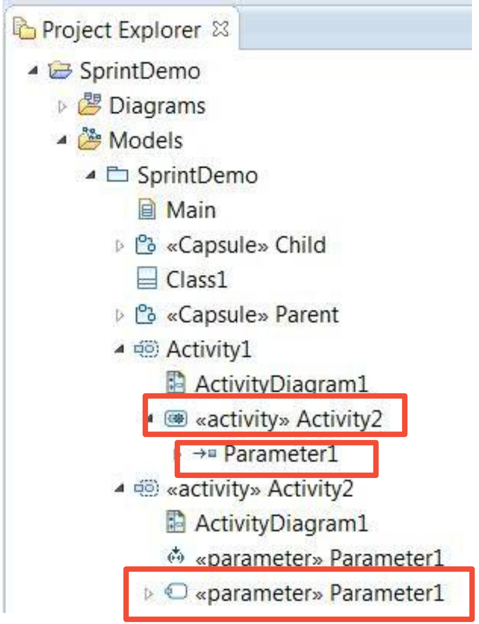


attribute2 : char[] [4] = { 1, 2, 3 }

Keyword Visualization for Activities in Project Explorer

- In previous versions the keyword for nodes in activities sometimes showed the keyword associated with referenced entity or no keyword at all.
- Now the keyword associated with the node itself is shown.

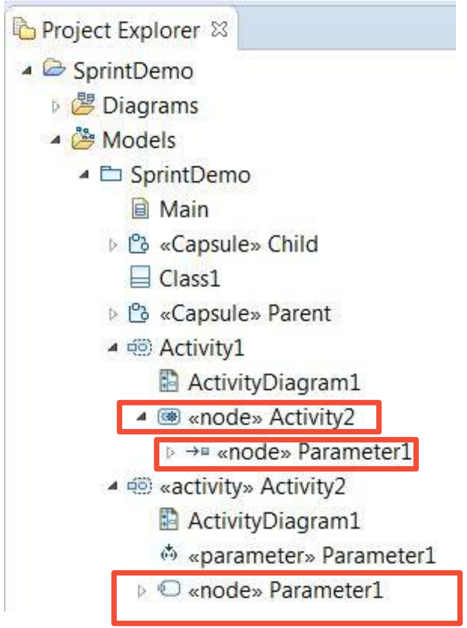
before



Project Explorer

SprintDemo
Diagrams
Models
SprintDemo
Main
«Capsule» Child
Class1
«Capsule» Parent
Activity1
ActivityDiagram1
«activity» Activity2
Parameter1
«activity» Activity2
ActivityDiagram1
«parameter» Parameter1
«parameter» Parameter1

now

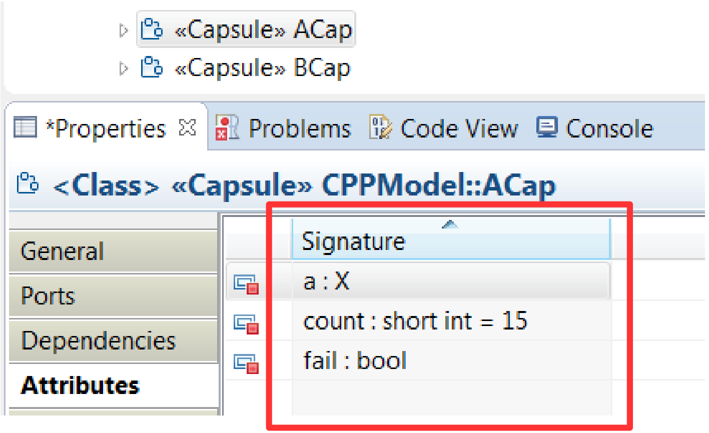


Project Explorer

SprintDemo
Diagrams
Models
SprintDemo
Main
«Capsule» Child
Class1
«Capsule» Parent
Activity1
ActivityDiagram1
«node» Activity2
«node» Parameter1
«activity» Activity2
ActivityDiagram1
«parameter» Parameter1
«node» Parameter1

Preserve Sorting Order in Properties View

- The sorting order of the Signature column is now kept when selecting a different model element
- This makes it easier to keep attributes and operations sorted alphabetically in the Properties view when switching the selection



«Capsule» ACap
«Capsule» BCap

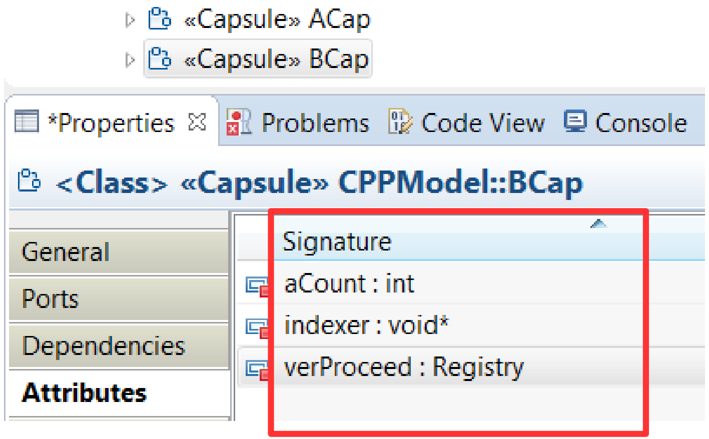
*Properties | Problems | Code View | Console

<Class> «Capsule» CPPModel::ACap

General
Ports
Dependencies
Attributes

Signature
a : X
count : short int = 15
fail : bool

Selecting capsule BCap



«Capsule» ACap
«Capsule» BCap

*Properties | Problems | Code View | Console

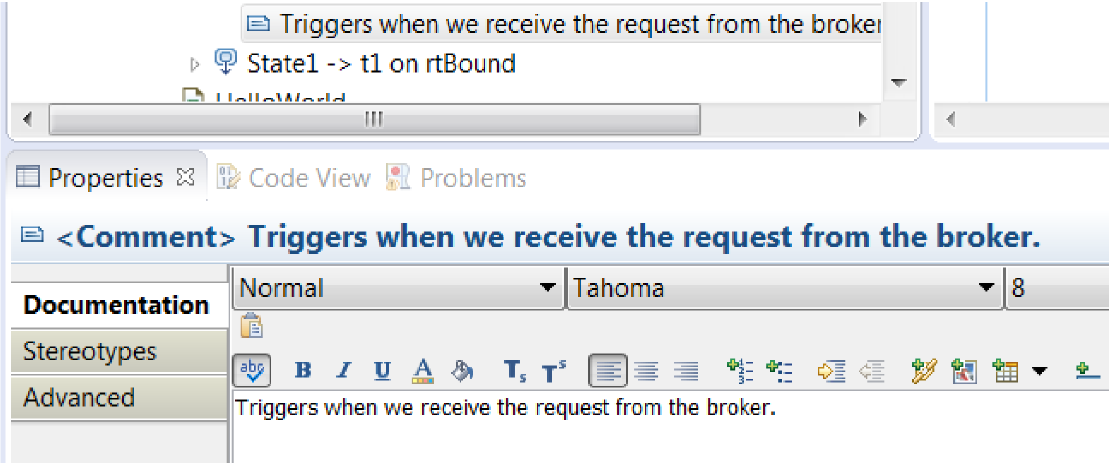
<Class> «Capsule» CPPModel::BCap

General
Ports
Dependencies
Attributes

Signature
aCount : int
indexer : void*
verProceed : Registry

Documentation Tab for Comments

- When a comment is selected in the Project Explorer, the Properties view now by default shows the Documentation tab where the comment can be viewed and edited
- The change compared to previously is that the Documentation tab now comes before the Stereotypes tab. This is reasonable for all kinds of elements since it’s more common to add a comment to an element than to apply a stereotype on it.

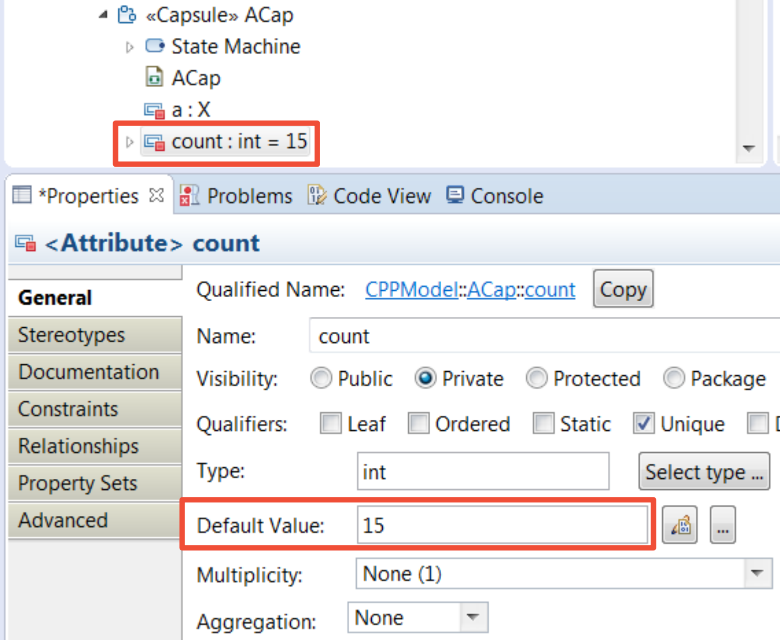


<Comment> Triggers when we receive the request from the broker.

Triggers when we receive the request from the broker.

Default Value Preserved when Changing Type

- Previously if changing the type of e.g. an attribute, the default value would be removed. The reason was that RSARTE could not verify that the default value would be compatible with the new type.
- Now the default value is preserved, and it’s the user’s responsibility to ensure that it is compatible with the new type.



<Attribute> count

Qualified Name: CPPModel::ACap::count

Name: count

Visibility: ○ Public ● Private ○ Protected ○ Package

Qualifiers: □ Leaf □ Ordered □ Static ☑ Unique □ I

Type: int

Default Value: 15

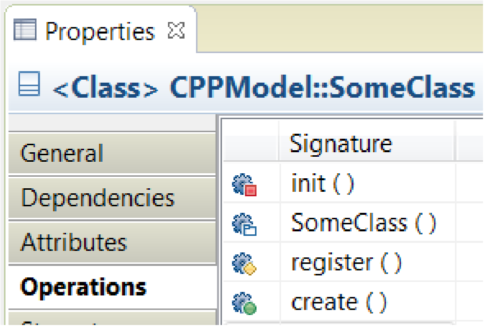
Multiplicity: None (1)

Aggregation: None

Support for a More C++ Like Textual Syntax

- A new preference is available: *UML Development – Real Time – Textual syntax*
- When set to ”C++” RSARTE supports a more C++ like textual syntax when creating and editing elements by typing text in diagrams, Project Explorer and Properties view
- Currently this preference affects the syntax used when creating operations
- UML textual syntax supports specifying operation visibility using a prefix character:

-init	Private visibility
~SomeClass	Package visibility
#register	Protected visibility
+create	Public visibility



<Class> CPPModel::SomeClass

Signature
init ()
SomeClass ()
register ()
create ()

- The C++ textual syntax is closer to standard C++ and does not recognize the special prefix characters. The prefix ~ can therefore be used for defining a destructor operation.
- Most RSARTE users should benefit from setting this preference to use the C++ like syntax, and our goal for the future is to make it identical to standard C++ syntax

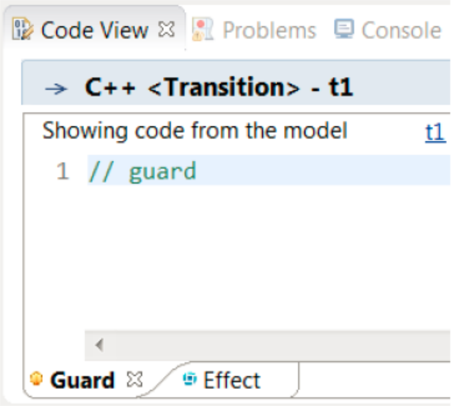
Increased Undo History

- Previously the Undo history for model-level commands was rather small. Only the 20 most recent changes to the model could be undone.
- It was easy to hit this limit, especially when working in diagrams
- The Undo stack is now increased to 200 (same as default Undo history size for text editing)

Fixed Order of Tabs in Code View

- By default Code View tabs are ordered so that non-empty tabs come before empty tabs
- However, there is now a preference which can be used if you prefer that the order of tabs are always the same, regardless of if the tabs contain any code or not
- The new preference is *Code View – Fixed Order of Tabs*
- Setting this preference can be helpful if you notice that you often happen to type code in the wrong tab when working with the Code View
- Example (for transition):

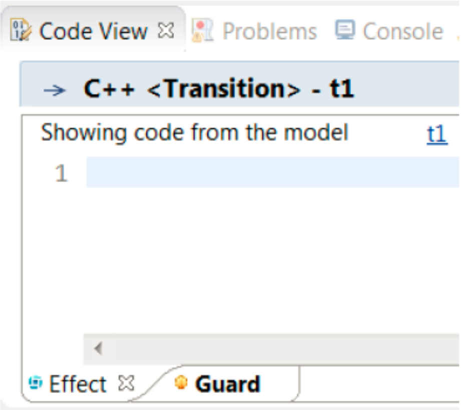
preference not set (default)



Showing code from the model t1

```
// guard
```

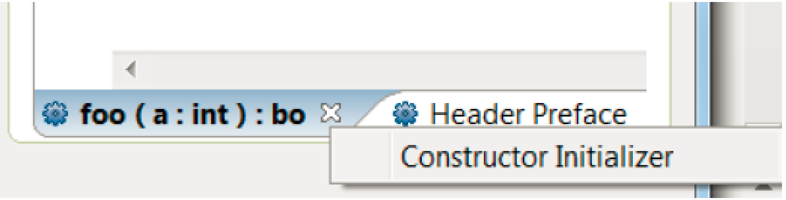
preference set



Showing code from the model t1

Code View Hides Irrelevant Code Snippets

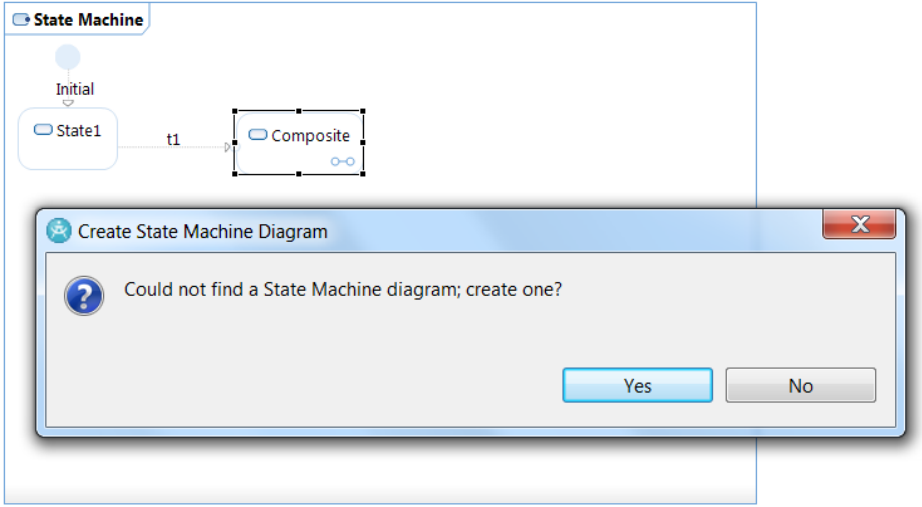
- The Code View now takes into account various properties of the edited element to hide code snippets that are irrelevant
  - **Constructor Initializer** only shown for constructor operations
  - **Public/Protected/Private Declarations** not shown for typedef classes
  - Type descriptor code snippets not shown for passive classes for which type descriptor generation has been disabled
  - **Header Preface/Ending** only shown for classes where a header file is generated
  - **Implementation Preface/Ending** only shown for classes where an implementation file is generated
- This reduces the number of Code view tabs that appears for a selected element.
- It's still possible to open hidden code snippets by using the context menu:
- Note: A non-empty code snippet is always considered relevant and is never hidden.



Navigation to Sub State Machine Improved

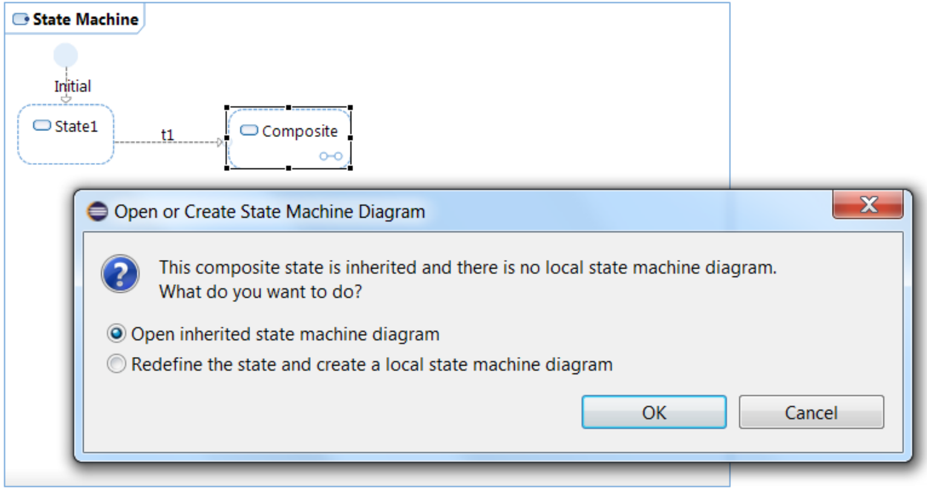
- It is now possible to navigate directly from a state symbol to an inherited state machine diagram

before



Could not find a State Machine diagram; create one?

now

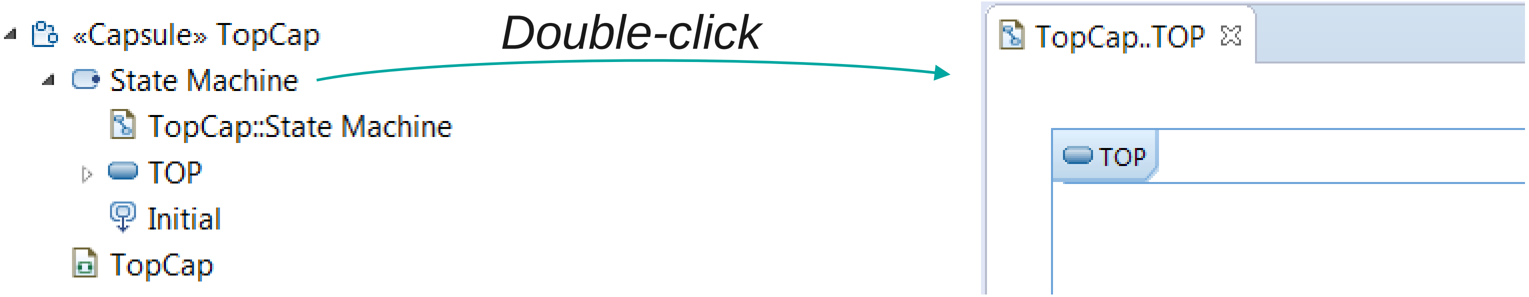


This composite state is inherited and there is no local state machine diagram.
What do you want to do?

● Open inherited state machine diagram
○ Redefine the state and create a local state machine diagram

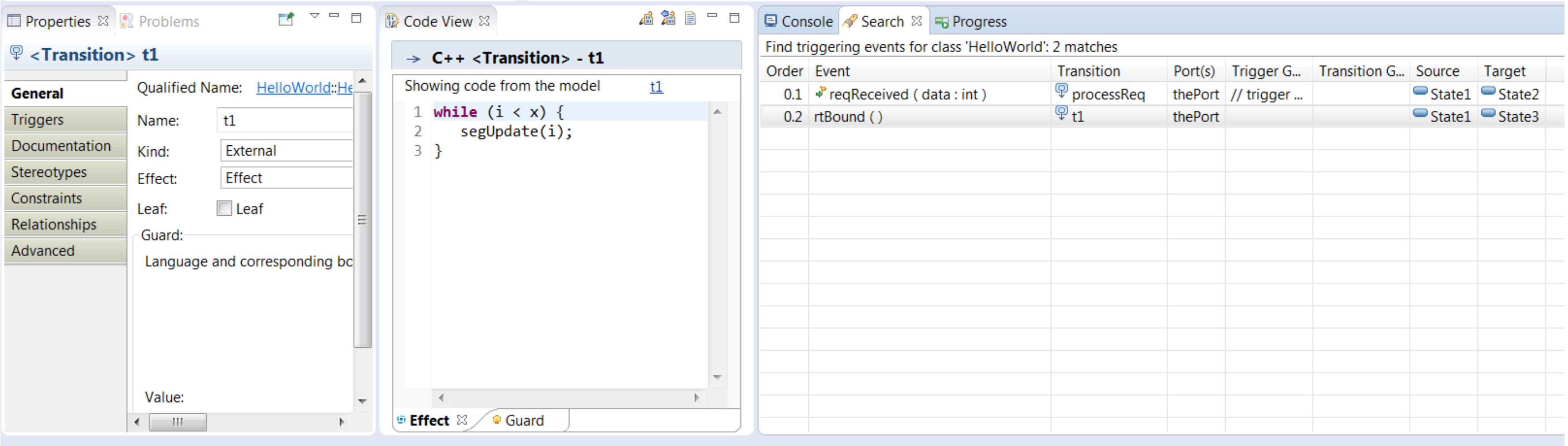
Navigate into TOP State Automatically

- This relates to the preference *UML Development – Real Time – Diagrams – State Chart – Let Open State Machine Diagram go inside TOP states automatically*
- When this preference is set, it’s now possible to navigate into the TOP state by double-clicking on the capsule state machine in the Project Explorer


Double-click

Find Commands Refresh Code and Properties Views

- When using the Find commands (Find Triggers, Find Triggering Events etc.) it’s now possible to just select an item in the Search view to see the properties and code snippets of the corresponding element
- This is more convenient than first having to navigate to the element in the Project Explorer



Qualified Name: HelloWorld::He

Name: t1

Kind: External

Effect: Effect

Leaf: ☐ Leaf

Guard:

Language and corresponding bo

Value:

Showing code from the model t1

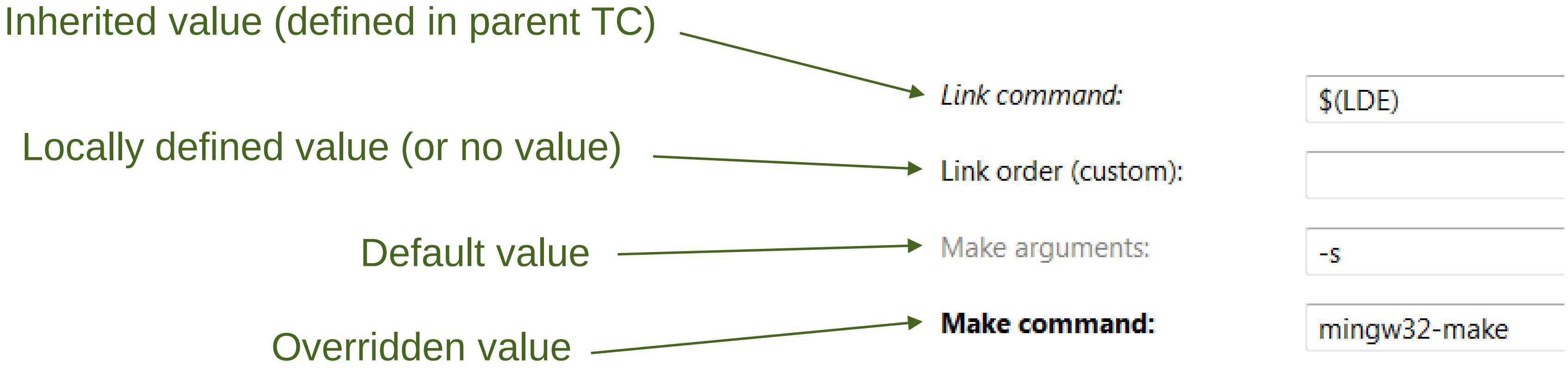
```
while (i < x) {
   segUpdate(i);
}
```

Find triggering events for class 'HelloWorld': 2 matches

Order	Event	Transition	Port(s)	Trigger G...	Transition G...	Source	Target
0.1	reqReceived (data : int)	processReq	thePort	// trigger ...		State1	State2
0.2	rtBound ( )	t1	thePort			State1	State3

Visual Indication of Values in a TC

- The TC editor now provides a visual indication to tell if a value is locally defined, inherited, overridden or if it is a default value
- This gives users a better understanding of where the value is defined, which for example is useful when values are inherited



Inherited value (defined in parent TC)

Locally defined value (or no value)

Default value

Overridden value

Link command: $(LDE)

Link order (custom):

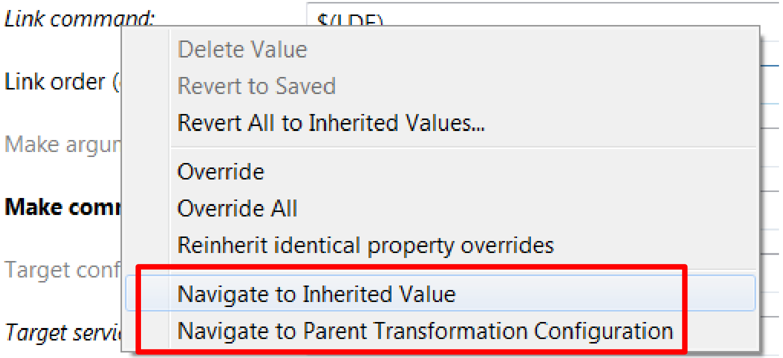
Make arguments: -s

Make command: mingw32-make

Navigation to Inherited Values in the TC Editor

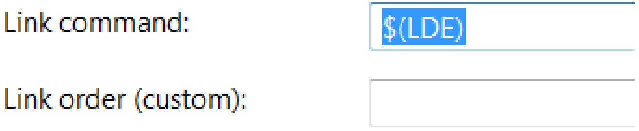
- When a TC inherits values for settings from an inherited TC, it is now possible to navigate to the value in that inherited TC.
- This is useful in order to find out from where a certain TC setting gets its value.

in child TC



Link command:
$(LDE)
Link order (
Make argur
Make comr
Target conf
Target servi
Delete Value
Revert to Saved
Revert All to Inherited Values...
Override
Override All
Reinherit identical property overrides
Navigate to Inherited Value
Navigate to Parent Transformation Configuration

in parent TC



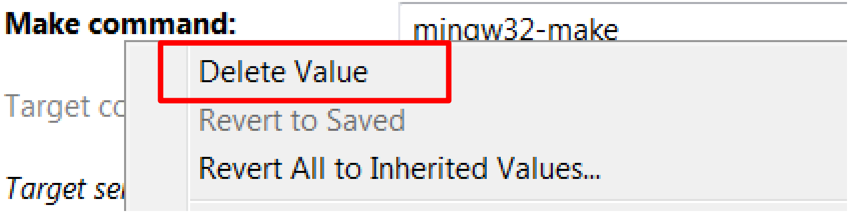
Link command: $(LDE)

Link order (custom):

- Note: Sometimes a value is not inherited but is the default value of the TC setting. In that case ”Navigate to Inherited Value” will be disabled, but you can still navigate to the parent TC using the command ”Navigate to Parent Transformation Configuration”.

Deleting Values in the TC Editor

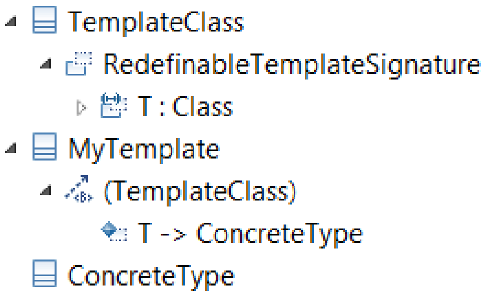
- Deleting a string value by just erasing the text makes the value an empty string
- Now it’s possible to actually delete the value from the TC
- This is useful for example when using multiple TC inheritance, to ensure that a value is inherited from a specific parent TC
- A new ”Delete Value” command is available in the TC editor context menu, and it has replaced the need for the old ”Revert to Inherited Value” command
- ”Delete Value” can also be used as a convenient way to delete a value from a TC even when no inheritance is involved



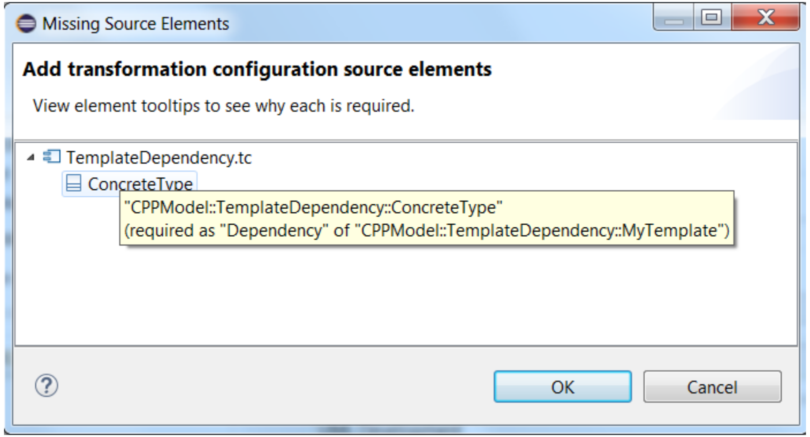
Make command: mingw32-make

Improved Support for Template Parameters

- References to types in actual template parameters are now taken into account when using the ”Detect Source Dependencies Automatically” feature
- Such references are also supported in other places where references are analyzed, such as the ”Organize Sources” and ”Analyze model and add missing dependencies” buttons, and in the ”Missing Source Elements” dialog



- TemplateClass
  - RedefinableTemplateSignature
    - T : Class
- MyTemplate
  - (TemplateClass)
    - T -> ConcreteType
- ConcreteType



Missing Source Elements

Add transformation configuration source elements

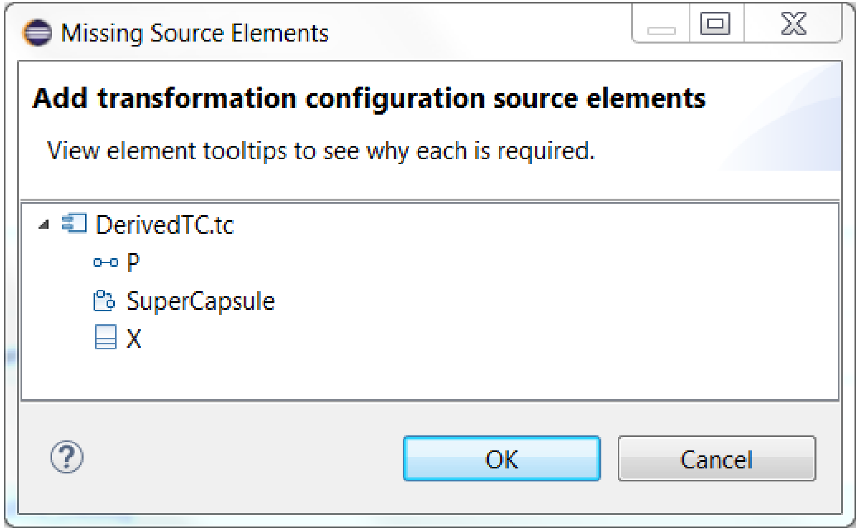
View element tooltips to see why each is required.

- TemplateDependency.tc
  - ConcreteType

"CPPModel::TemplateDependency::ConcreteType"
(required as "Dependency" of "CPPModel::TemplateDependency::MyTemplate")

Missing Source Elements when Building Inherited TCs

- When a TC inherits its Sources list, a build of that TC would previously update its own Sources list, if missing source elements were detected. Hence the Sources list would be overridden in the built TC, which usually is not wanted.
- Now the Sources list in the parent TC is updated instead
- Note that the ”Missing Source Elements” dialog still groups missing elements under the TC that is built (i.e. not the TC where missing elements will be added)



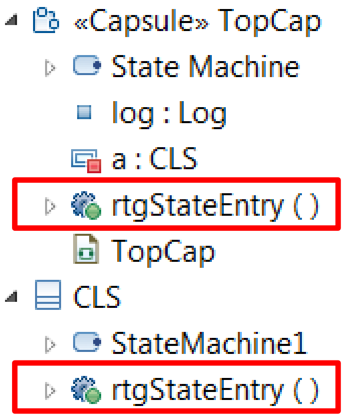
Missing Source Elements

Add transformation configuration source elements

View element tooltips to see why each is required.

State Entry Operation on Class/Capsule Level

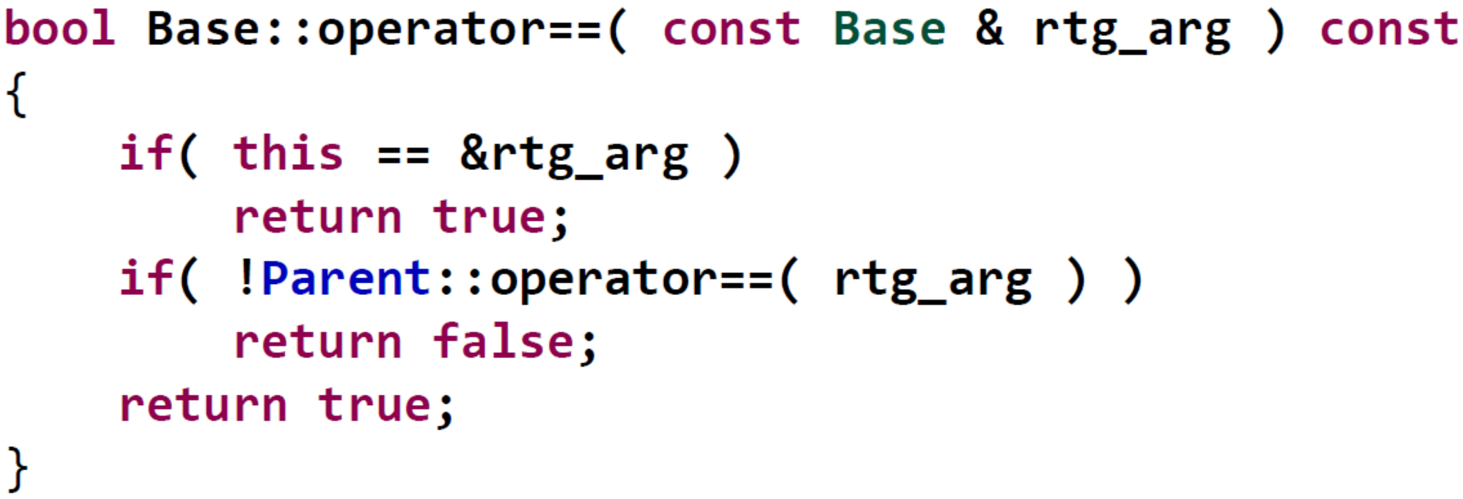
- Code that should execute whenever a new state becomes active in a state machine can now be placed in an operation called **rtgStateEntry()**.
- This is supported both for capsules and passive classes with state machines
- For capsules, this operation should be polymorphic (i.e. virtual)



```
◢ «Capsule» TopCap
    ▷ State Machine
      log : Log
      a : CLS
    ▷ rtgStateEntry ( )
      TopCap
◢ CLS
    ▷ StateMachine1
    ▷ rtgStateEntry ( )
```

Avoid Lint Warning for Equality Operator

- We have fixed an issue when using Lint to check applications generated by RSARTE
- Previously warnings was generated for compatibility between 0/1 and boolean false/true
- The code that is generated now is the following where true/false is returned as expected.



```
bool Base::operator==( const Base & rtg_arg ) const
{
    if( this == &rtg_arg )
        return true;
    if( !Parent::operator==( rtg_arg ) )
        return false;
    return true;
}
```

Removed Checks for Minimum Port Cardinality

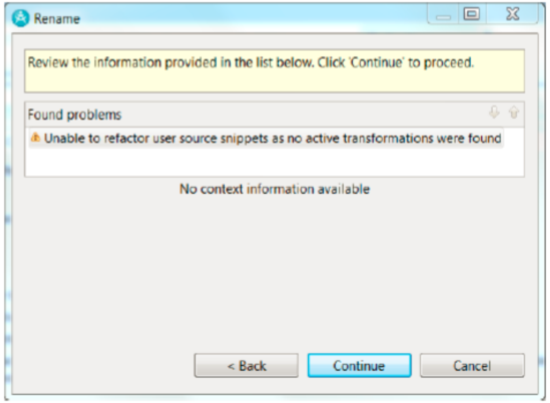
- For UML-RT style models the minimum port cardinality will not have any effect on the generated code. In previous versions some constraints were checked that caused confusion since everything worked even if the constrains were not satisfied.
- As part of an effort to simplify the usage of RSARTE for C++ development these checks have been removed.
- The removed constraints had the following messages:

In context of capsule "AAA", the port has minimum multiplicity XX, but is connected with other ports: BBB with total maximum multiplicity only YY.

In context of capsule "AAA", "BBB" end of this connector has minimum multiplicity XX, but "CCC' end has maximum multiplicity only YY.

Setting Active TC while Refactoring

- Previously a refactoring with support for updating code snippets could not be run unless an active TC had been set:



Rename

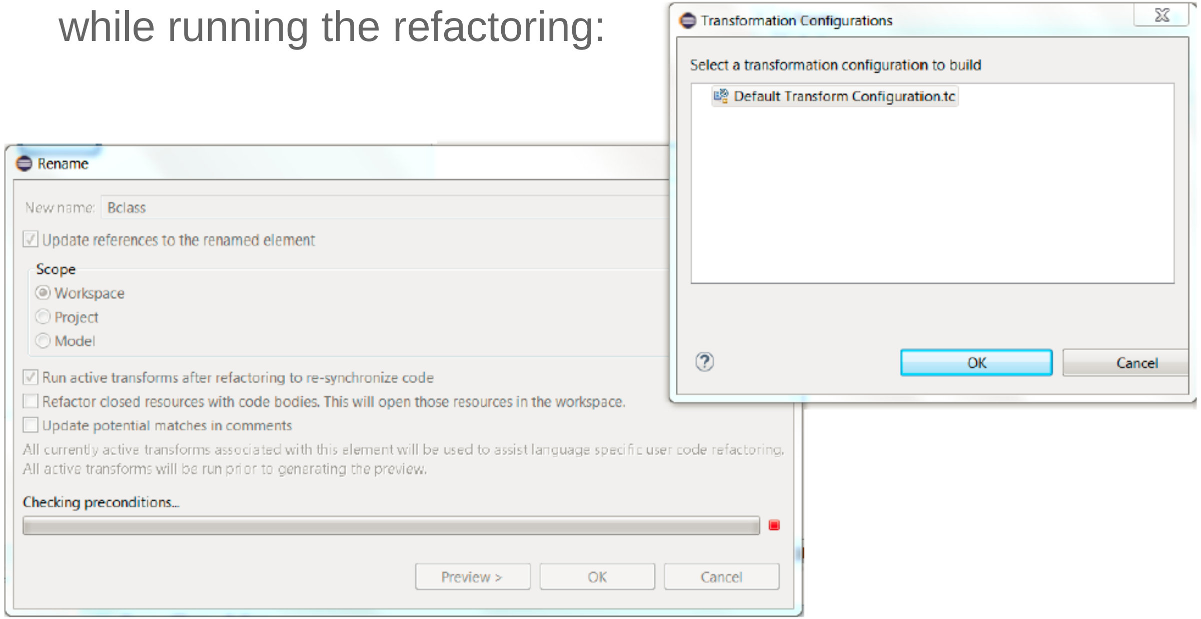
Review the information provided in the list below. Click 'Continue' to proceed.

Found problems

⚠ Unable to refactor user source snippets as no active transformations were found

No context information available

- This has now been improved so that an active TC can be set while running the refactoring:



Transformation Configurations

Select a transformation configuration to build

Default Transform Configuration.tc

Rename

New name: Bclass

☑ Update references to the renamed element

Scope

● Workspace

○ Project

○ Model

☑ Run active transforms after refactoring to re-synchronize code

□ Refactor closed resources with code bodies. This will open those resources in the workspace.

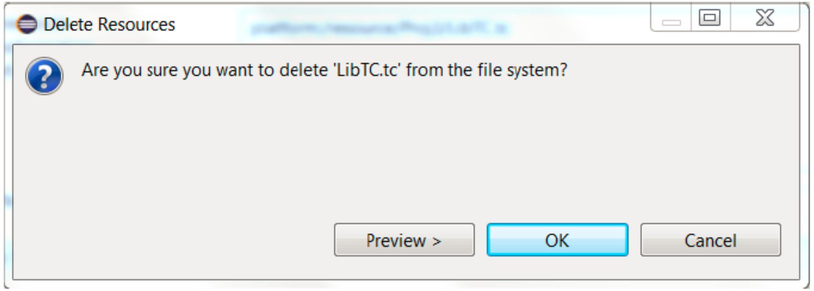
□ Update potential matches in comments

All currently active transforms associated with this element will be used to assist language specific user code refactoring. All active transforms will be run prior to generating the preview.

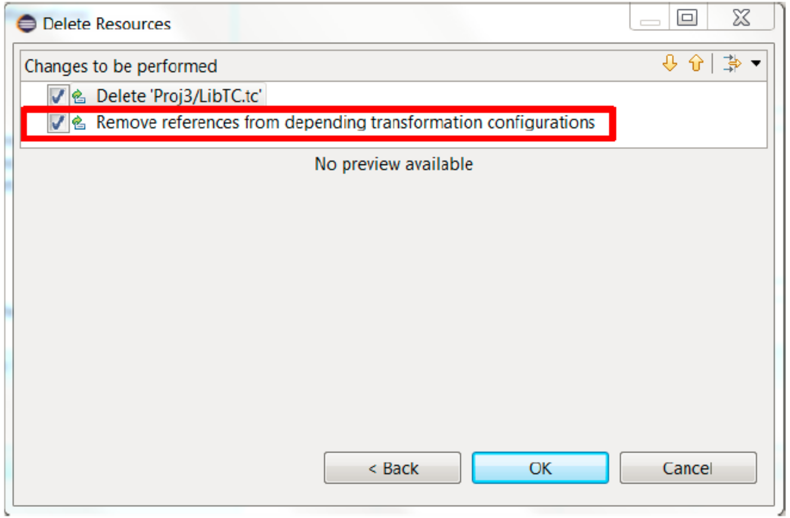
Checking preconditions...

Deleting a TC without Changing Referencing TCs

- When a TC is deleted you can choose whether you want to remove references to the deleted TC from other TCs in the workspace
- By default such references will be removed, but there is a checkbox in the Preview dialog which can be unchecked to keep them



Are you sure you want to delete 'LibTC.tc' from the file system?



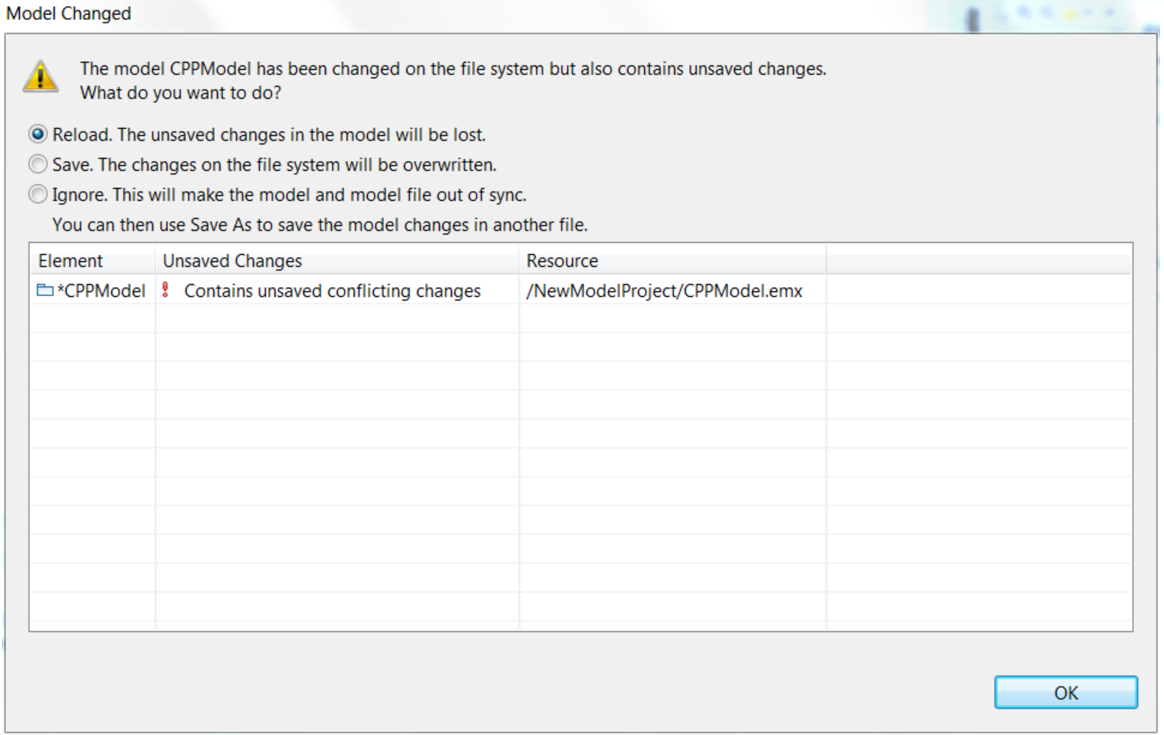
☑ Delete 'Proj3/LibTC.tc'

☑ Remove references from depending transformation configurations

No preview available

Improved “Model Changed” Dialog

- The previous behavior of the “Model Changed” dialog was confusing and did not provide sufficiently clear instructions on how to resolve the situation when an unsaved model was also changed on the file system.
- The dialog has now been improved and now provides clear options for what can be done when this happens.



Model Changed

The model CPPModel has been changed on the file system but also contains unsaved changes.
What do you want to do?

● Reload. The unsaved changes in the model will be lost.

○ Save. The changes on the file system will be overwritten.

○ Ignore. This will make the model and model file out of sync.
You can then use Save As to save the model changes in another file.

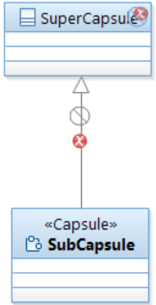
Element	Unsaved Changes	Resource
*CPPModel	! Contains unsaved conflicting changes	/NewModelProject/CPPModel.emx

Model Compiler

- A new command-line tool which can be used for building RSARTE models
  - It is independent of Eclipse (runs outside of RSARTE).
  - It can generate make files which completely drives the build of an RSARTE model (including both code generation, compilation and linking).
  - Generated code is the same as before (only C++ code is currently supported, but C code will be supported in the future)
- For more information read the documentation that is available on DeveloperWorks.
- This is currently an experimental feature. It is possible to use the new Model Compiler when building a TC in the RSARTE UI if enabling the experimental preference *UML Development – Model Compiler – Build models using the Model Compiler*.

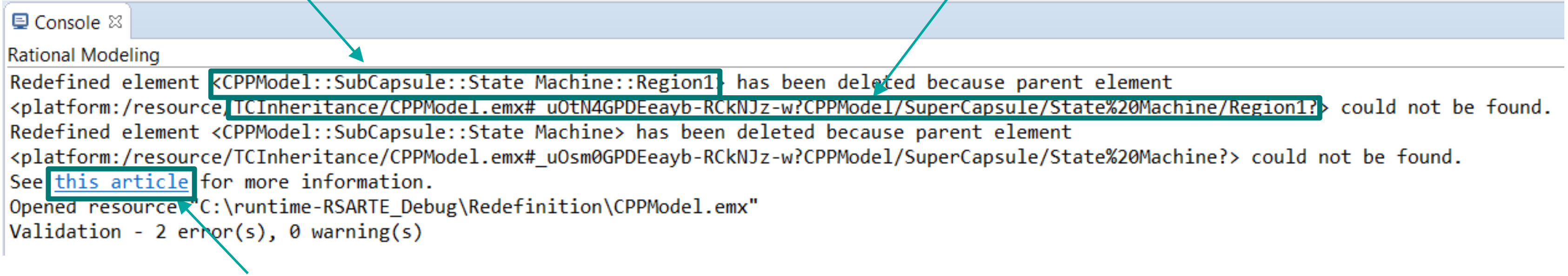
Broken References that Lead to Implicit Deletions

- RSARTE does not allow certain model references to be broken (for example state machine generalization and redefinition references).
- By default RSARTE deletes the element that contains the broken reference
- You are now alerted by a Console view printout when this happens, so you can take actions to resolve the broken references



Fully qualified name of the deleted element

URI of missing element (helps to know how to resolve the broken reference)

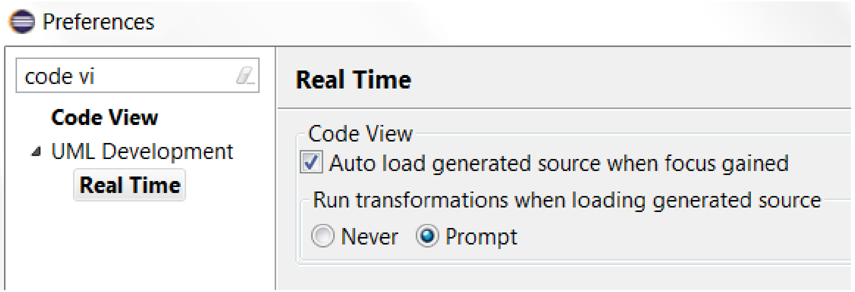


```
Rational Modeling
Redefined element <CPPModel::SubCapsule::State Machine::Region1> has been deleted because parent element
<platform:/resource/TCInheritance/CPPModel.emx#_uOtN4GPDEeayb-RCkNJz-w?CPPModel/SuperCapsule/State%20Machine/Region1?> could not be found.
Redefined element <CPPModel::SubCapsule::State Machine> has been deleted because parent element
<platform:/resource/TCInheritance/CPPModel.emx#_uOsm0GPDEeayb-RCkNJz-w?CPPModel/SuperCapsule/State%20Machine?> could not be found.
See this article for more information.
Opened resource "C:\runtime-RSARTE_Debug\Redefinition\CPPModel.emx"
Validation - 2 error(s), 0 warning(s)
```

An article is available on DeveloperWorks with more information, including the recommended steps to take when this happens

Filtering the Preferences Dialog using Keywords

- Previously the Preferences dialog could only be filtered by the title of available preferences pages
- Now filtering is also supported using certain keywords (corresponding to individual preferences that are commonly used)



Real Time

Code View

☑ Auto load generated source when focus gained

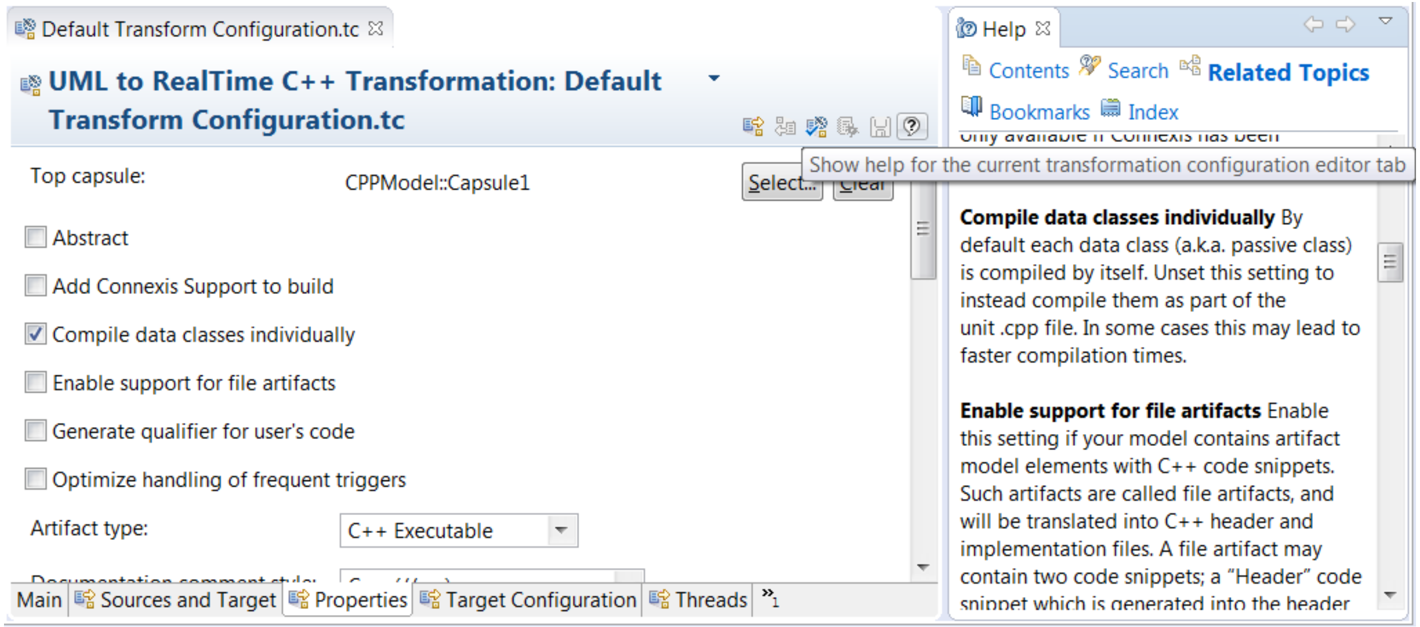
Run transformations when loading generated source

○ Never ● Prompt

For example, the ”Real Time” preference page contains a group of preferences called ”Code View”. It’s now possible to find these by filtering the Preferences dialog using the keyword ”code view”.

Improved Documentation within RSARTE

- Context sensitive help is now available for all tabs in the TC editor
  - All settings for C++ TCs are documented
  - A new toolbar button is available for opening context sensitive help


Default Transform Configuration.tc
UML to RealTime C++ Transformation: Default Transform Configuration.tc
Select...
Clear
Main
Sources and Target
Properties
Target Configuration
Threads
Help
Contents
Search
Related Topics
Bookmarks
Index
Show help for the current transformation configuration editor tab

Top capsule: CPPModel::Capsule1

□ Abstract

□ Add Connexis Support to build

☑ Compile data classes individually

□ Enable support for file artifacts

□ Generate qualifier for user's code

□ Optimize handling of frequent triggers

Artifact type: C++ Executable

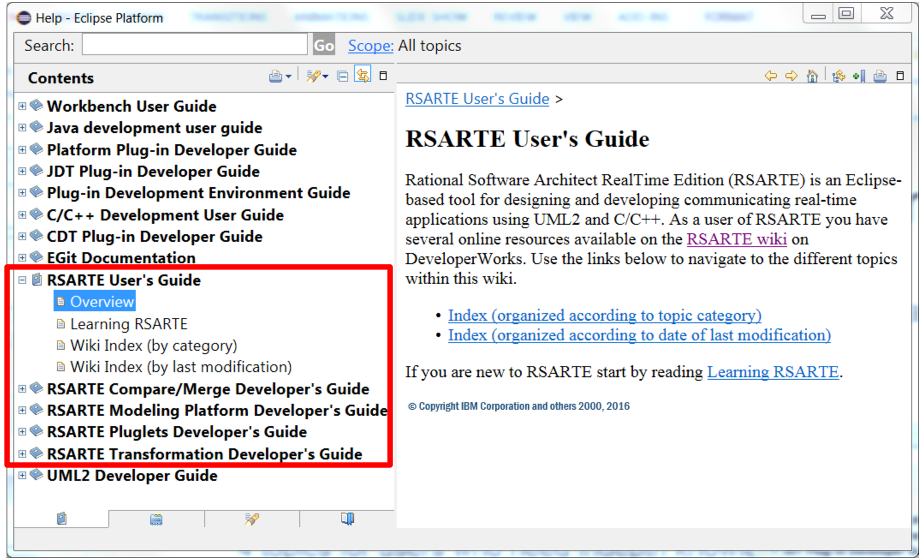
Only available if Connexis has been

Compile data classes individually By default each data class (a.k.a. passive class) is compiled by itself. Unset this setting to instead compile them as part of the unit .cpp file. In some cases this may lead to faster compilation times.

Enable support for file artifacts Enable this setting if your model contains artifact model elements with C++ code snippets. Such artifacts are called file artifacts, and will be translated into C++ header and implementation files. A file artifact may contain two code snippets; a "Header" code snippet which is generated into the header

Improved Documentation within RSARTE

- The built-in help (sometimes called ”online help”) was extended. It now contains 5 topics (all starting with ”RSARTE”):
  - **RSARTE User’s Guide**
    Provides index pages for navigating to the RSARTE wiki, as well as a page intended for new users that want to learn using RSARTE
  - **RSARTE Developer’s Guides**
    4 topics for users who need indepth knowledge of RSARTE, for example to extend it using plugins (includes JavaDoc for public APIs and available extension points)



- Workbench User Guide
- Java development user guide
- Platform Plug-in Developer Guide
- JDT Plug-in Developer Guide
- Plug-in Development Environment Guide
- C/C++ Development User Guide
- CDT Plug-in Developer Guide
- EGit Documentation
- RSARTE User's Guide
  - Overview
  - Learning RSARTE
  - Wiki Index (by category)
  - Wiki Index (by last modification)
- RSARTE Compare/Merge Developer's Guide
- RSARTE Modeling Platform Developer's Guide
- RSARTE Pluglets Developer's Guide
- RSARTE Transformation Developer's Guide
- UML2 Developer Guide

RSARTE User's Guide >

RSARTE User's Guide

Rational Software Architect RealTime Edition (RSARTE) is an Eclipse-based tool for designing and developing communicating real-time applications using UML2 and C/C++. As a user of RSARTE you have several online resources available on the RSARTE wiki on DeveloperWorks. Use the links below to navigate to the different topics within this wiki.

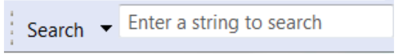
- Index (organized according to topic category)
- Index (organized according to date of last modification)

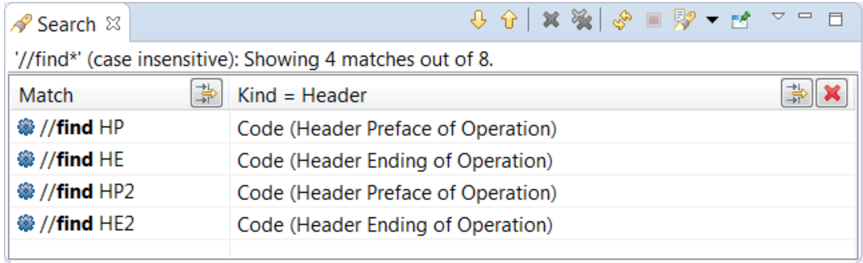
If you are new to RSARTE start by reading Learning RSARTE.

© Copyright IBM Corporation and others 2000, 2016

New Search

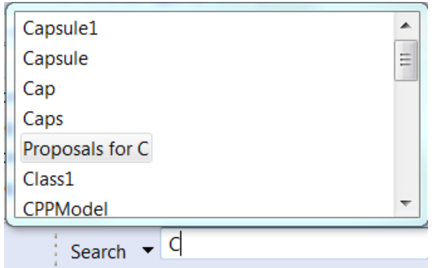
- Highlights of the new search implementation
  - Search is always available from the search field. No need to open a modal dialog and setting options before searching.
  - The model is automatically indexed in the background. No need to wait for the index to be synchronized when searching.
  - The search index is not stored in the workspace. Changing workspaces therefore do not impact performance of search.
  - The search index is distributed per project. It’s therefore possible to commit the search index for models that are infrequently modified to a CM system, such as Git.
  - The search result can be filtered using regular expressions (each column can be filtered separately)
  - RT concepts such as protocols and capsules are presented in the same way as in the Project Explorer
  - Indexing from the command-line is supported. This makes it possible to pre-index a workspace before it is opened.
  - The search result is a flat list (rather than a tree)
  - The items in the search result are ordered so that the best match appears first
  - Information in TC files can be searched, and navigation highlights the relevant part of the TC editor
  - The search field provides a proposals popup with recently and frequently used search strings
  - Libraries are indexed to allow references to library elements to be found
  - Possible to invoke ”Replace” on either all or some of the found elements





'//find*' (case insensitive): Showing 4 matches out of 8.

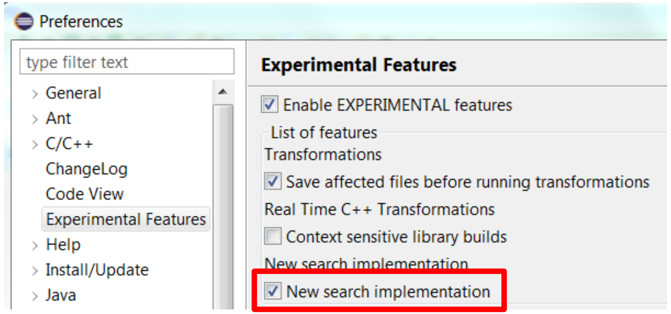
Match	Kind = Header
//**find** HP	Code (Header Preface of Operation)
//**find** HE	Code (Header Ending of Operation)
//**find** HP2	Code (Header Preface of Operation)
//**find** HE2	Code (Header Ending of Operation)



New Search

- The new search capability is an **experimental** feature. You are invited to try it out, but keep in mind its current limitations:
  - It’s necessary to restart RSARTE after enabling the experimental feature.
  - The new search implementation is only available from the new search field. There are no integrations with the other search commands yet.
  - It's not possible to limit search scope to anything except workspace and selected project.
  - Searching for exact whole-word matches is not currently supported (however, whole-word matches are placed before other matches in the search result)

  Also remember that it’s not always possible to directly compare the results of using the new and old search implementations since the new search uses a different strategy for presenting and organizing the search result.



Experimental Features

☑ Enable EXPERIMENTAL features

List of features

Transformations

☑ Save affected files before running transformations

Real Time C++ Transformations

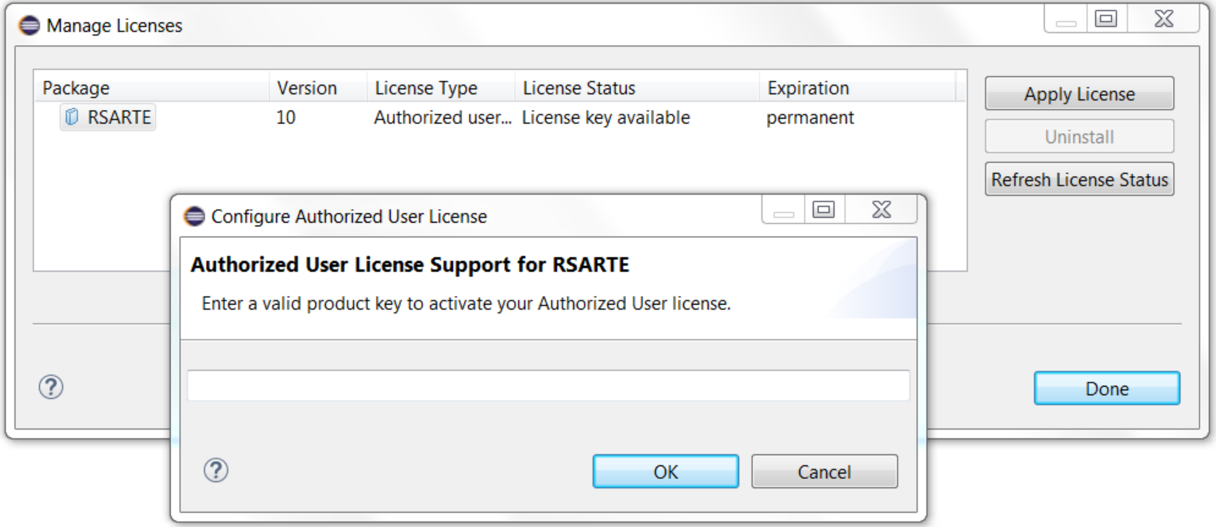
☐ Context sensitive library builds

New search implementation

☑ New search implementation

Licensing

- It’s now possible to use RSARTE without access to a license server, by means of ”Authorized User” licenses
- These are activated using license keys, and replace the ”Activation Kits” that were used in version 9.x
- Apply an ”authorized user” license in *Help – Manage Licenses – Apply License*
- To request an ”authorized user” license contact rsarte@se.ibm.com



Package	Version	License Type	License Status	Expiration
RSARTE	10	Authorized user...	License key available	permanent

Authorized User License Support for RSARTE

Enter a valid product key to activate your Authorized User license.

THANK YOU!