What’s New in HCL RTist 11.3

updated for release 2023.42

HCL SOFTWARE

Copyright © 2023 HCL Technologies Limited | www.hcltechsw.com

Overview

- RTist 11.3 is based on Eclipse 2022.06 (4.24)
- HCL RTist is 100% compatible with IBM RSARTE and all features in these two products are equivalent

HCL RTist
Version: 11.3.0.v20231023_1810
Release: 2023.42

(c) Copyright IBM Corporation 2004, 2016. All rights reserved.
(c) Copyright HCL Technologies Ltd. 2016, 2023. All rights reserved.
Visit https://RTist.hcldoc.com/help/topic/com.ibm.xtools.rsarte.webdoc/users-guide/overview.html

Copyright © 2023 HCL Technologies Limited | www.hcltechsw.com

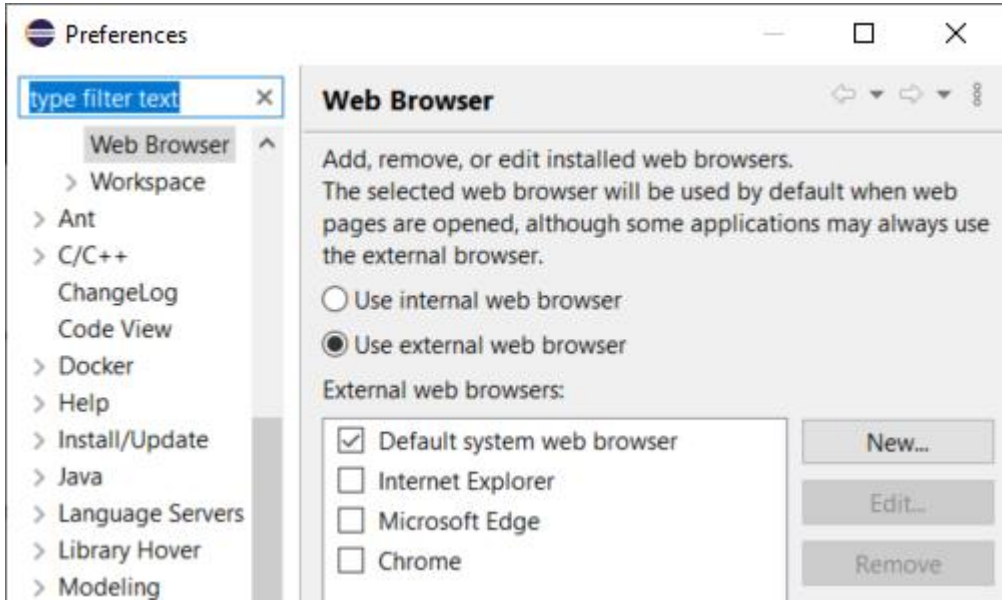
Eclipse 4.24 (2022.06)

- Compared to RTist 11.2, RTist 11.3 includes new features and bug fixes from 4 quarterly Eclipse releases:
  - 2021.09 (https://www.eclipse.org/eclipse/news/4.21/platform.php)
  - 2021.12 (https://www.eclipse.org/eclipse/news/4.22/platform.php)
  - 2022.03 (https://www.eclipse.org/eclipse/news/4.23/platform.php)
  - 2022.06 (https://www.eclipse.org/eclipse/news/4.24/platform.php)
- For full information about all improvements and changes in these Eclipse releases see the links above
  - Some highlights are listed in the next few slides...

Copyright © 2023 HCL Technologies Limited | www.hcltechsw.com

Eclipse 4.24 (2022.06)

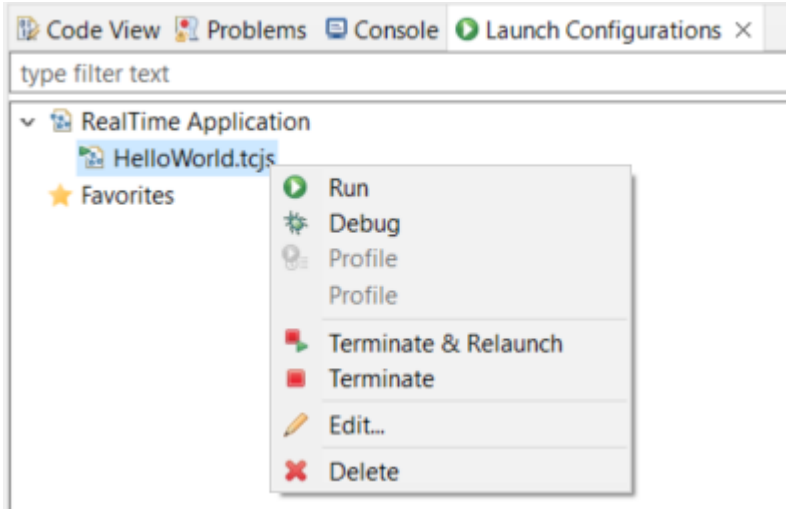
- Eclipse now by default uses the external web browser
  - The internal web browser has limitations and cannot show all web pages correctly
  - Usage of the external web browser is therefore recommended, and having it as the default makes it easier to get started with RTist without having to configure the preferences in **General - Web Browser**.



Copyright © 2023 HCL Technologies Limited | www.hcltechsw.com

Eclipse 4.24 (2022.06)

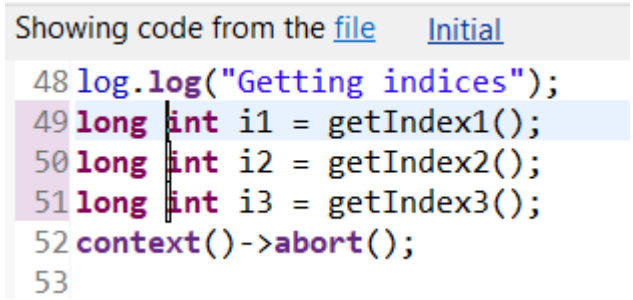
- A new Launch Configuration View makes it easier to, for example, launch model or C++ debug sessions
  - No need to first open the modal Launch Configuration dialog
  - Launch configurations will appear in the view automatically as they are created
  - Commonly useful commands for launching, terminating etc are available in the context menu
  - To start a model debug session, just double-click a "TC launch configuration" in that view

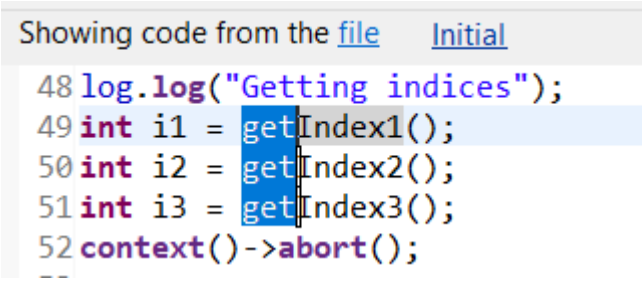


Copyright © 2023 HCL Technologies Limited | www.hcltechsw.com

Eclipse 4.24 (2022.06)

- Multiple text selections
  - You can now have multiple cursors in a text editor and make multiple selections
  - Add a new cursor by **Alt+click**
  - Multiple cursors can be useful when making the same change in multiple places in a file ("interactive find/replace")
  - Several new text editor commands are available for working with multiple text selections
    - For example: **Multi selection up/down relative to anchor selection** (for creating a multi-selection from a selected word)
    - Note: You need to assign a key binding to these commands to use them! Use the preference page **General - Keys** and search for commands with "multi" in their name in the "Text Editing" category.

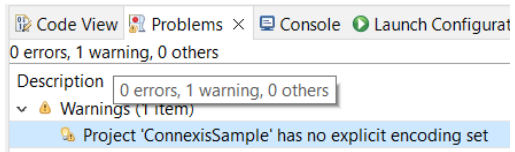




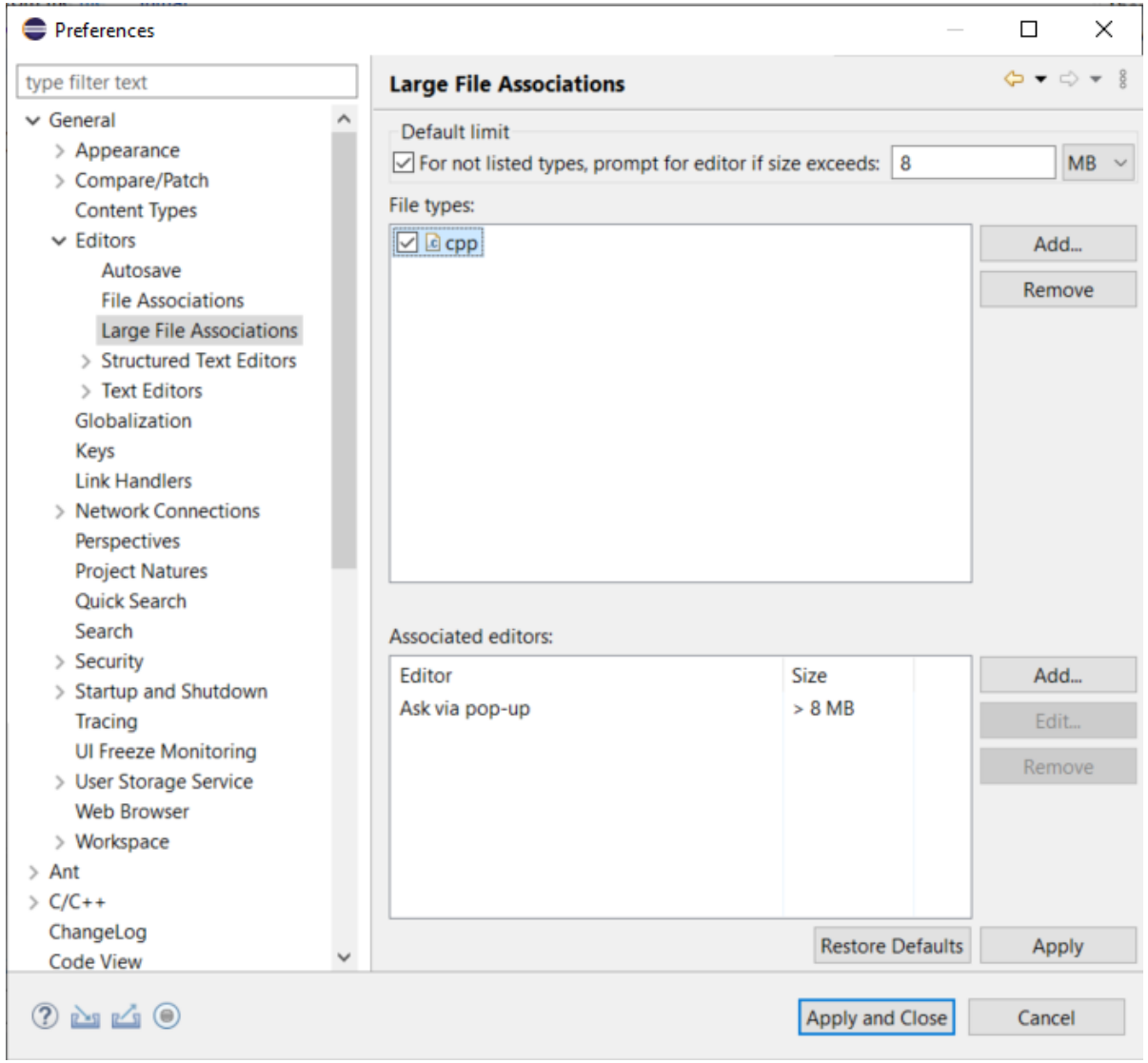
Copyright © 2023 HCL Technologies Limited | www.hcltechsw.com

Eclipse 4.24 (2022.06)

- Large file associations
  - A new preference page allows to specify special editors to use for large files: **General - Editors - Large File Associations**
  - Can help keeping a good performance in Eclipse even when opening large files
- Project encodings
  - Projects will now automatically get its encoding set to the workspace encoding (by default UTF-8) when they are created
  - For projects created in earlier versions of Eclipse a warning will appear



  - A Quick Fix is available for setting the project encoding to the workspace encoding

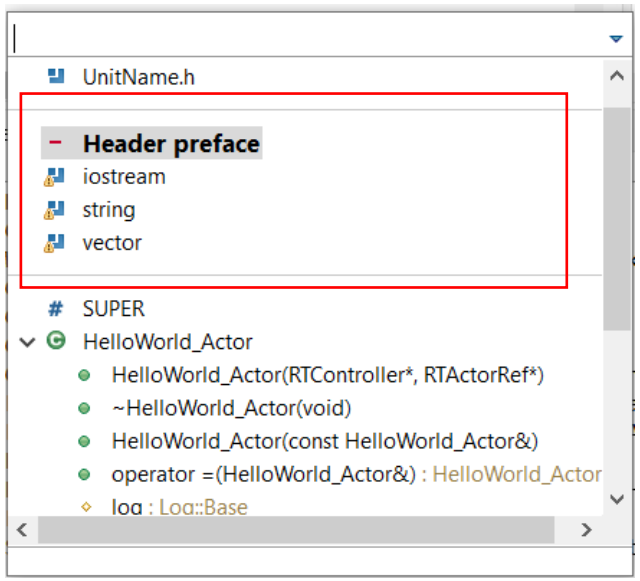


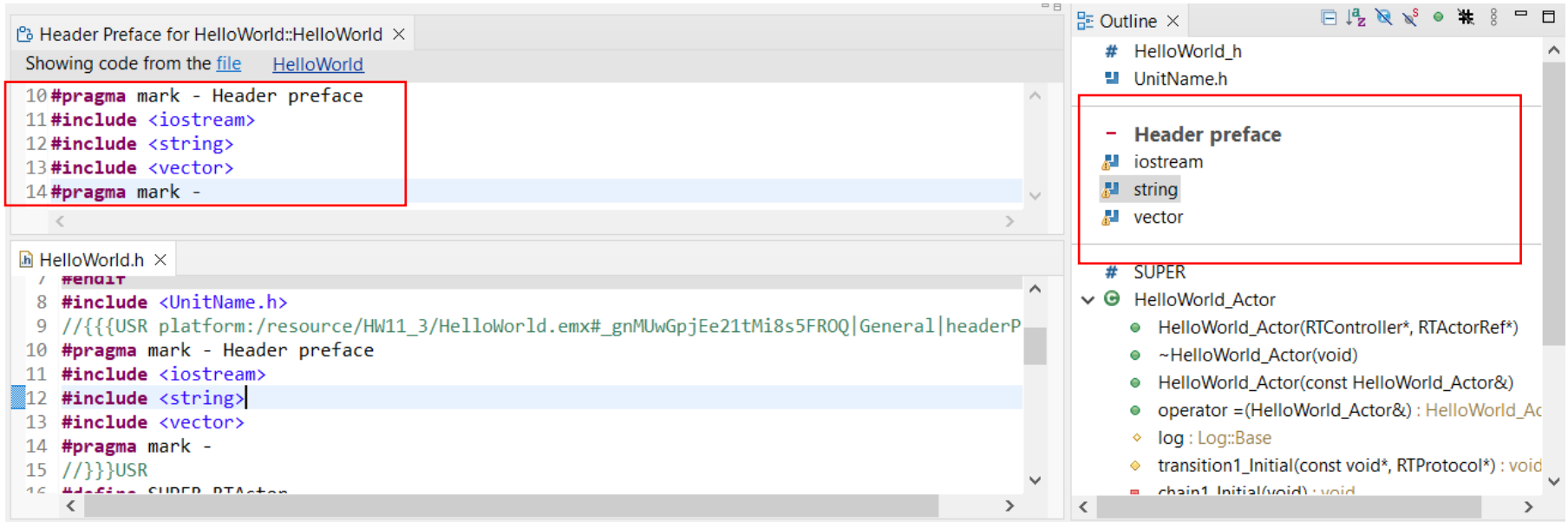
Copyright © 2023 HCL Technologies Limited | www.hcltechsw.com

HCL SOFTWARE

CDT 10.7 (included as part of Eclipse 2022.06)

- Separator lines in the Outline view
  - `#pragma mark` and `#pragma region` can be used for showing separator lines in the Outline view
  - Can help to more easily see and navigate to user code snippets in a generated C++ file
  - Automatically generating such separator lines for certain code snippets could be a future possibility...





Copyright © 2023 HCL Technologies Limited | www.hcltechsw.com

CDT 10.7 (included as part of Eclipse 2022.06)

- Improved code analysis for constexpr expressions
  - A number of GCC/Clang built-in functions can now be used without confusing the Code Analysis feature

- For more information about CDT improvements see
  https://wiki.eclipse.org/CDT/User/NewIn104
  https://wiki.eclipse.org/CDT/User/NewIn105
  https://wiki.eclipse.org/CDT/User/NewIn106
  https://wiki.eclipse.org/CDT/User/NewIn107

Copyright © 2023 HCL Technologies Limited | www.hcltechsw.com

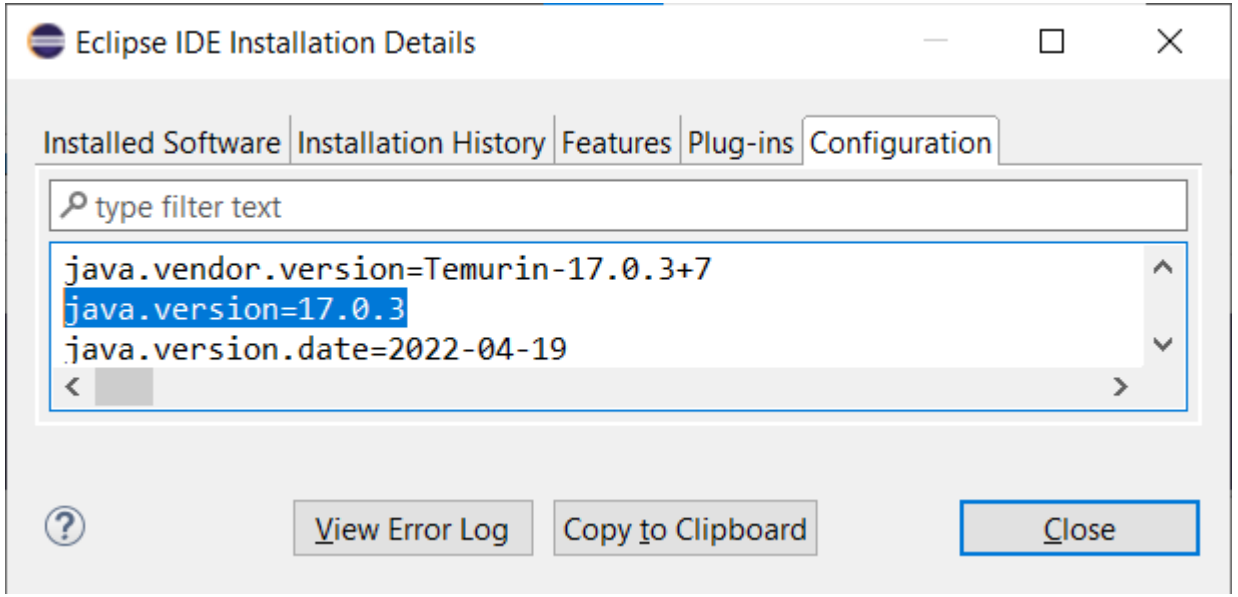
Newer EGit Version in the EGit Integration

- The EGit integration in RTist has upgraded EGit from 5.12 to 6.2
  - This is the recommended and latest version for Eclipse 2022.06
- This upgrade provides several new features and bug fixes
  - For detailed information about the changes see
    https://wiki.eclipse.org/EGit/New_and_Noteworthy/5.13
    https://wiki.eclipse.org/EGit/New_and_Noteworthy/6.0
    https://wiki.eclipse.org/EGit/New_and_Noteworthy/6.1
    https://wiki.eclipse.org/EGit/New_and_Noteworthy/6.2

Copyright © 2023 HCL Technologies Limited | www.hcltechsw.com

Java 17

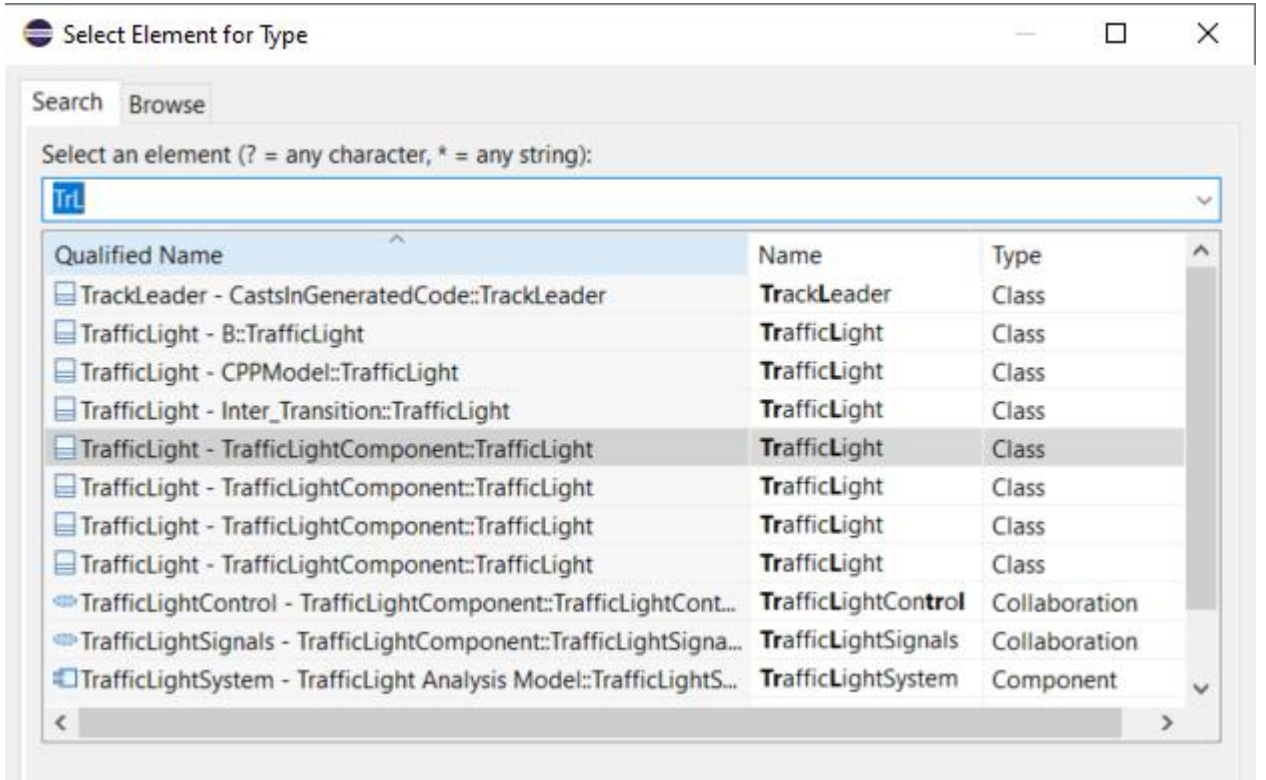
- RTist now should be run with a Java 17 JVM
  - Eclipse 2022.06 includes a Java 17 JVM which can be used. It's hence no longer necessary to update `eclipse.ini` to specify a different JVM for running RTist.
  - Refer to the System Requirements for more details
- Rebuild your plugins
  - If you have your own Eclipse plugins it's recommended to rebuild them with a Java 17 compiler before running them together with RTist 11.3.



Copyright © 2023 HCL Technologies Limited | www.hcltechsw.com

Improved Element Filtering in Views and Dialogs

- RTist provides several views and dialogs where a list of elements can be filtered by typing text in a filter box
  - Select Element dialog
  - External Projects Import view
  - Find NamedElement dialog
  - ...

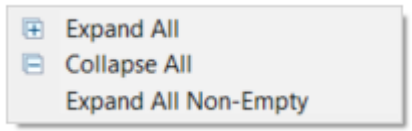


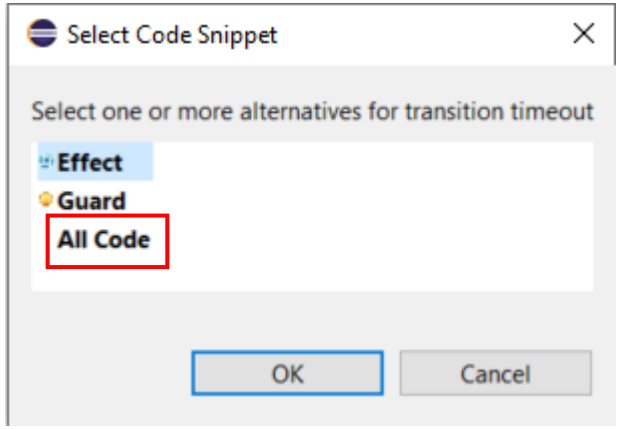
- These implementations are now harmonized
  - Fuzzy string matching is now supported. This helps when you only know approximately the name of the element you look for.
  - Columns in the Select Element dialog now allows sorting the elements on qualified name, name and type
  - The matched substring is marked with boldface

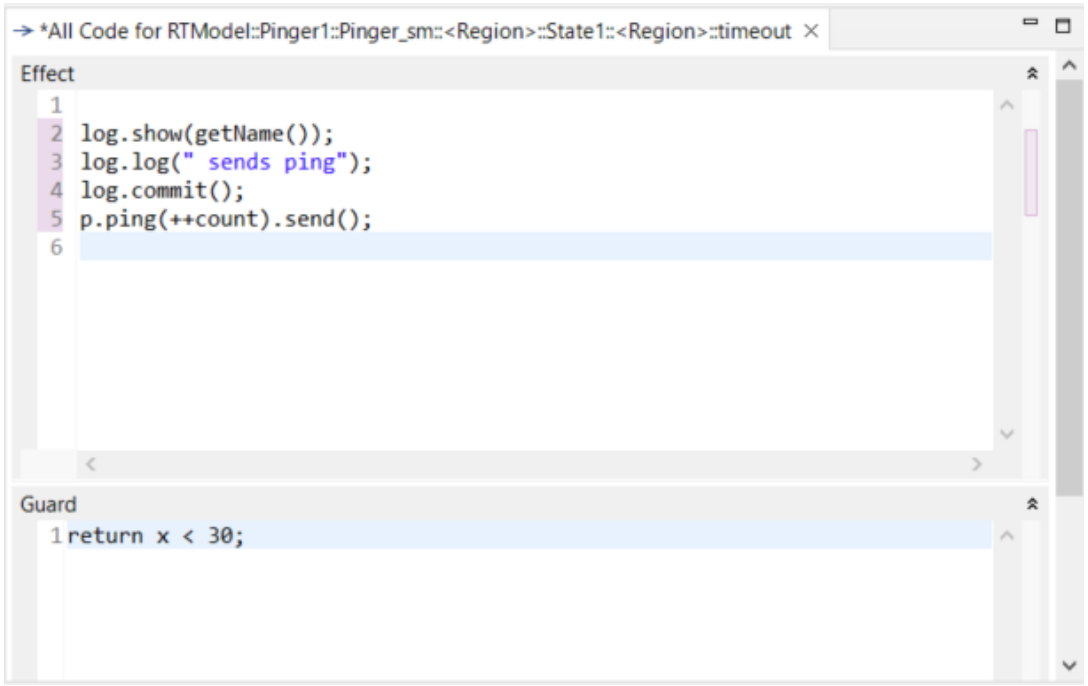
Copyright © 2023 HCL Technologies Limited | www.hcltechsw.com

Showing All Code Snippets for an Element in the Code Editor (1/2)

- You can now choose to show all code snippets for an element in a single Code editor
  - Significantly reduces the number of open code editors (e.g. a class has 14 different code snippets which now can be shown in a single code editor, instead of in 14 separate code editors)
  - Better overview by seeing related code snippets in a single place
  - Easier to copy/paste code between related code snippets
- Individual code snippets can be collapsed and expanded
  - By default empty code snippets are initially collapsed
  - The context menu on a code snippet heading (grey area) contains a few useful commands for collapsing/expanding code snippets:



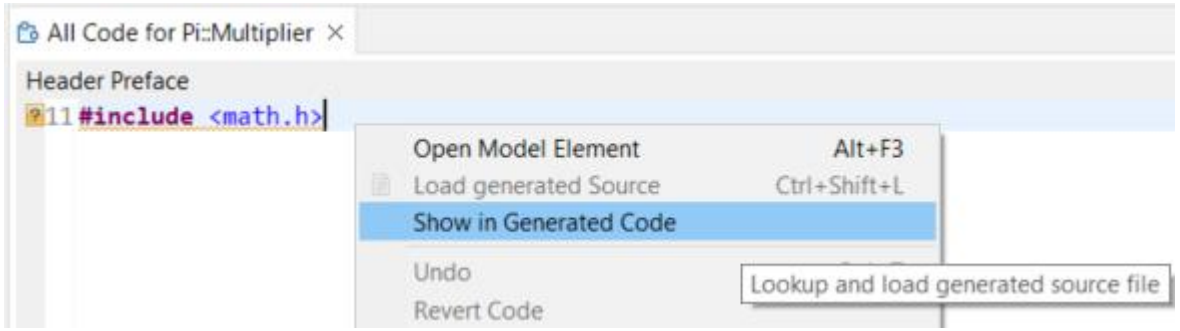




Copyright © 2023 HCL Technologies Limited | www.hcltechsw.com

Showing All Code Snippets for an Element in the Code Editor (2/2)

- Minimum height for each code snippet
  - A preference **RealTime Development - Code Editing - Code snippet minimum height in Code editor** can be set to avoid that a code snippet becomes too small for seeing at least some lines of code
  - By default it's 100 which corresponds to 4 lines of code (with default font size)
  - Raise the value (max 500) to see more lines of code in each code snippet (at the cost of more vertical scrolling)
- Navigate to generated C++ code
  - A context menu command **Show in Generated Code** is available
  - Works like the file hyperlink when the Code Editor shows a single code snippet
  - The command is disabled if generated code cannot be found and loaded



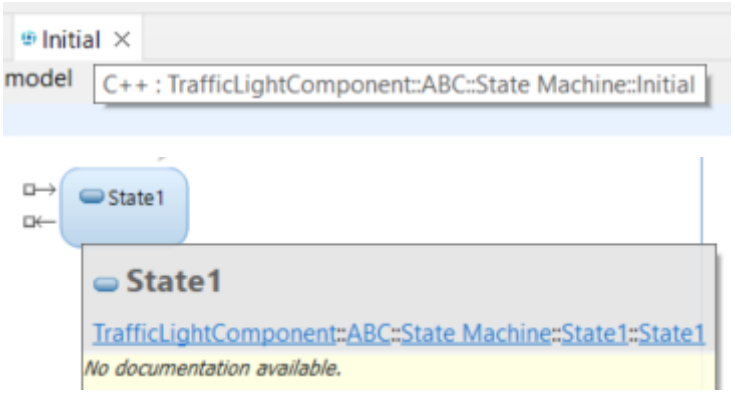
Copyright © 2023 HCL Technologies Limited | www.hcltechsw.com

Qualified Names According to Selected Viewpoint

- Qualified names shown in tooltips now respect the selected viewpoint
- In the Capsule Development viewpoint state machine regions are hidden and are therefore now excluded from the qualified names
  - Makes tooltips shorter and easier to read, especially for hierarchical state machines
  - This is now consistent with the Project Explorer and Properties view where the viewpoint already is respected
- Examples:

Model viewpoint

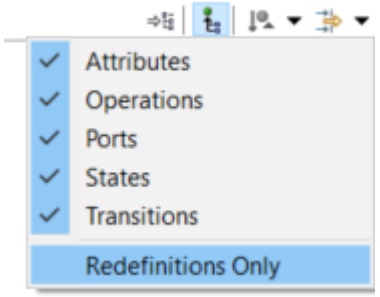
Capsule Development viewpoint

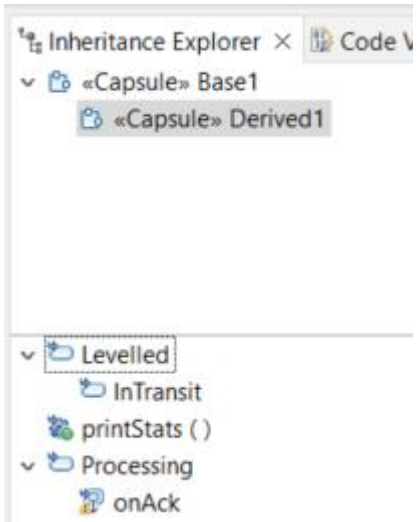


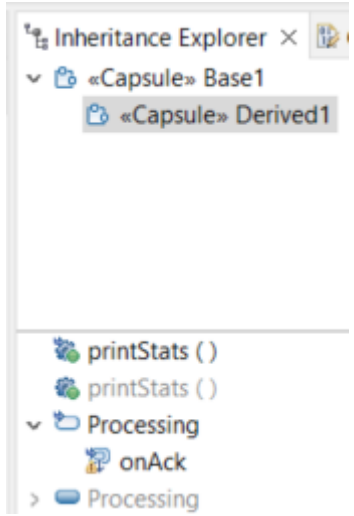
Copyright © 2023 HCL Technologies Limited | www.hcltechsw.com

Nested State Machines in the Inheritance Explorer

- The Inheritance Explorer now shows nested states and transitions
  - Inherited elements are shown with gray text
- The option **Redefinitions Only** can be used to easily see what states and transitions of an inherited state machine that have been redefined



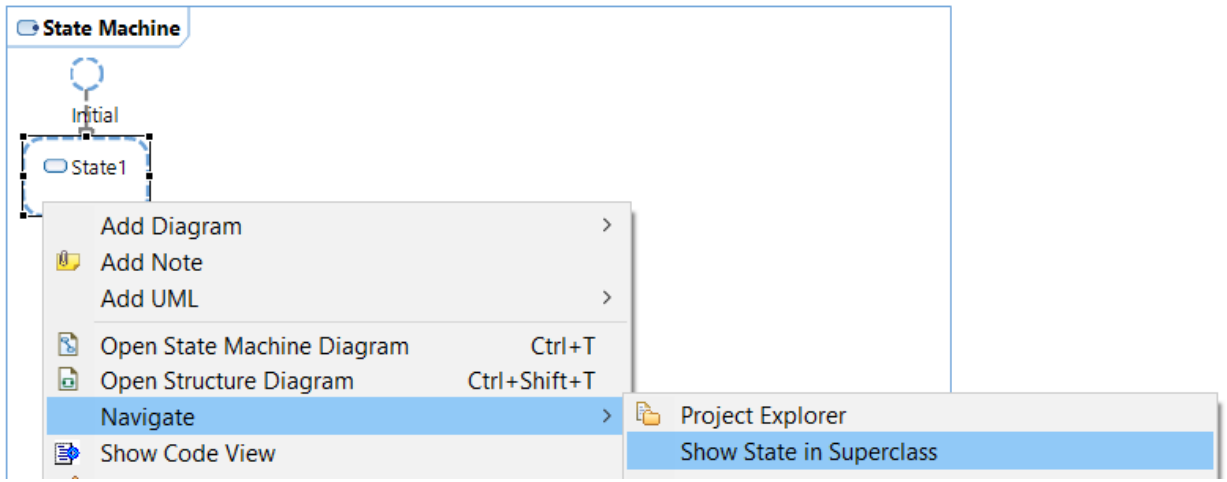




Copyright © 2023 HCL Technologies Limited | www.hcltechsw.com

Navigation to Inherited Elements

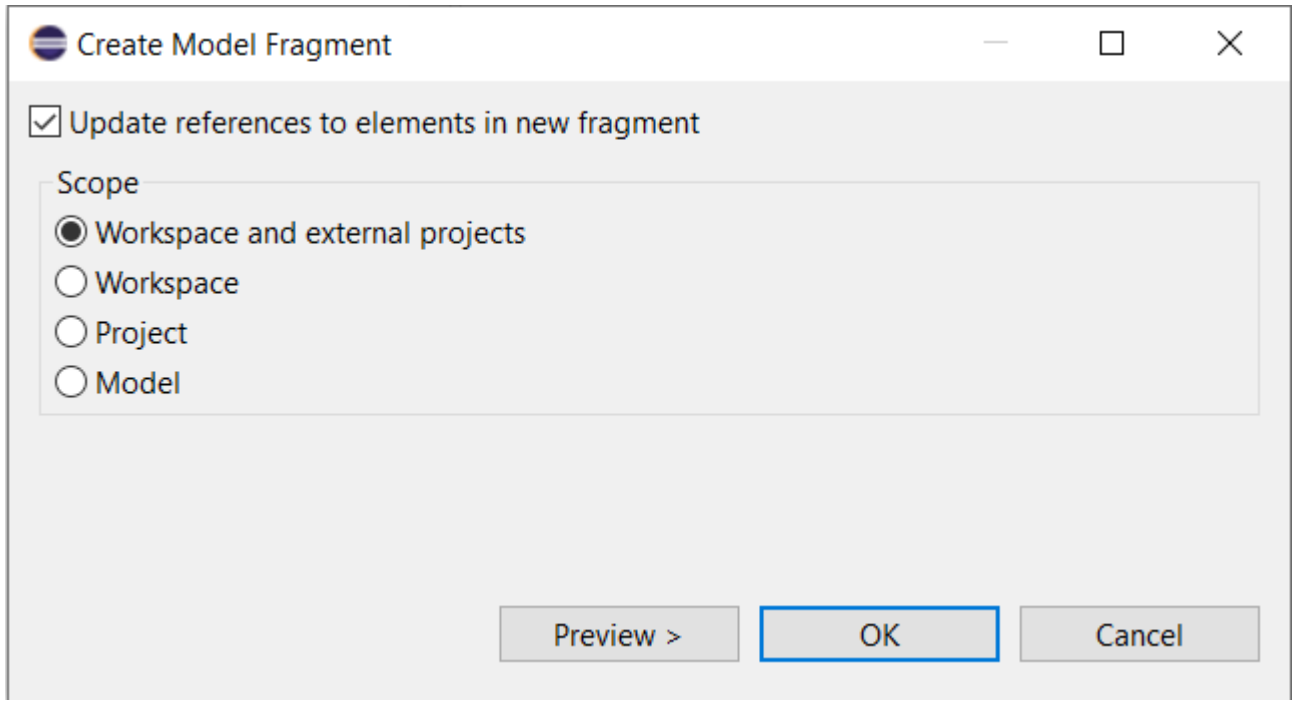
- The context menu command "Show in Superclass" was previously only capable of navigating from a redefining element to the redefined element in the super capsule
- Now it can also be used for navigating to inherited elements
  - Works for states, transitions and ports
  - Useful for navigating inheritance hierarchies
- The command is available in both the Project Explorer and diagram context menus
  - The inherited element will be highlighted in the Project Explorer, and from there you can use the context menu command **Navigate - Navigate to Diagram** to view it in a diagram.



Copyright © 2023 HCL Technologies Limited | www.hcltechsw.com

Improved Refactoring with External Projects

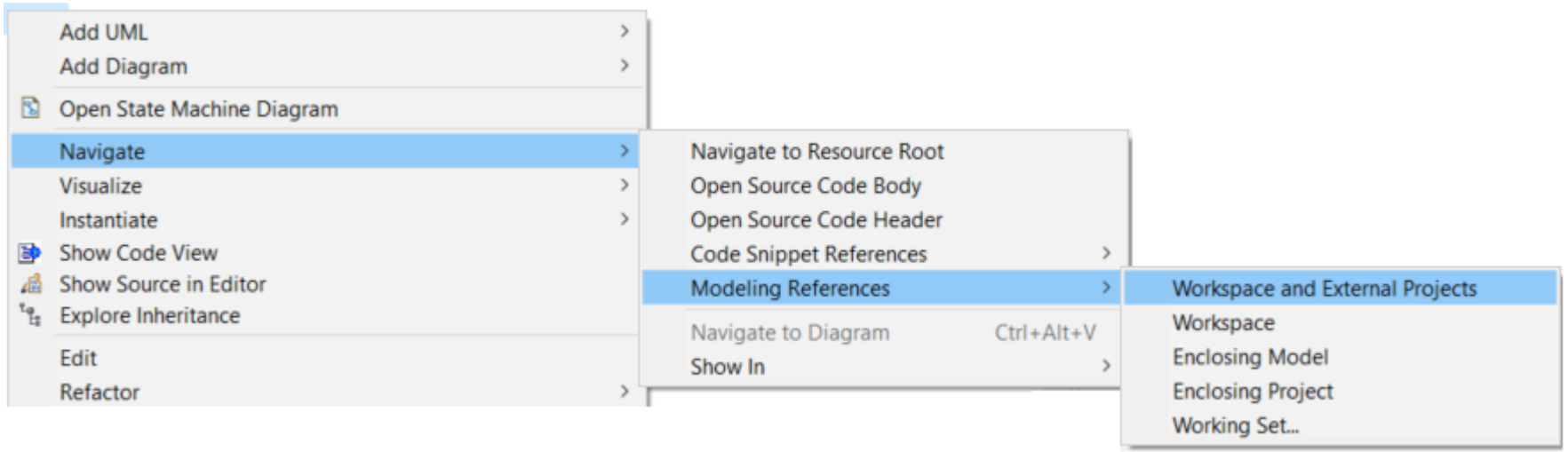
- Previously certain refactoring commands (e.g. creation of fragments) did not update external projects
  - This has now been solved



Copyright © 2023 HCL Technologies Limited | www.hcltechsw.com

Modeling References Improvements

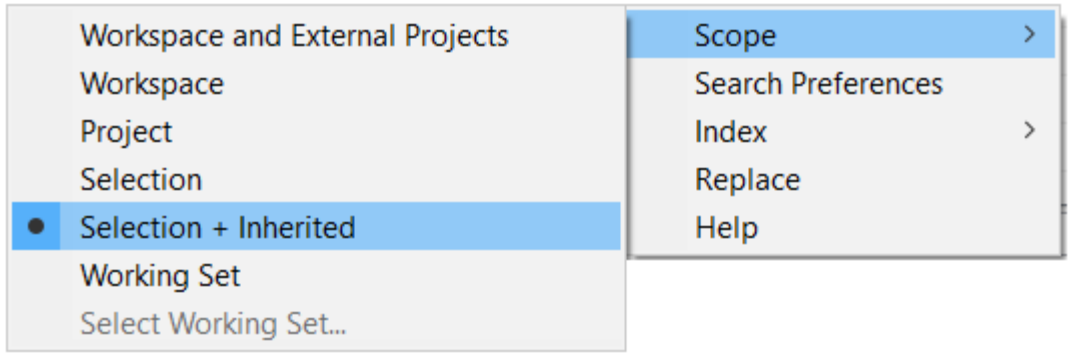
- The **Modeling References** command now supports searching for references in external projects
- The command has been moved under the **Navigate** context menu
  - Also the **Code Snippet References** command was moved
  - Makes the context menu shorter and more manageable

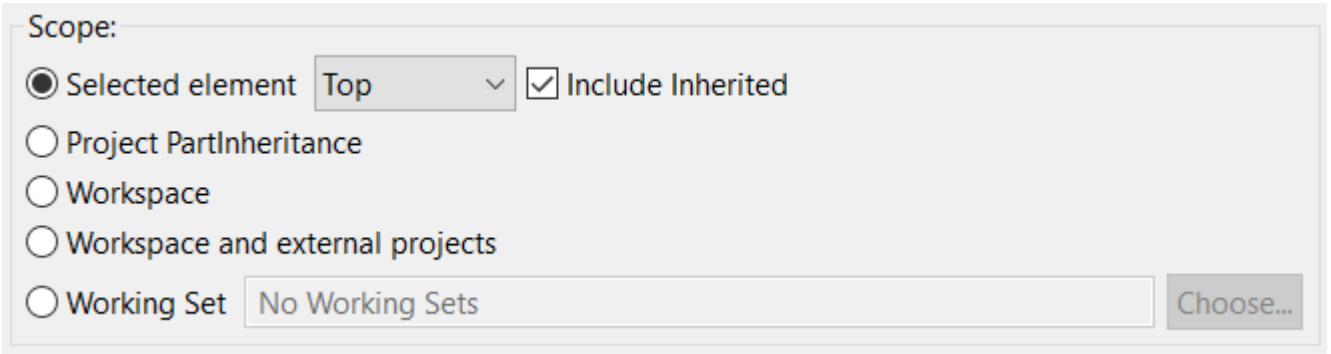


Copyright © 2023 HCL Technologies Limited | www.hcltechsw.com

Search in Inheritance Hierarchies

- It's now possible to search in the scope of an element plus all elements it inherits from
- For the Search field a new scope "Selection + Inherited" can be selected
- For the Search dialog a new checkbox "Include Inherited" can be checked
- This feature works for all elements that support inheritance: capsules, classes and interfaces



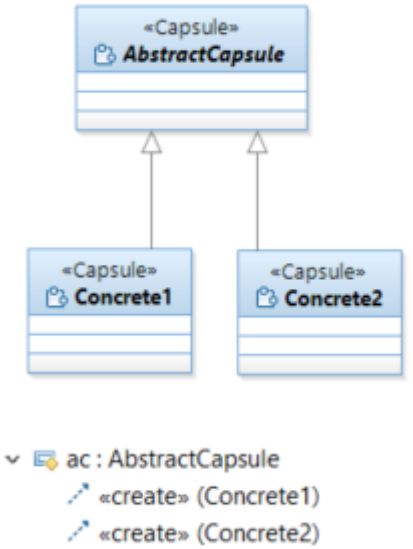


- When searching in inheritance hierarchies the search result is organized so that local elements come before inherited elements (i.e. the "distance" in the inheritance hierarchy is used for sorting the result)

Copyright © 2023 HCL Technologies Limited | www.hcltechsw.com

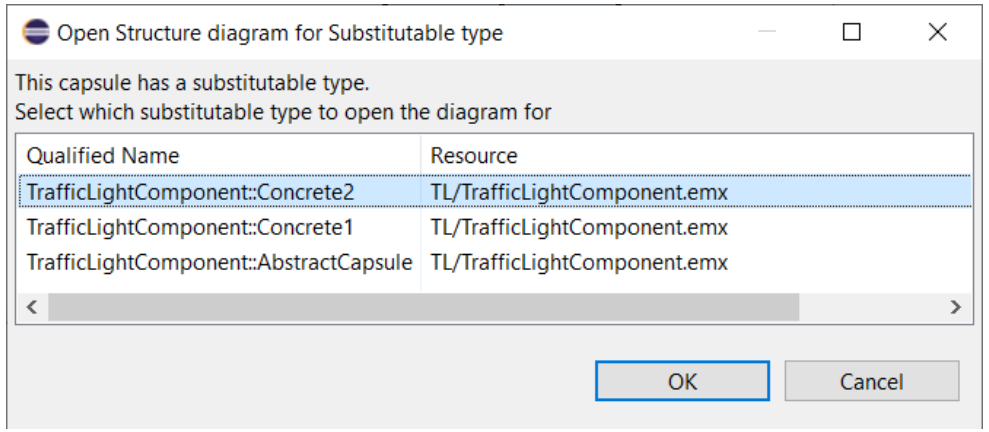
Navigation to Substitutable Types

- A capsule part that is typed by an abstract capsule, and with the Substitutable Type property set, will at run-time be incarnated by a concrete capsule that inherits from the abstract capsule



- It's now possible to use dependencies from the capsule part to those concrete capsules to express in the model which concrete capsules that will be incarnated into the part
  - <<create>> dependencies are used for this purpose, and you must use the Model viewpoint for creating those

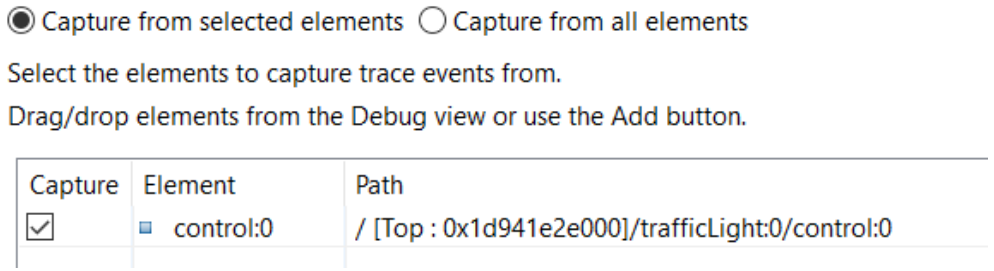
- When navigating to the composite structure or state machine diagram from such a capsule part, the following happens:
  - **No dependencies**
    The diagram of the abstract capsule is opened. This is the same behavior as before.
  - **At least one dependency to a concrete capsule**
    A dialog let's you choose which of the capsules to open the diagram for.

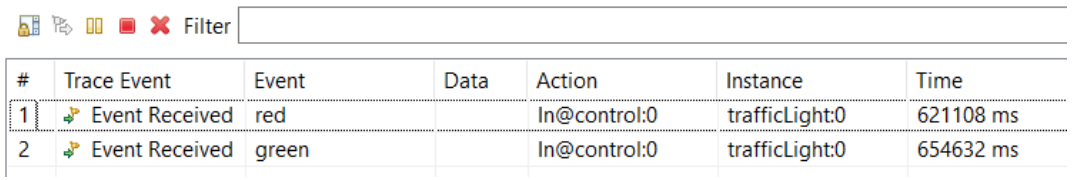


Copyright © 2023 HCL Technologies Limited | www.hcltechsw.com

Tracing Port Instances

- The Model Debugger now supports tracing events that are sent or received on specific port instances
  - Previously only the port itself could be added in the Capture tab in the Trace Editor, but now you can add one or several specific port instances



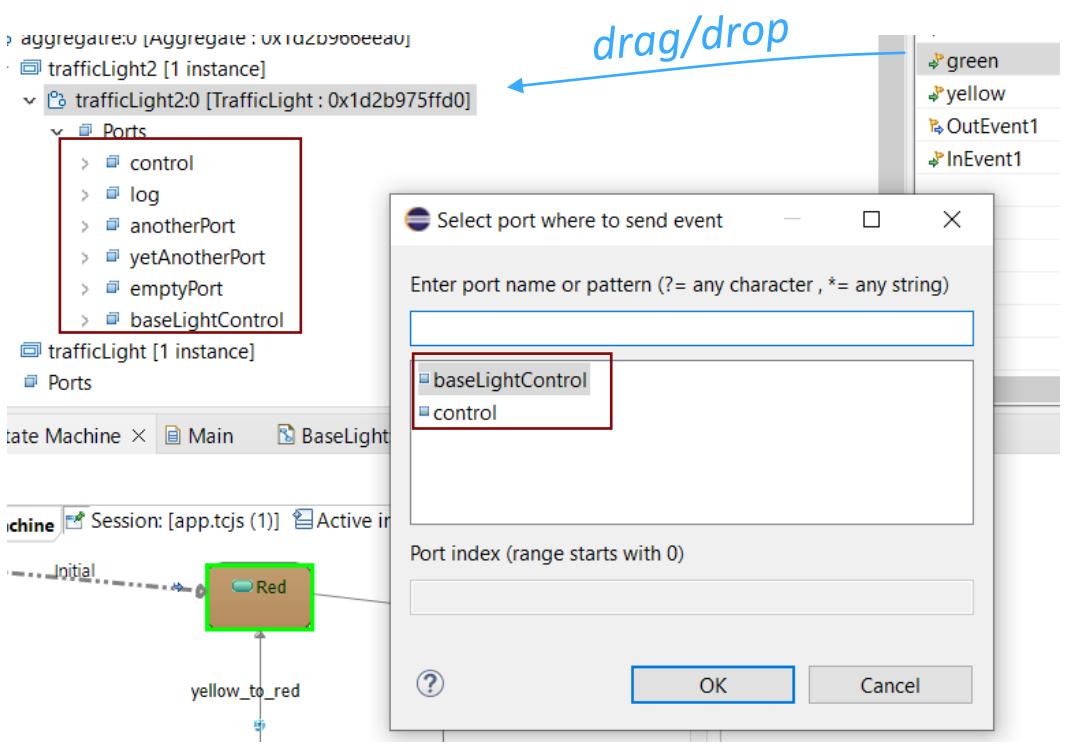


- The common case when tracing events sent or received on **all** ports of a capsule has been optimized
  - Accomplished by adding the capsule to the Capture tab (or selecting "Capture from all elements")
  - The trace performance is now better in case the capsule has many ports

Copyright © 2023 HCL Technologies Limited | www.hcltechsw.com

Easier to Send Events to Ports while Debugging

- A new dialog helps to send an event to the correct port
  - Only shows ports that can receive the event to be sent
  - Appears when you drag/drop an event from the Events view onto a capsule instance in the Debug view
  - Also appears when a capsule instance is selected in the Debug view and the context menu command **Send Event** is performed in the Events view context menu.
  - Also appears when you perform the **Send Event** context menu command on a capsule instance (then only ports that can receive <u>any</u> event are listed and you will be prompted in a subsequent dialog for which event to send)

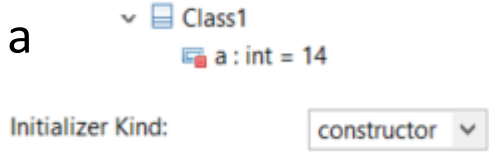


- If the target port is replicated (i.e. has multiplicity > 1) you can also specify the index of the port to receive the event
  - Leave the field blank to broadcast the event to all port instances

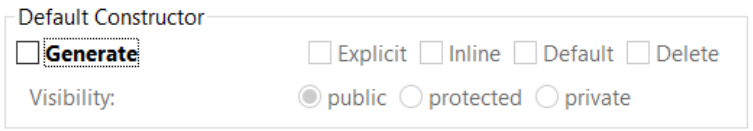
Copyright © 2023 HCL Technologies Limited | www.hcltechsw.com

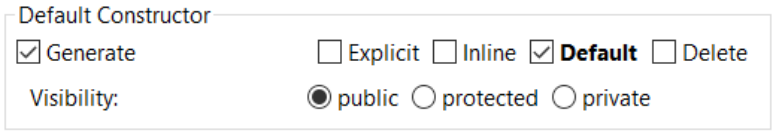
Warning if an Attribute Default Value is Ignored During Code Generation

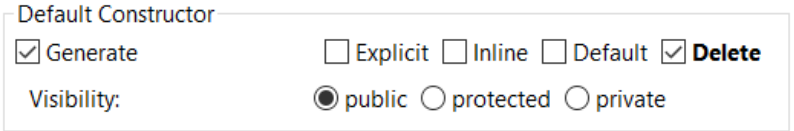
- A class attributes with "Initializer Kind" = "Constructor" that has a default value needs a constructor to be generated where the corresponding constructor initializer can be generated



- But it's possible (by means of other properties) to disable generation of a constructor that can have initializers. For example:







  - In this case the attribute will obviously not be initialized as expected (unless a user-defined constructor exists)

- The model compiler now detects this inconsistency and prints a warning

```
WARNING : HelloWorld::Class1::a : This attribute has a default value and the 'Initializer Kind'
property is set to 'Constructor', but no constructor will be generated where it can be
initialized. The attribute default value will be ignored.
```


Copyright © 2023 HCL Technologies Limited | www.hcltechsw.com

Configuration of Model Compiler Validation Rules (1/2)

- When generating code, the model compiler checks the input model against several validation rules
- It's now possible to configure which rules to be enabled or disabled, and which severity to use in case a rule fails and a problem is reported
  - New preference **RealTime Development - Build/Transformations - C++ - Rule Configuration**
  - A similar command-line argument for the model compiler can be used for batch builds: **--ruleConfiguration**
- Each rule has a unique id (4 digits). The rule configuration is a comma-separated list of rule ids, prefixed with a letter:
  - **X**: disable the rule
  - **E**: set the rule's severity to Error
  - **W**: set the rule's severity to Warning
  - **I**: set the rule's severity to Information
- The rule id is printed right after the rule's severity. For example:

```
WARNING[0001] : HelloWorld::Class1::a : This attribute has a default value and the 'Initializer Kind' property is set to
'Constructor', but no constructor will be generated where it can be initialized. The attribute default value will be ignored.
```

  - If no id is printed it means the rule cannot (yet) be configured.

Copyright © 2023 HCL Technologies Limited | www.hcltechsw.com

Configuration of Model Compiler Validation Rules (2/2)

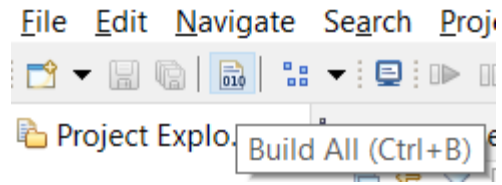
- Currently 5 validation rules can be configured:

Rule id	Rule name	Description	Default Severity
0001	AttributeInitialization	Signifies that the class does not contain a constructor where an attribute with a default value can be initialized. Hence, the attribute is ignored.	Warning
0002	TransitionUnreachable	Occurs when the model contains duplicate trigger events on more than one transition from the same origination point.	Warning
0003	EffectDuplicated	Indicates that a duplicate method has been generated as a result of the transition effect, and that the triggers are not compatible with the inherited method.	Warning
0004	GuardDuplicated	Indicates that a duplicate method has been generated for the guard of the transition as the triggers are not compatible with the inherited method.	Warning
0005	MultiplicityBadFormat	Indicates that the application is unable to parse multiplicity when an incorrect value of multiplicity is entered.	Warning

Copyright © 2023 HCL Technologies Limited | www.hcltechsw.com

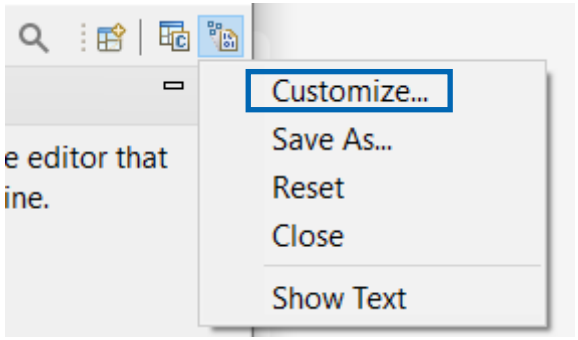
Build All Command Removed from Toolbar

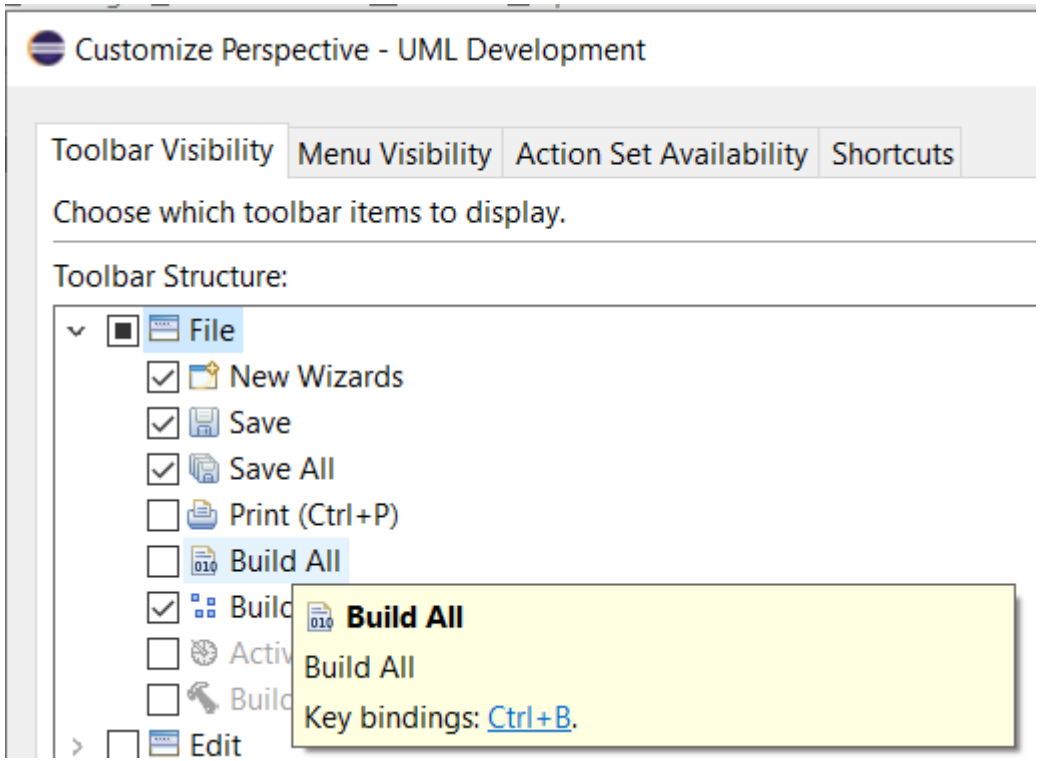
- The Build All command (contributed by CDT) has been removed from the toolbar in the UML Development perspective
  - It was too close to the "Build Active Transformation Configuration" button - easy to press the wrong button when building a TC



- With an old perspective you may need to do a Reset to get rid of the button

- If you want the button back, you can customize the UML Development perspective

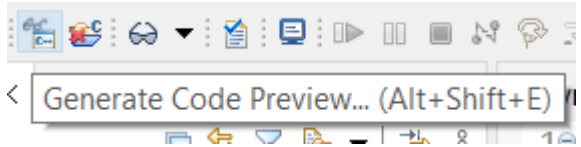




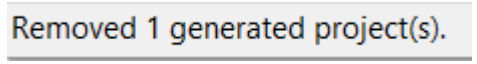
Copyright © 2023 HCL Technologies Limited | www.hcltechsw.com

UI Improvements for Code Generation

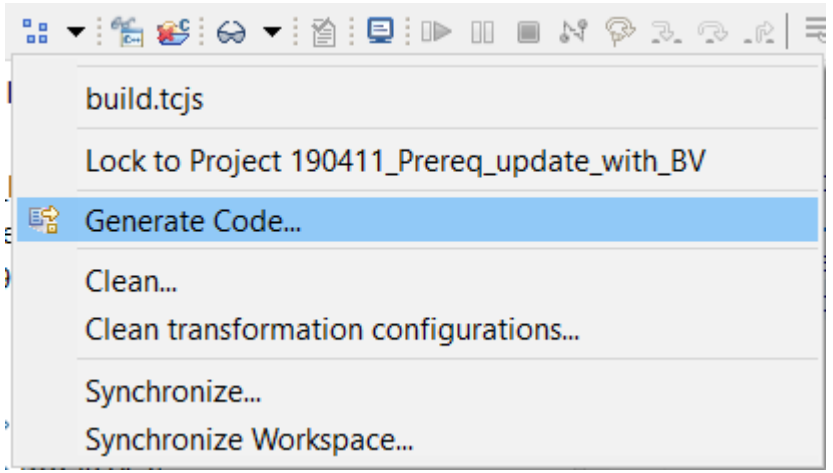
- New toolbar buttons were added to make it easier to invoke these commands:
  - **Generate Code Preview**
  - **Remove All Generated Projects**



- The number of generated projects that were removed is now shown in the status bar



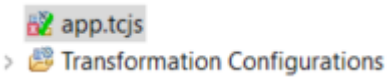
- A new command **Generate Code** was added
  - Available in the Build Active Transformation Configuration button menu
  - Also available in the context menu of a TC
  - The **Run Transformation** button in the TC editor was renamed to **Generate Code** for a consistent terminology

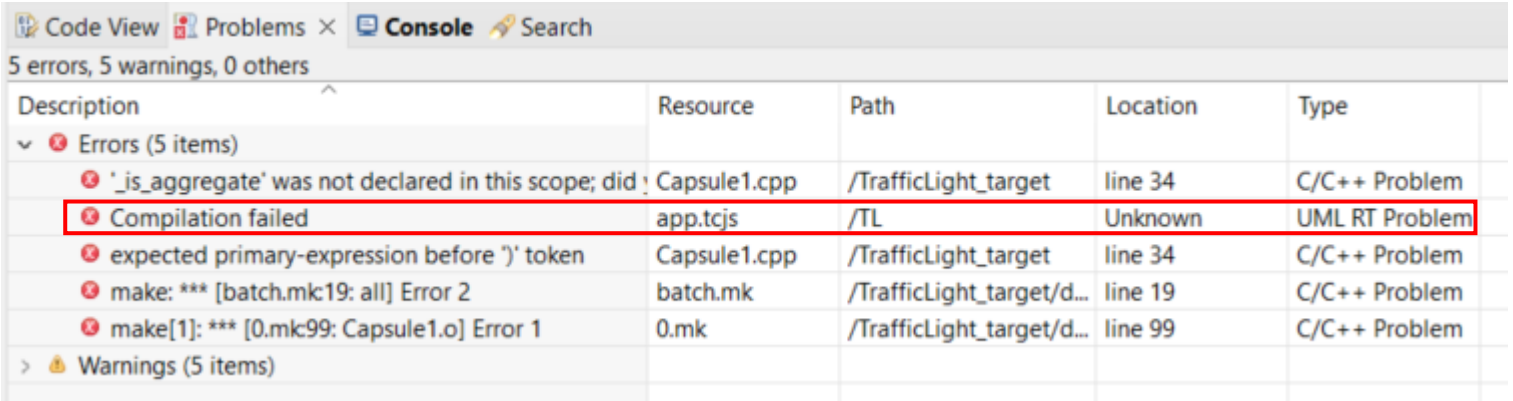


Copyright © 2023 HCL Technologies Limited | www.hcltechsw.com

Removal of Problem Markers when Cleaning a TC

- When a build fails, for example due to compilation errors, problem markers are set on the TC



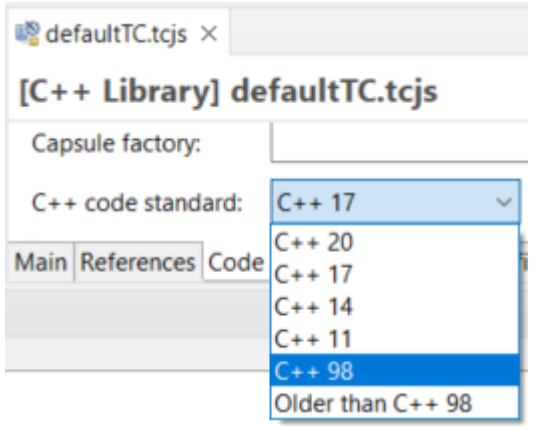


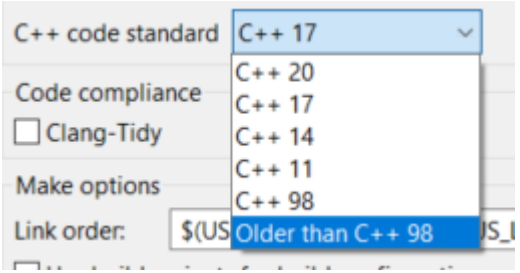
- This is done to make it more visible that the build failed
  - But note that these problem markers do not indicate that there is an error in the TC itself
- Now such problem markers are removed when the TC is cleaned
  - Keeping those problem markers after a Clean could previously confuse users and make them believe that Clean failed
- If the problems persist, new problem markers will be set on the TC the next time it is built

Copyright © 2023 HCL Technologies Limited | www.hcltechsw.com

Support for Old C++ Compilers

- Two new C++ code standards are now available
  - **C++ 98**
    Works like "Older than C++ 11" worked before, i.e. generates code compliant with the C++ 98 standard
  - **Older than C++ 98**
    Can be used if a very old C++ compiler is used that doesn't support all of C++ 98.
    For this code standard, C-style casts will be used instead of C++ casts.





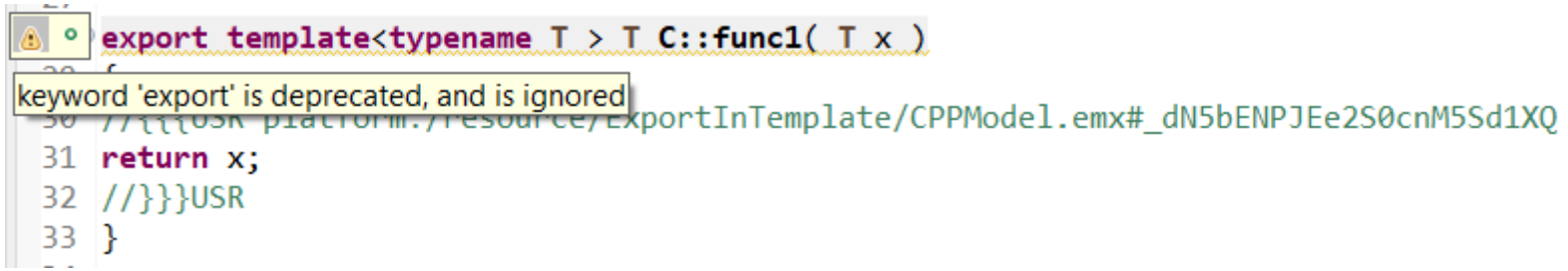
- The TargetRTS has been updated to allow it to be compiled with a compiler that doesn't support C++ 11
  - Previously it was necessary to use an older version of the TargetRTS (11.0 2020.22 or older), but now the latest version can be used
  - However, not all RTist features can be used if the compiler doesn't support C++ 11. If such a feature is used, an error message will be reported either by the Model Compiler or the C++ compiler. For example:

```
13:57:02 : INFO : Transforming model...
13:57:02 : ERROR : CPPModel::TOP::part1 : A Create Function Body requires C++ 11 or later
```


Copyright © 2023 HCL Technologies Limited | www.hcltechsw.com

Removal of "export" Keyword for Templates

- Starting from C++ 11 the `export` keyword is deprecated, and in C++ 20 it now has a different meaning
  - But even before C++ 11, very few compilers supported this language feature
  - Many modern compilers give warnings if templates use the `export` keyword.



- To avoid such warnings and be compliant with modern C++, the code generator no longer generates the `export` keyword

Copyright © 2023 HCL Technologies Limited | www.hcltechsw.com

Moving Event Data - Additional Constructor Generated

- The RTTypedValue struct that is generated for a user-defined type now has an additional constructor
- Allows moving (instead of copying) event data also for cases when the type descriptor needs to be explicitly provided by the user
  - For example when sending an event with data that is a subclass instance while the event parameter is typed by a superclass
- In the TargetRTS, RTTypedValue_RTString now also has such a constructor

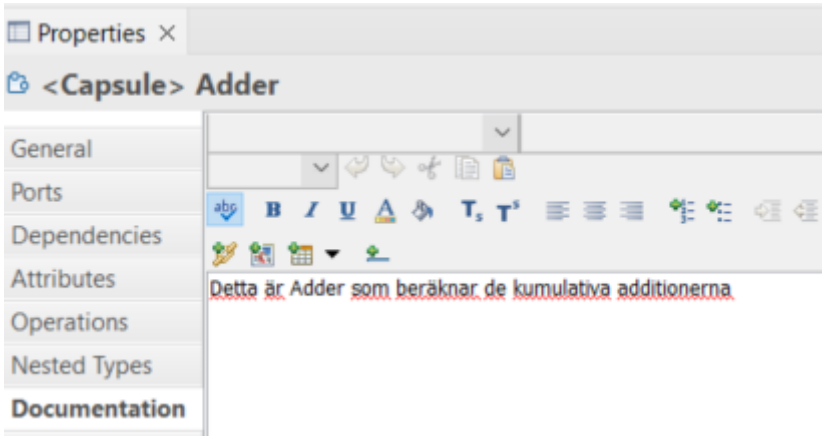
```
struct RTTypedValue_MyClass
{
    const void * data;
    const RTObject_class * type;
    const bool rValueRef = false;
    inline RTTypedValue_MyClass( const MyClass & rtg_value )
        : data( &rtg_value )
        , type( &RTType_MyClass )
    {
    }
    inline RTTypedValue_MyClass( const MyClass && rtg_value )
        : data( &rtg_value )
        , type( &RTType_MyClass )
        , rValueRef( true )
    {
    }
    inline RTTypedValue_MyClass( const MyClass & rtg_value, const RTObject_class * rtg_type )
        : data( &rtg_value )
        , type( rtg_type )
    {
    }
    inline RTTypedValue_MyClass( const MyClass && rtg_value, const RTObject_class * rtg_type )
        : data( &rtg_value )
        , type( rtg_type )
        , rValueRef( true )
    {
    }
    inline ~RTTypedValue_MyClass( void )
    {
    }
};
```

new constructor

Copyright © 2023 HCL Technologies Limited | www.hcltechsw.com

Encoding of Generated Files

- The Eclipse preference **General - Workspace - Text file encoding** is now UTF-8 by default
  - In older versions it was Cp1252 by default
  - It's recommended not to change this preference, but to keep the default and use UTF-8 as file encoding
- To be consistent with this preference, generated files are now also UTF-8 encoded
  - Applies for both C++ files and make files
  - Ensures that Unicode characters can be used, for example in comments and string literals




```
11 //{{{USR platform:/resource/Pi_original/HelloWorld.emx#_iDa14N9mEeCEzIVzLZLXAA|General|headerPreface
12 #pragma mark - HEADER PREFACE
13 // ∑ √(a + b)
14 #include <math.h>
15 #pragma mark -
16 //}}}USR
17 #define SUPER RTActor
18 // Detta är Adder som beräknar de kumulativa additionerna
19 class Adder_Actor : public RTActor
20 {
```


Copyright © 2023 HCL Technologies Limited | www.hcltechsw.com

More Merge-Friendly TC Files

- Lists in TC files now end with a trailing comma
  - Applies for example for the properties "Sources" and "Threads"

```
4  tc.sources = [
5      'platform:/resource/TrafficLightsDemo/TrafficLightComponent.emx#_rx2SkNXDEei3BJT-OqdJZA',
6      'platform:/resource/TrafficLightsDemo/TrafficLightComponent.emx#_Rjgg8NXEEei3BJT-OqdJZA',
7  ];
```

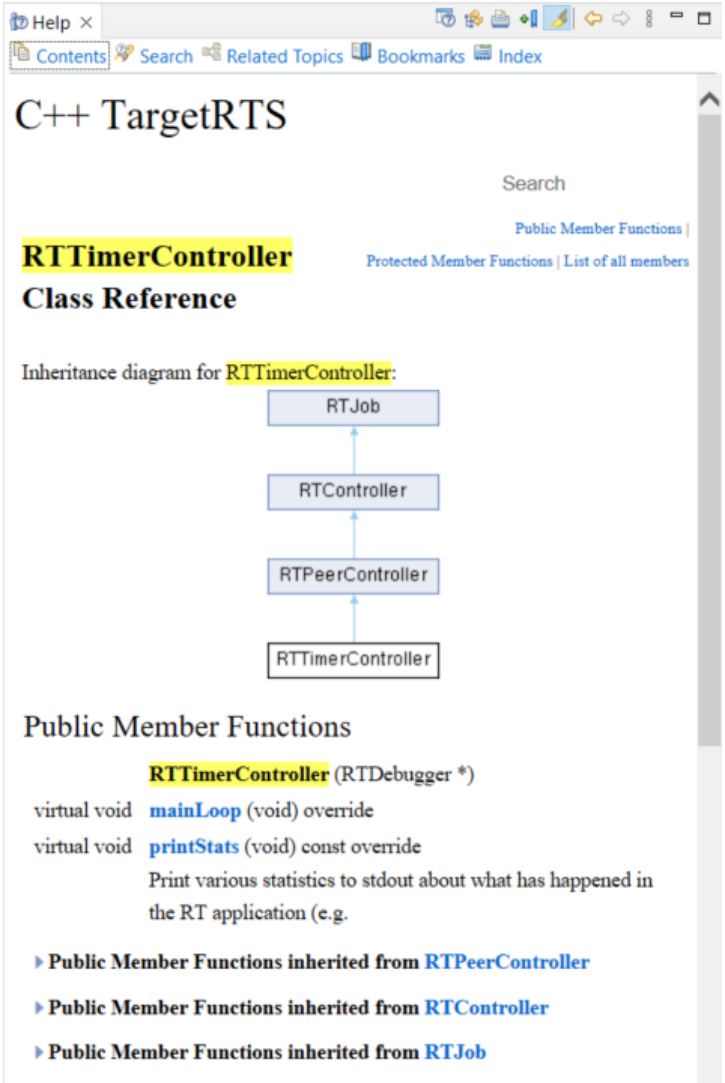
```
40      name: 'PushButtonThread',
41      implClass: 'RTPeerController',
42      stackSize: '20000',
43      priority: 'DEFAULT_MAIN_PRIORITY',
44      logical: [
45          'PushButtonThread',
46      ]
47  },
48  ];
```

- This reduces the risk for conflicts when merging TC files
  - Adding an element to such a property now only affects a single line
- But note that the first time you change and save an existing TC file, you will see these additional trailing commas in many places

Copyright © 2023 HCL Technologies Limited | www.hcltechsw.com

Extended TargetRTS Documentation

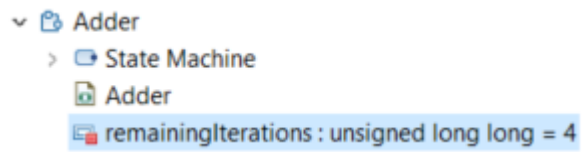
- Documentation for the TargetRTS is generated from its sources by means of Doxygen
- This documentation has now been extended to also include code in the folder `<TargetRTS>/src/include`
  - APIs provided by the code in this location are not directly used by generated code, but can still be useful to browse in the documentation



Copyright © 2023 HCL Technologies Limited | www.hcltechsw.com

Support for `long long` and `unsigned long long`

- These predefined types are available in CppPrimitiveDatatypes

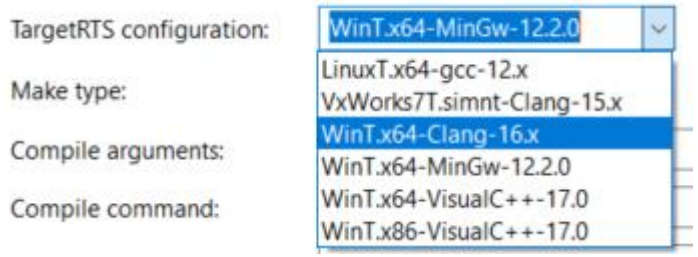


- The Model Compiler and TargetRTS now supports them
  - Previously a warning was printed if they were used since they did not have a type descriptor implementation in the TargetRTS
- Note: You can build the TargetRTS with a compiler that doesn't support C++ 11, but it must have support for `long long`.
  - Most old compilers supported `long long` even before C++ 11
  - If your compiler doesn't support `long long` you need to remove the code from the TargetRTS that relies on it, and make sure you application doesn't use this predefined type

Copyright © 2023 HCL Technologies Limited | www.hcltechsw.com

Support for the Clang Compiler

- The Clang 16.x C++ compiler for Windows (version mingw-w64-x86_64-clang) can now be directly used with RTist
  - A target configuration for this compiler is available
  - Prebuilt libraries for this compiler are included (TargetRTS, Connexis)
  - Note that the LibTCPServer is not yet supported for this compiler



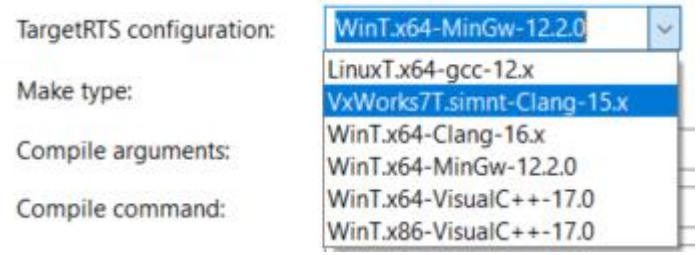
- The Clang 14.x C++ compiler for MacOS can now be directly used with RTist
  - A target configuration for this compiler is available
  - Prebuilt libraries for this compiler are included (TargetRTS, Connexis)

Copyright © 2023 HCL Technologies Limited | www.hcltechsw.com

VxWorks

- Older versions of RTist included support for the VxWorks RTOS
  - It was temporarily removed since it didn't work with modern versions of VxWorks

- Now there is built-in support for building with the Clang 15.x compiler for VxWorks 7
  - A new target configuration is available: VxWorks7T.simnt-Clang-15.x
  - Prebuilt libraries for this compiler are included (TargetRTS, Connexis) and can be used for building applications to be run on the VxWorks simulator on Windows
  - Note that the LibTCPServer is not yet supported for this compiler

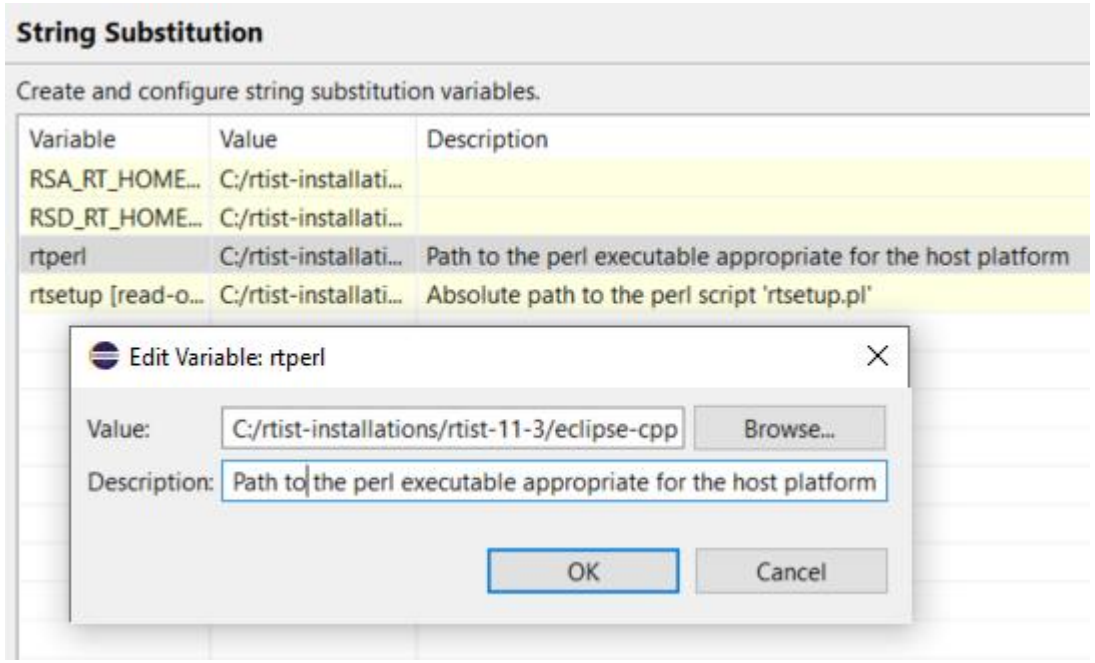


- Learn more about this in the documentation at **RTist User's Guide - Articles - Integrations - VxWorks Integration**

Copyright © 2023 HCL Technologies Limited | www.hcltechsw.com

Changing RTPerl

- The variable **rtperl** is now editable
  - If an rtperl executable is not available on the platform where you run the build, or if you don't want to use it, you can now change this variable (for example to instead use the regular perl)
  - Currently the rtperl executable is available for Linux and Windows only
  - On MacOS the variable is therefore by default set to just "perl", which means it is assumed that the regular perl is in the path



Copyright © 2023 HCL Technologies Limited | www.hcltechsw.com

HCL

Relationship™

BEYOND THE CONTRACT

$7 BILLION ENTERPRISE | **110,000** IDEAPRENEURS | **31** COUNTRIES

WATCH THE **FILM**