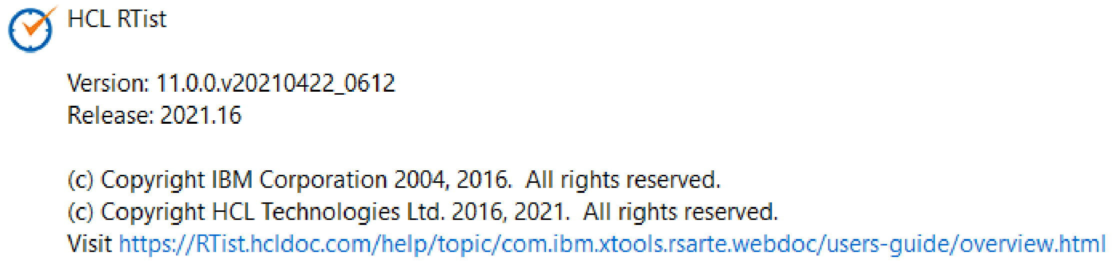
What's New in HCL RTist 11

updated for release 2021.16

HCL SOFTWARE

Overview

- RTist 11 is based on Eclipse 2019.06 (4.12)
- HCL RTist is 100% compatible with IBM RSARTE. All features in IBM RSARTE are also present in HCL RTist. However, HCL RTist contains a few features that do not exist in IBM RSARTE.
  - Those features are marked in this presentation by RTist ONLY


HCL RTist

Version: 11.0.0.v20210422_0612
Release: 2021.16

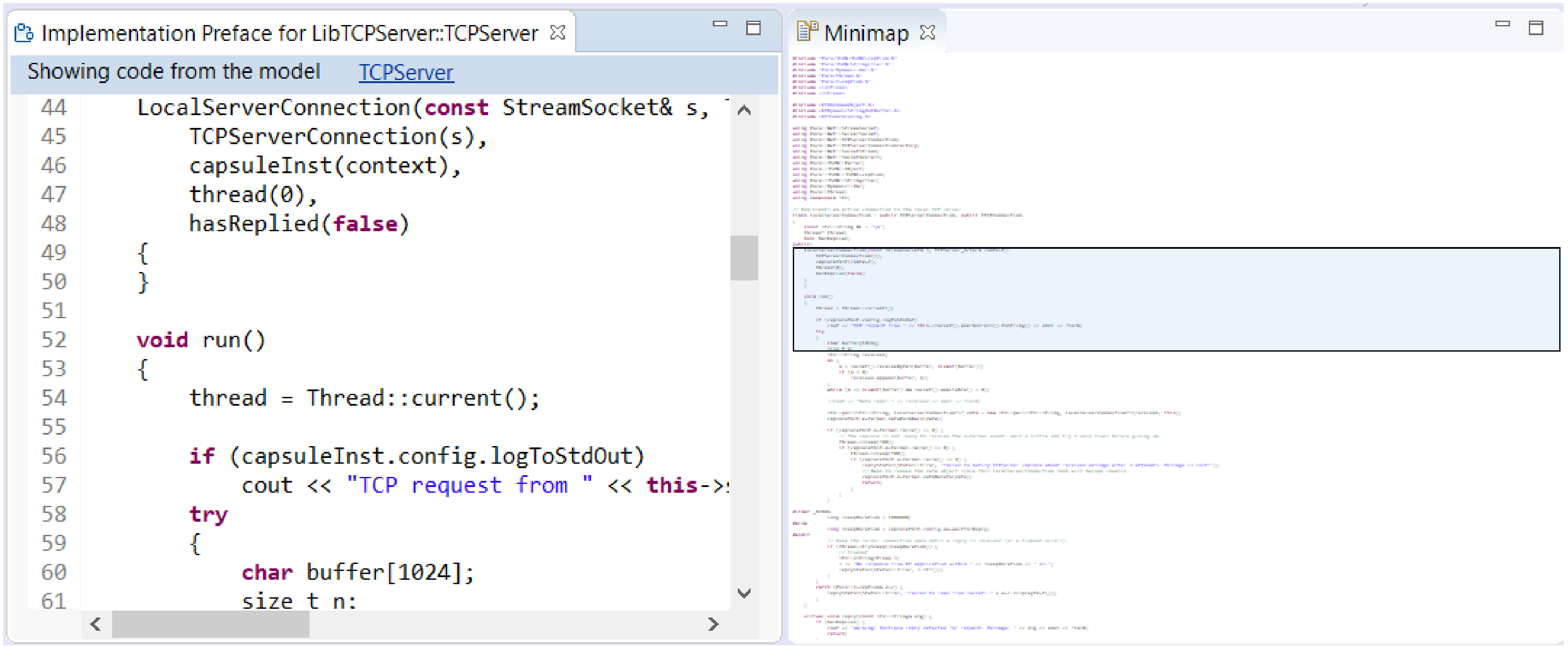
(c) Copyright IBM Corporation 2004, 2016. All rights reserved.
(c) Copyright HCL Technologies Ltd. 2016, 2021. All rights reserved.
Visit https://RTist.hcldoc.com/help/topic/com.ibm.xtools.rsarte.webdoc/users-guide/overview.html

Eclipse 4.12 (2019.06)

- Compared to RTist 10.3, RTist 11 includes new features from 4 quarterly Eclipse releases:
  - 2018.09 (https://www.eclipse.org/eclipse/news/4.9/platform.php)
  - 2018.12 (https://www.eclipse.org/eclipse/news/4.10/platform.php)
  - 2019.03 (https://www.eclipse.org/eclipse/news/4.11/platform.php)
  - 2019.06 (https://www.eclipse.org/eclipse/news/4.12/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...

Eclipse 4.12 (2019.06)

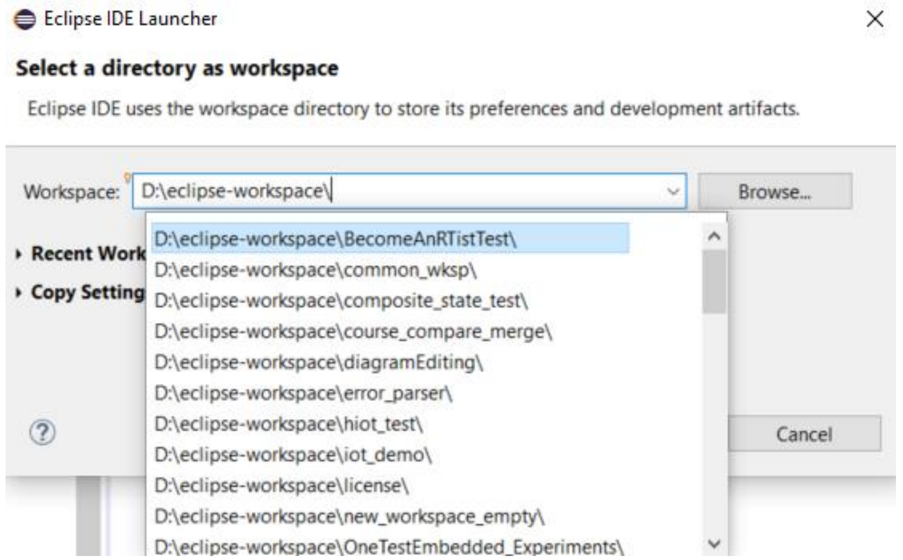
- A new Minimap view gives a better overview of the text editor contents (useful when it's large)
  - It's easy to launch it by typing "minimap" in the QuickAccess field


Implementation Preface for LibTCPServer::TCPServer
Showing code from the model TCPServer
Minimap

```
    LocalServerConnection(const StreamSocket& s, 
        TCPServerConnection(s),
        capsuleInst(context),
        thread(0),
        hasReplied(false)
    {
    }

    void run()
    {
        thread = Thread::current();

        if (capsuleInst.config.logToStdOut)
            cout << "TCP request from " << this->
        try
        {
            char buffer[1024];
            size_t n;
```

Eclipse 4.12 (2019.06)

- The Select Workspace dialog now supports auto-completion
  - Allows to pick a workspace more easily by only using the keyboard
- User-defined filtering of the Project Explorer
  - You can now create your own regular expressions for filtering out items from the Project Explorer
  - Can be used as an alternative to working sets

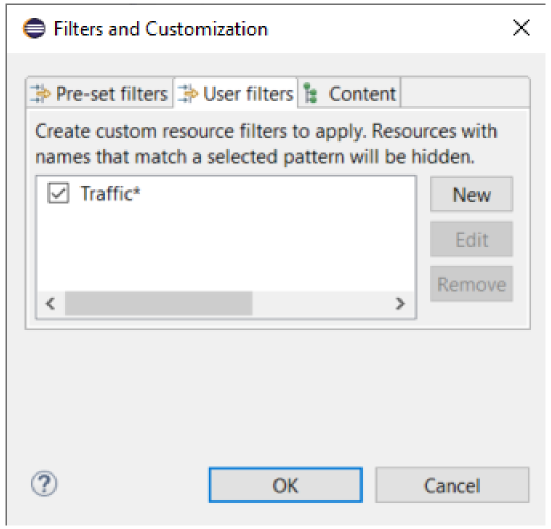

Eclipse IDE Launcher

Select a directory as workspace

Eclipse IDE uses the workspace directory to store its preferences and development artifacts.

Workspace: D:\eclipse-workspace\

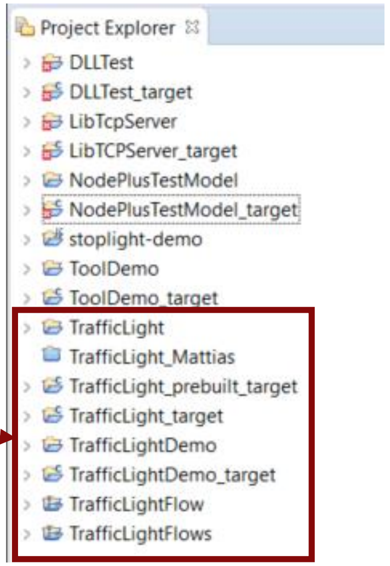
D:\eclipse-workspace\BecomeAnRTistTest\
D:\eclipse-workspace\common_wksp\
D:\eclipse-workspace\composite_state_test\
D:\eclipse-workspace\course_compare_merge\
D:\eclipse-workspace\diagramEditing\
D:\eclipse-workspace\error_parser\
D:\eclipse-workspace\hiot_test\
D:\eclipse-workspace\iot_demo\
D:\eclipse-workspace\license\
D:\eclipse-workspace\new_workspace_empty\
D:\eclipse-workspace\OneTestEmbedded_Experiments\


Filters and Customization

Create custom resource filters to apply. Resources with names that match a selected pattern will be hidden.

☑ Traffic*

will be hidden
by the user filter

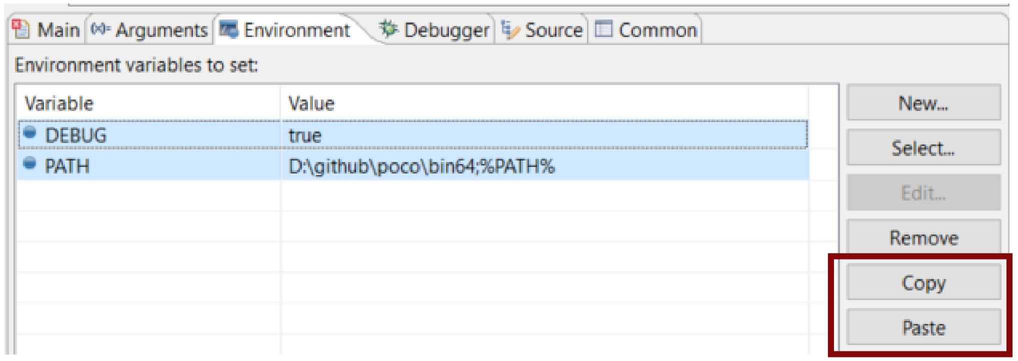

Project Explorer

DLLTest
DLLTest_target
LibTcpServer
LibTCPServer_target
NodePlusTestModel
NodePlusTestModel_target
stoplight-demo
ToolDemo
ToolDemo_target
TrafficLight
TrafficLight_Mattias
TrafficLight_prebuilt_target
TrafficLight_target
TrafficLightDemo
TrafficLightDemo_target
TrafficLightFlow
TrafficLightFlows

HCL SOFTWARE

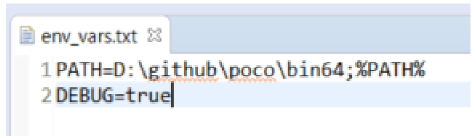
Eclipse 4.12 (2019.06)

- Copy/Paste of environment variables
  - Setting up environment variables in a launch configuration is now much easier thanks to copy/paste support
  - Environment variables can be copied from one launch configuration to another, or from a text editor to a launch configuration (or vice versa)



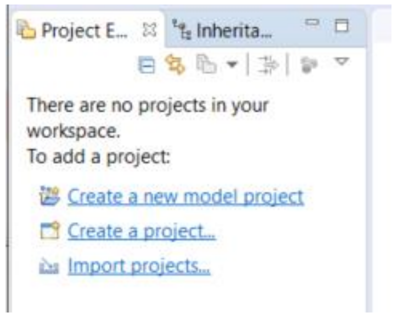
Environment variables to set:

Variable	Value
DEBUG	true
PATH	D:\github\poco\bin64;%PATH%


env_vars.txt

```
PATH=D:\github\poco\bin64;%PATH%
DEBUG=true
```

- Useful quick links for an empty workspace
  - Quick links for creating or importing projects make it easier for new users to get started with an empty workspace
  - Which quick links that are shown depend on the current perspective

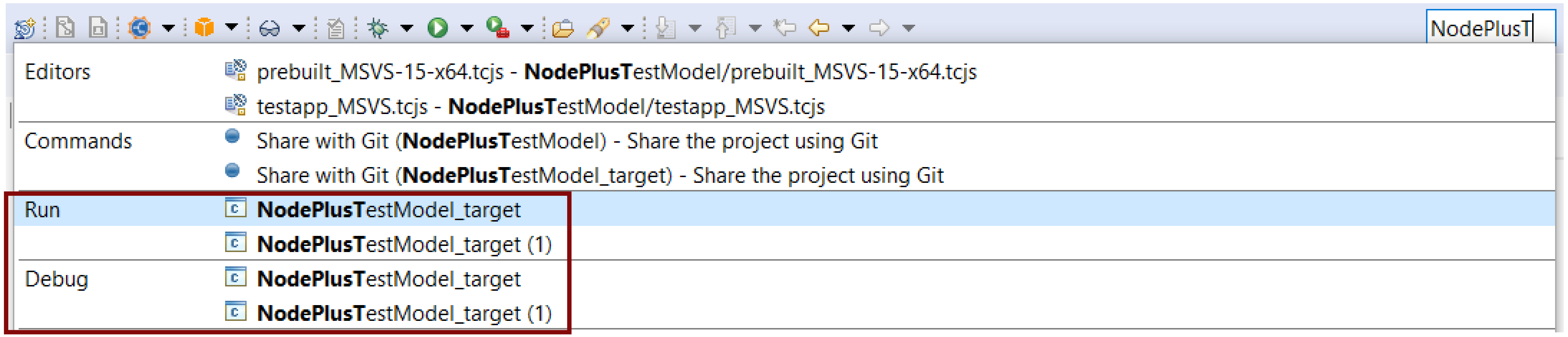


There are no projects in your workspace.
To add a project:

Create a new model project
Create a project...
Import projects...

Eclipse 4.12 (2019.06)

- Launch configurations are now accessible from Quick Access
  - Can start either a Run or a Debug session



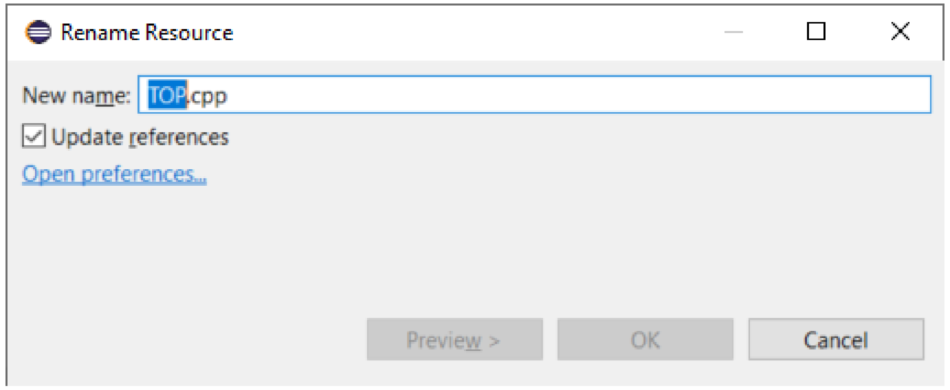
NodePlusT

Editors	prebuilt_MSVS-15-x64.tcjs - **NodePlusT**estModel/prebuilt_MSVS-15-x64.tcjs
	testapp_MSVS.tcjs - **NodePlusT**estModel/testapp_MSVS.tcjs
Commands	Share with Git (**NodePlusT**estModel) - Share the project using Git
	Share with Git (**NodePlusT**estModel_target) - Share the project using Git
Run	**NodePlusT**estModel_target
	NodePlusTestModel_target (1)
Debug	**NodePlusT**estModel_target
	NodePlusTestModel_target (1)

HCL SOFTWARE

CDT 9.8 (included as part of Eclipse 2019.06)

- Possible to rename a file without triggering a refactoring
  - A checkbox controls if references to the renamed file should be updated

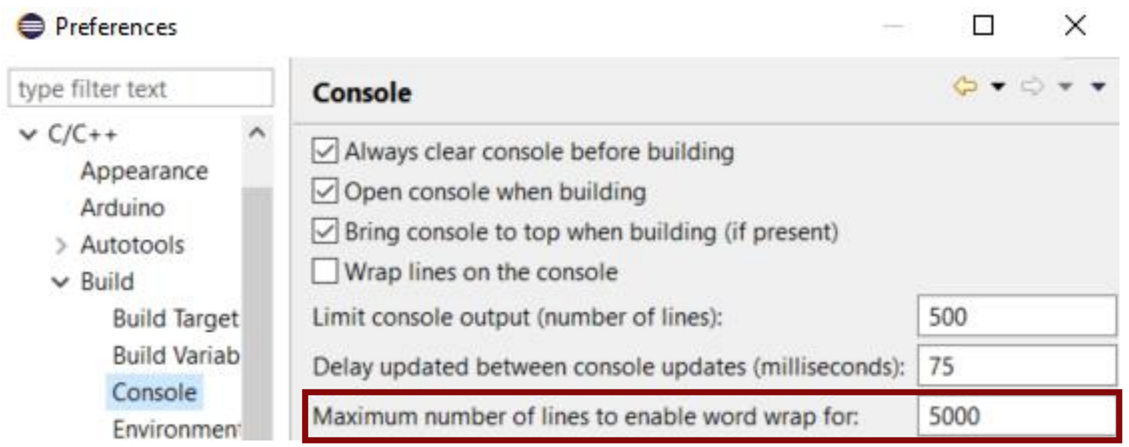

Rename Resource

New name: TOP.cpp

☑ Update references

Open preferences...

- New preference to address a performance issue with line wrapping a large number of lines in the console
  - *C/C++ - Build – Console – Maximum number of lines to enable word wrap for*
  - Don't set this too high if *Wrap lines on the console* is turned on


Preferences

Console

☑ Always clear console before building
☑ Open console when building
☑ Bring console to top when building (if present)
☐ Wrap lines on the console

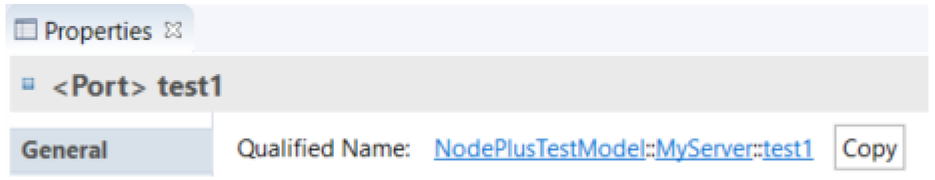
Limit console output (number of lines):	500
Delay updated between console updates (milliseconds):	75
Maximum number of lines to enable word wrap for:	5000

HCL SOFTWARE

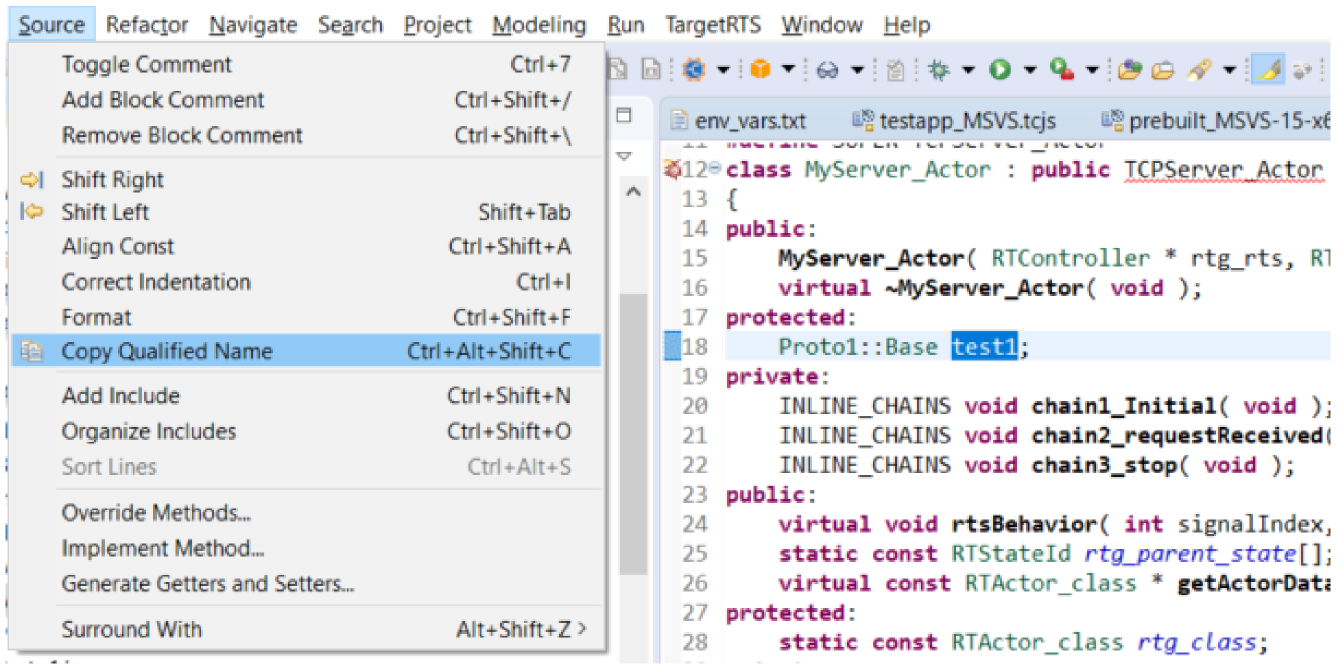
CDT 9.8 (included as part of Eclipse 2019.06)

- Improved dialog for attaching to a C++ application to debug
  - Command-line arguments for the process are now shown (allows to more easily find the process to debug)
  - The dialog now remembers a previously entered filter expression (to make it easier to attach to the same process multiple times)

- Copy the qualified name of a C++ declaration
  - A new command is available in the Source menu
  - Similar usecase as for the Copy button of the Qualified Name for a model element in the Properties view



Qualified Name: NodePlusTestModel::MyServer::test1



```
class MyServer_Actor : public TCPServer_Actor
{
public:
    MyServer_Actor( RTController * rtg_rts, RT
    virtual ~MyServer_Actor( void );
protected:
    Proto1::Base test1;
private:
    INLINE_CHAINS void chain1_Initial( void );
    INLINE_CHAINS void chain2_requestReceived(
    INLINE_CHAINS void chain3_stop( void );
public:
    virtual void rtsBehavior( int signalIndex,
    static const RTStateId rtg_parent_state[];
    virtual const RTActor_class * getActorData
protected:
    static const RTActor_class rtg_class;
```

CDT 9.8 (included as part of Eclipse 2019.06)

- CODAN improvements
  - Additional checks implemented, for example to detect C-style casts and goto-statements

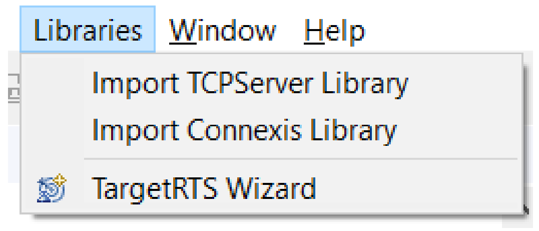
- For more information about CDT improvements see
  https://wiki.eclipse.org/CDT/User/NewIn96
  https://wiki.eclipse.org/CDT/User/NewIn97
  https://wiki.eclipse.org/CDT/User/NewIn98

Newer EGit Version in the EGit Integration

- The EGit integration in RTist has upgraded EGit from 5.0 to 5.4
  - This is the recommended and latest version for Eclipse 2019.06
- This upgrade provides several new features, performance improvements and bug fixes
  - For detailed information about the changes see
    https://wiki.eclipse.org/EGit/New_and_Noteworthy/5.1
    https://wiki.eclipse.org/EGit/New_and_Noteworthy/5.2
    https://wiki.eclipse.org/EGit/New_and_Noteworthy/5.3
    https://wiki.eclipse.org/EGit/New_and_Noteworthy/5.4

Libraries Menu

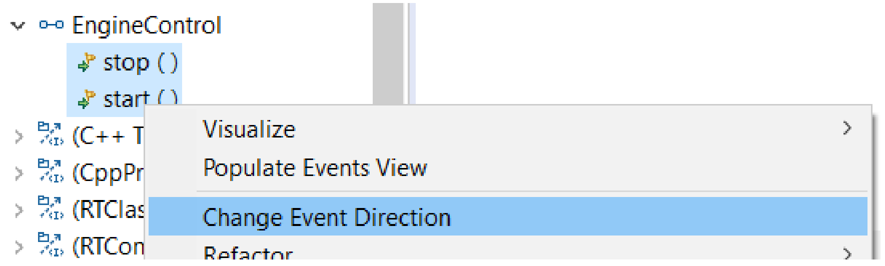
- A new menu in the menu bar with a few useful commands related to libraries
  - Contains the command for launching the TargetRTS wizard (i.e. the menu replaces the TargetRTS menu)



- Commands for importing the Connexis and TCPServer libraries
  - Available if these features have been installed
  - They import the Connexis or TCPServer library projects from the installation into the workspace

Converting Between In Events and Out Events

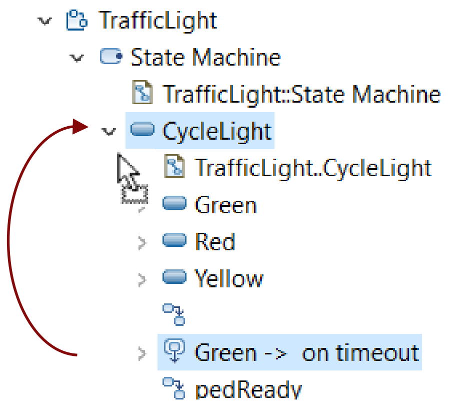
- It’s now possible to convert an In Event to an Out Event or an Out Event to an In Event
  - This can be useful for example if you want to change the conjugation of a port, and need to swap the direction of all events in the protocol of the port

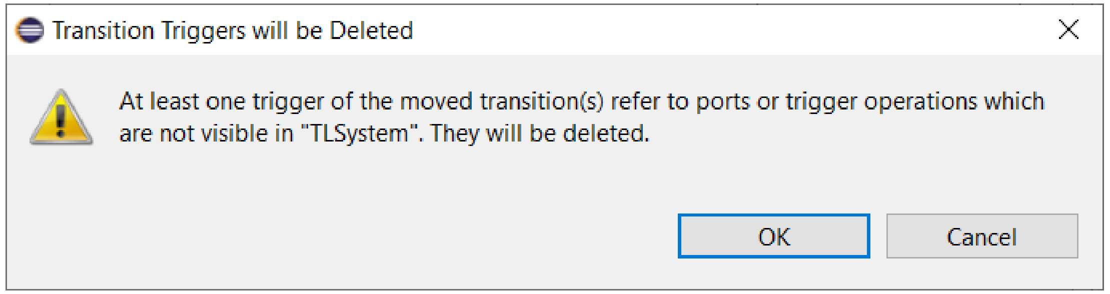


- Use the new “Change Event Direction” command in the Project Explorer context menu of a protocol event (or multiple events)

Moving Transitions in the Project Explorer

- You can now move a transition from one state to another using drag&drop in the Project Explorer
  - Useful for example when refactoring a state machine (moving a transition from a sub state to the container state, or vice versa)
  - Works for transitions in both capsule and passive class state machines
  - The moved transition will initially become a self-transition and a new target state can later be set if needed
- If events/operations referenced by triggers of the moved transition are not available in the target context, a dialog will inform that such triggers will be deleted



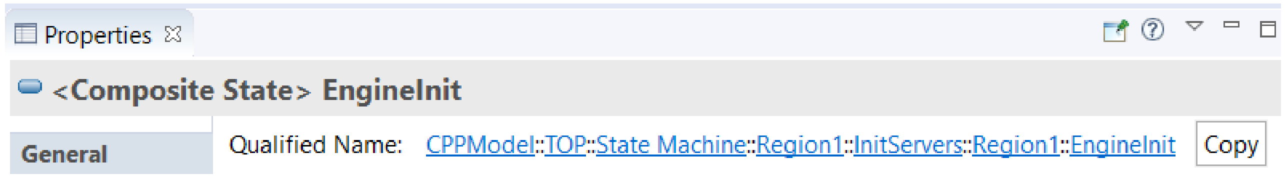

Transition Triggers will be Deleted

At least one trigger of the moved transition(s) refer to ports or trigger operations which are not visible in "TLSystem". They will be deleted.

HCL SOFTWARE

Shortened Qualifiers

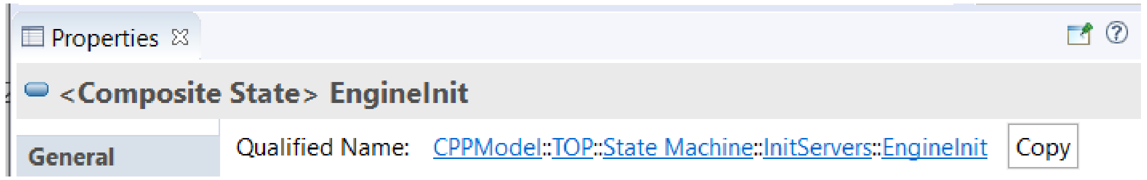
- Fully qualified names now take the current viewpoint into account
  - In the “Capsule Development Viewpoint” hidden elements such as regions are now excluded from the qualifiers
  - Makes fully qualified names shorter, in particular deeply nested states
  - Applies for a number of different views and dialogs (Project Explorer, Find Named Element, Diagram tooltips etc)



<Composite State> EngineInit

Qualified Name: CPPModel::TOP::State Machine::Region1::InitServers::Region1::EngineInit

before



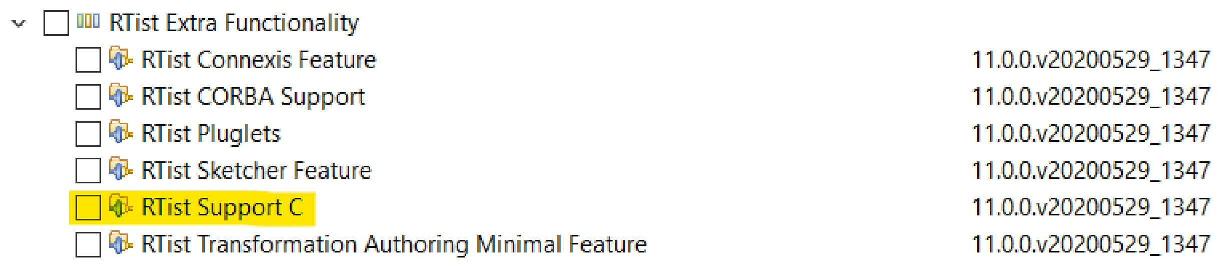
<Composite State> EngineInit

Qualified Name: CPPModel::TOP::State Machine::InitServers::EngineInit

now

Primitive Type Improvements in Search and Type Completion

- Previously duplicates of primitive C++ types appeared when doing search and type completion
  - As a consequence it was easy to by mistake use the wrong type (and also tedious to have to choose each time)
  - The reasons for getting these duplicates have now been addressed
- The “C++ Types” package and “C++ Transformations” profile have been removed from the installation
  - They were obsolete and no longer needed
- The support for C code generation is now an optional choice when installing RTist



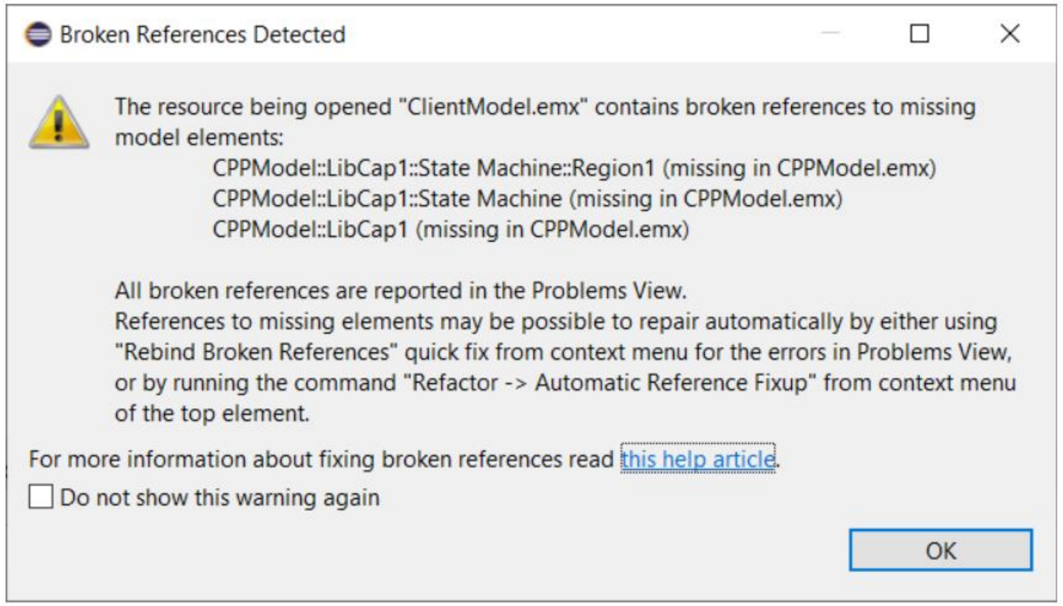
RTist Extra Functionality	
RTist Connexis Feature	11.0.0.v20200529_1347
RTist CORBA Support	11.0.0.v20200529_1347
RTist Pluglets	11.0.0.v20200529_1347
RTist Sketcher Feature	11.0.0.v20200529_1347
RTist Support C	11.0.0.v20200529_1347
RTist Transformation Authoring Minimal Feature	11.0.0.v20200529_1347

- New and updated model fixups are now available for updating existing models for these changes
  - Incorrect type references fixup (updated to fix references to removed C++ Types and, optionally, also C types)
  - Obsolete package import removal (new fixup)
  - Obsolete profile application removal (new fixup)
  - Add missing package imports (new fixup for adding the import of CppPrimitiveDatatypes, in case it’s missing)

Fixing Broken References (1/2)

- The "automatic cleanup" of certain elements with broken references is now removed
  - The preference *RealTime Development - Resource Resolution - Broken references auto-cleanup* has been removed

- By default, when a model with broken references are loaded a warning dialog now appears
  - This dialog replaces the previous Repair Workspace References dialog
  - To repair the broken references, by attempting to rebind them to elements that have been moved to a different file, follow one of the methods suggested by the dialog
  - You can disable this warning dialog using the new preference *Modeling – Resource Resolution – Show warning for broken references when loading a model*
  - Note: A current limitation is that broken references caused by deleted projects will not appear automatically when loading a model. To see such broken references perform the Validate command on the loaded model.



The resource being opened "ClientModel.emx" contains broken references to missing model elements:

CPPModel::LibCap1::State Machine::Region1 (missing in CPPModel.emx)
CPPModel::LibCap1::State Machine (missing in CPPModel.emx)
CPPModel::LibCap1 (missing in CPPModel.emx)

All broken references are reported in the Problems View.
References to missing elements may be possible to repair automatically by either using "Rebind Broken References" quick fix from context menu for the errors in Problems View, or by running the command "Refactor -> Automatic Reference Fixup" from context menu of the top element.

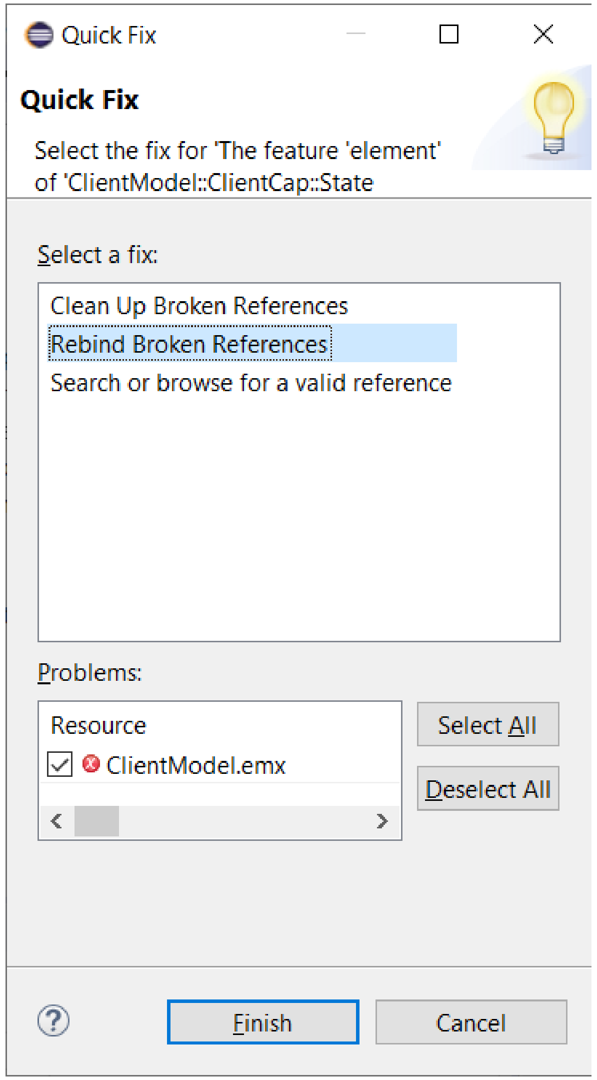
For more information about fixing broken references read this help article.

☐ Do not show this warning again

HCL SOFTWARE

Fixing Broken References (2/2)

- Broken references are listed in the Problems view. Right-click on any of them and run a Quick Fix for fixing broken references:
  - **Rebind Broken References** can fix references to elements that have been moved. It's for example useful after a refactoring that was done without having all referencing models available in the workspace.
  - **Clean Up Broken References** can fix references to elements that have been deleted (or for some other reason do not exist in the workspace). Broken references will be fixed by resetting them (or in some cases, by deleting the element that contains the reference).
  - **Search or browse for a valid reference** can be used to rebind the broken reference to a different target element using the Select Element dialog. This Quick Fix only fixes the selected broken reference.
    Note: To be able to browse to the new target element for the reference you may need to first switch to the Model viewpoint (in case the Capsule Development view point filters out the desired target element).


Quick Fix

Select the fix for 'The feature 'element' of 'ClientModel::ClientCap::State

Select a fix:

Clean Up Broken References
Rebind Broken References
Search or browse for a valid reference

Problems:

Resource
✓ ClientModel.emx

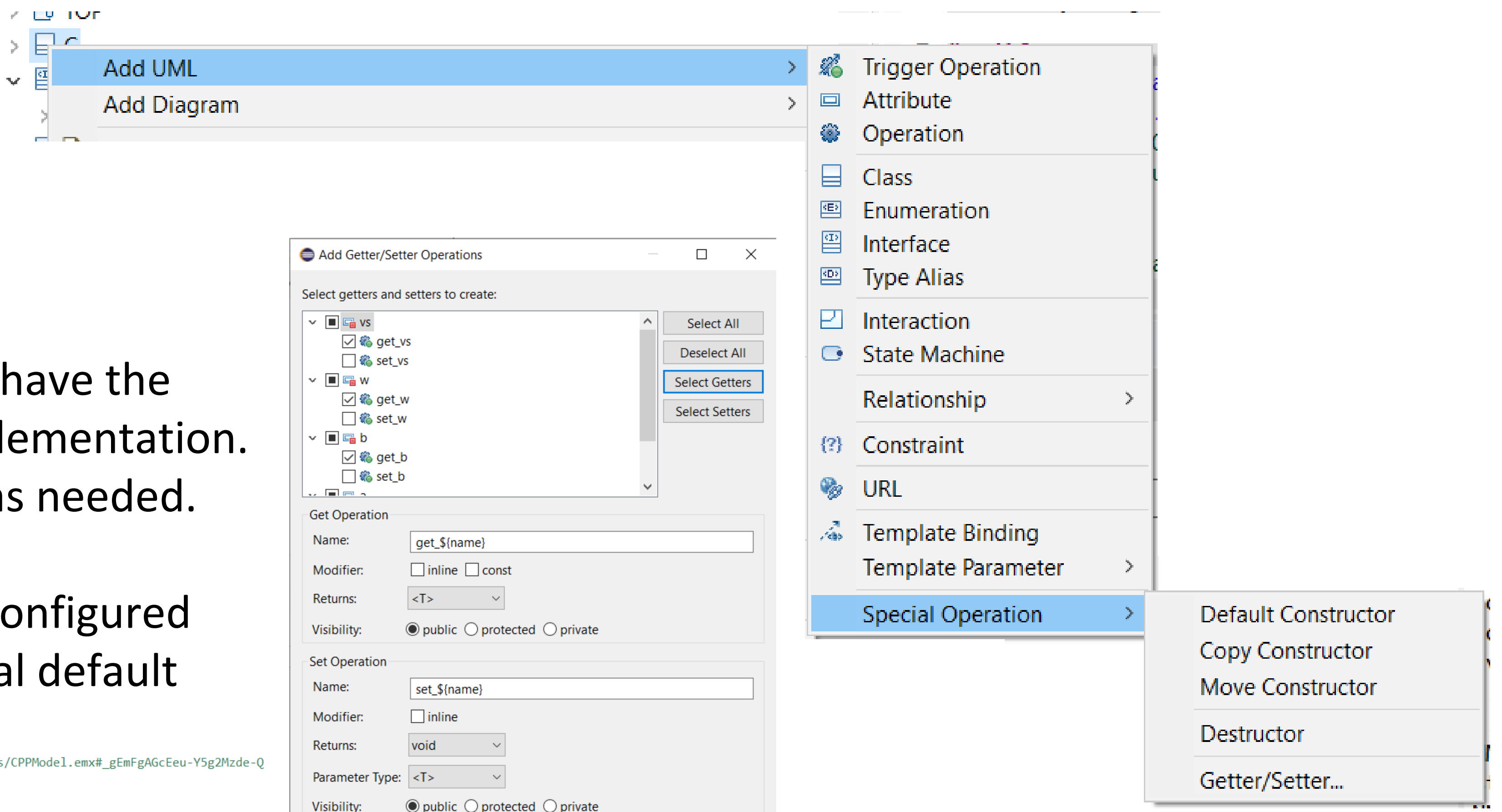
HCL SOFTWARE

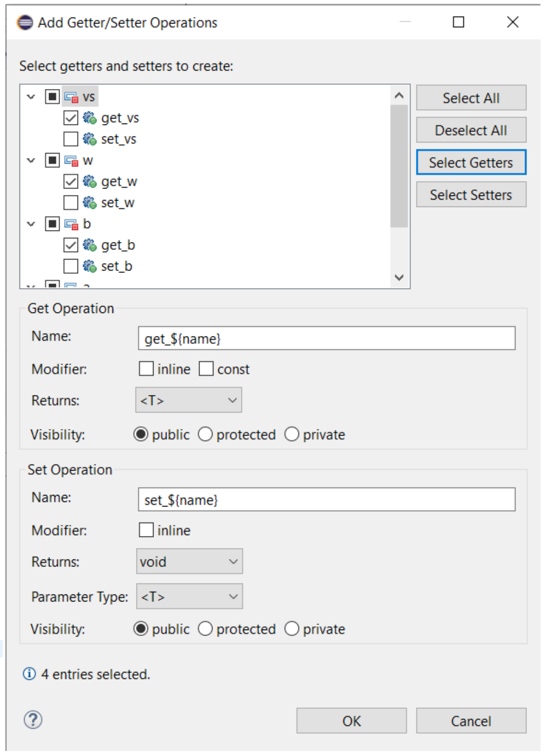
Creating Special Operations

- New commands are now available in the Add UML context menu for creating “special operations” on a capsule or class
  - Constructors (default, copy, move)
  - Destructor
  - Getter/Setter operations

- Constructors and destructors will have the correct C++ signature, but no implementation. Write your own implementation as needed.

- Getter/Setter operations can be configured using a dialog, and will have typical default implementations





Add Getter/Setter Operations

Select getters and setters to create:

- vs
  - [✓] get_vs
  - [ ] set_vs
- w
  - [✓] get_w
  - [ ] set_w
- b
  - [✓] get_b
  - [ ] set_b

Select All | Deselect All | Select Getters | Select Setters

Get Operation

Name: get_${name}

Modifier: [ ] inline [ ] const

Returns: <T>

Visibility: [✓ public] protected private

Set Operation

Name: set_${name}

Modifier: [ ] inline

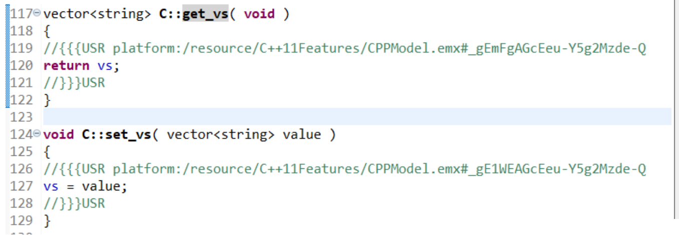
Returns: void

Parameter Type: <T>

Visibility: [✓ public] protected private

4 entries selected.

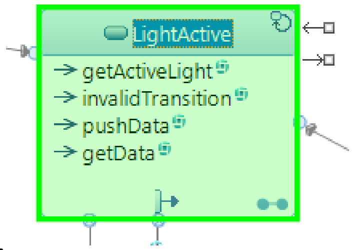
OK | Cancel



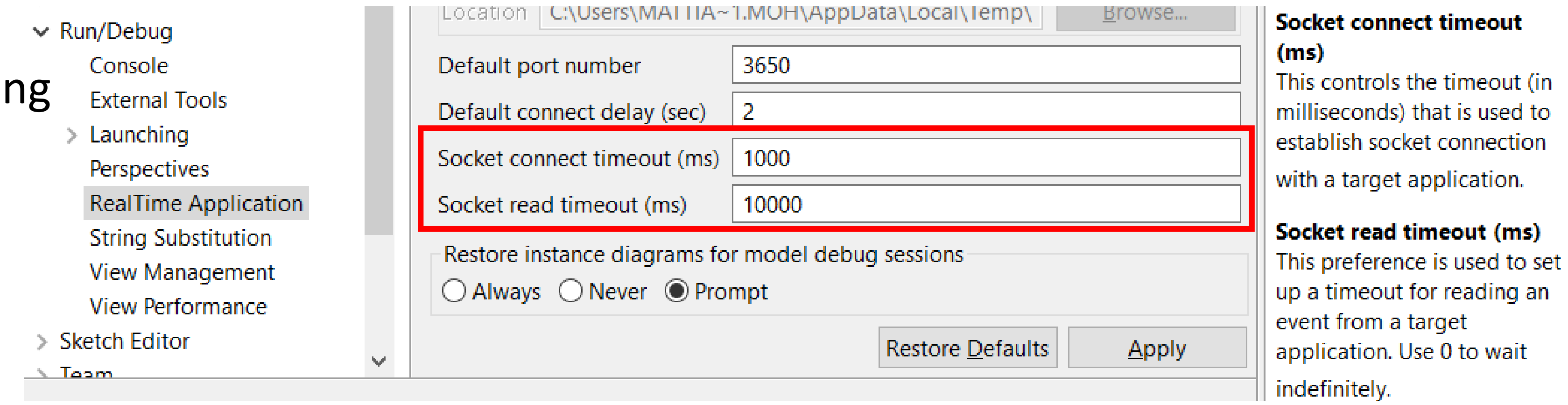
```
vector<string> C::get_vs( void )
{
//{{{USR platform:/resource/C++11Features/CPPModel.emx#_gEmFgAGcEeu-Y5g2Mzde-Q
return vs;
//}}}USR
}

void C::set_vs( vector<string> value )
{
//{{{USR platform:/resource/C++11Features/CPPModel.emx#_gE1WEAGcEeu-Y5g2Mzde-Q
vs = value;
//}}}USR
}
```

HCL SOFTWARE

Model Debugger Improvements (1/3)



- State instance diagram for a sub state machine can now be opened by double-click
  - Previously the command "Open State Machine Diagram" (Ctrl +T) was the only easy option
  - Now it also works to double-click on the state name (same as when the model is not being debugged)
- The socket timeouts used for communicating with the debugged application are now configurable
  - New preferences on the *Run/Debug – RealTime Application* preference page
  - When debugging remotely over slow connections a too small socket read timeout can cause the debug session to terminate unexpectedly
- The performance of configuring the Trace Editor was significantly improved
  - Now by default only the listed elements are traced



Location C:\Users\MATTIA~1.MOH\AppData\Local\Temp\

Default port number	3650
Default connect delay (sec)	2
Socket connect timeout (ms)	1000
Socket read timeout (ms)	10000

Restore instance diagrams for model debug sessions
Always Never [✓ Prompt]

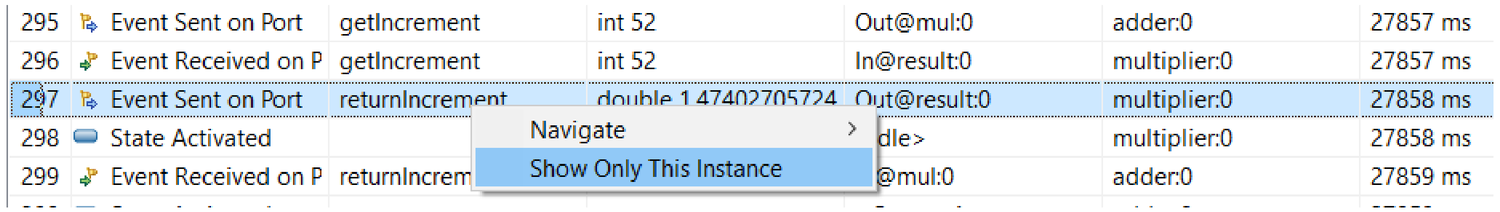
Socket connect timeout (ms)
This controls the timeout (in milliseconds) that is used to establish socket connection with a target application.

Socket read timeout (ms)
This preference is used to set up a timeout for reading an event from a target application. Use 0 to wait indefinitely.

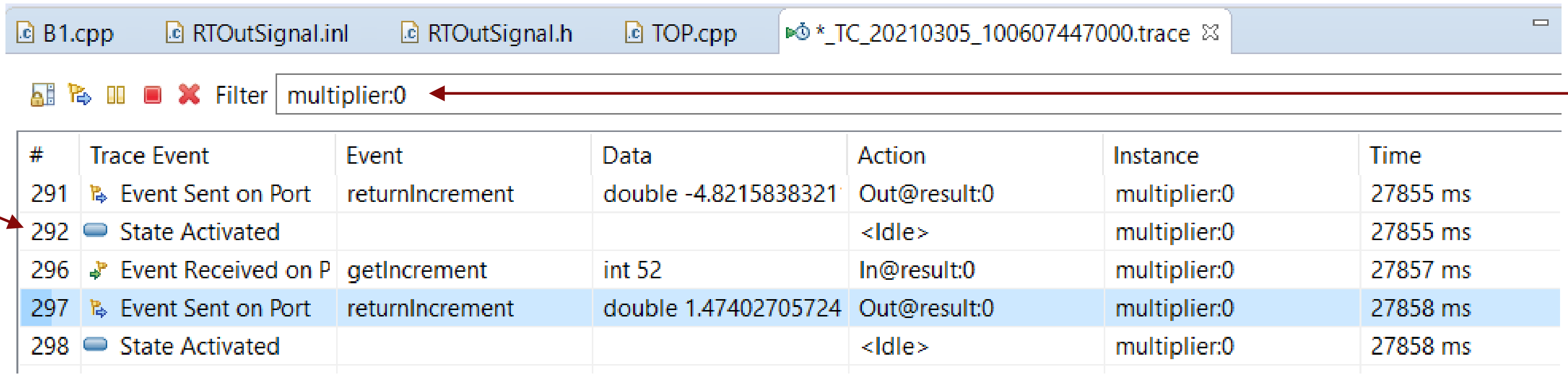
HCL SOFTWARE

Model Debugger Improvements (2/3)

- A new command “Show Only This Instance” in the Trace Editor context menu
  - Makes it easy to filter the trace to only see trace events related to the selected capsule instance
  - For example useful in order to find out which state a capsule instance is in when sending or receiving an event



295	Event Sent on Port	getIncrement	int 52	Out@mul:0	adder:0	27857 ms
296	Event Received on P	getIncrement	int 52	In@result:0	multiplier:0	27857 ms
297	Event Sent on Port	returnIncrement	double 1.47402705724	Out@result:0	multiplier:0	27858 ms
298	State Activated			dle>	multiplier:0	27858 ms
299	Event Received on P	returnIncrem		@mul:0	adder:0	27859 ms



Filter: multiplier:0

#	Trace Event	Event	Data	Action	Instance	Time
291	Event Sent on Port	returnIncrement	double -4.8215838321	Out@result:0	multiplier:0	27855 ms
292	State Activated			<Idle>	multiplier:0	27855 ms
296	Event Received on P	getIncrement	int 52	In@result:0	multiplier:0	27857 ms
297	Event Sent on Port	returnIncrement	double 1.47402705724	Out@result:0	multiplier:0	27858 ms
298	State Activated			<Idle>	multiplier:0	27858 ms

Active state when sending the event

Remove the filter to again show all trace events

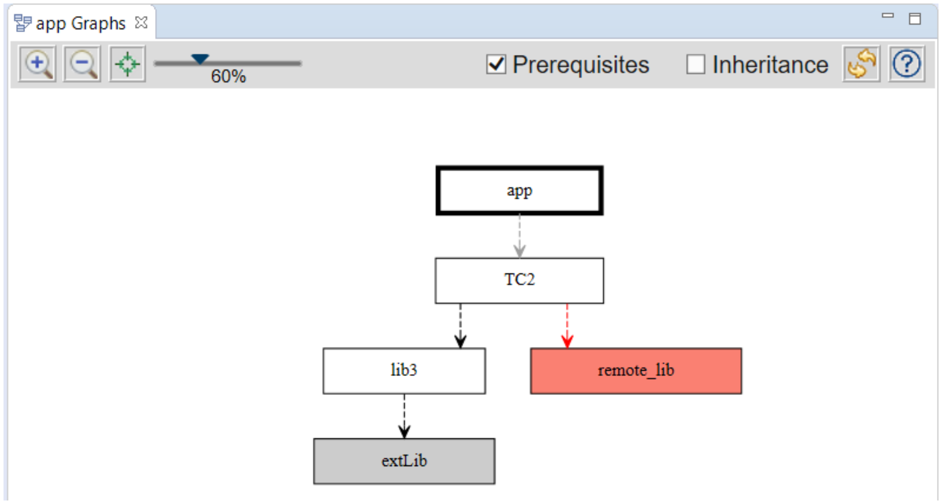
HCL SOFTWARE

Model Debugger Improvements (3/3)

- Tracing everything related to a capsule
  - Adding a capsule to the list of elements to trace is now a shorthand for adding all its ports and states (including sub states). Previously only incarnation of the capsule was traced in this case.
- Tracing internal transitions
  - Now a state activation event will not be reported when an internal transition executes (since the state remains active while it runs)

Graphic Visualization of TC Relationships

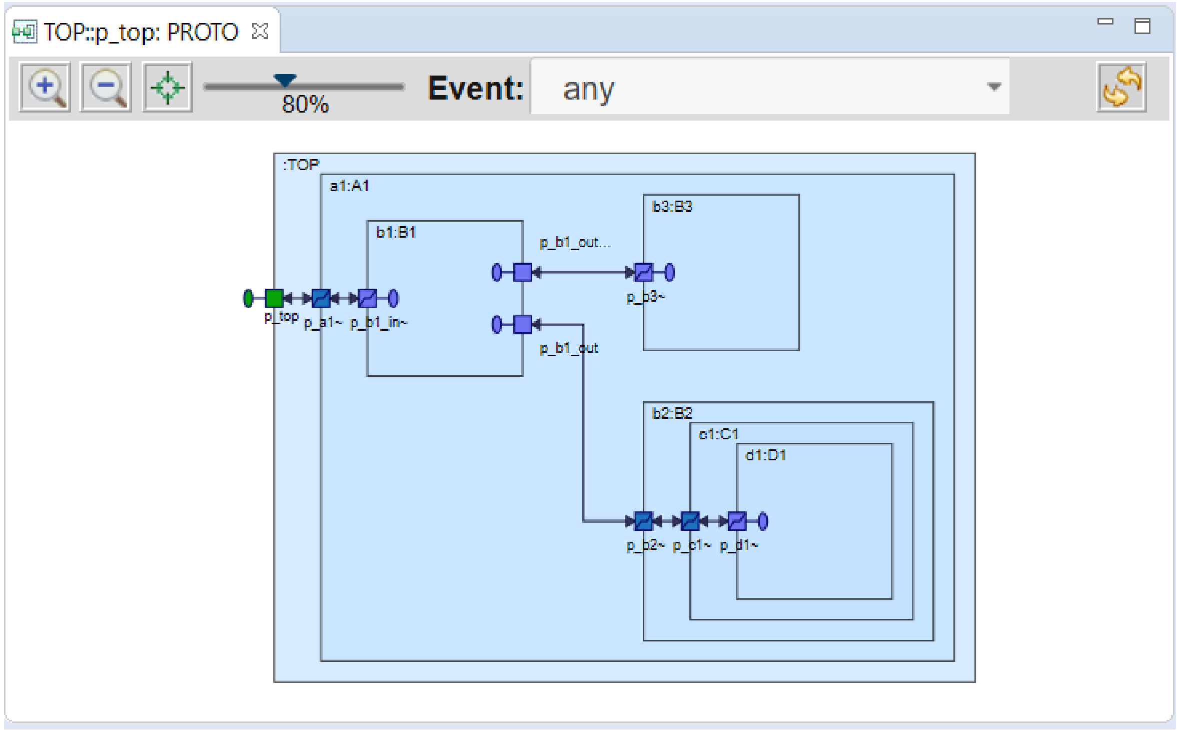
- The possibility to visualize TC relationships that was available in version 10.2 for TCs in the old file format, is now also supported for TCs in the new .tcjs file format
  - A new command **Visualize – Explore in Graphs** is available in the Project Explorer context menu of a TC
- The visualization uses Graphviz as an external tool for doing the graph layout
  - Graphviz must be separately installed. Specify the path to its "dot" utility in *Preferences – RealTime Development - Visualization*
- The graph is rendered using SVG in an embedded web browser
  - Support for panning and zooming
  - You can search for texts in the rendered diagram using Ctrl+F
- Prerequisites and/or inheritance relationships can be shown
- Double-click on TCs in the graph to open the TC editor



HCL SOFTWARE

Event Routing Visualization

- Visualize graphically how a protocol event can be routed in the composite structure of a capsule
  - A new command **Visualize – Event Routing** is available in the context menu of a port (in Project Explorer, diagrams or Code/CDT Editor)
  - Select an event in the drop down menu to see how that event can be routed in the composite structure
- Explore the composite structure interactively
  - Use commands "Go Inside" and "Go Outside" in the ports' context menu in the diagram
- If there are multiple capsule parts typed by the same capsule, they will be replaced by hyperlinks in the diagram to save space

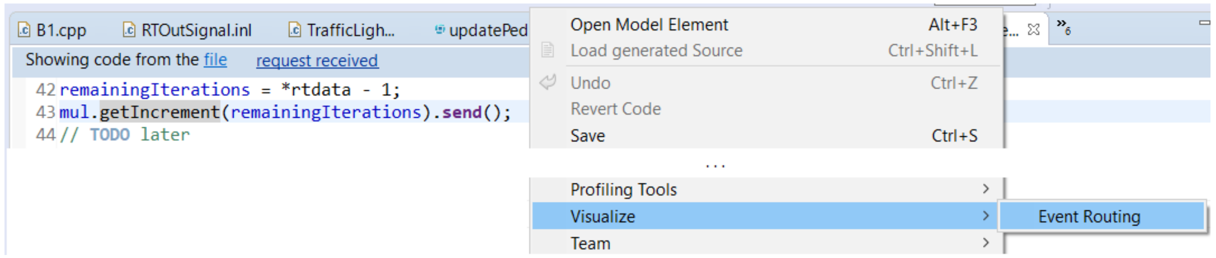


- The graph uses SVG so panning, zooming and text search is supported

Navigation from Code Sending an Event to the Transition it Triggers

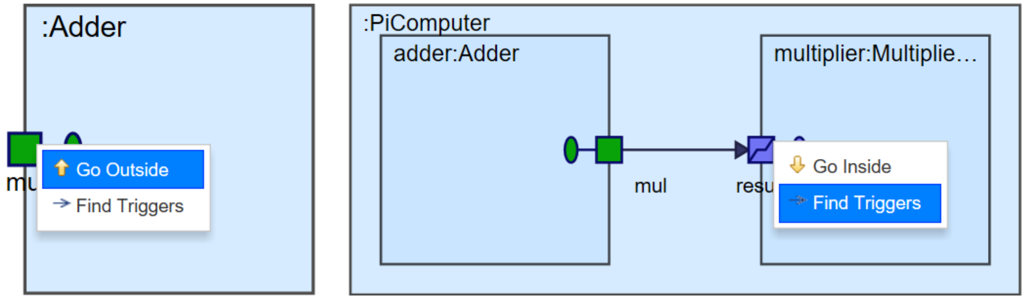
The Event Routing Visualization feature makes it easy to navigate from a place in the code where an event is sent, to the transition(s) that may receive it:

1. Right-click in the Code or CDT editor on a line where an event is sent (or invoked or replied to) and open the Event Routing Visualization graph.

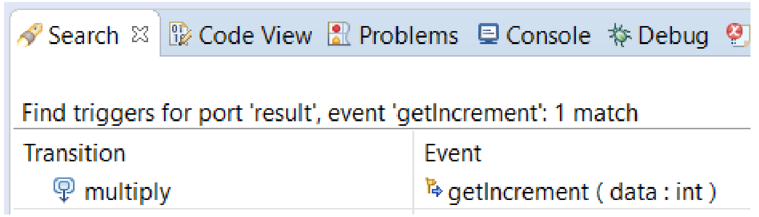


```
remainingIterations = *rtdata - 1;
mul.getIncrement(remainingIterations).send();
// TODO later
```

2. Follow the arrows from the green port (the port on which the event is sent) to find the possible ports that may receive the event. To see also arrows to ports on enclosing capsules, perform **Go Outside** on the green port to reveal more of the composite structure.



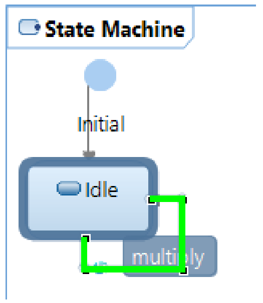
3. Right-click on such a port and perform the command **Find Triggers**. All transitions which may trigger when the event arrives at the port will be shown in the Search View.



Find triggers for port 'result', event 'getIncrement': 1 match

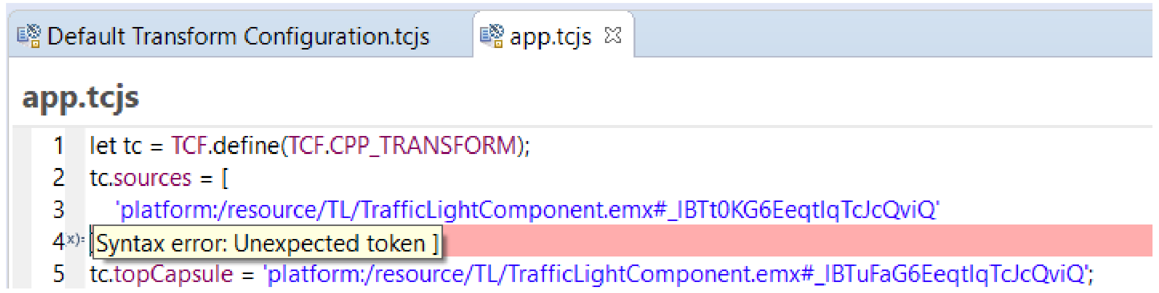
Transition	Event
multiply	getIncrement (data : int)

4. Double-click on a found transition to navigate to it in a diagram



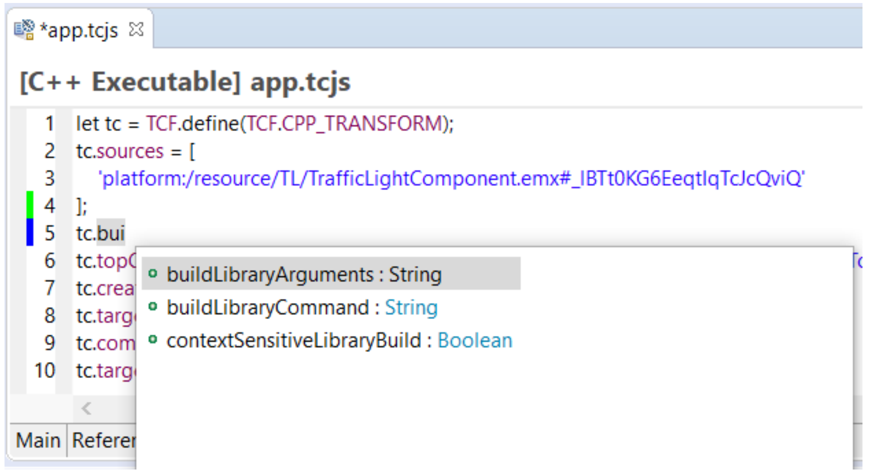
Transformation Configuration Editor Improvements (1/2)

- Syntax highlighting is now available in the Code tab
  - JavaScript parse errors are shown with red color and margin marker tooltip



```
let tc = TCF.define(TCF.CPP_TRANSFORM);
tc.sources = [
    'platform:/resource/TL/TrafficLightComponent.emx#_IBTt0KG6EeqtIqTcJcQviQ'
[Syntax error: Unexpected token ]
tc.topCapsule = 'platform:/resource/TL/TrafficLightComponent.emx#_IBTuFaG6EeqtIqTcJcQviQ';
```

- Content assist ("code completion") is now available in the Code tab
  - Triggered automatically when typing certain "key" characters ('.', '=' etc)
  - Can also be manually triggered using Ctrl + Space
  - Controlled by a new preference *RealTime Development – Transformation Configuration Editor – Content Assist – Enable auto activation*



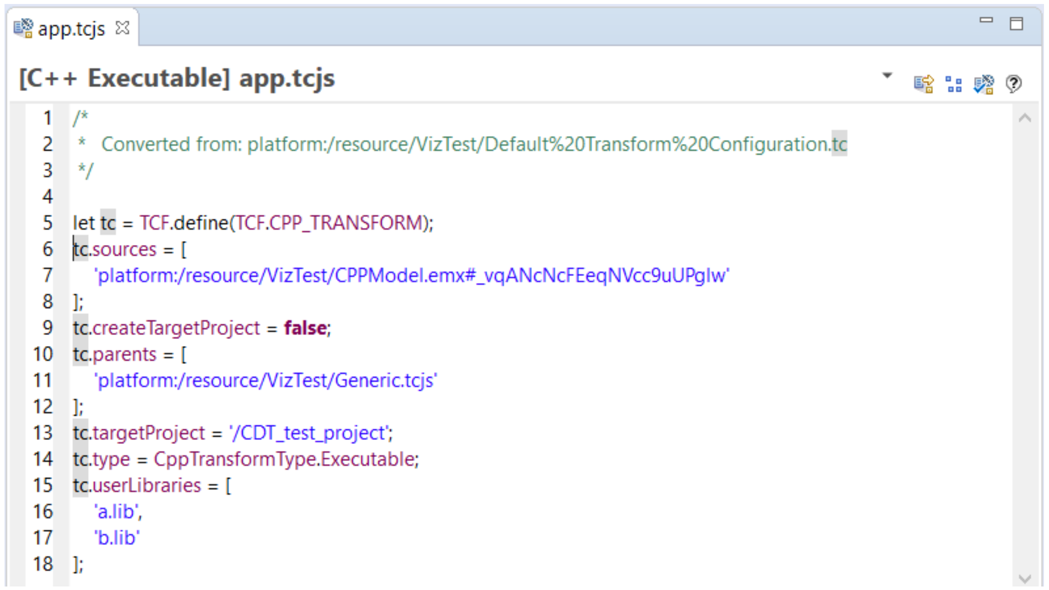
```
let tc = TCF.define(TCF.CPP_TRANSFORM);
tc.sources = [
    'platform:/resource/TL/TrafficLightComponent.emx#_IBTt0KG6EeqtIqTcJcQviQ'
];
tc.bui
tc.topC
tc.crea
tc.targ
tc.com
tc.targ
```

```
buildLibraryArguments : String
buildLibraryCommand : String
contextSensitiveLibraryBuild : Boolean
```

HCL SOFTWARE

Transformation Configuration Editor Improvements (2/2)

- Marking of occurrences of selected element in the Code tab
  - Controlled by a new preference *RealTime Development – Transformation Configuration Editor – Mark Occurrences – Mark occurrences of selected element*
- Incremental Find is now supported in the Code tab (with the same keyboard bindings as in other Eclipse text editors)

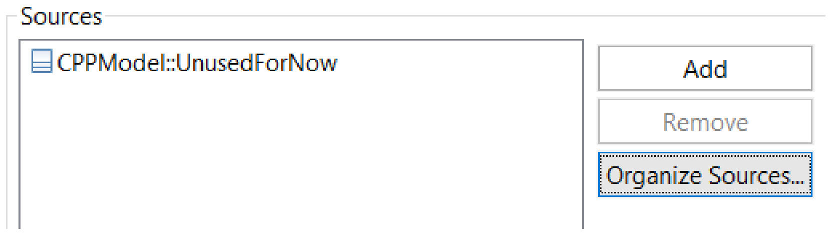

app.tcjs

[C++ Executable] app.tcjs

```
/*
 *  Converted from: platform:/resource/VizTest/Default%20Transform%20Configuration.tc
 */

let tc = TCF.define(TCF.CPP_TRANSFORM);
tc.sources = [
    'platform:/resource/VizTest/CPPModel.emx#_vqANcNcFEeqNVcc9uUPglw'
];
tc.createTargetProject = false;
tc.parents = [
    'platform:/resource/VizTest/Generic.tcjs'
];
tc.targetProject = '/CDT_test_project';
tc.type = CppTransformType.Executable;
tc.userLibraries = [
    'a.lib',
    'b.lib'
];
```

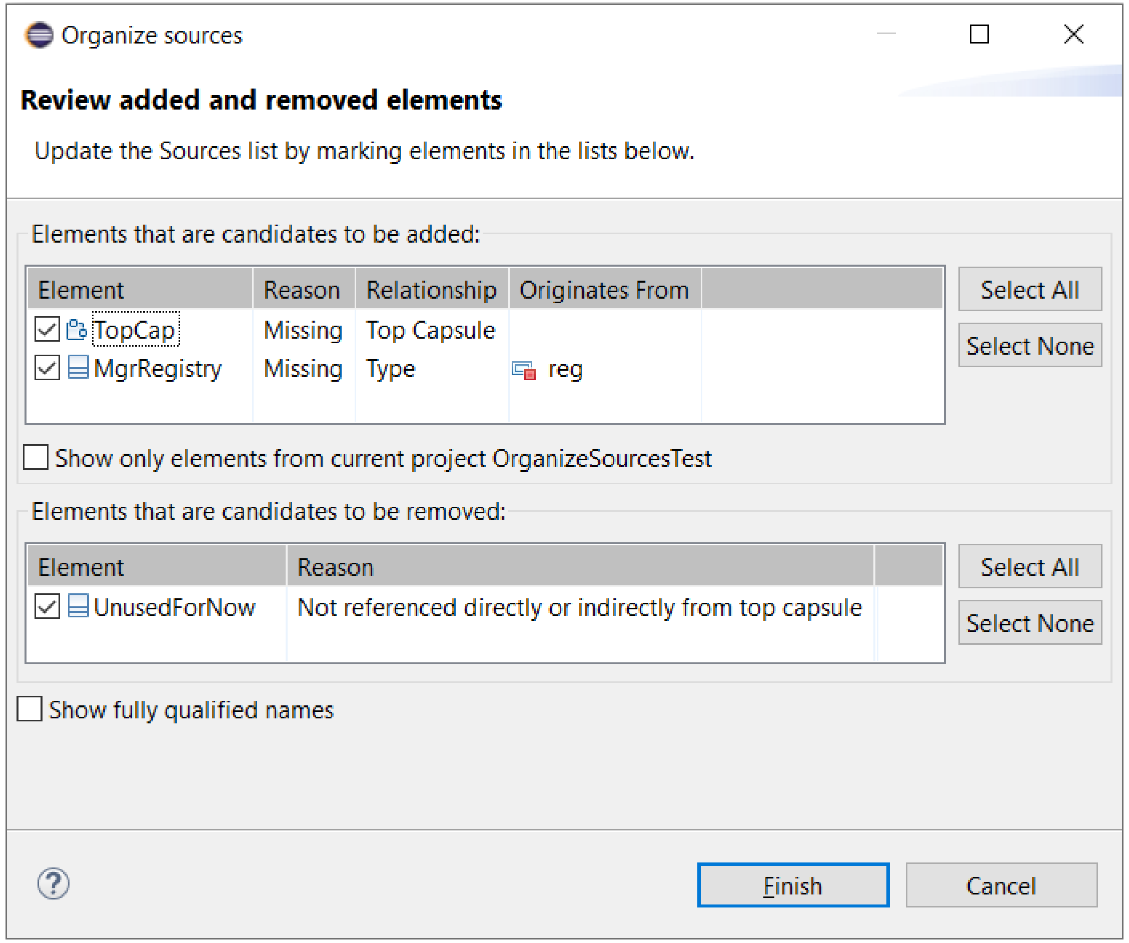
Organize Sources

- A new Organize Sources dialog is provided to assist in setting up the Sources of a TC optimally



Sources

CPPModel::UnusedForNow



Organize sources

Review added and removed elements

Update the Sources list by marking elements in the lists below.

Elements that are candidates to be added:

Element	Reason	Relationship	Originates From
✓ TopCap	Missing	Top Capsule	
✓ MgrRegistry	Missing	Type	reg

☐ Show only elements from current project OrganizeSourcesTest

Elements that are candidates to be removed:

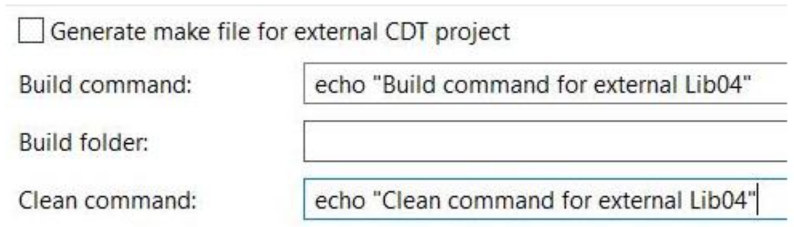
Element	Reason
✓ UnusedForNow	Not referenced directly or indirectly from top capsule

☐ Show fully qualified names

- This is an improved and simplified implementation compared to the similar feature in 10.2
- Elements are proposed to be added to or removed from the Sources list to ensure that no unnecessary elements will be built by the TC. The suggestions are made by analyzing references in the model.
- This feature is also available from the command-line by means of a new model compiler command-line argument `--organizeSources`
- It can also be launched from the TC Editor toolbar menu (useful for TCs edited only textually in the Code tab)

Cleaning TCs

- External Library TCs have a new "Clean Command" property
  - Use for example "make clean", invoke a "clean script" or something else
- When it has been specified it's possible to use the Clean command also on external library TCs
  - Currently this only works when cleaning from the command-line (using the makefile generated by the model compiler)
- When build variants are used, the Clean dialog now shows the Build Variants user interface
  - Allows setting the build variant settings which may influence on the behavior of clean, and clean build

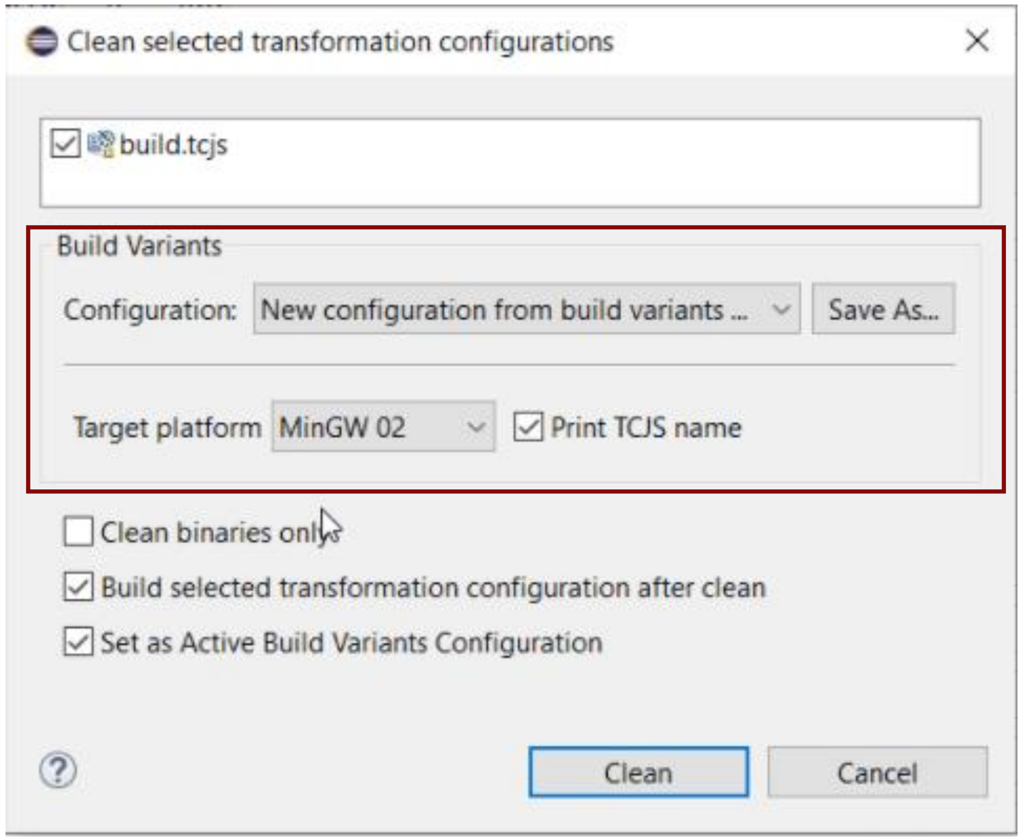


☐ Generate make file for external CDT project

Build command: echo "Build command for external Lib04"

Build folder:

Clean command: echo "Clean command for external Lib04"



Clean selected transformation configurations

☑ build.tcjs

Build Variants

Configuration: New configuration from build variants ...

Target platform: MinGW 02 ☑ Print TCJS name

☐ Clean binaries only

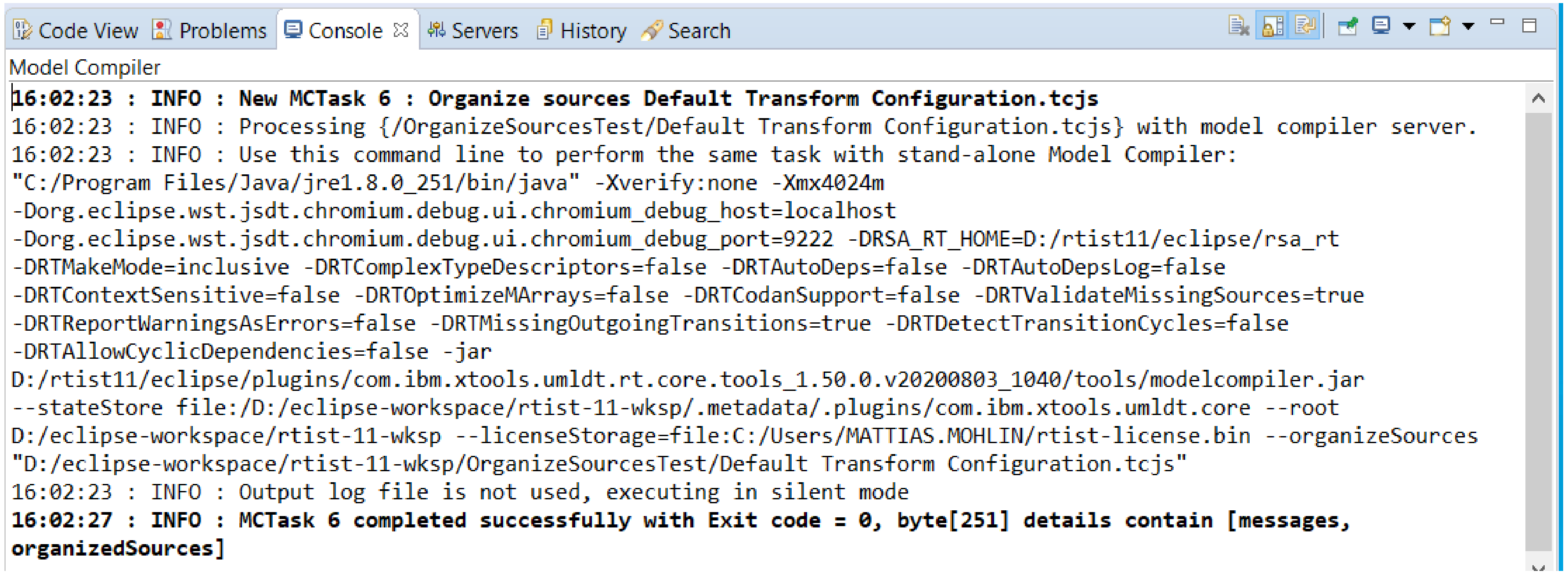
☑ Build selected transformation configuration after clean

☑ Set as Active Build Variants Configuration

HCL SOFTWARE

Model Compiler Console

- A new ”Model Compiler” console is now available
  - Helps troubleshooting commands that involves calls to the model compiler
  - Makes it easier to know how a model-compiler command should be invoked from the command-line



```
16:02:23 : INFO : New MCTask 6 : Organize sources Default Transform Configuration.tcjs
16:02:23 : INFO : Processing {/OrganizeSourcesTest/Default Transform Configuration.tcjs} with model compiler server.
16:02:23 : INFO : Use this command line to perform the same task with stand-alone Model Compiler:
"C:/Program Files/Java/jre1.8.0_251/bin/java" -Xverify:none -Xmx4024m
-Dorg.eclipse.wst.jsdt.chromium.debug.ui.chromium_debug_host=localhost
-Dorg.eclipse.wst.jsdt.chromium.debug.ui.chromium_debug_port=9222 -DRSA_RT_HOME=D:/rtist11/eclipse/rsa_rt
-DRTMakeMode=inclusive -DRTComplexTypeDescriptors=false -DRTAutoDeps=false -DRTAutoDepsLog=false
-DRTContextSensitive=false -DRTOptimizeMArrays=false -DRTCodanSupport=false -DRTValidateMissingSources=true
-DRTReportWarningsAsErrors=false -DRTMissingOutgoingTransitions=true -DRTDetectTransitionCycles=false
-DRTAllowCyclicDependencies=false -jar
D:/rtist11/eclipse/plugins/com.ibm.xtools.umldt.rt.core.tools_1.50.0.v20200803_1040/tools/modelcompiler.jar
--stateStore file:/D:/eclipse-workspace/rtist-11-wksp/.metadata/.plugins/com.ibm.xtools.umldt.core --root
D:/eclipse-workspace/rtist-11-wksp --licenseStorage=file:C:/Users/MATTIAS.MOHLIN/rtist-license.bin --organizeSources
"D:/eclipse-workspace/rtist-11-wksp/OrganizeSourcesTest/Default Transform Configuration.tcjs"
16:02:23 : INFO : Output log file is not used, executing in silent mode
16:02:27 : INFO : MCTask 6 completed successfully with Exit code = 0, byte[251] details contain [messages,
organizedSources]
```

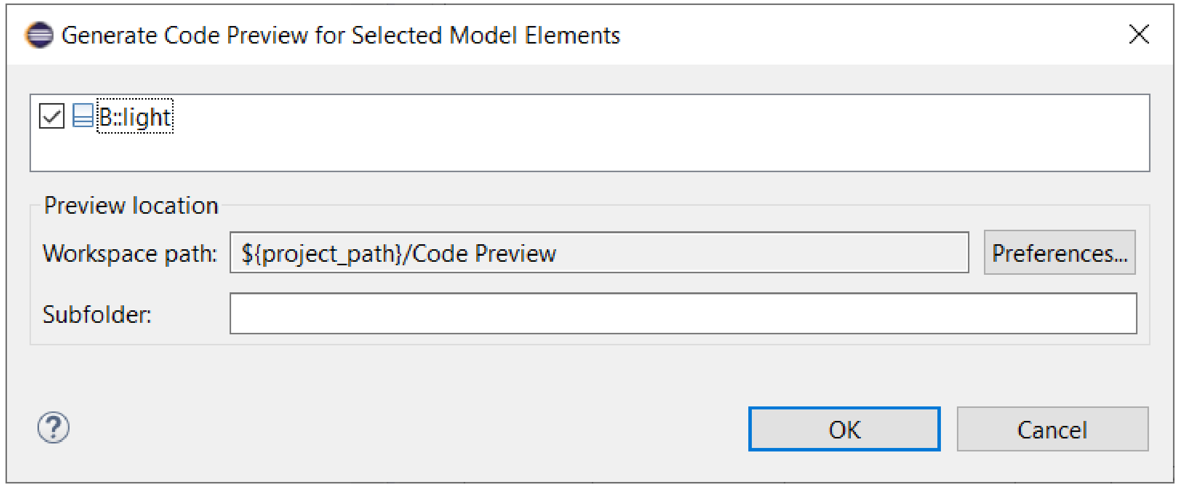
HCL SOFTWARE

Incremental Compilation for Models with External Code

- This feature worked in the classic builder by means of a Perl script (rtcppdep.pl)
- The model compiler now also supports this, by relying on the compiler's ability to generate dependency files
  - Currently only supported for GCC compilers, and only for recursive make files
- The file `libset.mk` in the TargetRTS must be updated to enable this feature
  - The variable $(DEP_OPTS) should be added to LIBSETCCFLAGS
  - Note that $(DEP_OPTS) is a new variable in the file `default.mk`. In this file is also another variable $(DEP_EXT) which specifies the file extension for dependency files (.d). If this variable is not defined, dependency files will not be included in the generated makefile.
- Note: A clean build will take slightly longer when this feature is enabled (due to dependency file generation), but subsequent builds will be faster since only the minimal number of .cpp files will be recompiled when something in the external code has been changed.

Code Preview (1/2)

- A preview of what generated code will look like can now be generated
  - Use the new command **Generate Code Preview** in the Project Explorer context menu of a model element
  - The default keybinding is `Alt+Shift+E`
- Preview code is by default placed in a "Code Preview" folder in the current project
  - You can change this location in the Preferences, and you can also specify a subfolder to be used
- Old versions of preview code files are marked with ~
  - You can compare old and new file versions using **Compare With – Each Other** to easily understand how changes in the model have affected the generated code

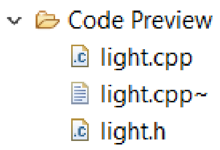

Generate Code Preview for Selected Model Elements

[✓] B::light

Preview location

Workspace path: ${project_path}/Code Preview

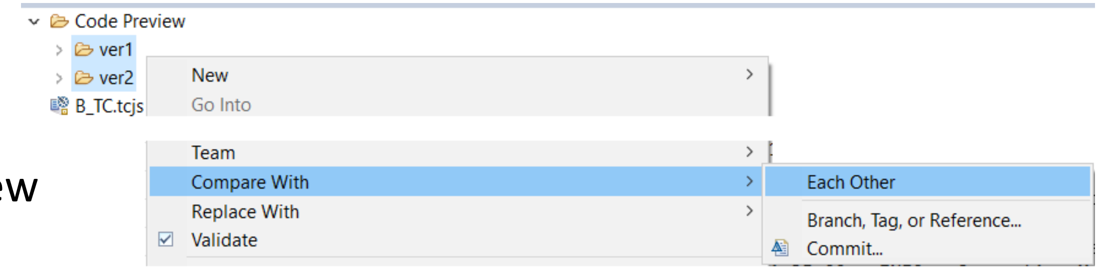
Subfolder:


Code Preview
light.cpp
light.cpp~
light.h

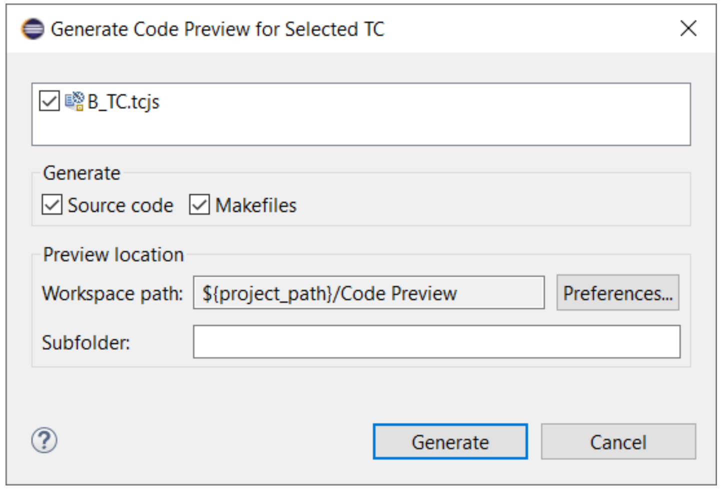
HCL SOFTWARE

Code Preview (2/2)

- You can also use the Code Preview feature as a way to compare generated code, as an alternative to comparing models. For example you can compare the contents of two Git branches:
  - First checkout the first branch and generate a code preview into one subfolder
  - Then checkout the second branch and generate a code preview into another subfolder
  - Finally select the two subfolders and perform the command **Compare With – Each Other**



- Both code and makefiles can be previewed and compared in the same way
  - You can run the Code Preview command on a TC and select if generated source code or makefiles or both should be generated
  - Build variants are also supported by this dialog


Generate Code Preview for Selected TC

☑ B_TC.tcjs

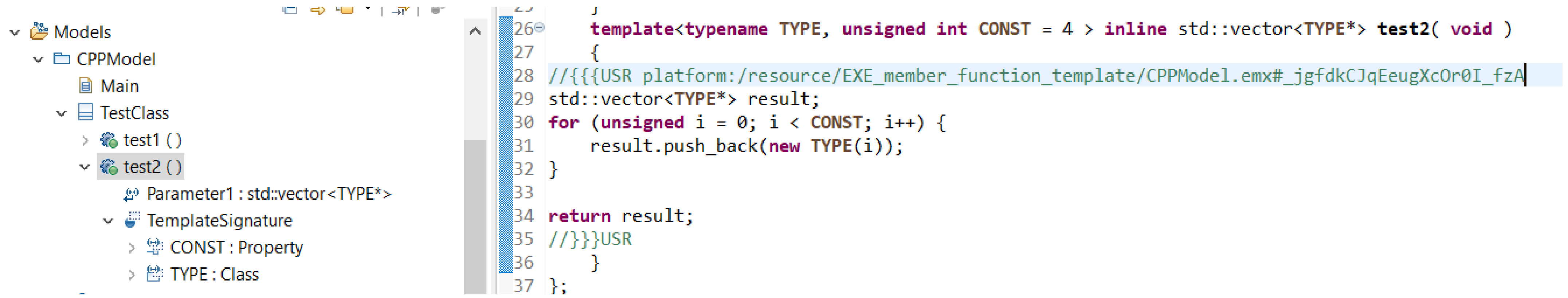
Generate
☑ Source code ☑ Makefiles

Preview location
Workspace path: ${project_path}/Code Preview
Subfolder:

- To remove all code preview when you no longer need it, use the command **File – Remove All Code Preview**

Improved Support for Templates (1/2)

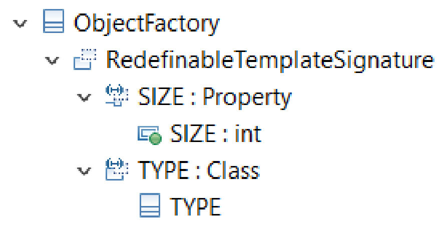
- The model compiler now supports member functions with template parameters
  - Both type and non-type template parameters are supported
  - Note that most compilers require such operations to be declared as **inline**



- Models
  - CPPModel
    - Main
    - TestClass
      - test1 ( )
      - test2 ( )
        - Parameter1 : std::vector<TYPE*>
        - TemplateSignature
          - CONST : Property
          - TYPE : Class

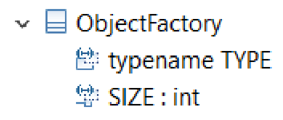
```
    template<typename TYPE, unsigned int CONST = 4 > inline std::vector<TYPE*> test2( void )
    {
//{{{USR platform:/resource/EXE_member_function_template/CPPModel.emx#_jgfdkCJqEeugXcOr0I_fzA
std::vector<TYPE*> result;
for (unsigned i = 0; i < CONST; i++) {
    result.push_back(new TYPE(i));
}

return result;
//}}}USR
    }
};
```

- The Project Explorer representation of template parameters is now more compact
  - Easier to recognize type template parameters by means of the word “typename” in the label



- ObjectFactory
  - RedefinableTemplateSignature
    - SIZE : Property
      - SIZE : int
    - TYPE : Class
      - TYPE

before

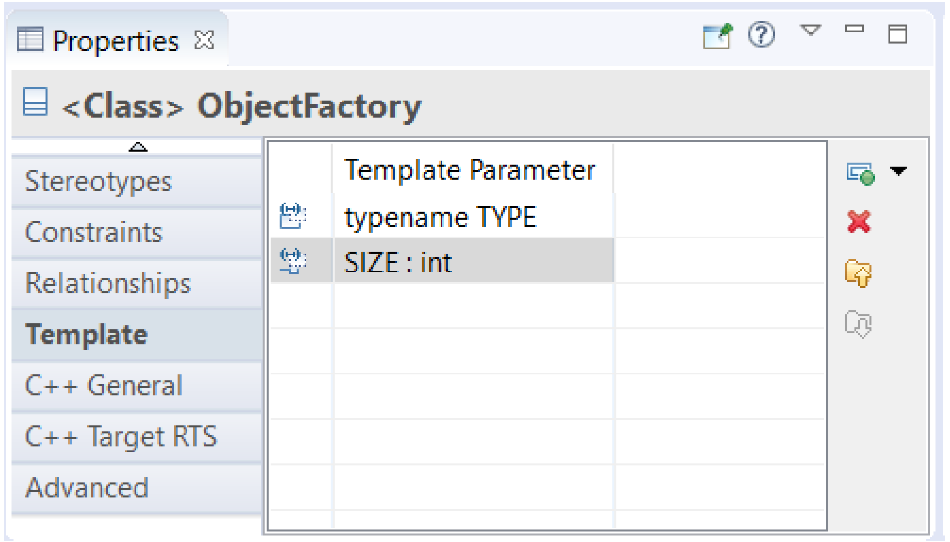


- ObjectFactory
  - typename TYPE
  - SIZE : int

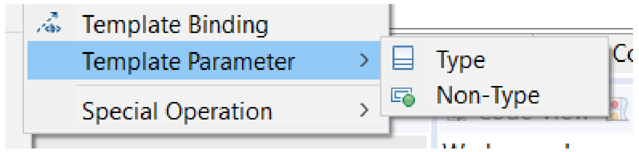
now

Improved Support for Templates (2/2)

- The Templates page of the Properties view has been much improved
  - Consistent look-and-feel with other property pages
  - Support for reordering template parameters using either up/down buttons or by drag&drop
  - Improved textual representation of template parameters, more aligned with the C++ syntax
- Improved context menu for creating template parameters
  - Now aligned with C++ terminology (type and non-type template parameters)



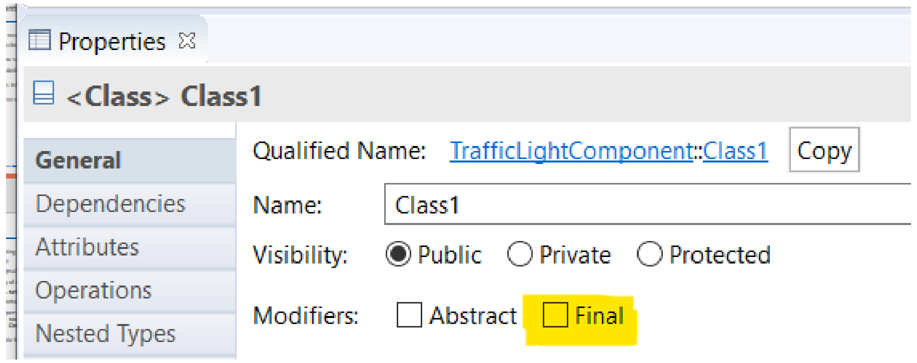
Template Parameter
typename TYPE
SIZE : int



HCL SOFTWARE

C++ 11 Improvements (1/4)

- The **final** modifier can now be used on classes, capsules and operations
  - Use it to specify that a class/capsule cannot be inherited from, or an operation cannot be redefined



<Class> Class1

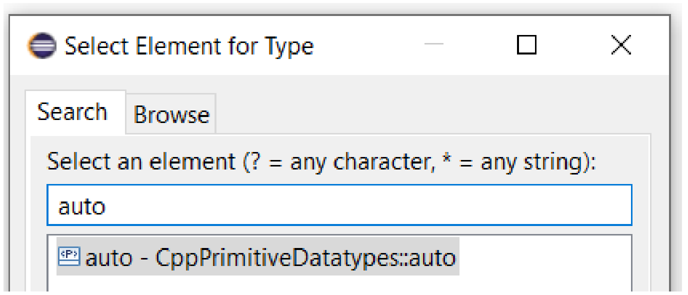
Qualified Name: TrafficLightComponent::Class1

Name: Class1

Visibility: [✓ Public] Private Protected

Modifiers: Abstract Final

- **auto** attributes are now supported by the model compiler
  - The C++ compiler will deduce the type of the corresponding member variable from its initializer value
  - Type completion now works for auto, and it can be selected as any other type



Select an element (? = any character, * = any string):

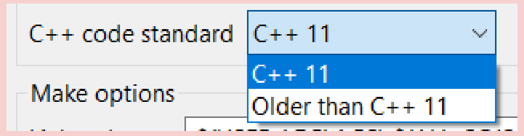
auto

auto - CppPrimitiveDatatypes::auto

- Attributes with in-class initialization (brace or equal) are no longer also initialized in the constructor
- Some C++ 11 types were missing in the CppPrimitiveDatatypes package and have now been added
  - long long
  - unsigned long long

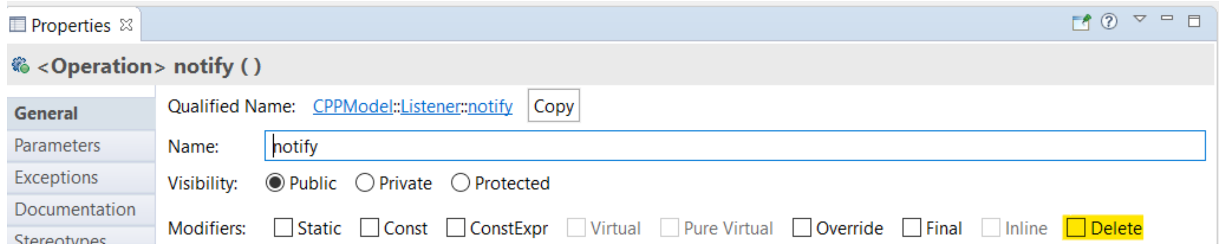
C++ 11 Improvements (2/4)

- The **override** keyword is now specified for functions that redefine virtual functions from the TargetRTS. Also, the **nullptr** constant is now used instead of 0 for representing a null pointer.
  - Both generated code and the TargetRTS code was changed
  - Avoids warnings when compiling generated code
  - For GCC compilers, the warnings `–Wsuggest-override` and `–Wzero-as-null-pointer-constant` are now set when compiling to detect these problems also in user code snippets
  - **Note:** Starting with these changes it's now required to use a C++ 11 compiler for compiling generated code. You can use the new preference *RealTime Development – Build/Transformation – C++ code standard* if you need to compile generated code with a compiler that doesn't support C++ 11.

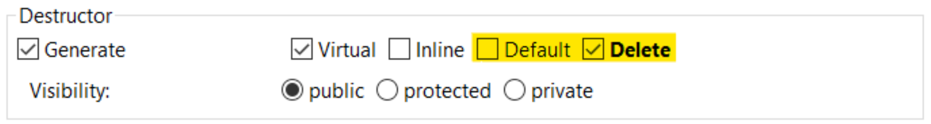


C++ code standard: C++ 11
Options: [✓ C++ 11] / Older than C++ 11
Make options

- Operations can now be marked as deleted, and destructors can be marked as default or delete



Properties
<Operation> notify ()
General | Parameters | Exceptions | Documentation | Stereotypes
Qualified Name: CPPModel::Listener::notify Copy
Name: notify
Visibility: [✓ Public] Private Protected
Modifiers: ☐ Static ☐ Const ☐ ConstExpr ☐ Virtual ☐ Pure Virtual ☐ Override ☐ Final ☐ Inline ☐ Delete

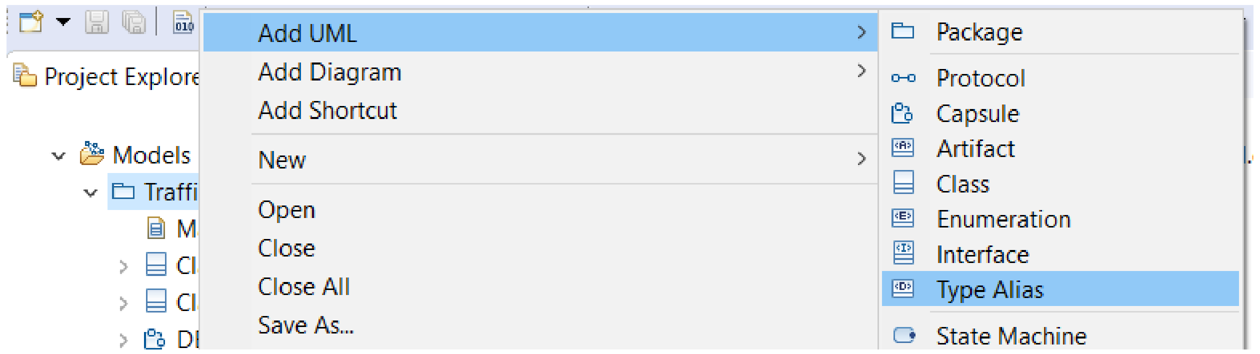


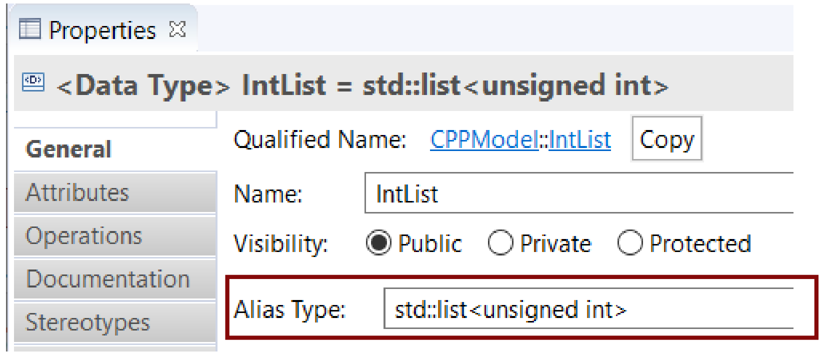
Destructor
☑ Generate ☑ Virtual ☐ Inline ☐ Default ☑ Delete
Visibility: [✓ public] protected private

HCL SOFTWARE

C++ 11 Improvements (3/4)

- C++ 11 type aliases are now supported in the user interface and model compiler





<Data Type> IntList = std::list<unsigned int>

Qualified Name: CPPModel::IntList

Name: IntList

Visibility: [✓ Public] Private Protected

Alias Type: std::list<unsigned int>

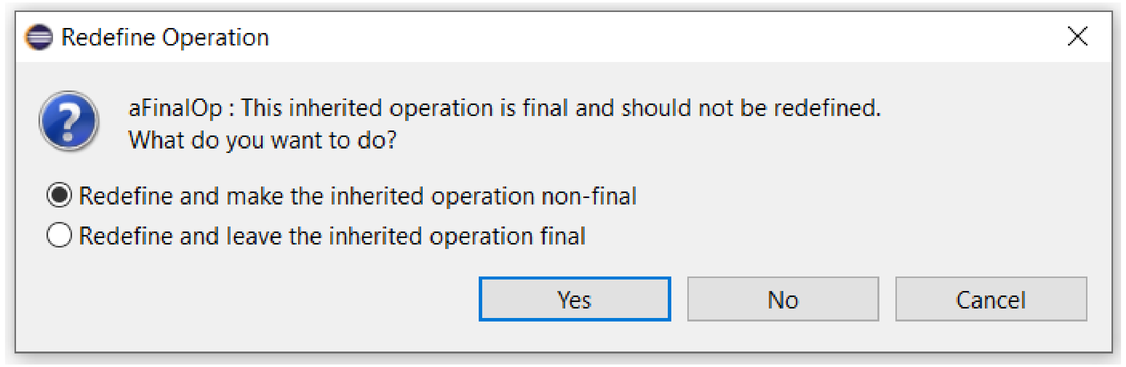
  - In the model a DataType with a new ”Alias Type” property is used
  - The main benefit with using type aliases instead of traditional typedefs is that they can have template parameters

```
template<typename TYPE > using GenericList = std::list<TYPE*>;
```

  - You can write a type descriptor for type aliases in the same way as you currently can for typedefs. The ”Type Descriptor Hint” property is also supported, to automatically generate a type descriptor for type aliases of commonly used STL collection types (list, set, map etc).

C++ 11 Improvements (4/4)

- When overriding an operation in the UI, its 'override' property is automatically set
  - If the overridden operation is final, a new dialog appears and suggests to make it non-final

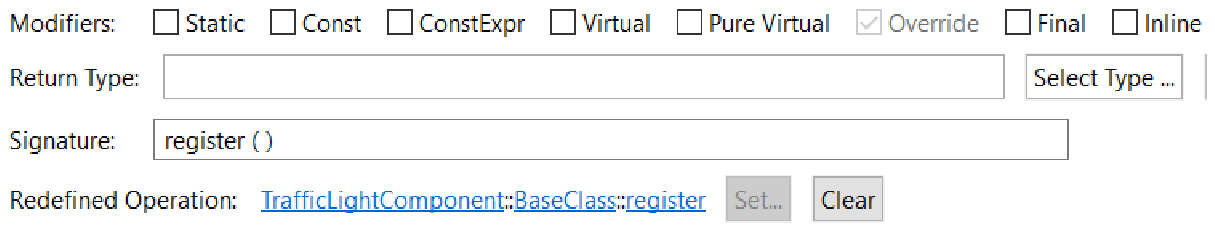


Redefine Operation

aFinalOp : This inherited operation is final and should not be redefined.
What do you want to do?

[✓ Redefine and make the inherited operation non-final]
Redefine and leave the inherited operation final

  - For redefined operations, the "override" property is automatically set (and cannot be unset)



Modifiers: Static, Const, ConstExpr, Virtual, Pure Virtual, [✓ Override], Final, Inline

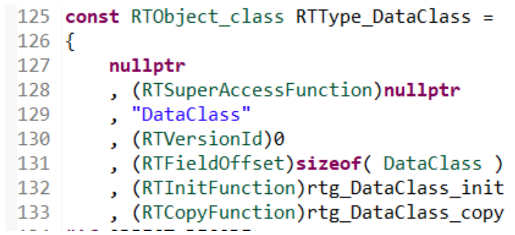
Return Type:

Signature: register ( )

Redefined Operation: TrafficLightComponent::BaseClass::register

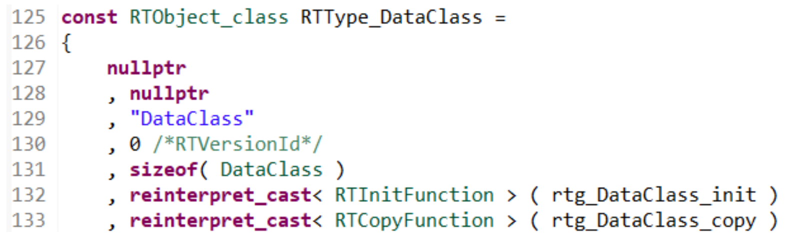
Removal of C-style Casts

- Both generated code and the TargetRTS have been changed by removing all C-style casts
- Each cast has either been removed, or replaced with a corresponding C++ cast
  - `static_cast` (in most cases) or `reinterpret_cast` (in a few cases)
  - Note: We don't generate `dynamic_cast` since we cannot assume code is compiled with run-time type information



```
const RTObject_class RTType_DataClass =
{
    nullptr
    , (RTSuperAccessFunction)nullptr
    , "DataClass"
    , (RTVersionId)0
    , (RTFieldOffset)sizeof( DataClass )
    , (RTInitFunction)rtg_DataClass_init
    , (RTCopyFunction)rtg_DataClass_copy
```

before



```
const RTObject_class RTType_DataClass =
{
    nullptr
    , nullptr
    , "DataClass"
    , 0 /*RTVersionId*/
    , sizeof( DataClass )
    , reinterpret_cast< RTInitFunction > ( rtg_DataClass_init )
    , reinterpret_cast< RTCopyFunction > ( rtg_DataClass_copy )
```

now

- A benefit is that the TargetRTS now can be compiled without the `–fpermissive` flag with GCC (without warnings)
  - Helped to detect a couple of bugs in the TargetRTS specific to 64-bit platforms – now fixed!
- With GCC the TargetRTS is now compiled with new flags to detect issues related to casts
  - `-Wuseless-cast` and `-Wold-style-cast`

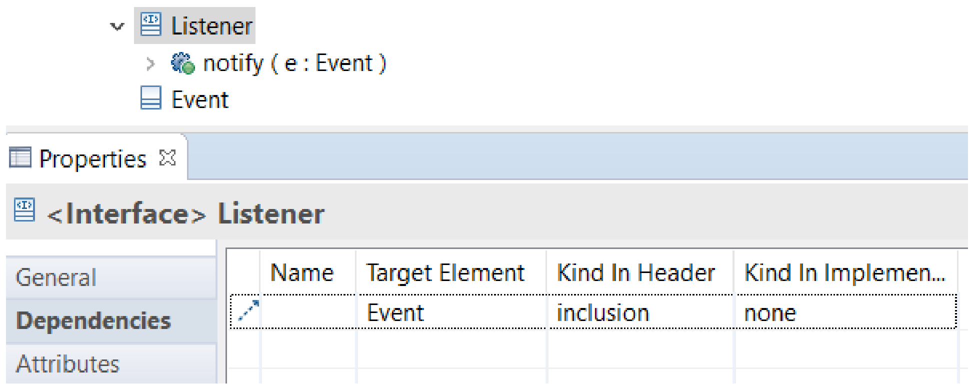
HCL SOFTWARE

Removed Compatibility with Old Rose-RT Versions

- The TargetRTS file `RTCompatibility.h` was removed
  - It contained macros, typedefs etc. for compatibility with very old versions of Rose-RT (6.4 and earlier)
- Migrated models that still use old Rose-RT names should be updated
  - RTDATA -> rtdata
  - Integer -> RTInteger
  - ... etc.
- It's recommended to update your models, but if you want to keep using these old names, you can keep `RTCompatibility.h` from the old TargetRTS version

Dependencies for Interfaces and Enumerations

- The Dependencies property page is now available also for Interfaces and Enumerations
  - Makes it easier to specify dependencies for these elements
  - Helps for example when an interface operation references another type which must be included or forward declared in the interface header file



Listener
notify (e : Event)
Event

Properties

<Interface> Listener

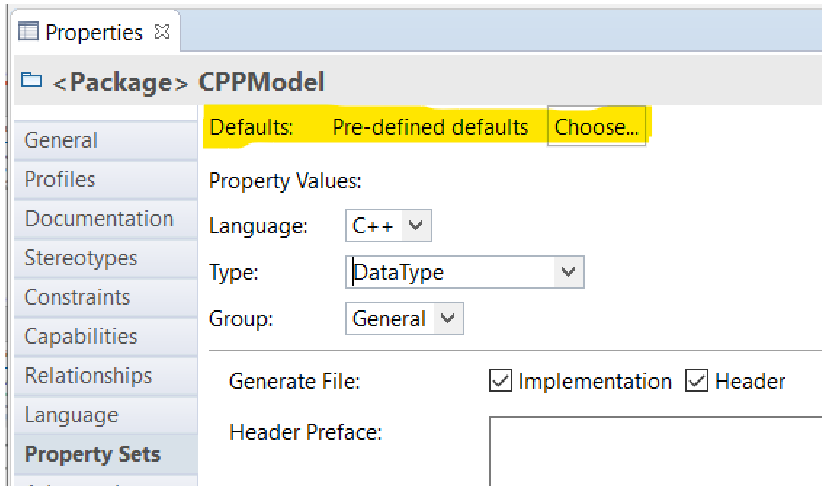
General
Dependencies
Attributes

Name	Target Element	Kind In Header	Kind In Implemen...
	Event	inclusion	none

HCL SOFTWARE

Changing Default Values for Property Set Properties

- Most of the properties shown on the “C++ General” and “C++ Target RTS” property pages have customizable default values
  - Before, the only way to change the default values of these so called “property set” properties was to do it from the Property Sets property page on the top-level package.
  - The drawback was that it was necessary to do this on all models that needed a customized default value!


Properties

<Package> CPPModel

Defaults: Pre-defined defaults Choose...

Property Values:

Language: C++

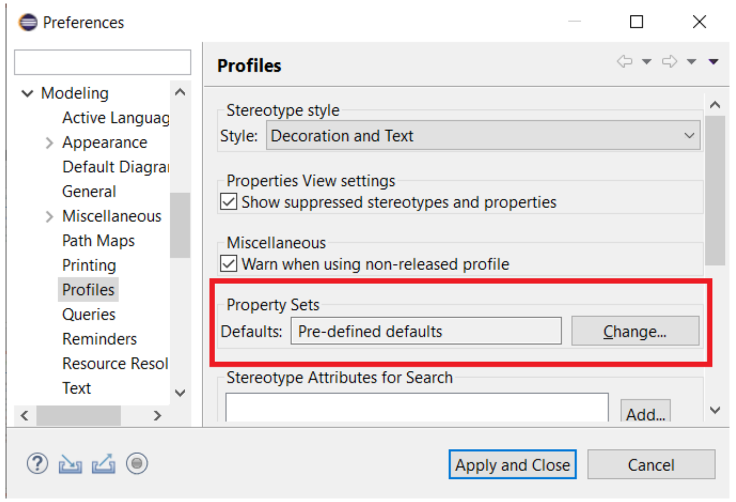
Type: DataType

Group: General

Generate File: ☑ Implementation ☑ Header

Header Preface:

- Now, a new preference is available which makes it possible to set new default values globally, without modifying any model
  - *Modeling – Profiles – Property Sets – Defaults*


Preferences

Profiles

Stereotype style
Style: Decoration and Text

Properties View settings
☑ Show suppressed stereotypes and properties

Miscellaneous
☑ Warn when using non-released profile

Property Sets
Defaults: Pre-defined defaults Change...

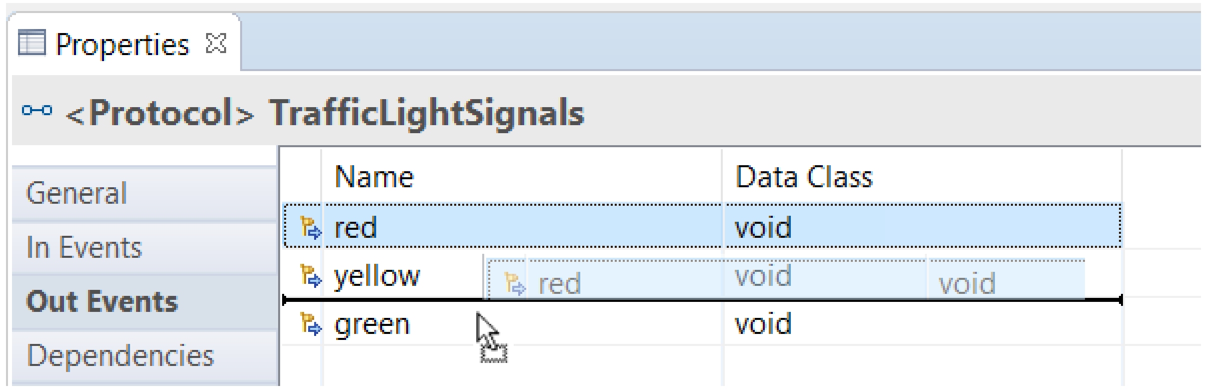
Stereotype Attributes for Search

- An example of when this can be useful is when you want custom names for generated getter/setter operations

HCL SOFTWARE

Moving Elements in the Properties View using Drag&Drop

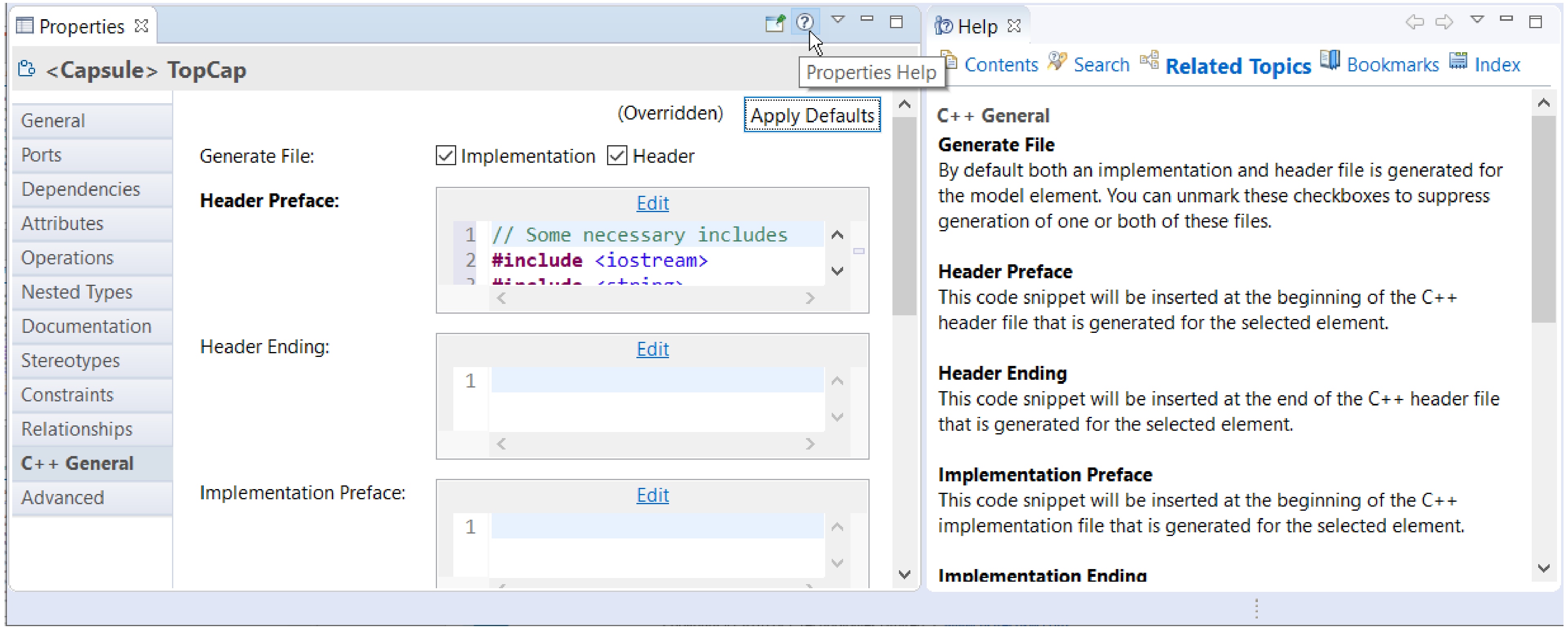
- It’s now possible to move elements in tables of the Properties view by means of drag&drop
  - More convenient than using the Up / Down buttons if there are many elements in the table
  - Multiple elements can be moved at the same time and will then all be inserted at the drop location (shown by a line in the table)



Name	Data Class
red	void
yellow	void
green	void

Help for the Properties View

- The Properties View now provides context sensitive help
  - Use the new toolbar button for showing help for the properties currently visible in the view (or press F1)
  - The new property documentation helps understand what properties mean, and how they affect the generated code



Generate File: ☑ Implementation ☑ Header

Header Preface: Edit

```
// Some necessary includes
#include <iostream>
```

Header Ending: Edit

Implementation Preface: Edit

C++ General

Generate File
By default both an implementation and header file is generated for the model element. You can unmark these checkboxes to suppress generation of one or both of these files.

Header Preface
This code snippet will be inserted at the beginning of the C++ header file that is generated for the selected element.

Header Ending
This code snippet will be inserted at the end of the C++ header file that is generated for the selected element.

Implementation Preface
This code snippet will be inserted at the beginning of the C++ implementation file that is generated for the selected element.

Implementation Ending

HCL SOFTWARE

UML-RT Java APIs

- The Java APIs for creating UML-RT models programmatically have been extended
  - Creating a capsule part

```
RTCapsulePart RTCapsule.createCapsulePart(String name, RTCapsule type);
```

  - Setting the multiplicity of a capsule part, and its kind (fixed, optional or plugin)

```
void RTCapsulePart.setMultiplicityAndKind(int lower, int upper, AggregationKind kind);
```

  - Creating a connector between two ports on capsule parts

```
Connector RTModelOperations.createConnector(RTCapsulePart sourcePart,
RTPortRedefinition sourcePort, RTCapsulePart targetPart, RTPortRedefinition
targetPort) throws ConnectionError;
```

- The UML RT SDK sample has been updated to use these new APIs

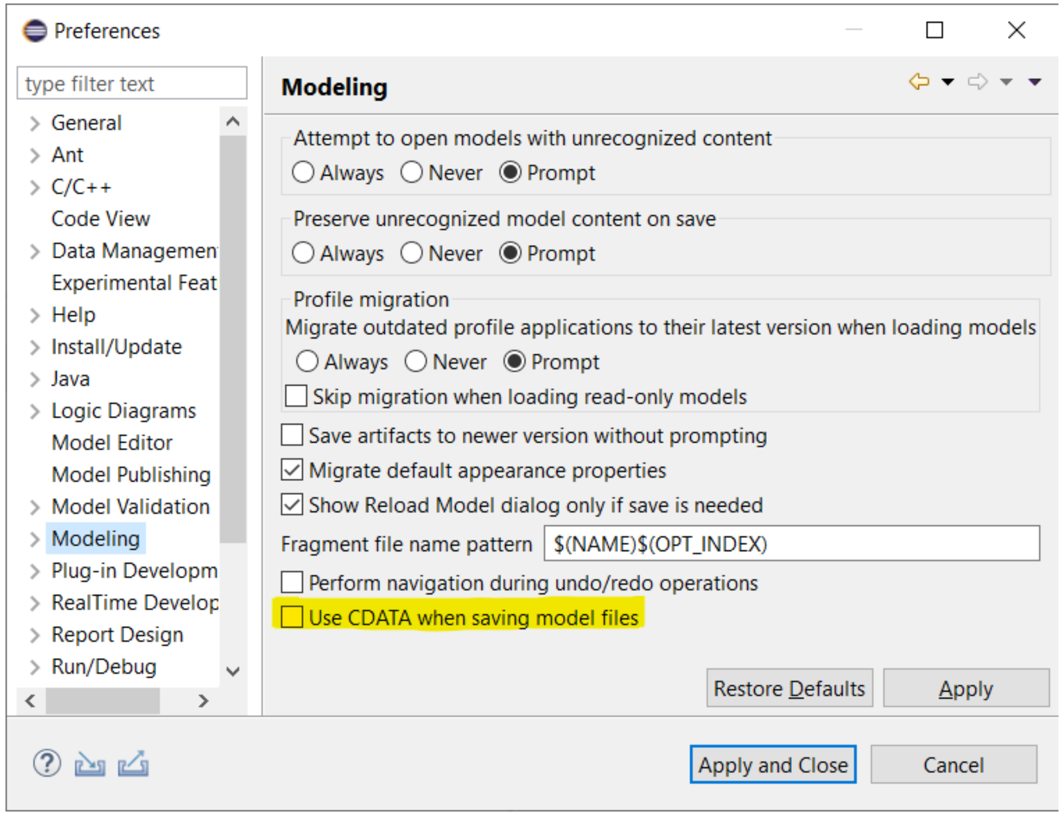
Usage of CDATA when Storing Models

- A new preference can be set to store code snippets and documentation texts using CDATA in the model XML files
- This avoids use of XML escape characters which makes such texts more readable when viewed in a text editor
  - Example (without CDATA)

```
<headerPreface xmi:id="_irLY4NitEeqes4xYsqINtA"
body="// Some necessary
includes&#xD;&#xA;#include
&lt;iostream>&#xD;&#xA;#include
&lt;string>&#xD;&#xA;&#xD;&#xA;#define OPEN
1&#xD;&#xA;#define CLOSED 0"/>
```

  - Example (with CDATA)

```
    <headerPreface xmi:id="_irLY4NitEeqes4xYsqINtA">
      <body><![CDATA[// Some necessary includes
#include <iostream>
#include <string>

#define OPEN 1
#define CLOSED 0]]></body>
    </headerPreface>
```



Preferences

Modeling

Attempt to open models with unrecognized content: Always / Never / [✓ Prompt]

Preserve unrecognized model content on save: Always / Never / [✓ Prompt]

Profile migration
Migrate outdated profile applications to their latest version when loading models: Always / Never / [✓ Prompt]

- [ ] Skip migration when loading read-only models
- [ ] Save artifacts to newer version without prompting
- [x] Migrate default appearance properties
- [x] Show Reload Model dialog only if save is needed

Fragment file name pattern: $(NAME)$(OPT_INDEX)

- [ ] Perform navigation during undo/redo operations
- [ ] Use CDATA when saving model files

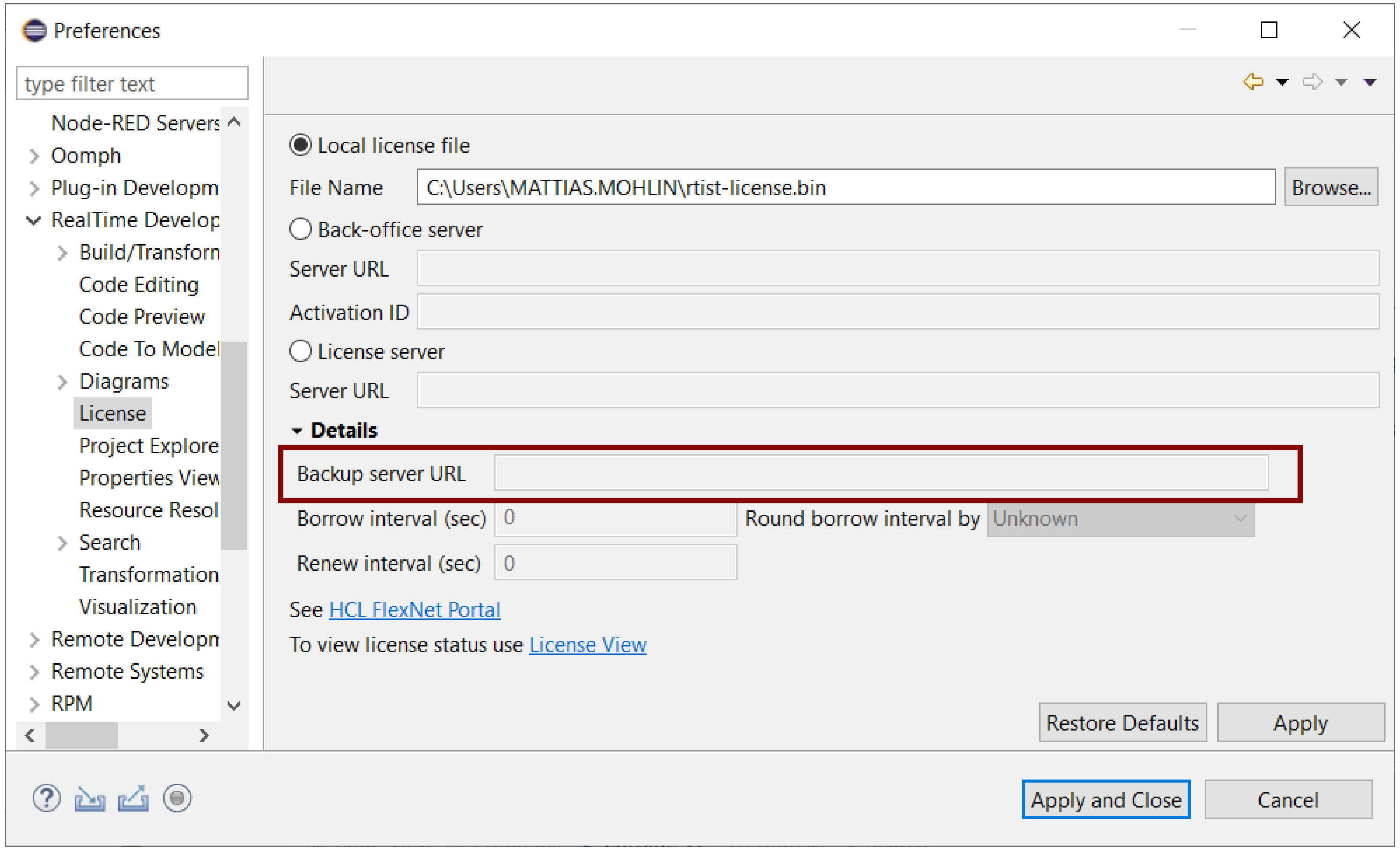
Note: Use of CDATA does not affect backwards compatibility.

Old versions of RTist can still load model files that use CDATA. But if saved in old tool versions, CDATA will of course not be used. A new model fixup is available for converting all model files to use the new file format with CDATA.

HCL SOFTWARE

License Handling

- A backup license server can now be specified
  - Will be used if the main license server is not available


Preferences

- [✓ Local license file]
  - File Name: C:\Users\MATTIAS.MOHLIN\rtist-license.bin
- Back-office server
  - Server URL:
  - Activation ID:
- License server
  - Server URL:

Details

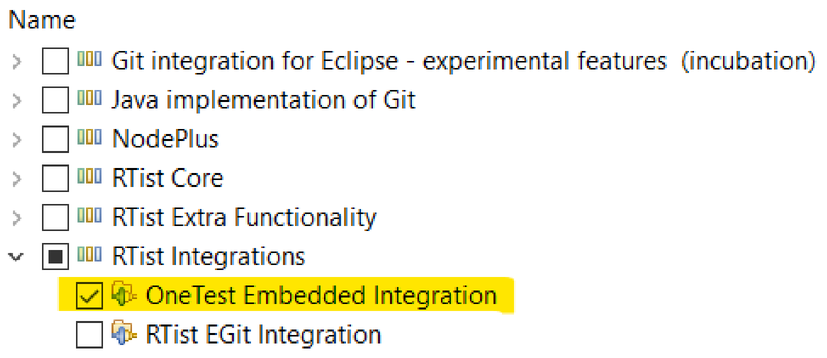
- Backup server URL:
- Borrow interval (sec): 0
- Round borrow interval by: Unknown
- Renew interval (sec): 0

See HCL FlexNet Portal

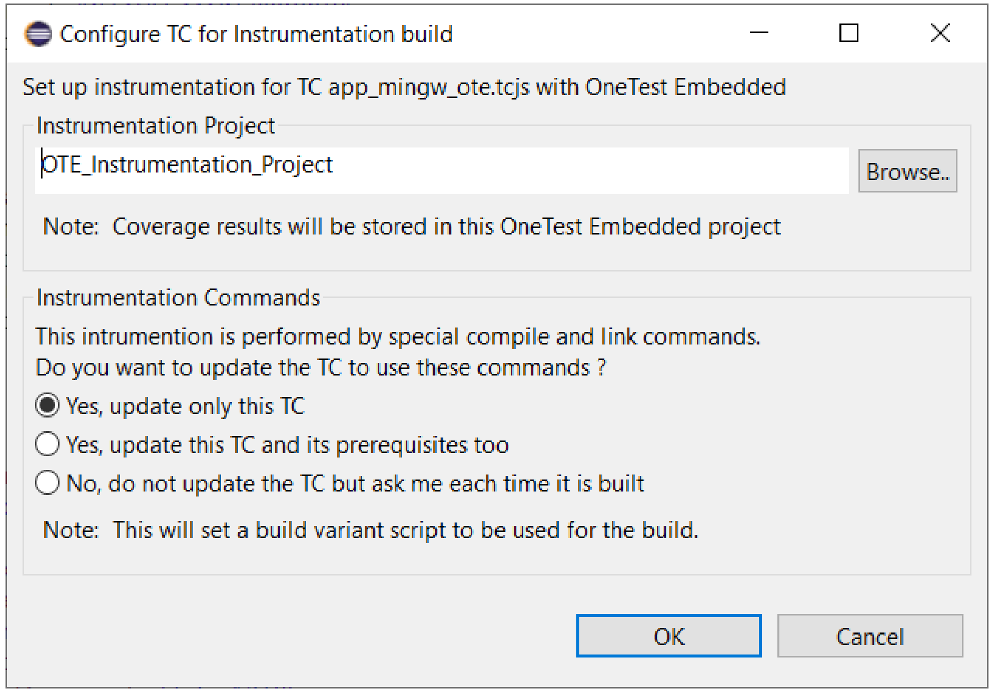
To view license status use License View

OneTest Embedded Integration (1/2)

- A new install component of HCL RTist
  - Note: The OneTest Embedded Integration is not available in IBM RSARTE
- Computes and visualizes run-time coverage for UML-RT state machines
  - Which states and transitions have been executed in the state machine?
  - What parts have not executed at all? May indicate too little test coverage.
  - What parts have executed the most? Special focus may be spent on optimizing the performance of those parts.
- Based on building an instrumented version of the RTist application that collects coverage data when it runs
  - You can either create special versions of your TC(s) for building an instrumented application, or use build variants which will prompt you each time it's built



Name
- [ ] Git integration for Eclipse - experimental features (incubation)
- [ ] Java implementation of Git
- [ ] NodePlus
- [ ] RTist Core
- [ ] RTist Extra Functionality
- ■ RTist Integrations
  - [✓] OneTest Embedded Integration
  - [ ] RTist EGit Integration



Configure TC for Instrumentation build

Set up instrumentation for TC app_mingw_ote.tcjs with OneTest Embedded

Instrumentation Project

OTE_Instrumentation_Project

Note: Coverage results will be stored in this OneTest Embedded project

Instrumentation Commands

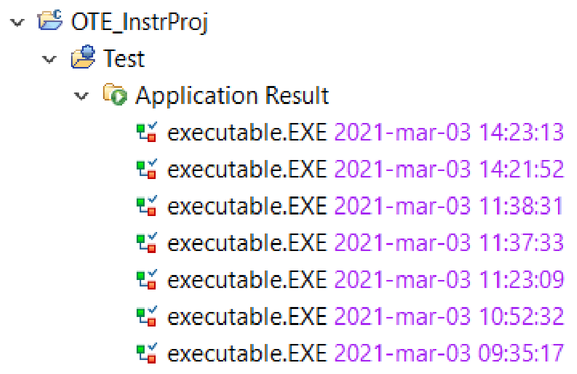
This intrumentation is performed by special compile and link commands.
Do you want to update the TC to use these commands ?

- [✓ Yes, update only this TC]
- Yes, update this TC and its prerequisites too
- No, do not update the TC but ask me each time it is built

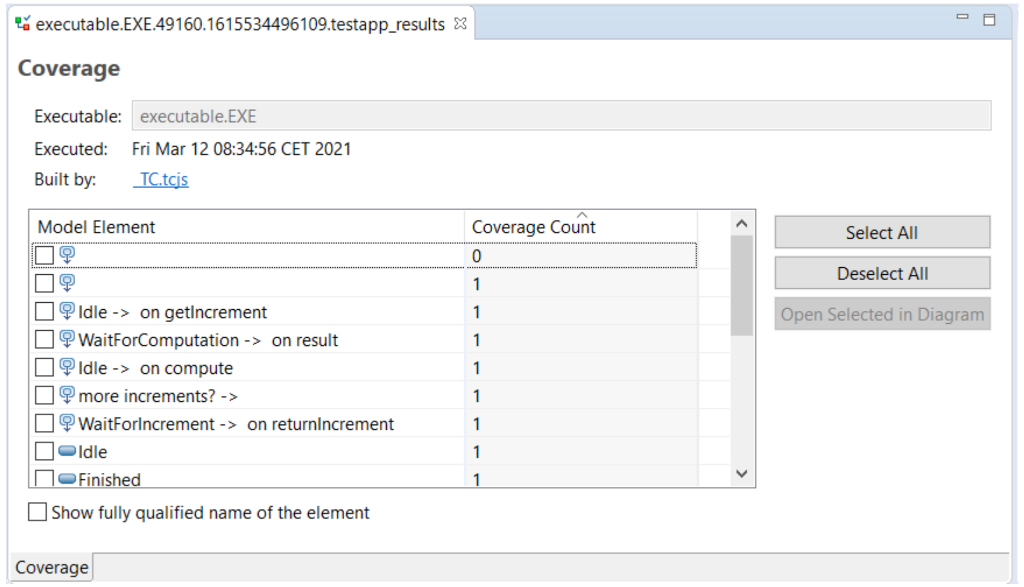
Note: This will set a build variant script to be used for the build.

OneTest Embedded Integration (2/2)

- Coverage can be accumulated from several executions of the RT application (e.g. running multiple tests)
  - You can look at the combined coverage from multiple executions of the application
- Coverage can be shown in a table
  - Gives full details of how many times different elements have executed



- OTE_InstrProj
  - Test
    - Application Result
      - executable.EXE 2021-mar-03 14:23:13
      - executable.EXE 2021-mar-03 14:21:52
      - executable.EXE 2021-mar-03 11:38:31
      - executable.EXE 2021-mar-03 11:37:33
      - executable.EXE 2021-mar-03 11:23:09
      - executable.EXE 2021-mar-03 10:52:32
      - executable.EXE 2021-mar-03 09:35:17



executable.EXE.49160.1615534496109.testapp_results

Coverage

Executable: executable.EXE
Executed: Fri Mar 12 08:34:56 CET 2021
Built by: TC.tcjs

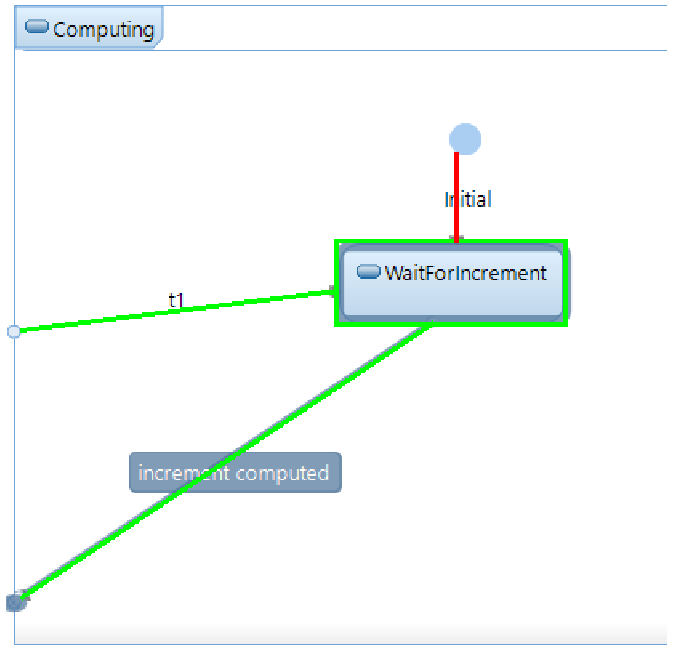
Model Element	Coverage Count
☐	0
☐	1
☐ Idle -> on getIncrement	1
☐ WaitForComputation -> on result	1
☐ Idle -> on compute	1
☐ more increments? ->	1
☐ WaitForIncrement -> on returnIncrement	1
☐ Idle	1
☐ Finished	1

Select All
Deselect All
Open Selected in Diagram

☐ Show fully qualified name of the element

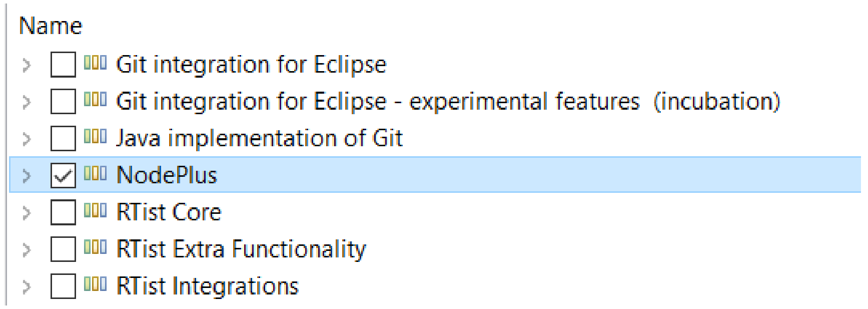
Coverage

- Coverage can be shown in state chart diagrams
  - Gives an overview of what elements have or have not executed at least once



NodePlus

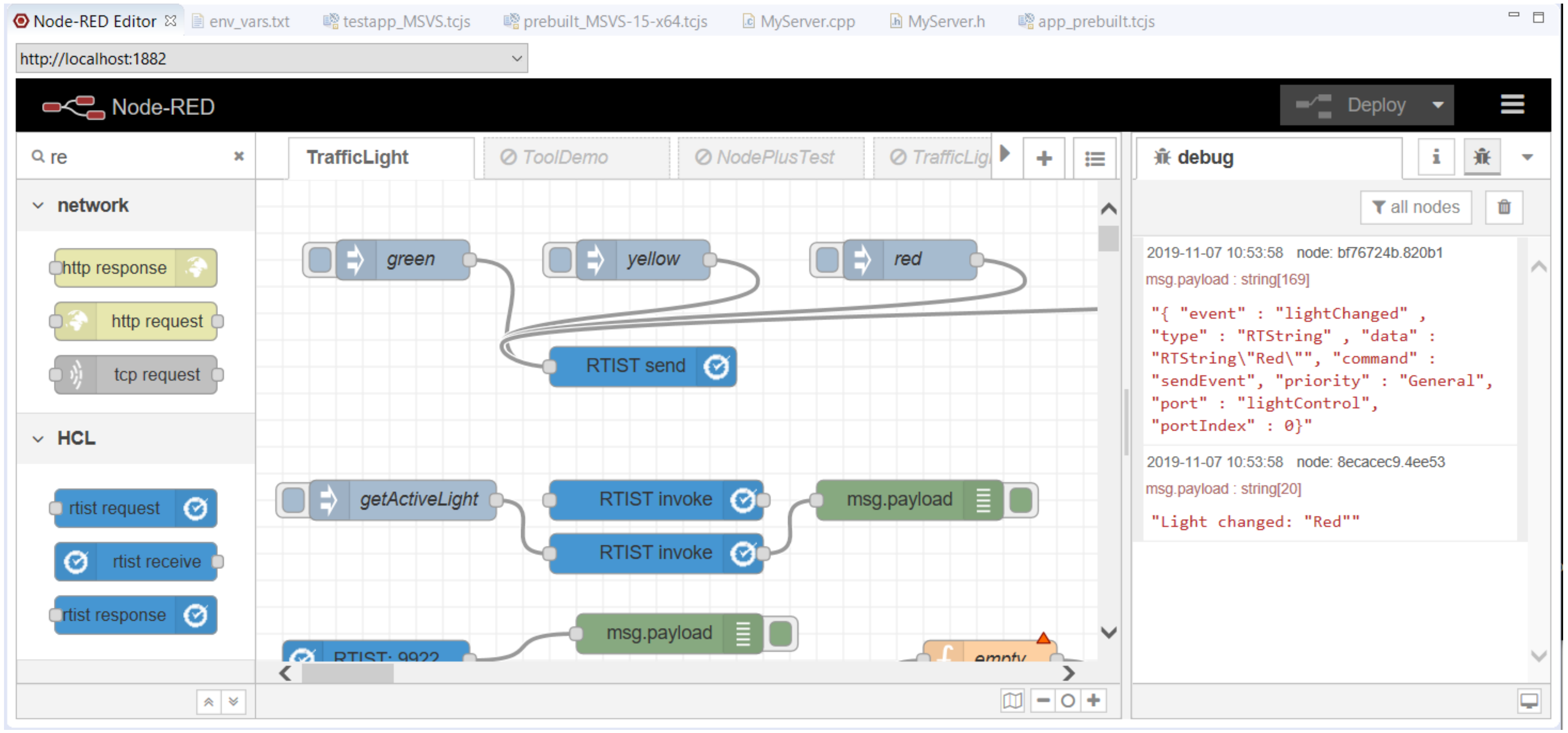
- HCL NodePlus is a new install component of HCL RTist
  - Note: NodePlus is not available in IBM RSARTE
- NodePlus provides all necessary tooling for developing IoT applications using Node-RED (and related technologies)
  - Node-RED projects - both for creating Node-RED nodes and flows of interconnected nodes
  - Node-RED server - for convenient deployment of Node-RED flows within the Eclipse workbench
  - Node.js projects - for creating Node.js applications
  - Node.js server - for convenient deployment of Node.js applications within the Eclipse workbench
  - Swagger - for specifying APIs using Swagger documents
  - Node.js Debugger - for debugging Node.js applications
  - Pug (formerly known as Jade) – for template-based rendering of web pages
  - Unit test
  - Code coverage



Name
- [ ] Git integration for Eclipse
- [ ] Git integration for Eclipse - experimental features (incubation)
- [ ] Java implementation of Git
- [✓] NodePlus
- [ ] RTist Core
- [ ] RTist Extra Functionality
- [ ] RTist Integrations

Node-RED Tools in NodePlus

- Node-RED editor
  - Configure nodes and wire them together into flows
  - Deploy changes to the Node-RED server
  - View debug output and execution within the flow
  - Install new nodes to the palette (e.g. from the public library)

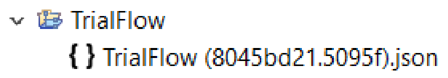


```
2019-11-07 10:53:58  node: bf76724b.820b1
msg.payload : string[169]
"{ "event" : "lightChanged" ,
"type" : "RTString" , "data" :
"RTString\"Red\"", "command" :
"sendEvent", "priority" : "General",
"port" : "lightControl",
"portIndex" : 0}"

2019-11-07 10:53:58  node: 8ecacec9.4ee53
msg.payload : string[20]
"Light changed: "Red""
```

Node-RED Tools in NodePlus

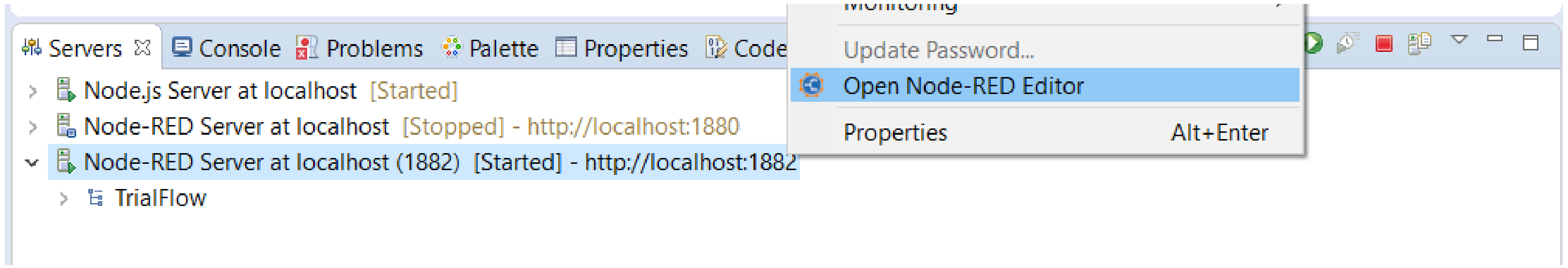
RTist ONLY

- Node-RED Flow Projects
  - Flows can be created directly in the Node-RED editor, but creating them as Eclipse projects has many benefits (e.g. team development using SCM integration)
  - An intuitive wizard for creating Node-RED Flow projects
  - Either create a flow from scratch or start from one of the existing flows in the public library



TrialFlow
{ } TrialFlow (8045bd21.5095f).json

- Node-RED Server
  - A dedicated Servers view lets you create, start and stop Node-RED local and remote servers within the Eclipse workbench
  - Deploy a Node-RED flow onto a selected server and open its Node-RED editor



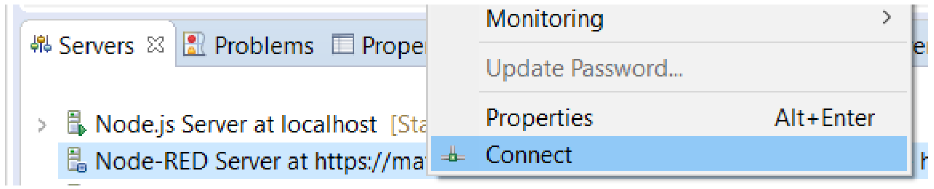
Servers | Console | Problems | Palette | Properties | Code

Node.js Server at localhost [Started]
Node-RED Server at localhost [Stopped] - http://localhost:1880
Node-RED Server at localhost (1882) [Started] - http://localhost:1882
TrialFlow

HCL SOFTWARE

Node-RED Tools in NodePlus

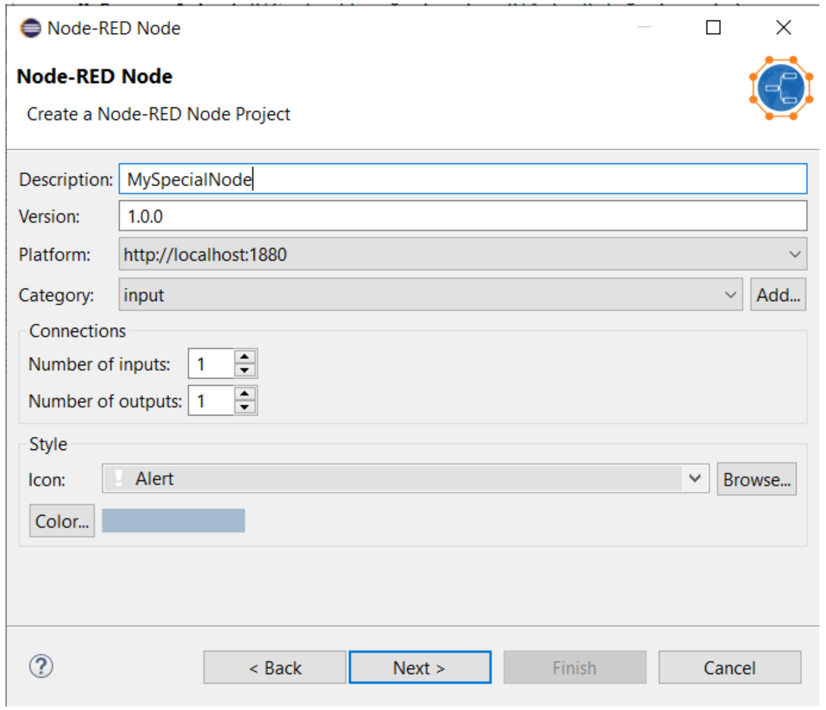
- Remote Node-RED Server
  - You don’t need to run the Node-RED server locally. You can instead connect to a server that runs remotely (for example on a server or on a device)
  - Create a new server in the Servers view and enter the URL to the remote server
  - You can then connect to the server in order to open the Node-RED editor for it (other operations on a remote server are currently not supported).



Node-RED Tools in NodePlus

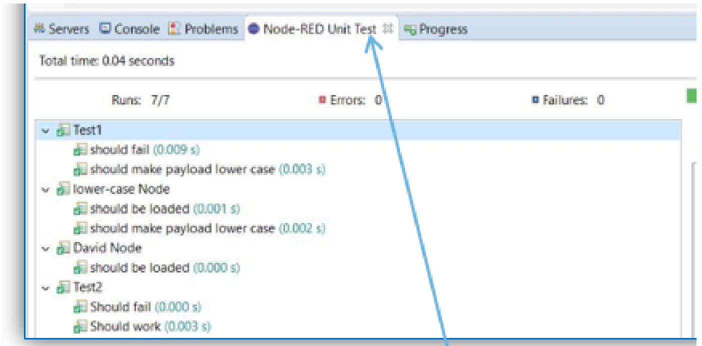
RTist ONLY

- Node-RED Node Projects
  - More than 2500 nodes exist in the public library, but sometimes you may need to create you own specific node
  - An intuitive wizard for creating Node-RED Node projects
  - Benefit from all the usual Eclipse features when developing your node (e.g. SCM integration)
  - Either create a node from scratch or start from one of the existing nodes in the public library



Description:	MySpecialNode
Version:	1.0.0
Platform:	http://localhost:1880
Category:	input
Number of inputs:	1
Number of outputs:	1
Icon:	Alert

- Unit test of Node-RED nodes
  - Unit tests are created using the Mocha framework and results are shown in a special Node-RED Unit Test view

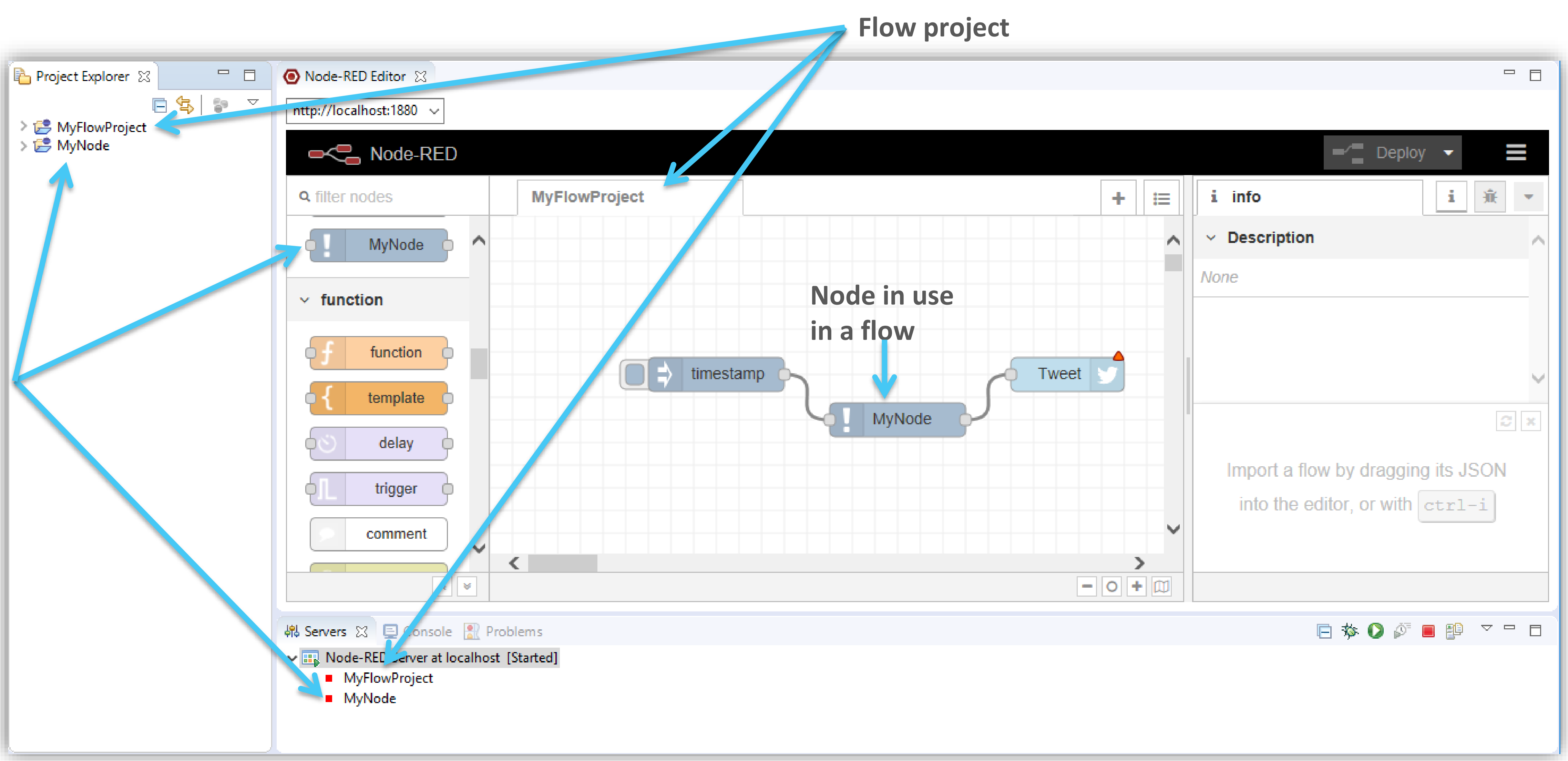


```
Total time: 0.04 seconds
Runs: 7/7    Errors: 0    Failures: 0
Test1
    should fail (0.009 s)
    should make payload lower case (0.003 s)
lower-case Node
    should be loaded (0.001 s)
    should make payload lower case (0.002 s)
David Node
    should be loaded (0.000 s)
Test2
    Should fail (0.000 s)
    Should work (0.003 s)
```

Custom "Unit Test" view

HCL SOFTWARE

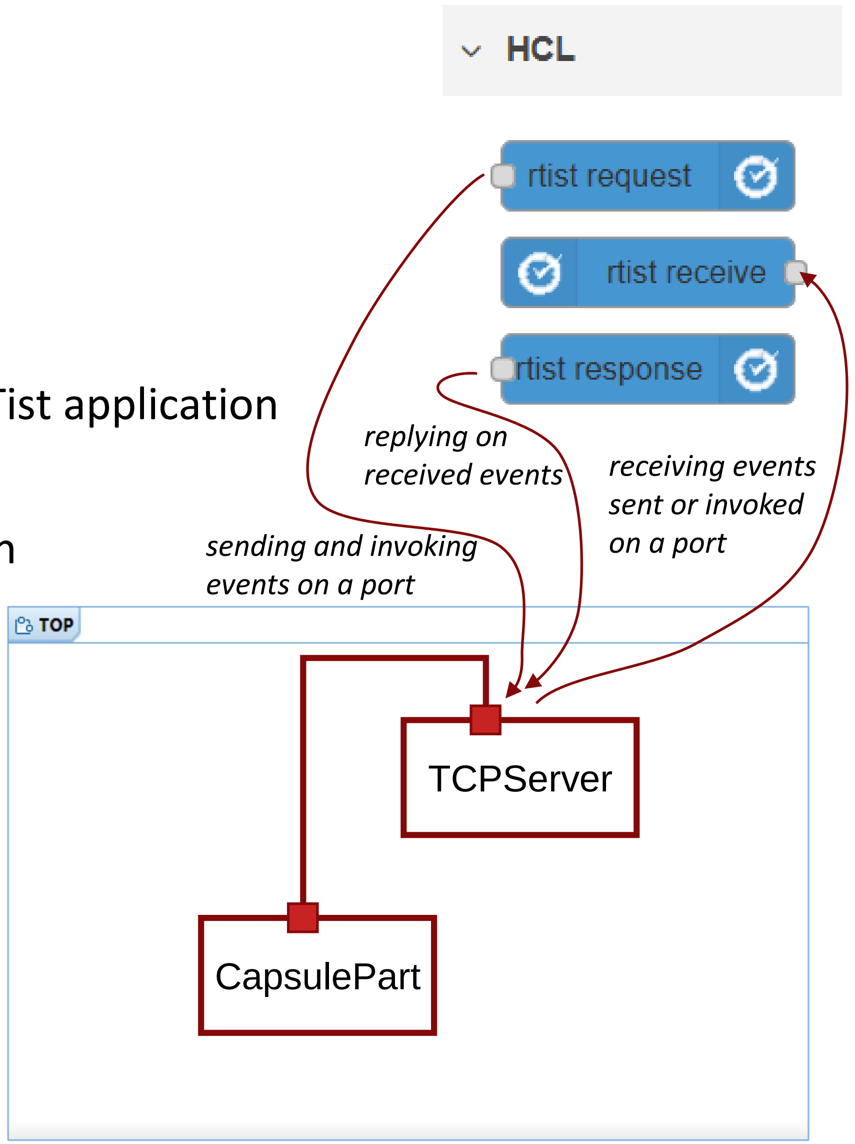
Node-RED Tools in NodePlus



Node-RED Tools in NodePlus

- Nodes for communicating with an RTist application
  - **rtist request**
    Let a flow send or invoke an event into an RTist application
  - **rtist receive**
    Trigger a flow when receiving an event that is sent or invoked from an RTist application
  - **rtist response**
    Reply to a received event that is sent or invoked from an RTist application

- These nodes make use of the JSON API provided by lib-tcp-server (i.e. the RTist application must use this library)
  - The nodes can be statically configured in the Node-RED editor
  - It's also possible to provide many of the node properties in the message payload for a more dynamic behavior


HCL
rtist request
rtist receive
rtist response
replying on received events
receiving events sent or invoked on a port
sending and invoking events on a port
TOP
TCPServer
CapsulePart

Swagger Support in NodePlus

- Document APIs using Swagger in a new Swagger editor

Link source and design panes

Support for JSON and YAML syntax

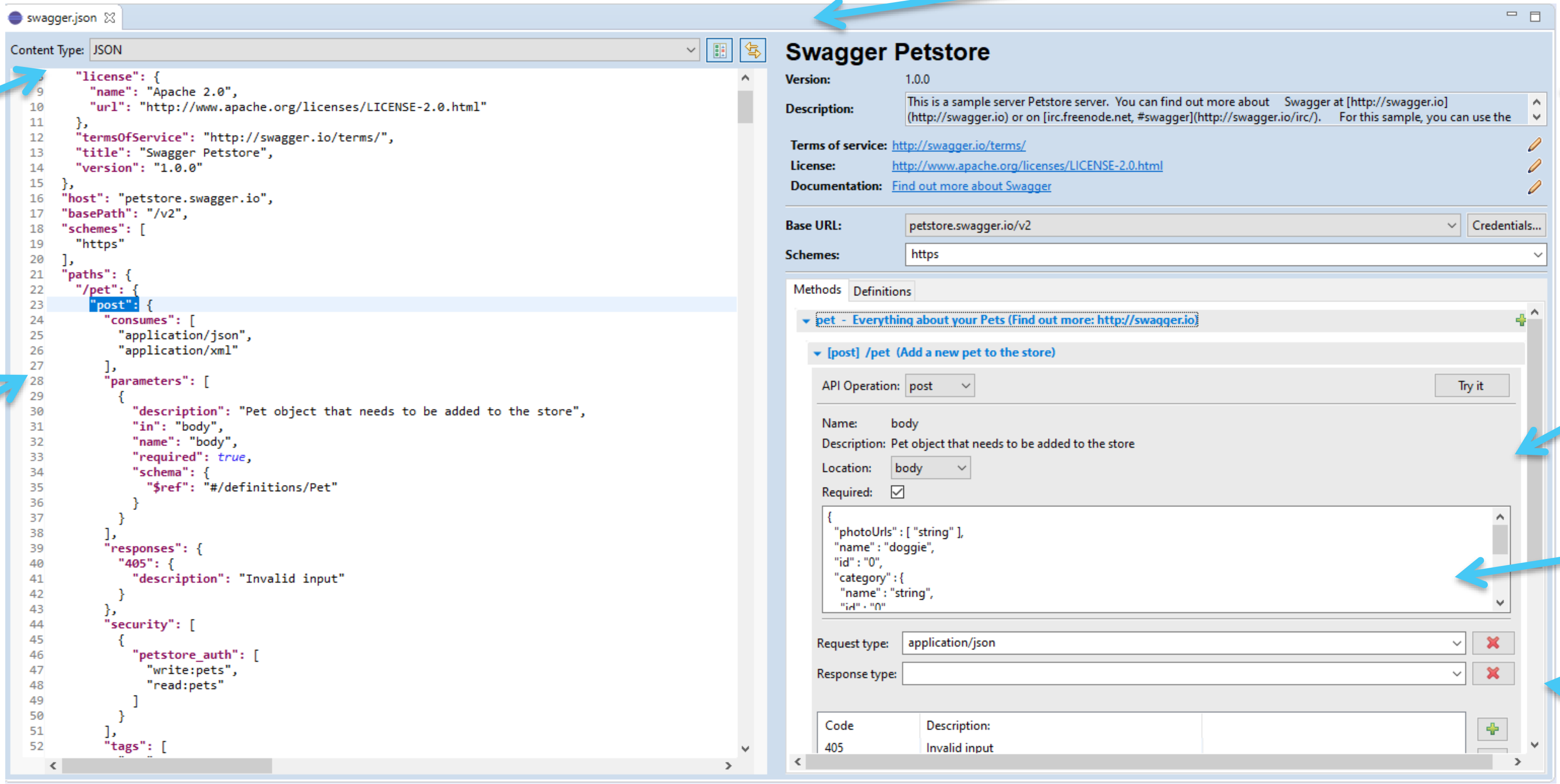
Syntax highlight and content assist

Visual API representation

Support for executing / testing API

Automated generation of example data

Content types selections



Content Type: JSON

```
    "license": {
      "name": "Apache 2.0",
      "url": "http://www.apache.org/licenses/LICENSE-2.0.html"
    },
    "termsOfService": "http://swagger.io/terms/",
    "title": "Swagger Petstore",
    "version": "1.0.0"
  },
  "host": "petstore.swagger.io",
  "basePath": "/v2",
  "schemes": [
    "https"
  ],
  "paths": {
    "/pet": {
      "post": {
        "consumes": [
          "application/json",
          "application/xml"
        ],
        "parameters": [
          {
            "description": "Pet object that needs to be added to the store",
            "in": "body",
            "name": "body",
            "required": true,
            "schema": {
              "$ref": "#/definitions/Pet"
            }
          }
        ],
        "responses": {
          "405": {
            "description": "Invalid input"
          }
        },
        "security": [
          {
            "petstore_auth": [
              "write:pets",
              "read:pets"
            ]
          }
        ],
        "tags": [
```

Swagger Petstore

Version: 1.0.0

Description: This is a sample server Petstore server. You can find out more about Swagger at [http://swagger.io] (http://swagger.io) or on [irc.freenode.net, #swagger](http://swagger.io/irc/). For this sample, you can use the

Terms of service: http://swagger.io/terms/

License: http://www.apache.org/licenses/LICENSE-2.0.html

Documentation: Find out more about Swagger

Base URL: petstore.swagger.io/v2

Schemes: https

pet - Everything about your Pets (Find out more: http://swagger.io)

[post] /pet (Add a new pet to the store)

API Operation: post

Name: body

Description: Pet object that needs to be added to the store

Location: body

Required: ☑

```
{
 "photoUrls" : [ "string" ],
 "name" : "doggie",
 "id" : "0",
 "category" : {
  "name" : "string",
```

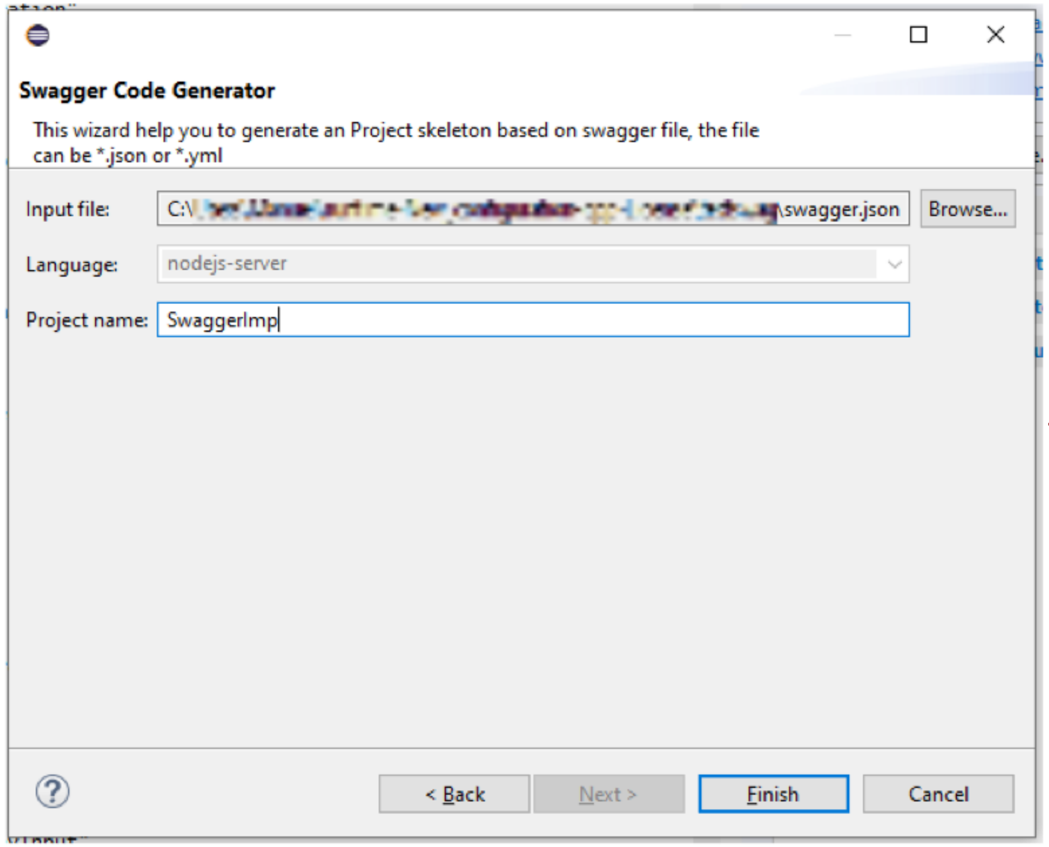
Request type: application/json

Response type:

Code	Description:
405	Invalid input

Swagger Code Generator

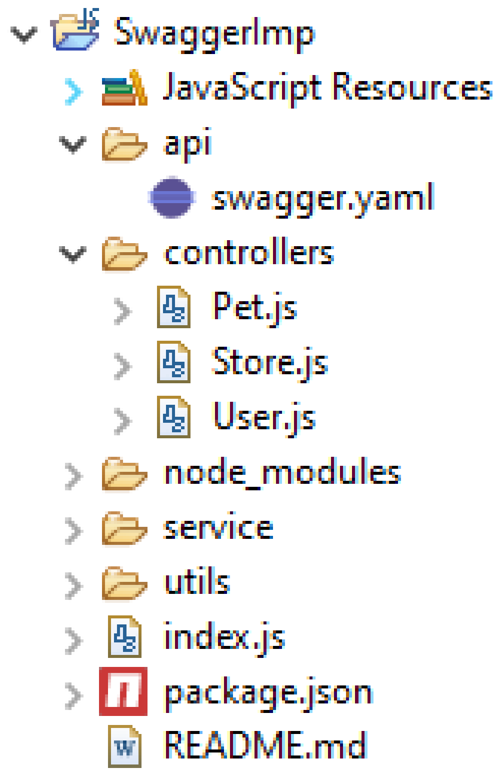
- Create development projects in different languages, such as Java, JavaScript, NodeJS, Python, etc. from a swagger specification with the Swagger Code Generator Wizard



Swagger Code Generator

This wizard help you to generate an Project skeleton based on swagger file, the file can be *.json or *.yml

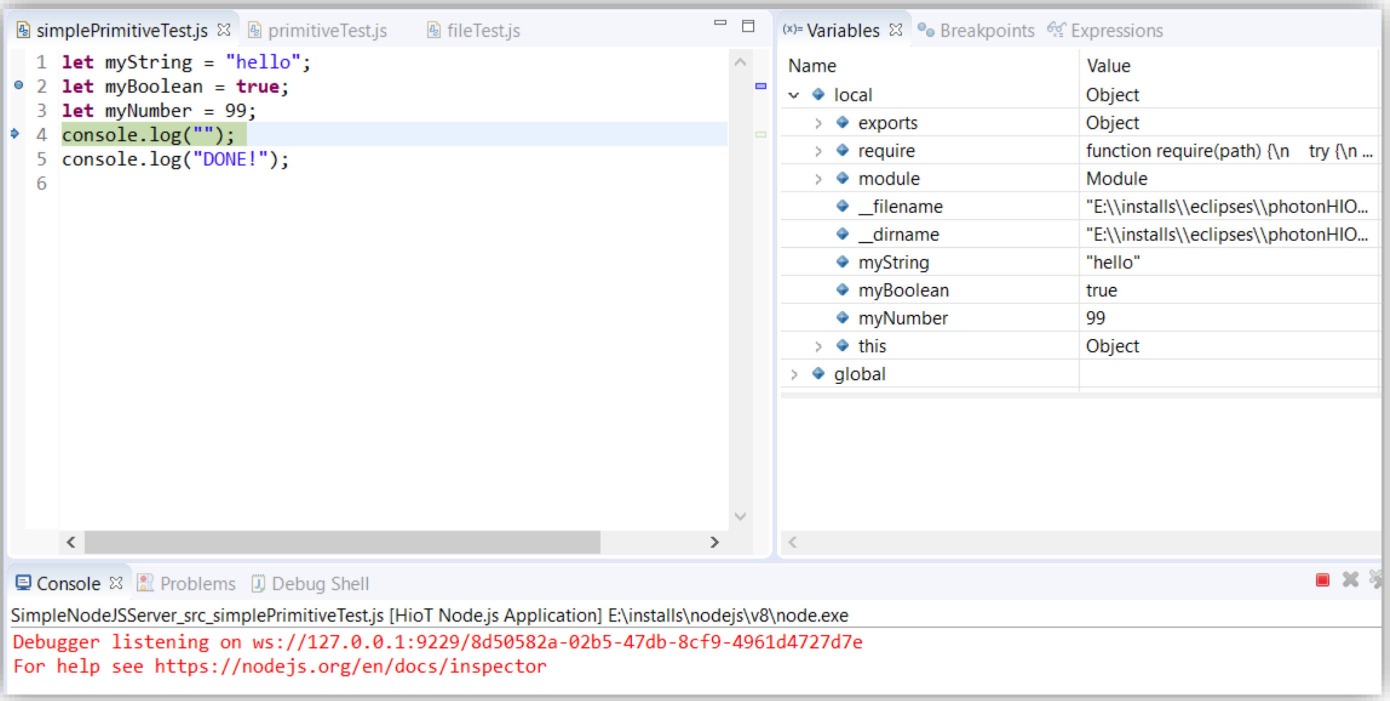
Input file:	C:\[redacted]\swagger.json
Language:	nodejs-server
Project name:	SwaggerImp



```
SwaggerImp
  JavaScript Resources
  api
    swagger.yaml
  controllers
    Pet.js
    Store.js
    User.js
  node_modules
  service
  utils
  index.js
  package.json
  README.md
```

Node.js Tools in NodePlus

- Wizard for creating Node.js projects
- Create, start and stop a Node.js server from within the Eclipse workbench
- Debug Node.js applications with a modern JavaScript debugger
- New editor for developing HTML pages for your Node.js application using the Pug template engine



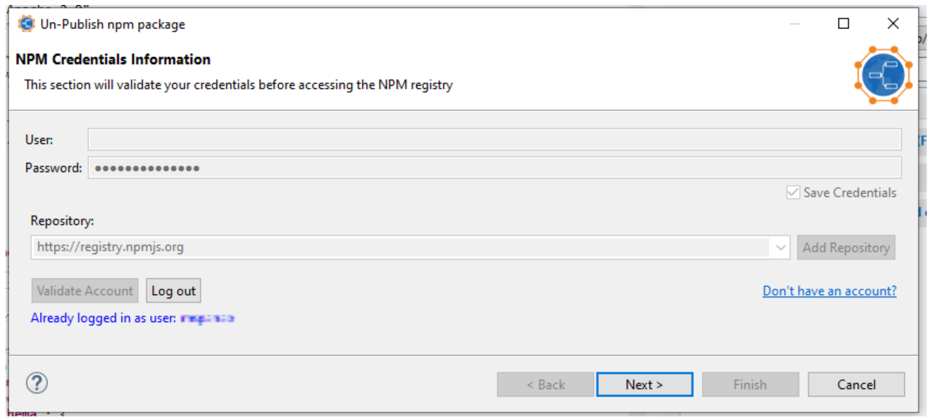
```
let myString = "hello";
let myBoolean = true;
let myNumber = 99;
console.log("");
console.log("DONE!");
```

Name	Value
local	Object
exports	Object
require	function require(path) {\n try {\n ...
module	Module
__filename	"E:\\installs\\eclipses\\photonHIO...
__dirname	"E:\\installs\\eclipses\\photonHIO...
myString	"hello"
myBoolean	true
myNumber	99
this	Object
global	

```
SimpleNodeJSServer_src_simplePrimitiveTest.js [HioT Node.js Application] E:\installs\nodejs\v8\node.exe
Debugger listening on ws://127.0.0.1:9229/8d50582a-02b5-47db-8cf9-4961d4727d7e
For help see https://nodejs.org/en/docs/inspector
```

Node.js Tools in NodePlus

- Deploy NPM packages into the NPM repository
- Deprecate / Unpublish NPM packages from the NPM repository



Un-Publish npm package

NPM Credentials Information

This section will validate your credentials before accessing the NPM registry

User:	
Password:	••••••••••••••

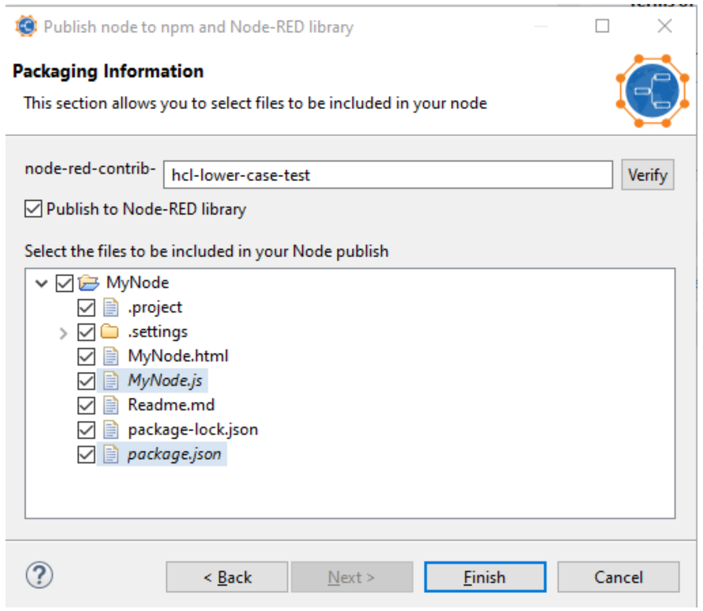
☑ Save Credentials

Repository: https://registry.npmjs.org — Add Repository

Validate Account | Log out — Don't have an account?

Already logged in as user: [redacted]

< Back | Next > | Finish | Cancel



Publish node to npm and Node-RED library

Packaging Information

This section allows you to select files to be included in your node

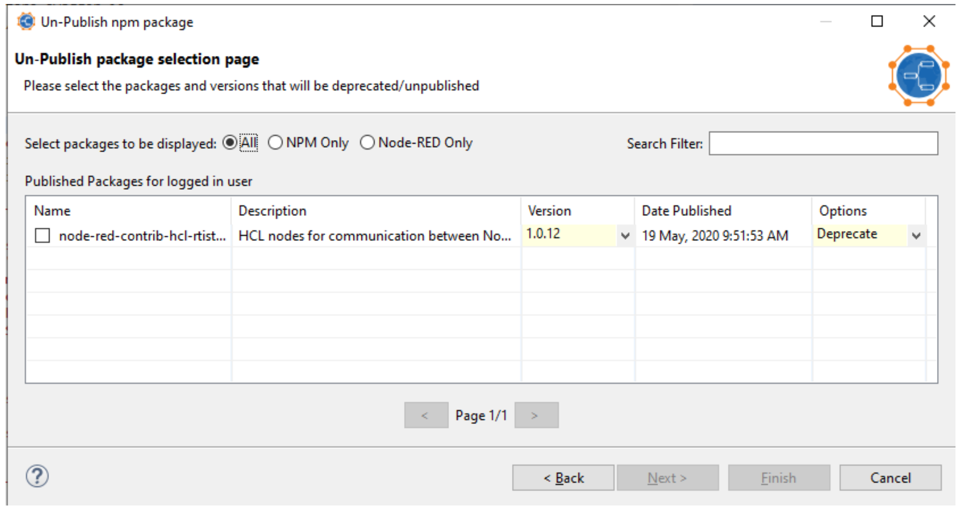
node-red-contrib- hcl-lower-case-test — Verify

☑ Publish to Node-RED library

Select the files to be included in your Node publish

- ☑ MyNode
  - ☑ .project
  - ☑ .settings
  - ☑ MyNode.html
  - ☑ *MyNode.js*
  - ☑ Readme.md
  - ☑ package-lock.json
  - ☑ *package.json*

< Back | Next > | Finish | Cancel



Un-Publish npm package

Un-Publish package selection page

Please select the packages and versions that will be deprecated/unpublished

Select packages to be displayed: [✓ All] NPM Only Node-RED Only — Search Filter:

Published Packages for logged in user

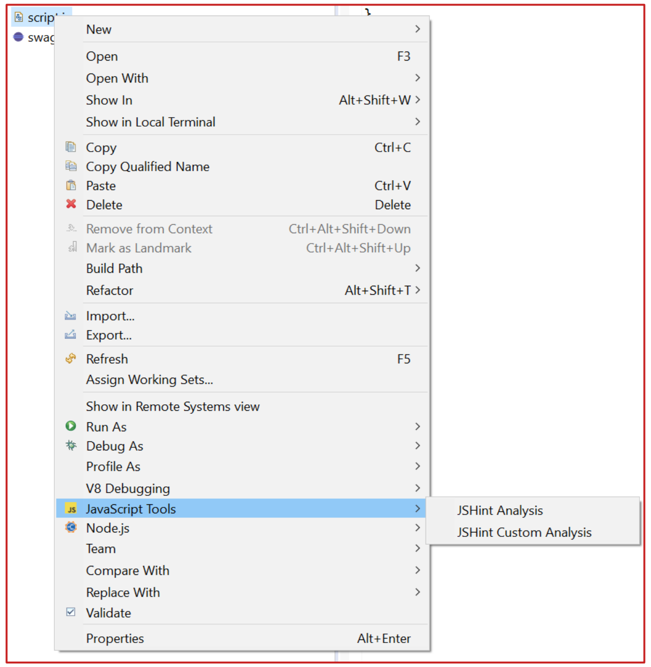
Name	Description	Version	Date Published	Options
☐ node-red-contrib-hcl-rtist...	HCL nodes for communication between No...	1.0.12	19 May, 2020 9:51:53 AM	Deprecate

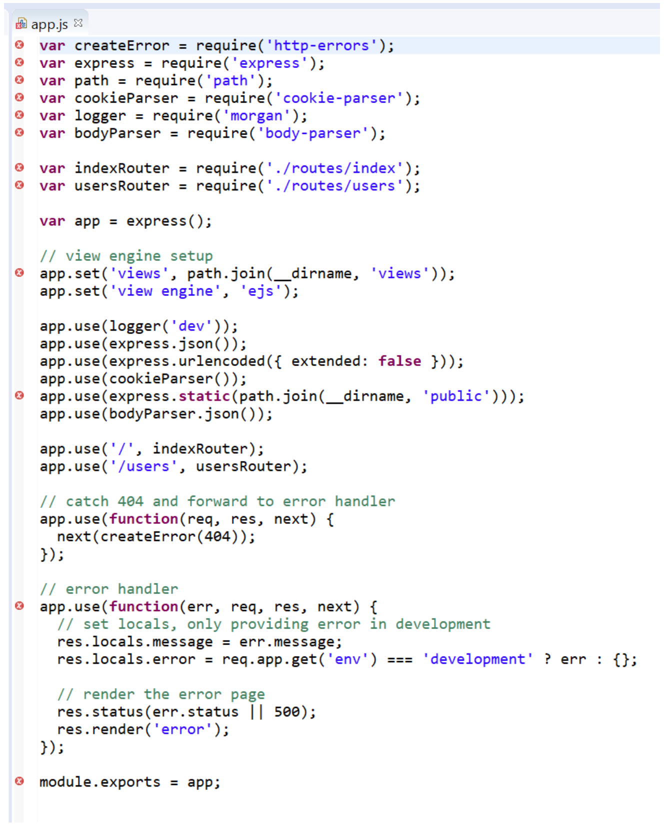
< Page 1/1 >

< Back | Next > | Finish | Cancel

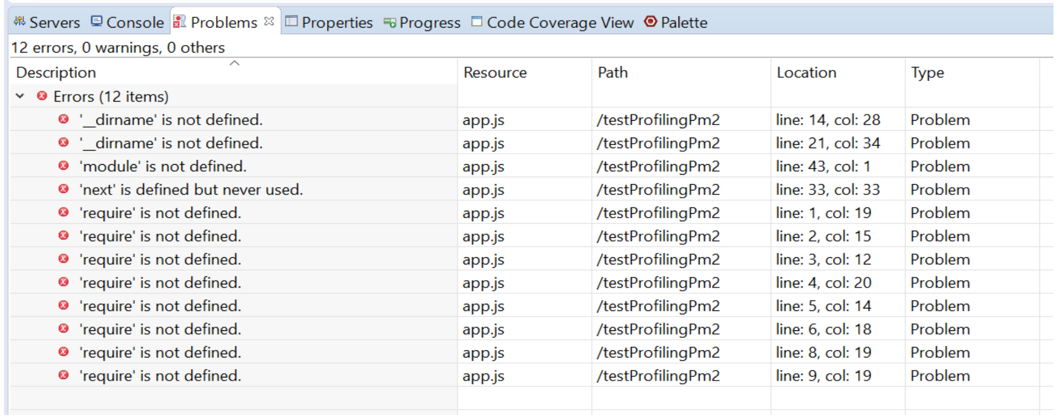
Node.js Tools in NodePlus

- Static Analysis for JavaScript files and Projects to validate coding rules





```
var createError = require('http-errors');
var express = require('express');
var path = require('path');
var cookieParser = require('cookie-parser');
var logger = require('morgan');
var bodyParser = require('body-parser');

var indexRouter = require('./routes/index');
var usersRouter = require('./routes/users');

var app = express();

// view engine setup
app.set('views', path.join(__dirname, 'views'));
app.set('view engine', 'ejs');

app.use(logger('dev'));
app.use(express.json());
app.use(express.urlencoded({ extended: false }));
app.use(cookieParser());
app.use(express.static(path.join(__dirname, 'public')));
app.use(bodyParser.json());

app.use('/', indexRouter);
app.use('/users', usersRouter);

// catch 404 and forward to error handler
app.use(function(req, res, next) {
  next(createError(404));
});

// error handler
app.use(function(err, req, res, next) {
  // set locals, only providing error in development
  res.locals.message = err.message;
  res.locals.error = req.app.get('env') === 'development' ? err : {};

  // render the error page
  res.status(err.status || 500);
  res.render('error');
});

module.exports = app;
```



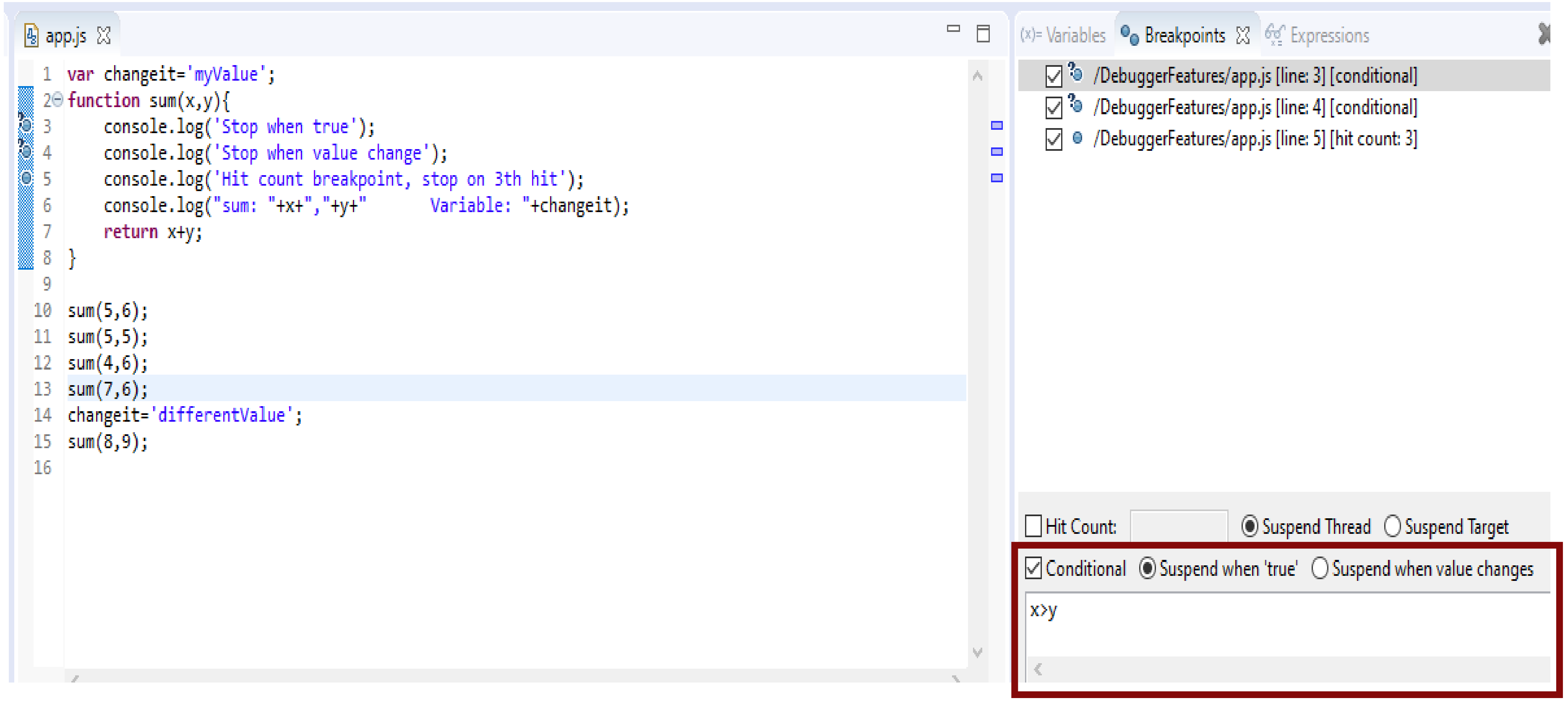
12 errors, 0 warnings, 0 others

Description	Resource	Path	Location	Type
Errors (12 items)				
'__dirname' is not defined.	app.js	/testProfilingPm2	line: 14, col: 28	Problem
'__dirname' is not defined.	app.js	/testProfilingPm2	line: 21, col: 34	Problem
'module' is not defined.	app.js	/testProfilingPm2	line: 43, col: 1	Problem
'next' is defined but never used.	app.js	/testProfilingPm2	line: 33, col: 33	Problem
'require' is not defined.	app.js	/testProfilingPm2	line: 1, col: 19	Problem
'require' is not defined.	app.js	/testProfilingPm2	line: 2, col: 15	Problem
'require' is not defined.	app.js	/testProfilingPm2	line: 3, col: 12	Problem
'require' is not defined.	app.js	/testProfilingPm2	line: 4, col: 20	Problem
'require' is not defined.	app.js	/testProfilingPm2	line: 5, col: 14	Problem
'require' is not defined.	app.js	/testProfilingPm2	line: 6, col: 18	Problem
'require' is not defined.	app.js	/testProfilingPm2	line: 8, col: 19	Problem
'require' is not defined.	app.js	/testProfilingPm2	line: 9, col: 19	Problem

HCL SOFTWARE

Node.js Debugger

- Support for conditional breakpoints
  - The Node.js debugger can be set to suspend when the conditional expression is true or when it changes



```
var changeit='myValue';
function sum(x,y){
    console.log('Stop when true');
    console.log('Stop when value change');
    console.log('Hit count breakpoint, stop on 3th hit');
    console.log("sum: "+x+","+y+"       Variable: "+changeit);
    return x+y;
}

sum(5,6);
sum(5,5);
sum(4,6);
sum(7,6);
changeit='differentValue';
sum(8,9);
```

```
/DebuggerFeatures/app.js [line: 3] [conditional]
/DebuggerFeatures/app.js [line: 4] [conditional]
/DebuggerFeatures/app.js [line: 5] [hit count: 3]
```

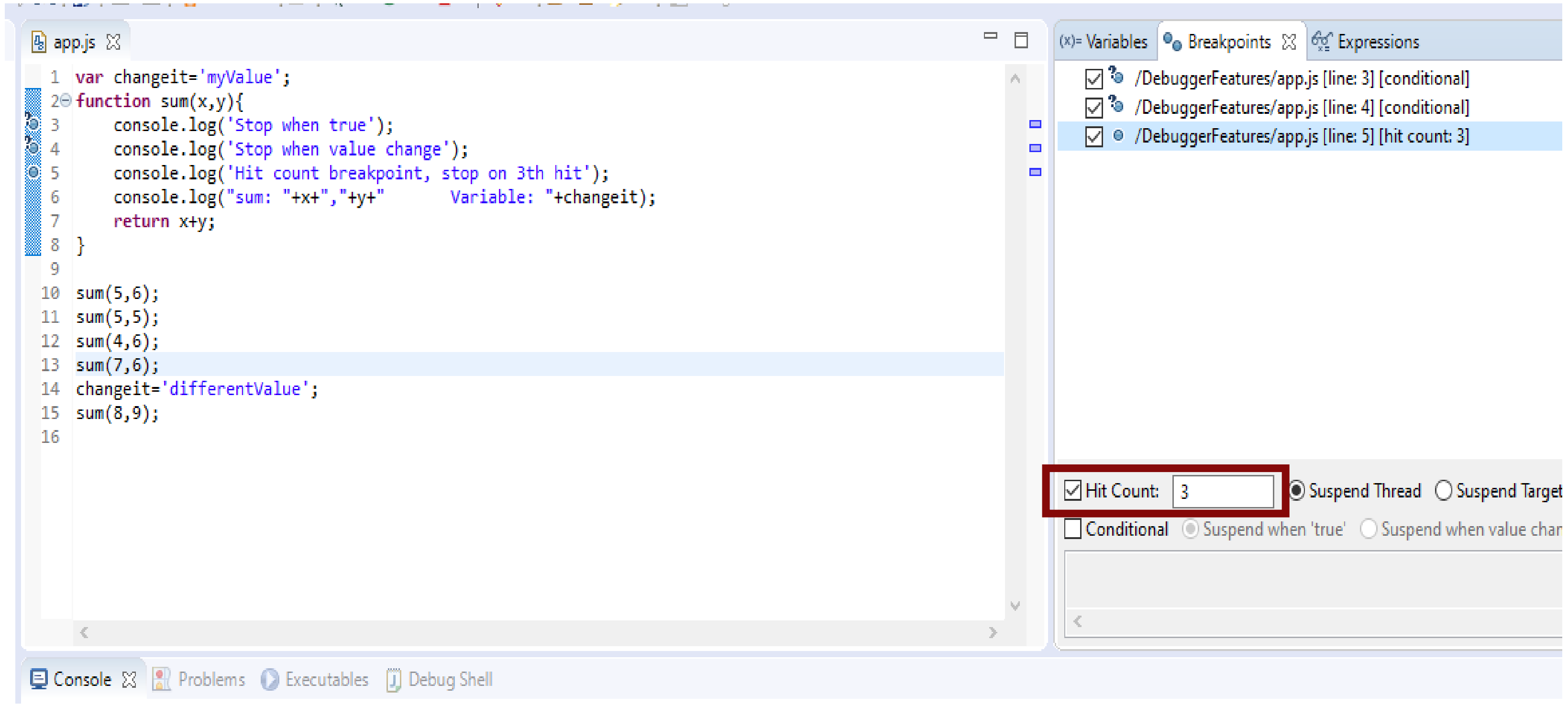
☐ Hit Count: [✓ Suspend Thread] Suspend Target

☑ Conditional [✓ Suspend when 'true'] Suspend when value changes

```
x>y
```

Node.js Debugger

- Support for Hit Count breakpoints
  - The Node.js debugger can be set to suspend when a breakpoint has been hit a certain number of times



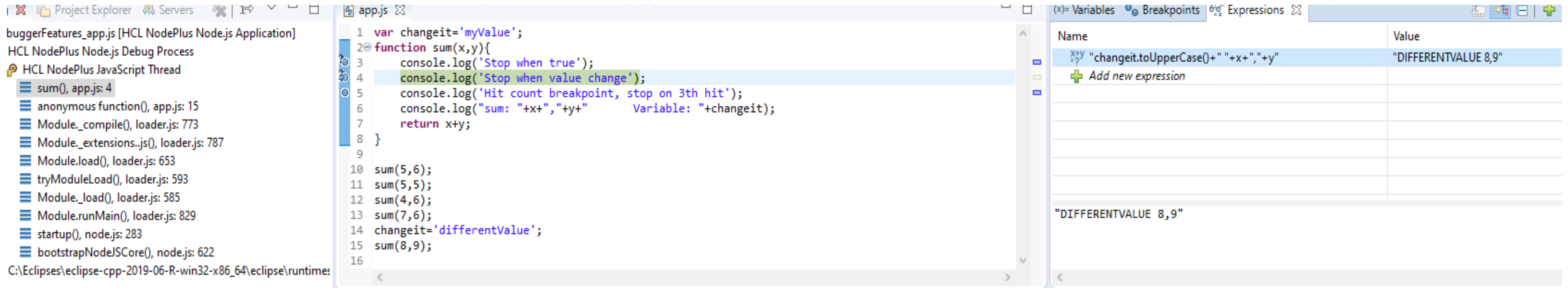
```
var changeit='myValue';
function sum(x,y){
    console.log('Stop when true');
    console.log('Stop when value change');
    console.log('Hit count breakpoint, stop on 3th hit');
    console.log("sum: "+x+","+y+"       Variable: "+changeit);
    return x+y;
}

sum(5,6);
sum(5,5);
sum(4,6);
sum(7,6);
changeit='differentValue';
sum(8,9);
```

/DebuggerFeatures/app.js [line: 3] [conditional]
/DebuggerFeatures/app.js [line: 4] [conditional]
/DebuggerFeatures/app.js [line: 5] [hit count: 3]

[✓] Hit Count: 3 [● Suspend Thread] ○ Suspend Target
[ ] Conditional ○ Suspend when 'true' ○ Suspend when value char

Node.js Debugger

- In the Expressions tab the Node.js debugger can now evaluate a JavaScript expression in real time



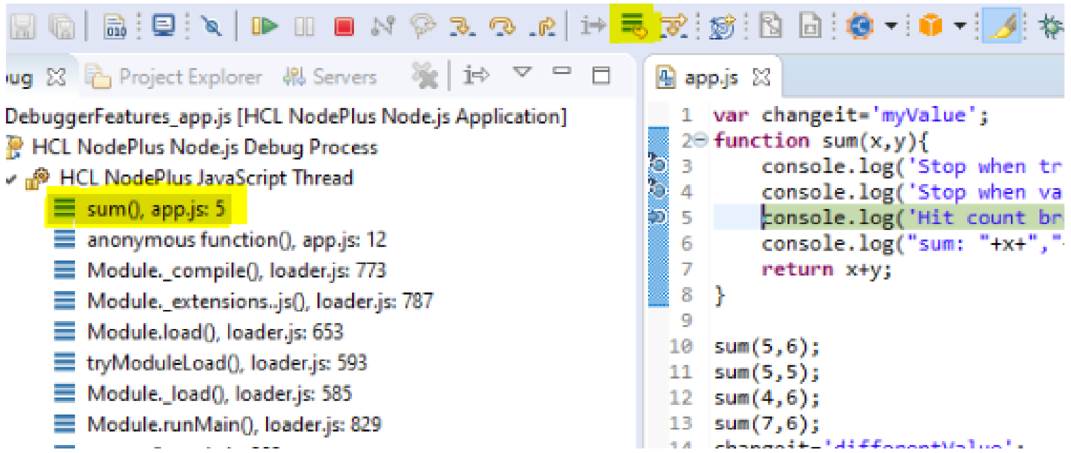
```
buggerFeatures_app.js [HCL NodePlus Node.js Application]
HCL NodePlus Node.js Debug Process
HCL NodePlus JavaScript Thread
    sum(), app.js: 4
    anonymous function(), app.js: 15
    Module._compile(), loader.js: 773
    Module._extensions..js(), loader.js: 787
    Module.load(), loader.js: 653
    tryModuleLoad(), loader.js: 593
    Module._load(), loader.js: 585
    Module.runMain(), loader.js: 829
    startup(), node.js: 283
    bootstrapNodeJSCore(), node.js: 622
C:\Eclipses\eclipse-cpp-2019-06-R-win32-x86_64\eclipse\runtimes
```

```js
var changeit='myValue';
function sum(x,y){
    console.log('Stop when true');
    console.log('Stop when value change');
    console.log('Hit count breakpoint, stop on 3th hit');
    console.log("sum: "+x+","+y+"       Variable: "+changeit);
    return x+y;
}

sum(5,6);
sum(5,5);
sum(4,6);
sum(7,6);
changeit='differentValue';
sum(8,9);
```

Name	Value
"changeit.toUpperCase()+" "+x+","+y"	"DIFFERENTVALUE 8,9"
Add new expression	

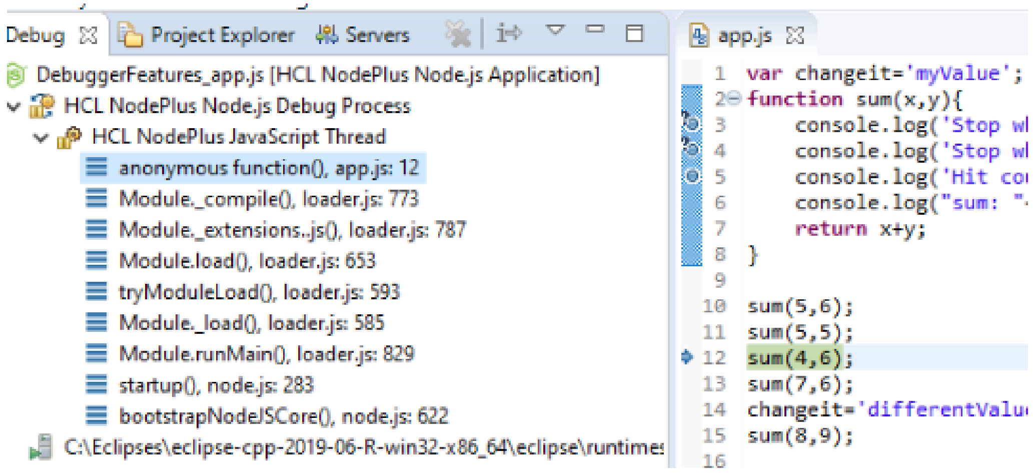
```
"DIFFERENTVALUE 8,9"
```

- Support for “Drop to Frame”
  - When the application is suspended you can select a stack frame and press the “Drop to Frame” button to restart execution from there



```
DebuggerFeatures_app.js [HCL NodePlus Node.js Application]
HCL NodePlus Node.js Debug Process
HCL NodePlus JavaScript Thread
    sum(), app.js: 5
    anonymous function(), app.js: 12
    Module._compile(), loader.js: 773
    Module._extensions..js(), loader.js: 787
    Module.load(), loader.js: 653
    tryModuleLoad(), loader.js: 593
    Module._load(), loader.js: 585
    Module.runMain(), loader.js: 829
```

```js
var changeit='myValue';
function sum(x,y){
    console.log('Stop when tr
    console.log('Stop when va
    console.log('Hit count br
    console.log("sum: "+x+","
    return x+y;
}

sum(5,6);
sum(5,5);
sum(4,6);
sum(7,6);
```

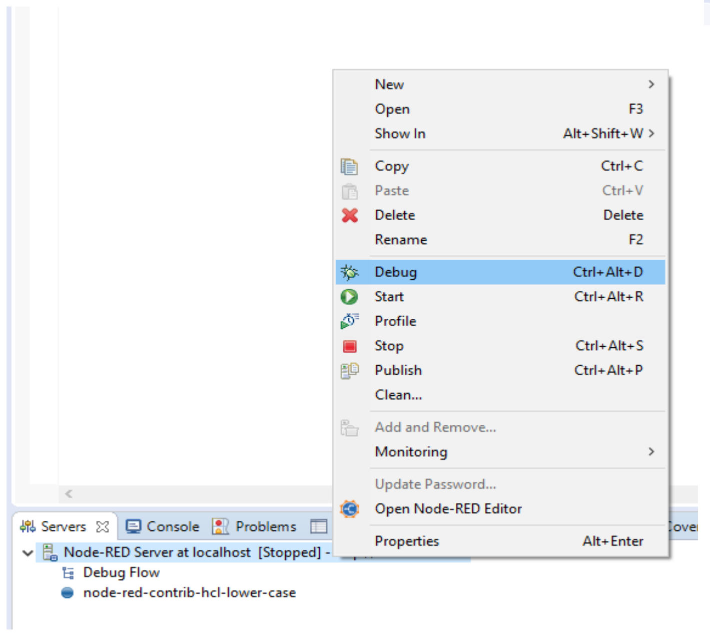


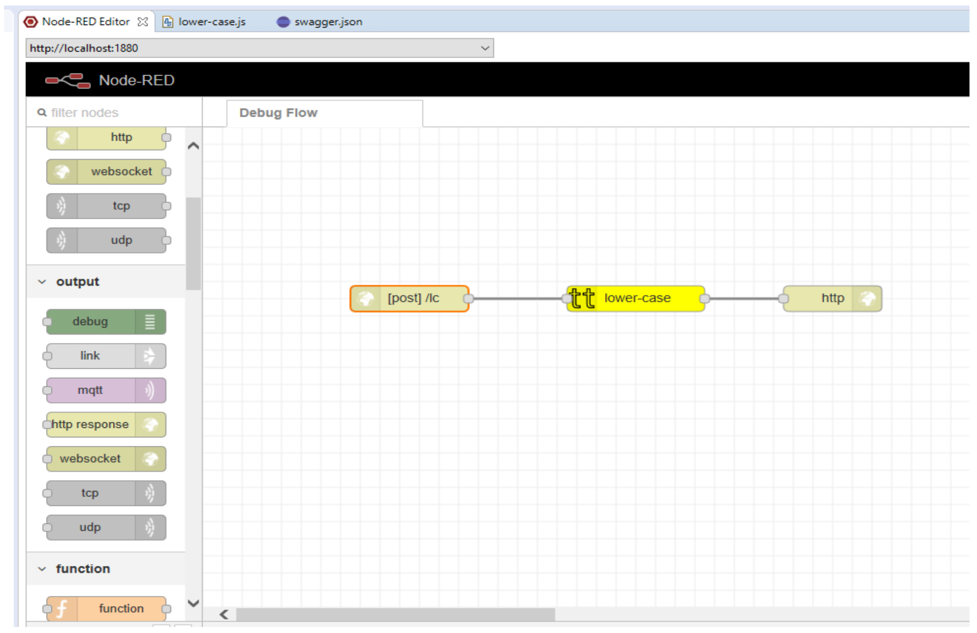
```
DebuggerFeatures_app.js [HCL NodePlus Node.js Application]
HCL NodePlus Node.js Debug Process
HCL NodePlus JavaScript Thread
    anonymous function(), app.js: 12
    Module._compile(), loader.js: 773
    Module._extensions..js(), loader.js: 787
    Module.load(), loader.js: 653
    tryModuleLoad(), loader.js: 593
    Module._load(), loader.js: 585
    Module.runMain(), loader.js: 829
    startup(), node.js: 283
    bootstrapNodeJSCore(), node.js: 622
C:\Eclipses\eclipse-cpp-2019-06-R-win32-x86_64\eclipse\runtimes
```

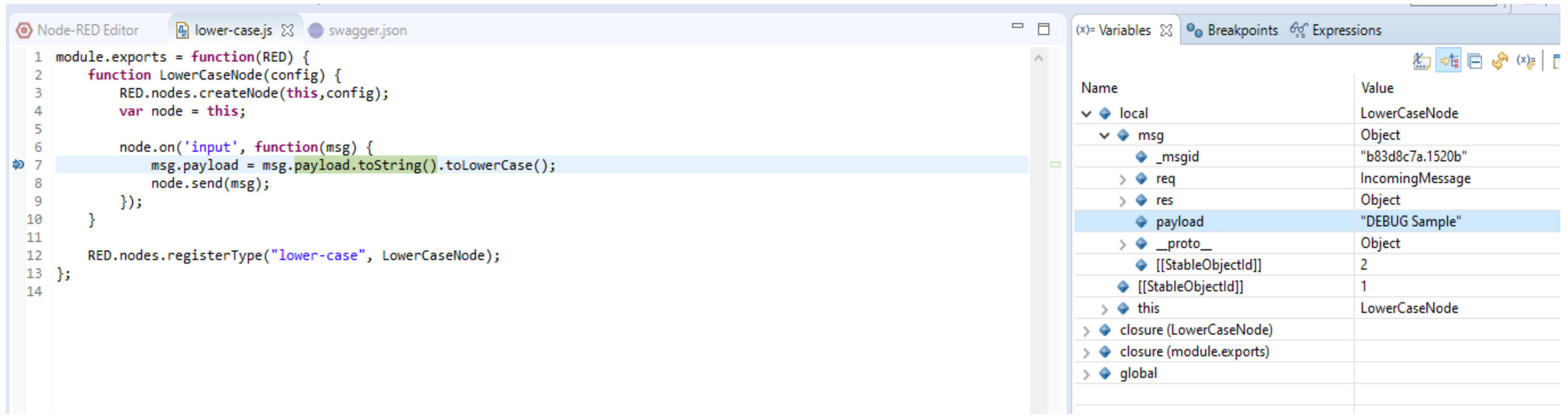
```js
var changeit='myValue';
function sum(x,y){
    console.log('Stop wl
    console.log('Stop wl
    console.log('Hit co
    console.log("sum: "
    return x+y;
}

sum(5,6);
sum(5,5);
sum(4,6);
sum(7,6);
changeit='differentValu
sum(8,9);
```

Node-RED Debugger

- You can use NodePlus to debug nodes deployed into Node-RED by enabling 'run in debug' mode





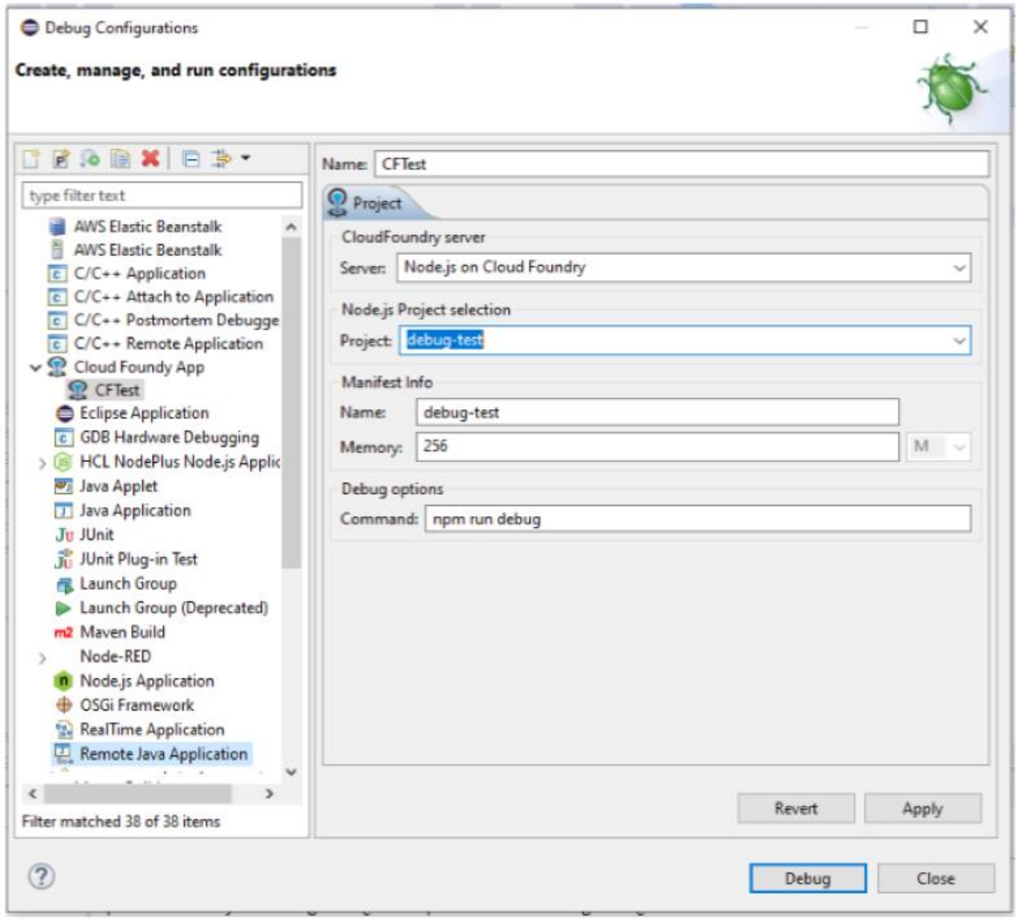


```
module.exports = function(RED) {
    function LowerCaseNode(config) {
        RED.nodes.createNode(this,config);
        var node = this;

        node.on('input', function(msg) {
            msg.payload = msg.payload.toString().toLowerCase();
            node.send(msg);
        });
    }

    RED.nodes.registerType("lower-case", LowerCaseNode);
};
```

Name	Value
local	LowerCaseNode
msg	Object
_msgid	"b83d8c7a.1520b"
req	IncomingMessage
res	Object
payload	"DEBUG Sample"
__proto__	Object
[[StableObjectId]]	2
[[StableObjectId]]	1
this	LowerCaseNode
closure (LowerCaseNode)	
closure (module.exports)	
global	

Debug Node.js Applications on Cloud Foundry

- You can now debug Node.js applications that are running on Cloud Foundry



Debug Configurations

Create, manage, and run configurations

type filter text

- AWS Elastic Beanstalk
- AWS Elastic Beanstalk
- C/C++ Application
- C/C++ Attach to Application
- C/C++ Postmortem Debugge
- C/C++ Remote Application
- Cloud Foundy App
  - CFTest
- Eclipse Application
- GDB Hardware Debugging
- HCL NodePlus Node.js Applic
- Java Applet
- Java Application
- JUnit
- JUnit Plug-in Test
- Launch Group
- Launch Group (Deprecated)
- Maven Build
- Node-RED
- Node.js Application
- OSGi Framework
- RealTime Application
- Remote Java Application

Filter matched 38 of 38 items

Name:	CFTest

Project

CloudFoundry server

Server:	Node.js on Cloud Foundry

Node.js Project selection

Project:	debug-test

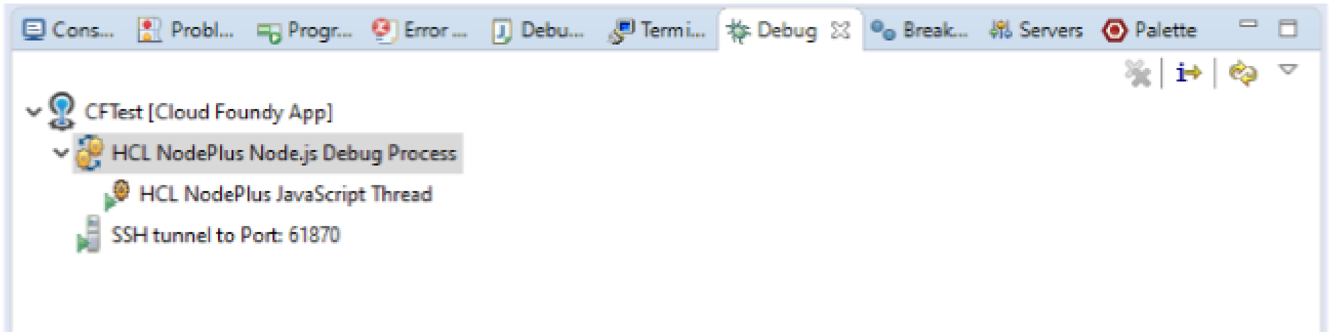
Manifest Info

Name:	debug-test
Memory:	256 M

Debug options

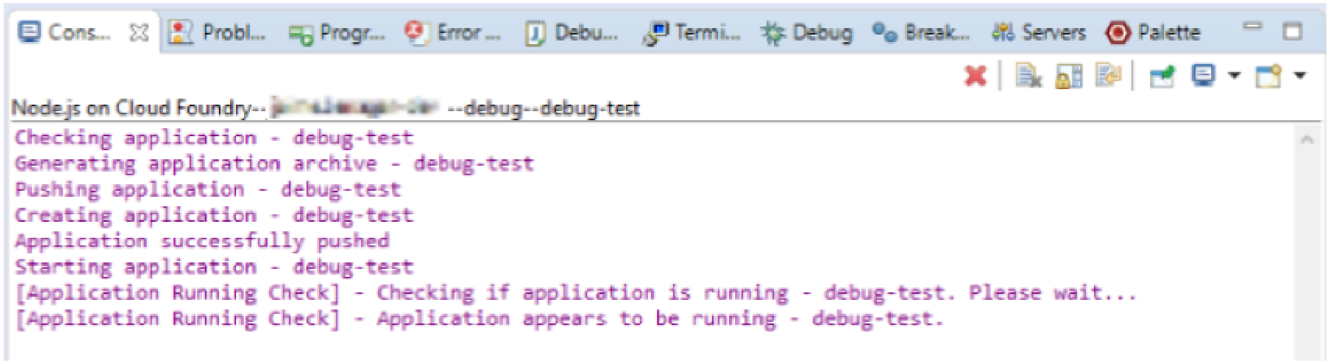
Command:	npm run debug

Revert | Apply | Debug | Close



Cons... Probl... Progr... Error ... Debu... Termi... Debug Break... Servers Palette

- CFTest [Cloud Foundy App]
  - HCL NodePlus Node.js Debug Process
    - HCL NodePlus JavaScript Thread
  - SSH tunnel to Port: 61870



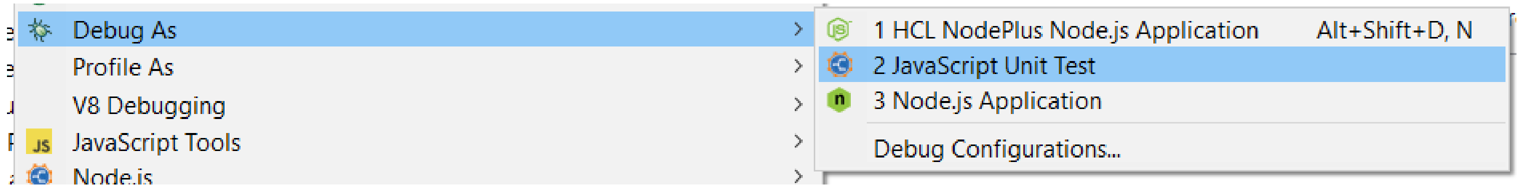
Cons... Probl... Progr... Error ... Debu... Termi... Debug Break... Servers Palette

Node.js on Cloud Foundry--[redacted] --debug--debug-test

```
Checking application - debug-test
Generating application archive - debug-test
Pushing application - debug-test
Creating application - debug-test
Application successfully pushed
Starting application - debug-test
[Application Running Check] - Checking if application is running - debug-test. Please wait...
[Application Running Check] - Application appears to be running - debug-test.
```

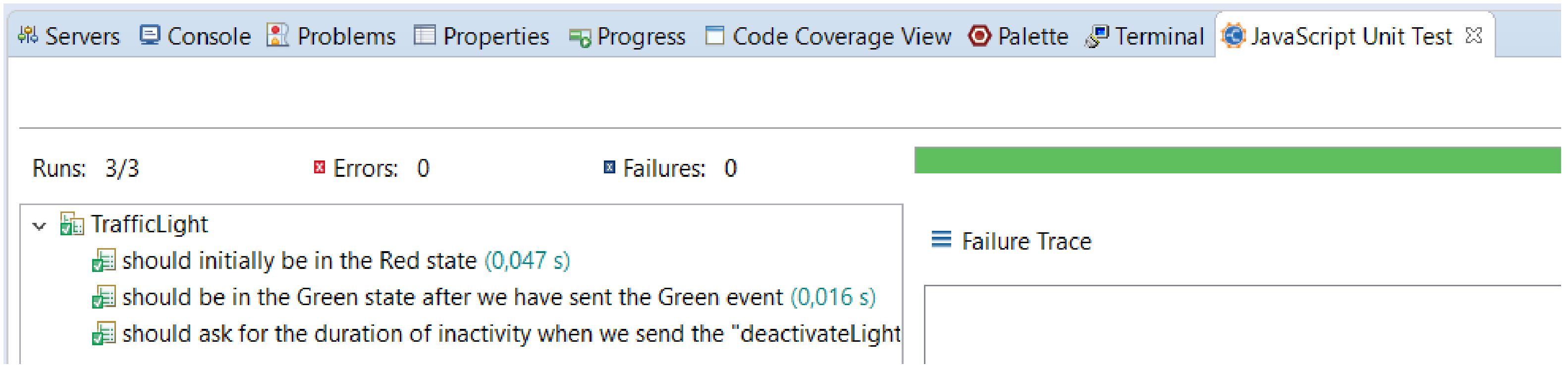
Unit Testing with Mocha

- Node-RED nodes and Node.js applications can be unit tested using the Mocha framework
  - Unit tests can also be debugged



- Using the TCP Server library in an RTist application makes it possible to test it too with Mocha
  - An example is here https://github.com/hcl-pnp-rtist/rt-test-probe

- Results from unit testing are shown in a JavaScript Unit Test view



Runs: 3/3 Errors: 0 Failures: 0

TrafficLight
- should initially be in the Red state (0,047 s)
- should be in the Green state after we have sent the Green event (0,016 s)
- should ask for the duration of inactivity when we send the "deactivateLight

HCL

Relationship™

BEYOND THE CONTRACT

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

WATCH THE **FILM**