HCL

What’s New in HCL RTist 11

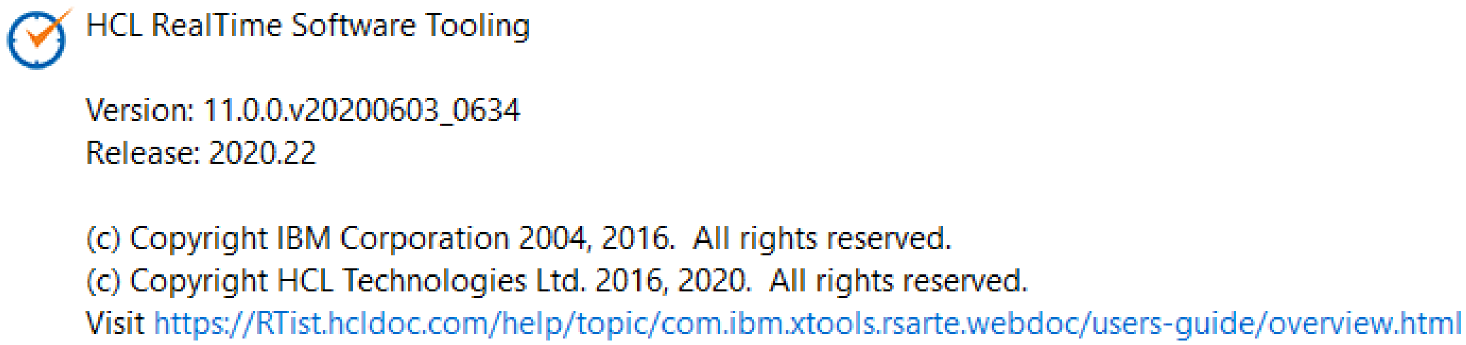
updated for release 2020.22

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 RealTime Software Tooling

Version: 11.0.0.v20200603_0634
Release: 2020.22

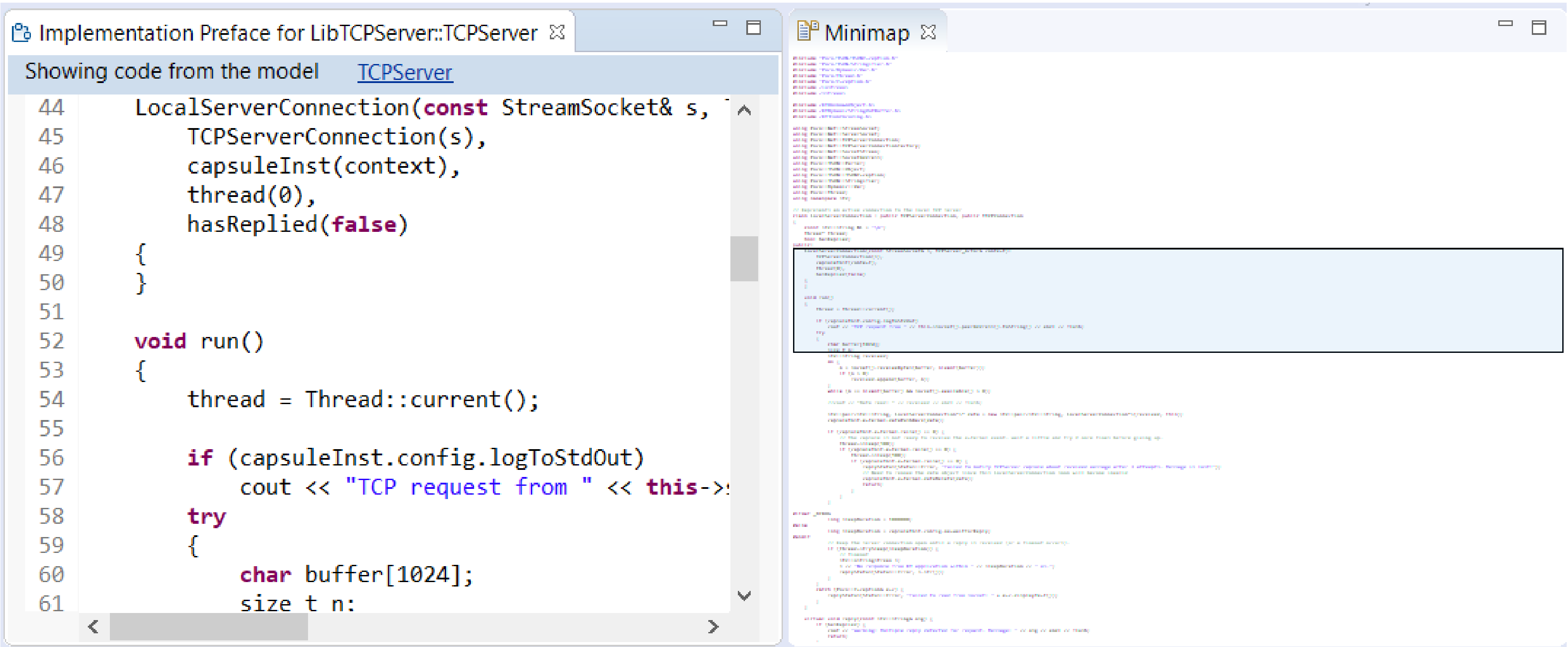
(c) Copyright IBM Corporation 2004, 2016. All rights reserved.
(c) Copyright HCL Technologies Ltd. 2016, 2020. All rights reserved.
Visit https://RTist.hcldoc.com/help/topic/com.ibm.xtools.rsarte.webdoc/users-guide/overview.html

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

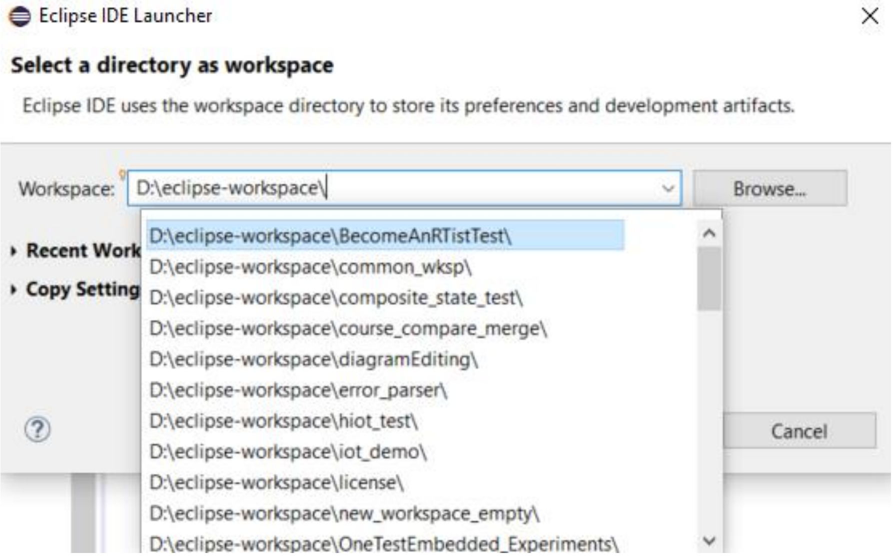

Implementation Preface for LibTCPServer::TCPServer
Showing code from the model
TCPServer
Minimap

```
    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;
```

HCL SOFTWARE

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



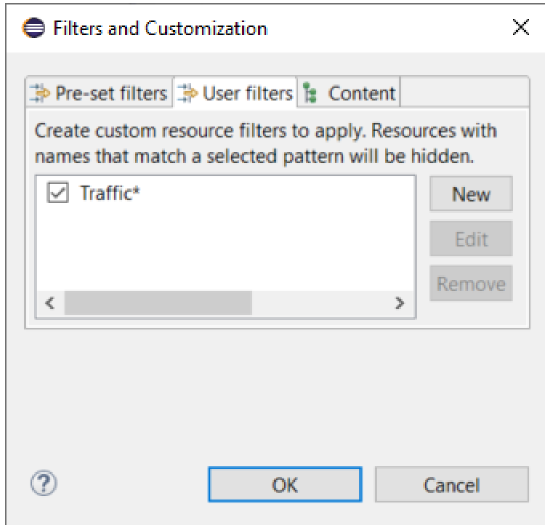
Eclipse IDE Launcher

Select a directory as workspace

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

Workspace: D:\eclipse-workspace\

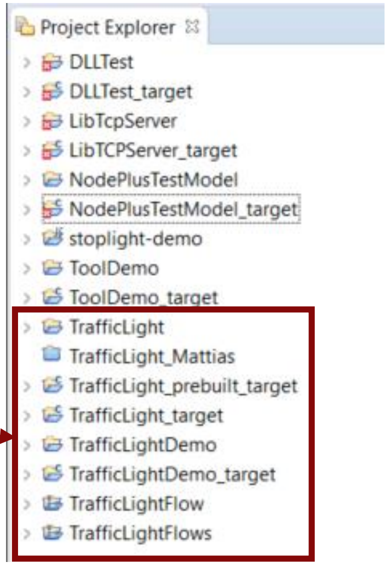
D:\eclipse-workspace\BecomeAnRTistTest\
D:\eclipse-workspace\common_wksp\
D:\eclipse-workspace\composite_state_test\
D:\eclipse-workspace\course_compare_merge\
D:\eclipse-workspace\diagramEditing\
D:\eclipse-workspace\error_parser\
D:\eclipse-workspace\hiot_test\
D:\eclipse-workspace\iot_demo\
D:\eclipse-workspace\license\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



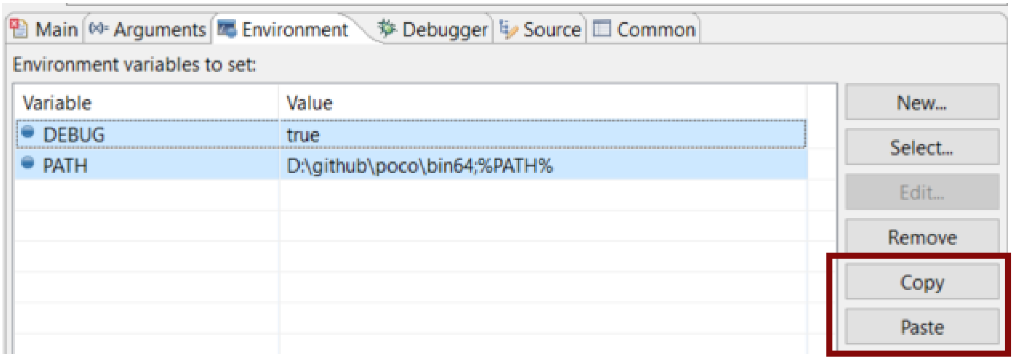
Project Explorer

- 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

will be hidden
by the user filter

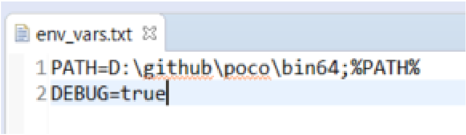
Eclipse 4.12 (2019.06)

- 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)



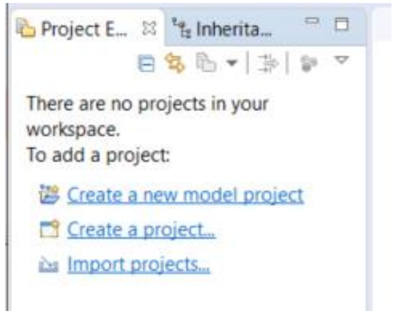
Environment variables to set:

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



```
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

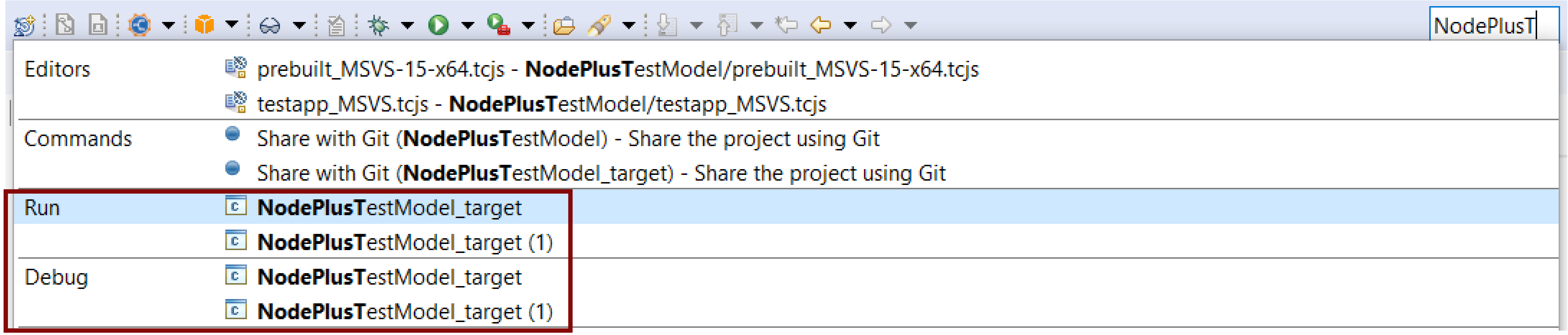


There are no projects in your workspace.
To add a project:

- 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

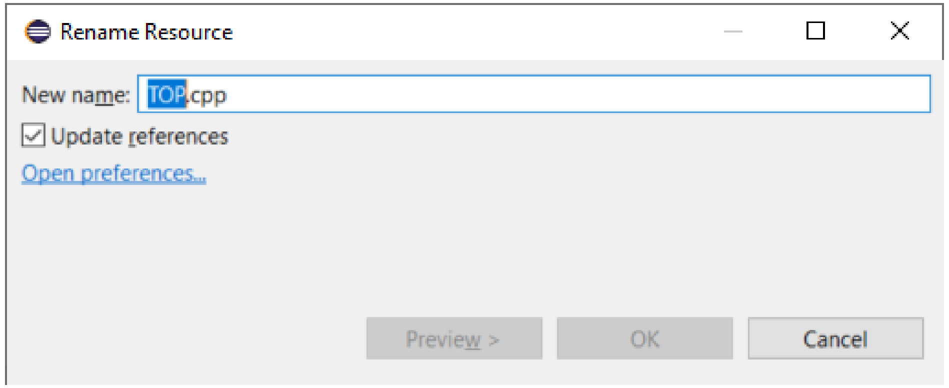


NodePlusT

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)

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



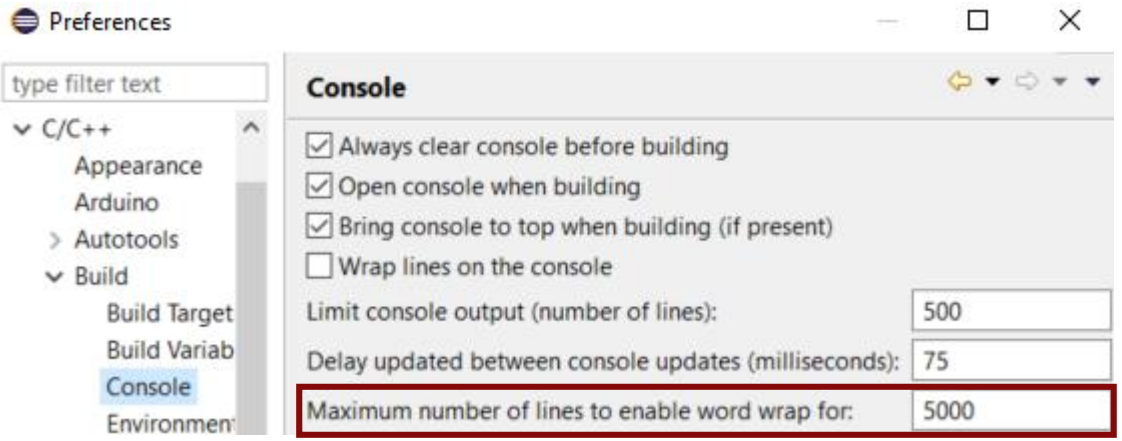
Rename Resource

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



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

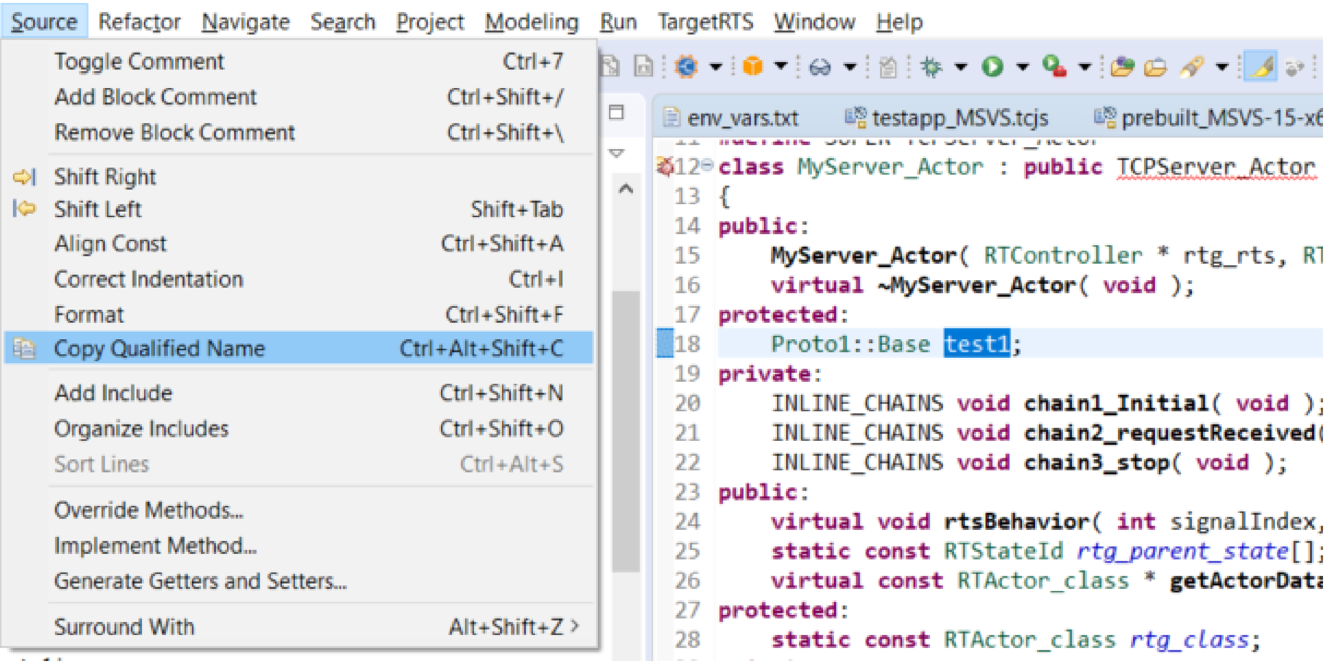
Delay updated between console updates (milliseconds): 75

Maximum number of lines to enable word wrap for: 5000

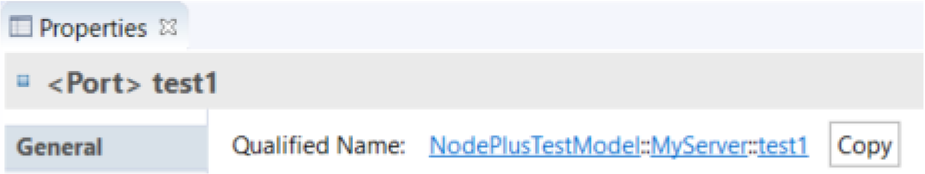
CDT 9.8 (included as part of Eclipse 2019.06)

- 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



```
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 * getActorDat
protected:
    static const RTActor_class rtg_class;
```



<Port> test1

Qualified Name: NodePlusTestModel::MyServer::test1

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

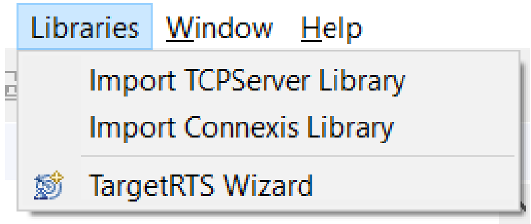
Newer EGit Version in the EGit Integration

- 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

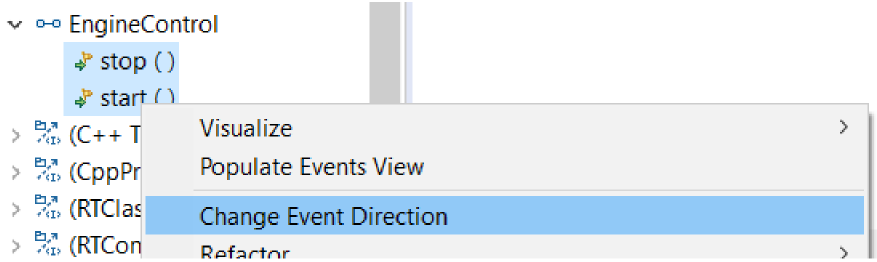
- A new menu in the menu bar with a few useful commands related to libraries
  - Contains the command for launching the TargetRTS wizard (i.e. the menu replaces the TargetRTS menu)



- 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



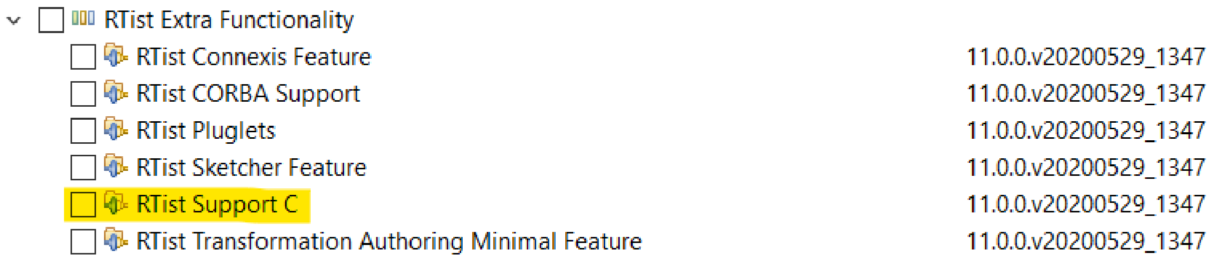
- Use the new “Change Event Direction” command in the Project Explorer context menu of a protocol event (or multiple events)

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

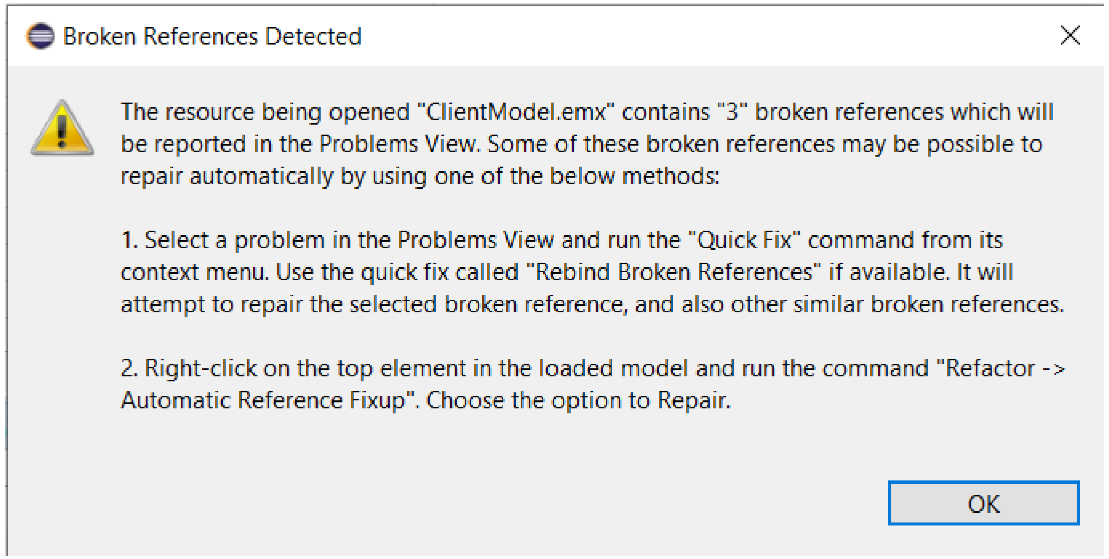


□ RTist Extra Functionality	
□ RTist Connexis Feature	11.0.0.v20200529_1347
□ RTist CORBA Support	11.0.0.v20200529_1347
□ RTist Pluglets	11.0.0.v20200529_1347
□ RTist Sketcher Feature	11.0.0.v20200529_1347
□ RTist Support C	11.0.0.v20200529_1347
□ RTist Transformation Authoring Minimal Feature	11.0.0.v20200529_1347

- 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

- 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.



Broken References Detected

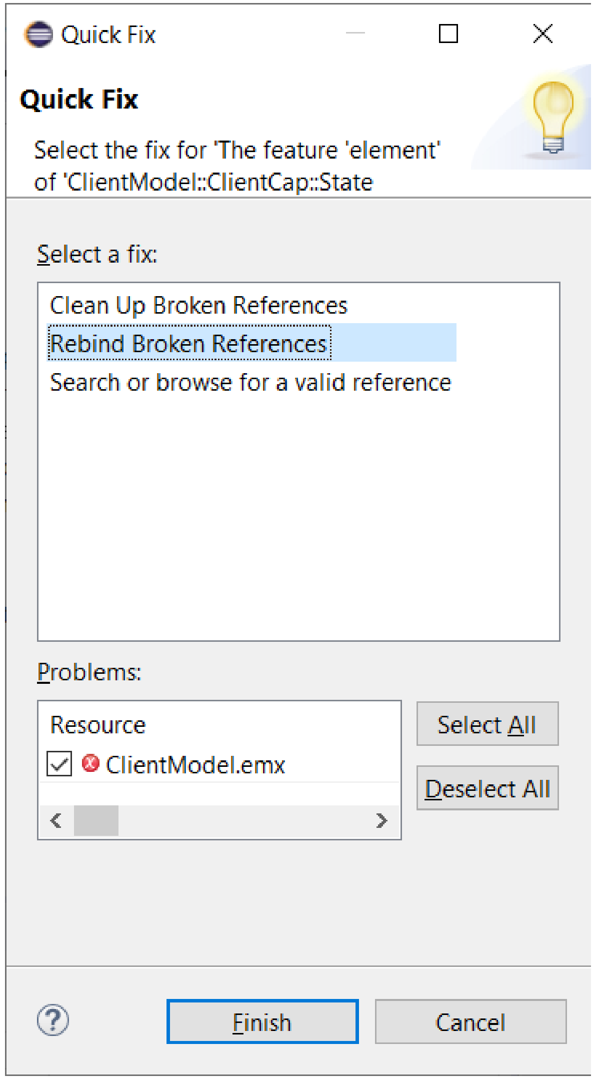
The resource being opened "ClientModel.emx" contains "3" broken references which will be reported in the Problems View. Some of these broken references may be possible to repair automatically by using one of the below methods:

1. Select a problem in the Problems View and run the "Quick Fix" command from its context menu. Use the quick fix called "Rebind Broken References" if available. It will attempt to repair the selected broken reference, and also other similar broken references.

2. Right-click on the top element in the loaded model and run the command "Refactor -> Automatic Reference Fixup". Choose the option to Repair.

Fixing Broken References

- 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

Quick Fix

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

Select a fix:

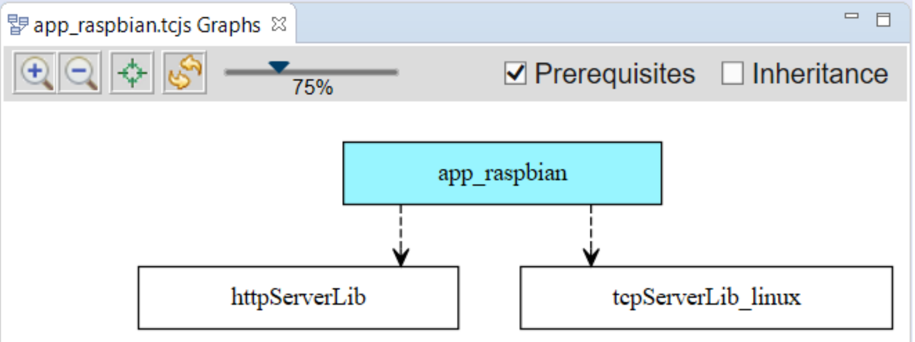
Clean Up Broken References
Rebind Broken References
Search or browse for a valid reference

Problems:

Resource
☑ ClientModel.emx

Graphic Visualization of TC Relationships

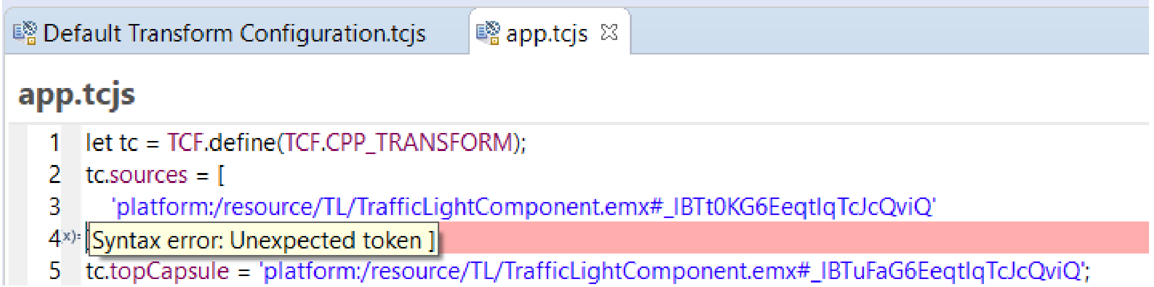
- The possibility to visualize TC relationships that was available in version 10.2 for TCs in the old file format, is now also supported for TCs in the new .tcjs file format
  - A new command **Visualize – Explore in Graphs** is available in the Project Explorer context menu of a TC
- The visualization 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



- You can double-click on TCs in the graph to open the TC in the TC editor
- This is currently an experimental feature

Transformation Configuration Editor Improvements

- 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'
tc.topCapsule = 'platform:/resource/TL/TrafficLightComponent.emx#_IBTuFaG6EeqtIqTcJcQviQ';
```

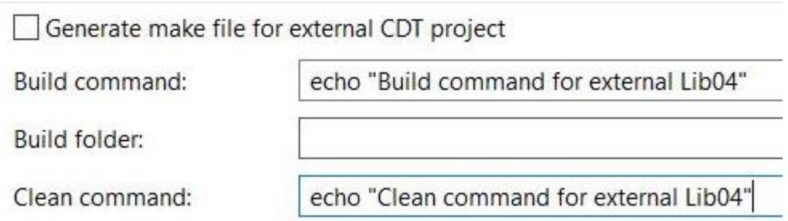
Syntax error: Unexpected token]

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

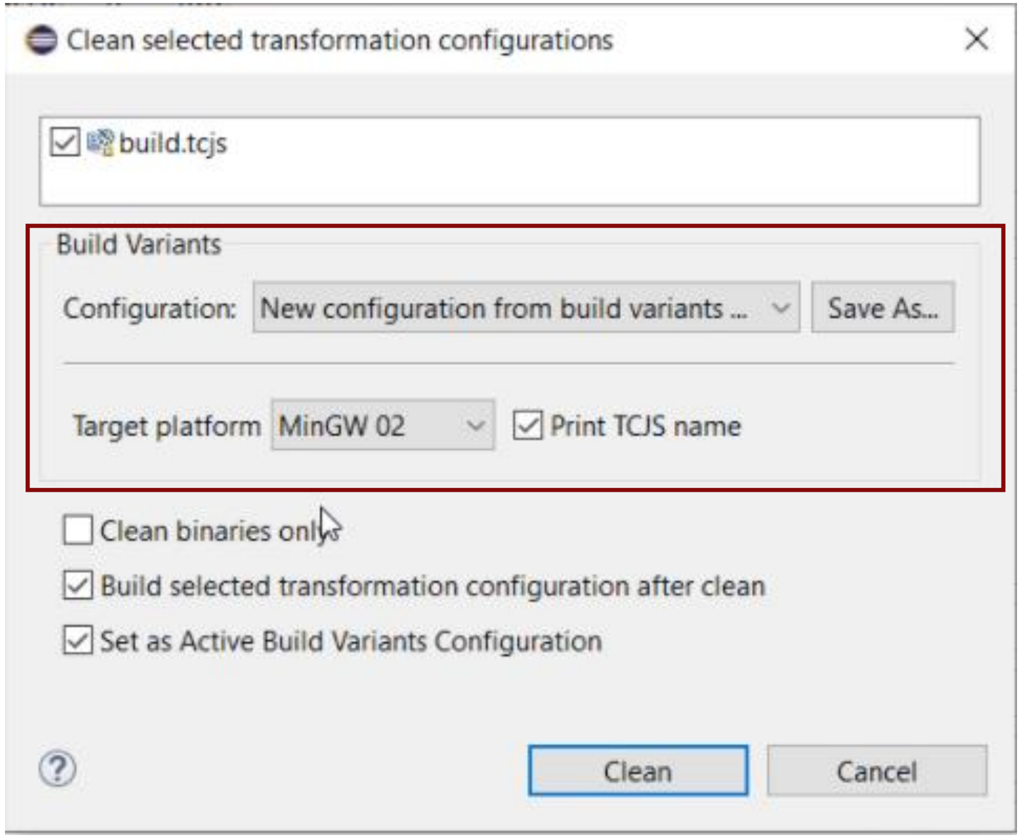


☐ Generate make file for external CDT project

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

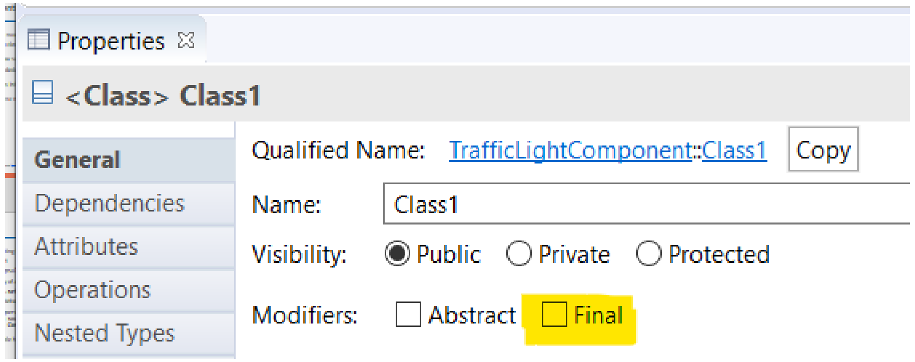
☐ Clean binaries only

☑ Build selected transformation configuration after clean

☑ Set as Active Build Variants Configuration

C++ 11 Improvements

- 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



Properties

<Class> Class1

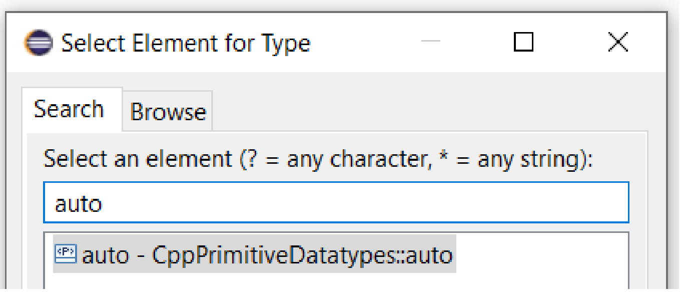
Qualified Name: TrafficLightComponent::Class1

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 Element for 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

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);
```

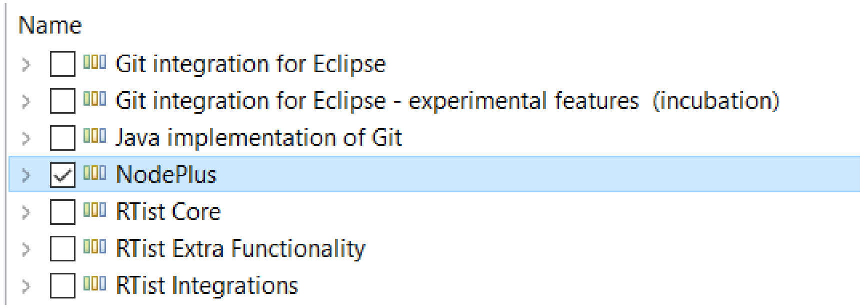
  - Creating a connector between two ports on capsule parts

```
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

NodePlus

RTist ONLY



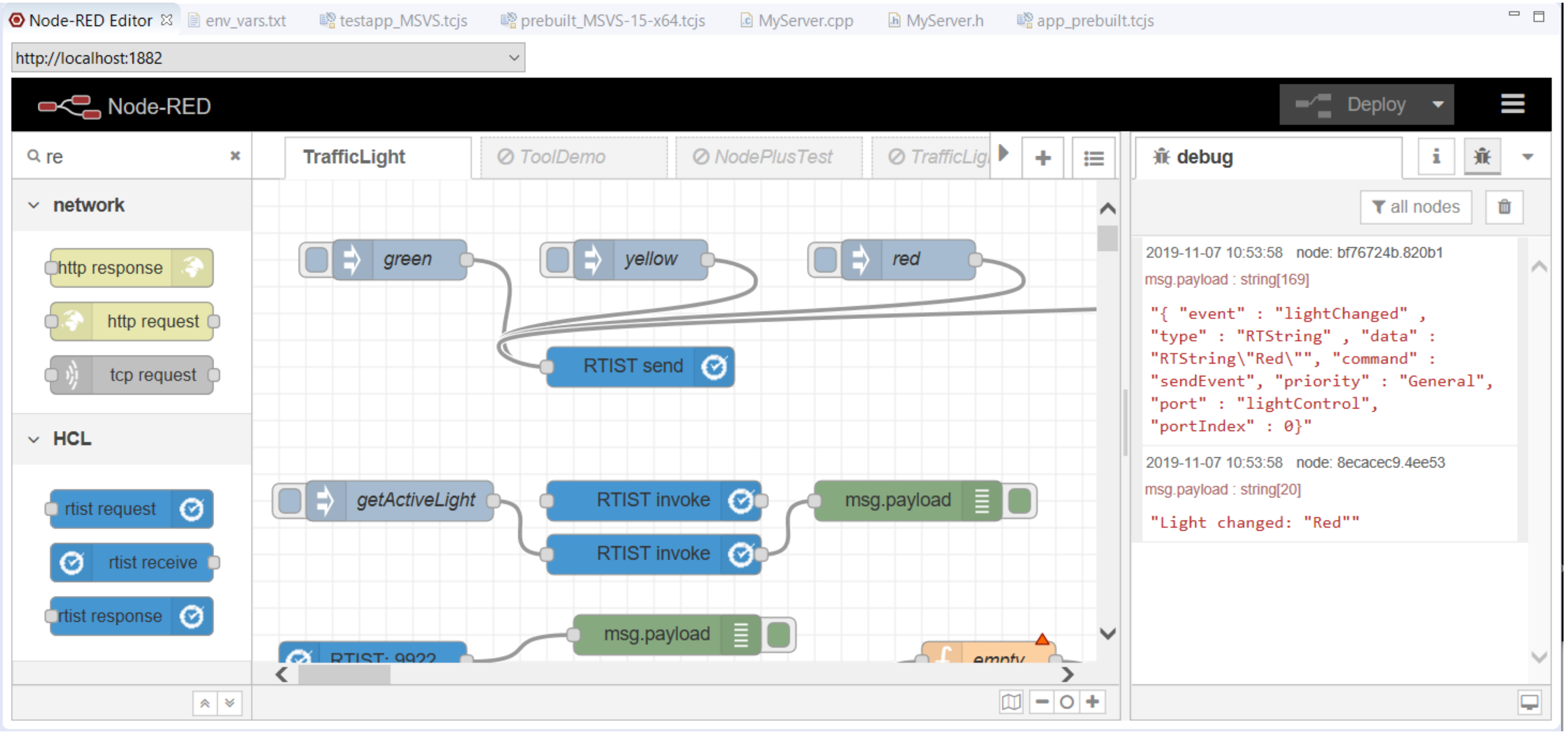
Name

- □ Git integration for Eclipse
- □ Git integration for Eclipse - experimental features (incubation)
- □ Java implementation of Git
- ☑ NodePlus
- □ RTist Core
- □ RTist Extra Functionality
- □ RTist Integrations

- HCL NodePlus is a new install component of HCL RTist
  - Note: Currently 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

Node-RED Tools in NodePlus

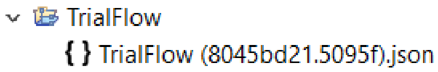
- Node-RED editor
  - Configure nodes and wire them together into flows
  - Deploy changes to the Node-RED server
  - View debug output and execution within the flow
  - Install new nodes to the palette (e.g. from the public library)



```
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""
```

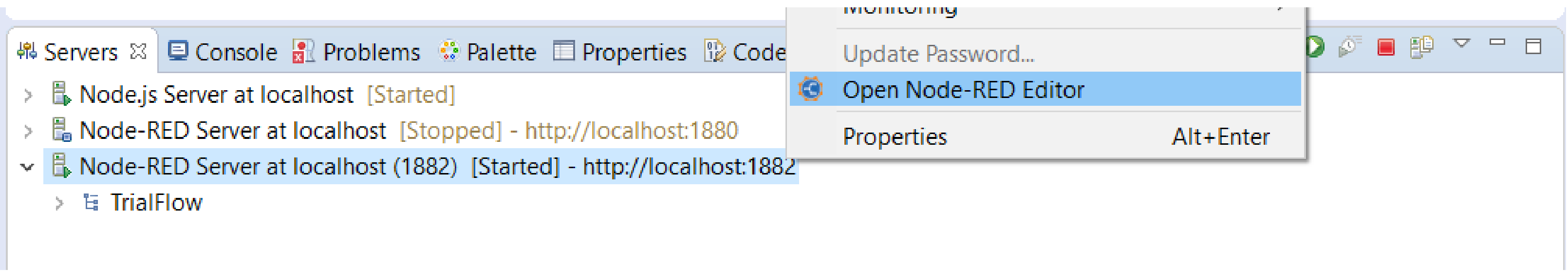
Node-RED Tools in NodePlus

- 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 servers within the Eclipse workbench
  - Deploy a Node-RED flow onto a selected server and open its Node-RED editor

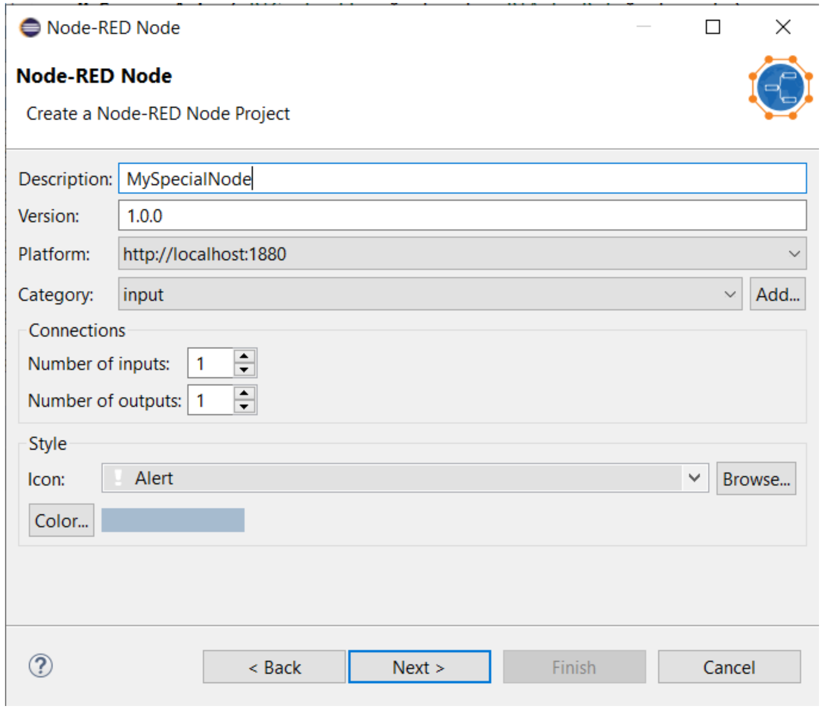


Node.js Server at localhost [Started]
Node-RED Server at localhost [Stopped] - http://localhost:1880
Node-RED Server at localhost (1882) [Started] - http://localhost:1882
TrialFlow

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



Node-RED Node

Node-RED Node

Create a Node-RED Node Project

Description: MySpecialNode

Version: 1.0.0

Platform: http://localhost:1880

Category: input

Connections

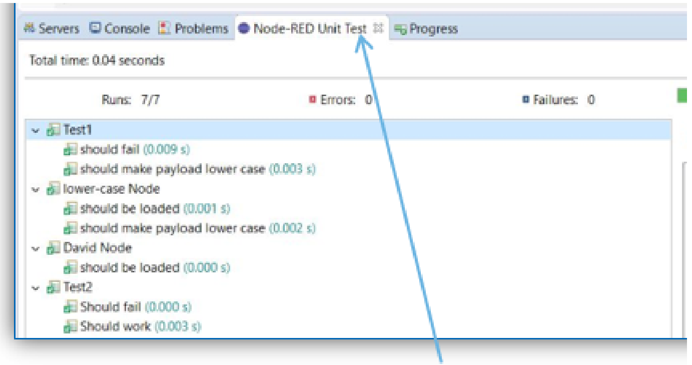
Number of inputs: 1

Number of outputs: 1

Style

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

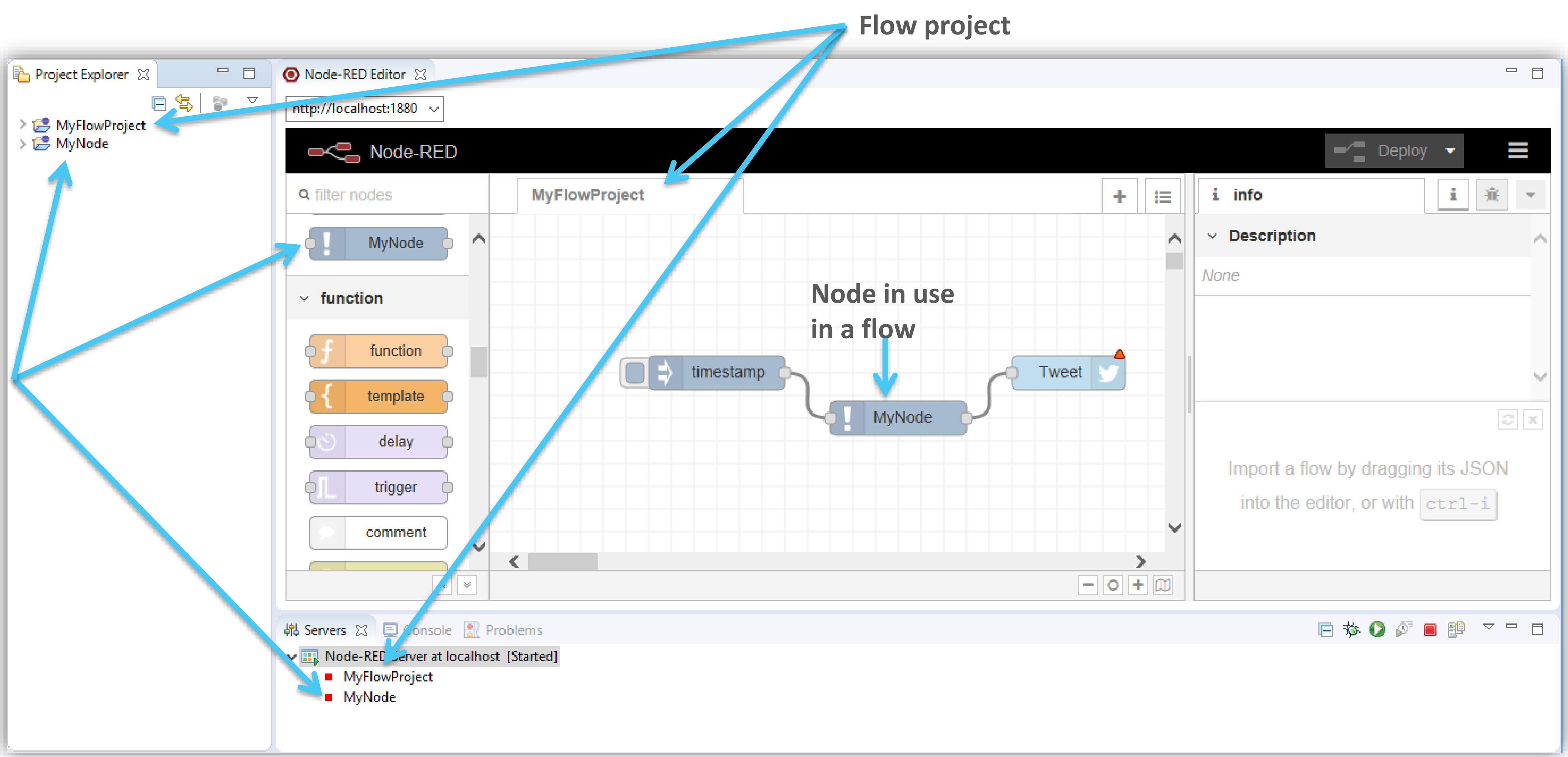


```
Total time: 0.04 seconds

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

Node-RED Tools in NodePlus

RTist ONLY



Flow project

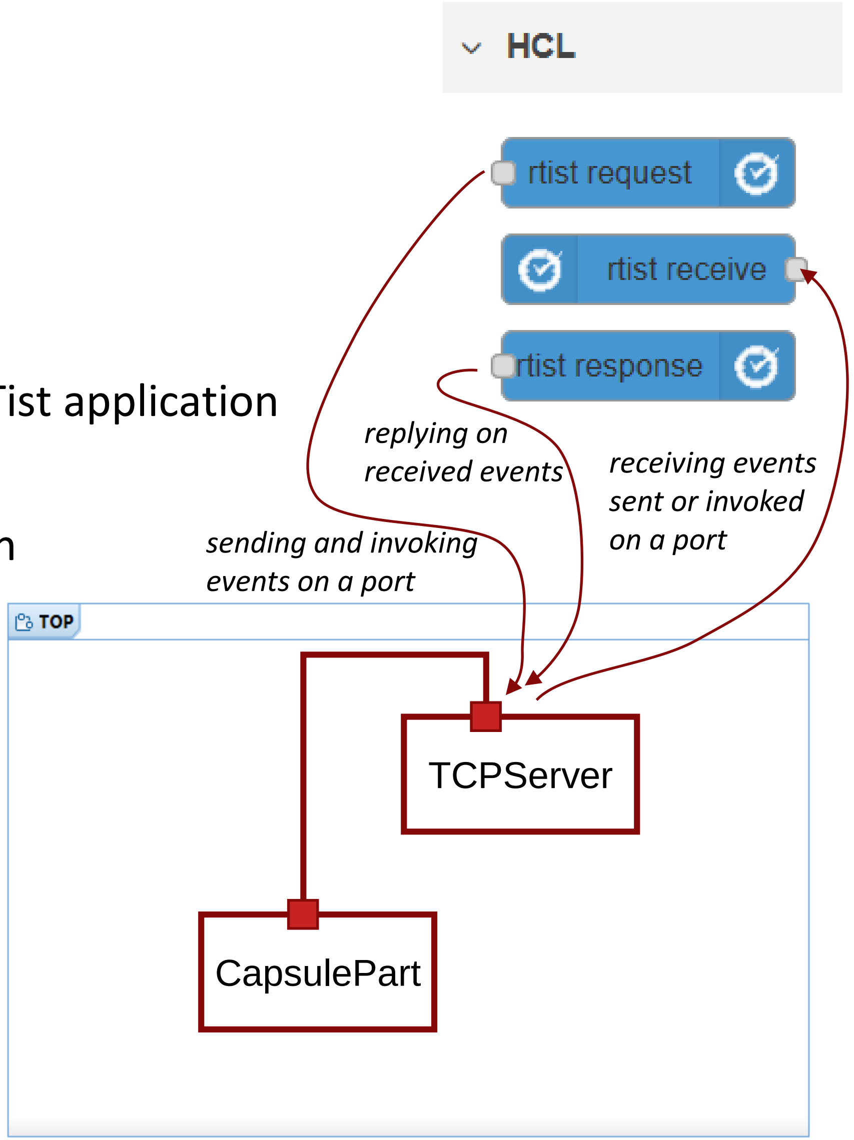
Node project

Node in use in a flow

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



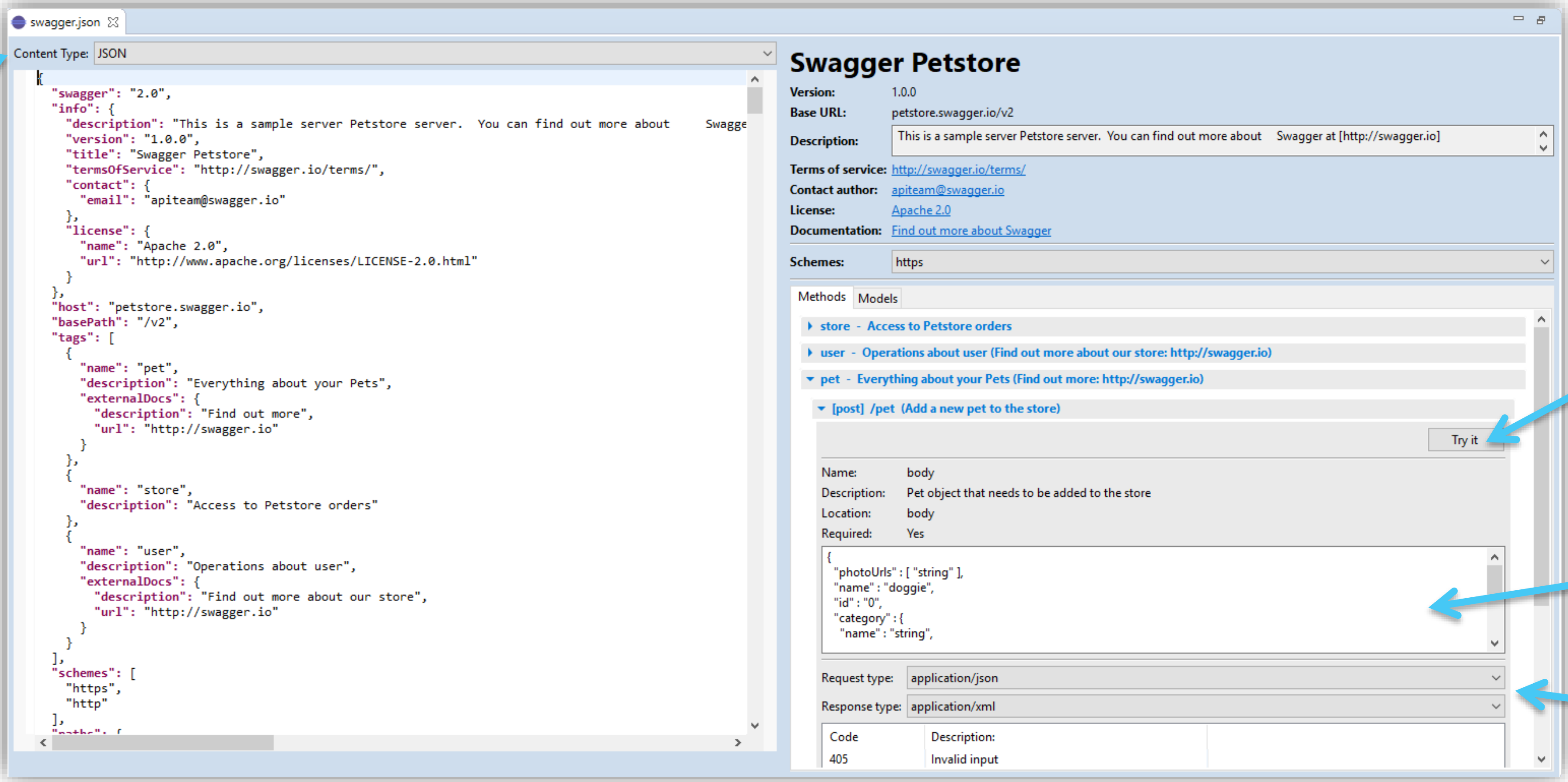
replying on received events

receiving events sent or invoked on a port

sending and invoking events on a port

Swagger Support in NodePlus

- Document APIs using Swagger in a new Swagger editor



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

swagger.json

Content Type: JSON

```
{
  "swagger": "2.0",
  "info": {
    "description": "This is a sample server Petstore server.  You can find out more about     Swagge
    "version": "1.0.0",
    "title": "Swagger Petstore",
    "termsOfService": "http://swagger.io/terms/",
    "contact": {
      "email": "apiteam@swagger.io"
    },
    "license": {
      "name": "Apache 2.0",
      "url": "http://www.apache.org/licenses/LICENSE-2.0.html"
    }
  },
  "host": "petstore.swagger.io",
  "basePath": "/v2",
  "tags": [
    {
      "name": "pet",
      "description": "Everything about your Pets",
      "externalDocs": {
        "description": "Find out more",
        "url": "http://swagger.io"
      }
    },
    {
      "name": "store",
      "description": "Access to Petstore orders"
    },
    {
      "name": "user",
      "description": "Operations about user",
      "externalDocs": {
        "description": "Find out more about our store",
        "url": "http://swagger.io"
      }
    }
  ],
  "schemes": [
    "https",
    "http"
  ],
  "paths": {
```

Swagger Petstore

Version: 1.0.0

Base URL: petstore.swagger.io/v2

Description: This is a sample server Petstore server. You can find out more about   Swagger at [http://swagger.io]

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

Contact author: apiteam@swagger.io

License: Apache 2.0

Documentation: Find out more about Swagger

Schemes: https

- store - Access to Petstore orders
- user - Operations about user (Find out more about our store: http://swagger.io)
- pet - Everything about your Pets (Find out more: http://swagger.io)
  - [post] /pet (Add a new pet to the store)

Name: body

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

Location: body

Required: Yes

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

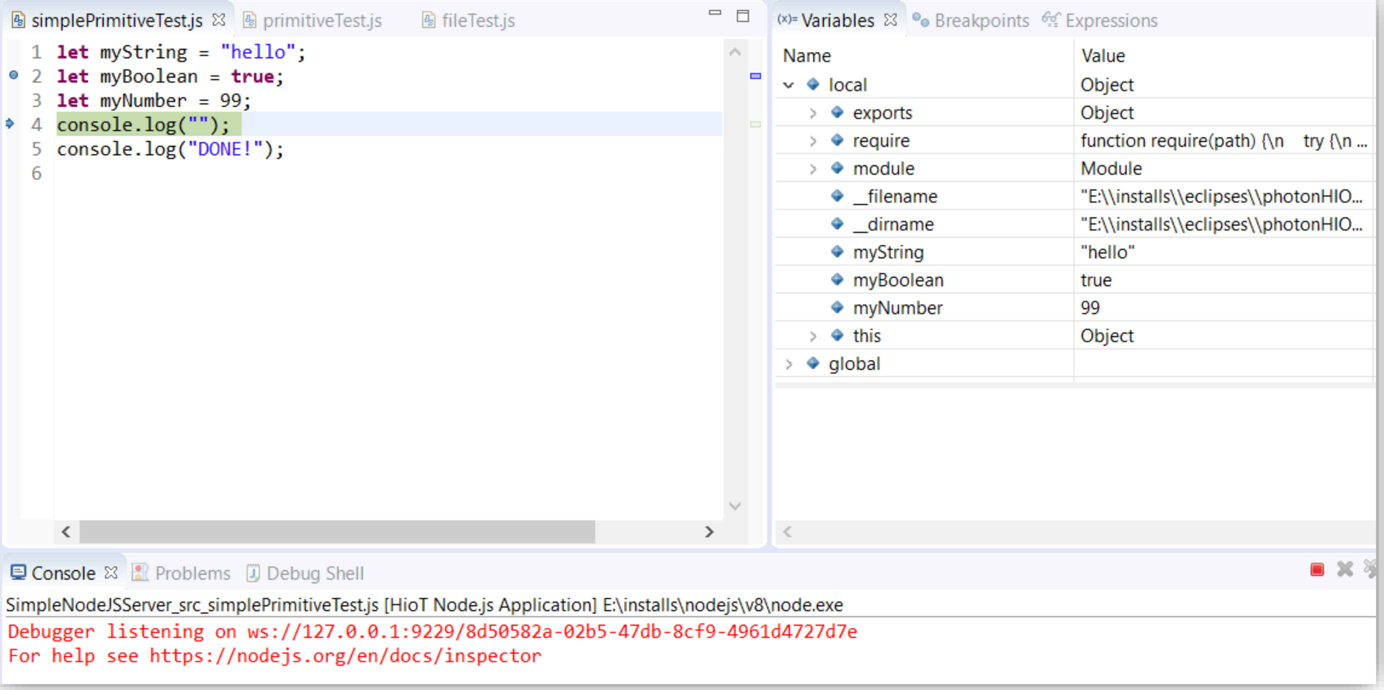
Request type: application/json

Response type: application/xml

Code	Description:
405	Invalid input

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



```javascript
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
```

HCL

Relationship™

BEYOND THE CONTRACT

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

▶ WATCH THE **FILM**