HCLSoftware

What's New in DevOps Model RealTime 12.0.3

updated for the DevOps 2024.09 release

Overview

▶ Model RealTime 12.0.3 is based on Eclipse 2023-06 (4.28)

▶ Two brandings of the product are available (HCL and IBM). There is no differences in functionality between them.

- Only difference is in the licensing mechanism and branding (e.g. documentation)

DevOps Model RealTime
Version: 12.0.3 (part of DevOps 2024.09)
Build: 12.0.3.v20240913_1209

(c) Copyright IBM Corporation 2004, 2016. All rights reserved.
(c) Copyright HCL Technologies Ltd. 2016, 2024. All rights reserved.
Visit https://shorturl.hcl-software.com/ibm-model-realtime-overview

2

HCL SOFTWARE

Eclipse 4.28 (2023.06)

▶ Compared to RSARTE/RTist 11.3, Model RealTime 12 includes new features and bug fixes from 4 quarterly Eclipse releases:

  ▪ 2022.09 (https://www.eclipse.org/eclipse/news/4.25/platform.php)

  ▪ 2022.12 (https://www.eclipse.org/eclipse/news/4.26/platform.php)

  ▪ 2023.03 (https://www.eclipse.org/eclipse/news/4.27/platform.php)

  ▪ 2023.06 (https://www.eclipse.org/eclipse/news/4.28/platform.php)

▶ For full information about all improvements and changes in these Eclipse releases see the links above

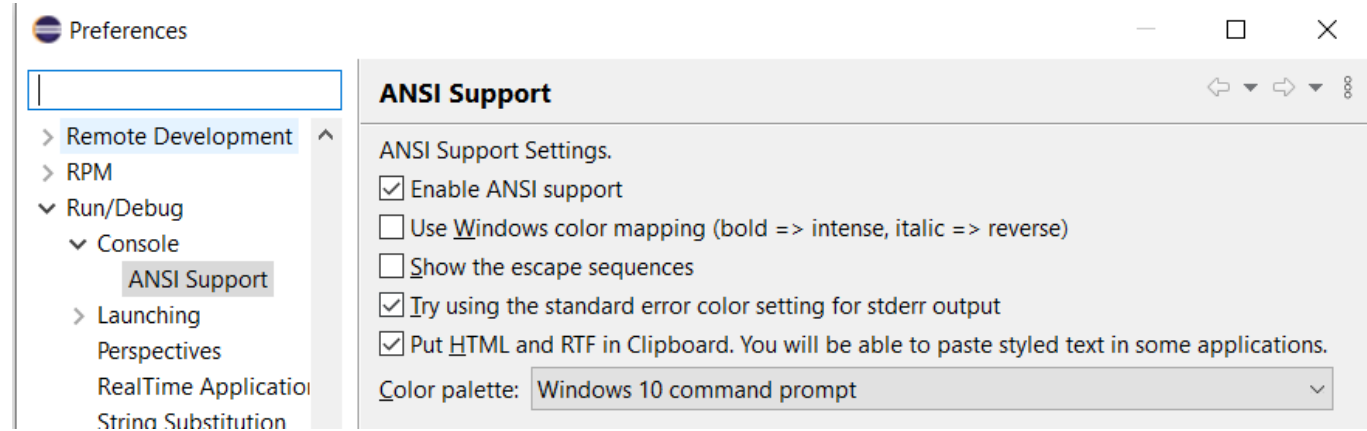 ▪ Some highlights are listed in the next few slides...

HCL SOFTWARE

Eclipse 4.28 (2023.06)

▸ New preference to enable word wrap automatically

- **General - Editors - Text Editors - Enable word wrap when opening an editor**

- Applies to all text editors including the Code editor

HCL SOFTWARE
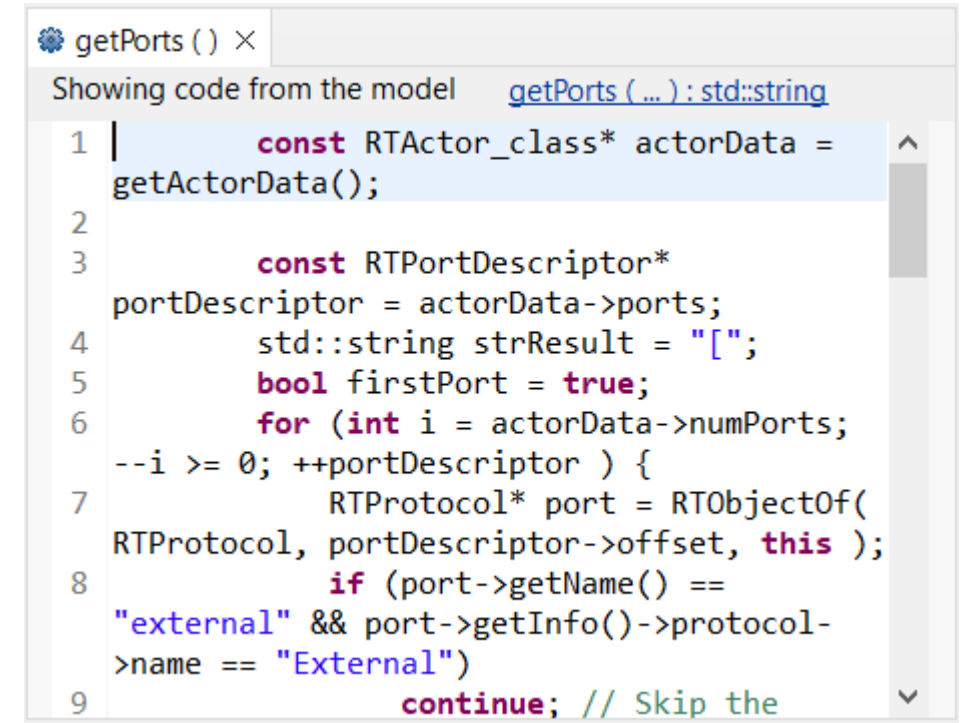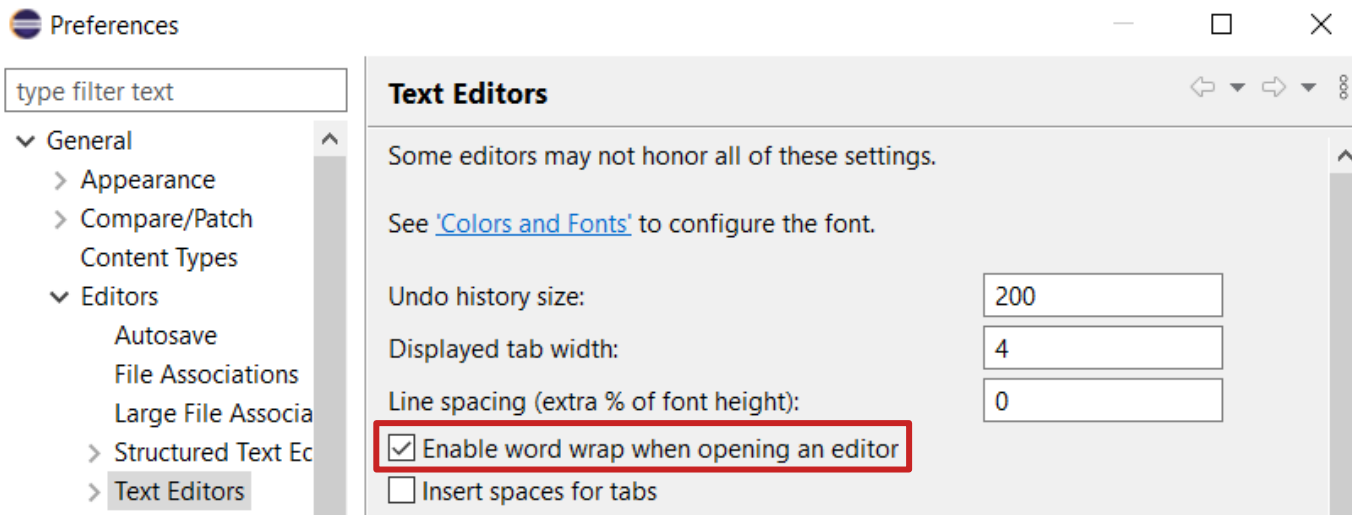
Eclipse 4.28 (2023.06)

▸ The Eclipse console now supports ANSI escape codes for producing styled output

- The SGR (Select Graphic Rendition) control sequences are supported

- Rendering can be controlled by preferences in **Run/Debug - Console - ANSI Support**

- You can use this to make application printouts easier to read

HCL SOFTWARE

Eclipse 4.28 (2023.06)

▶ Changes in the Open Resource dialog

- Now the string you type in this dialog is searched in all parts of file names

- It's no longer necessary to prefix the search string with an asterisk (*) to accomplish this

- You can use a leading angle bracket (>) to enforce the old behavior (can be useful in case of too many matches)

HCL SOFTWARE
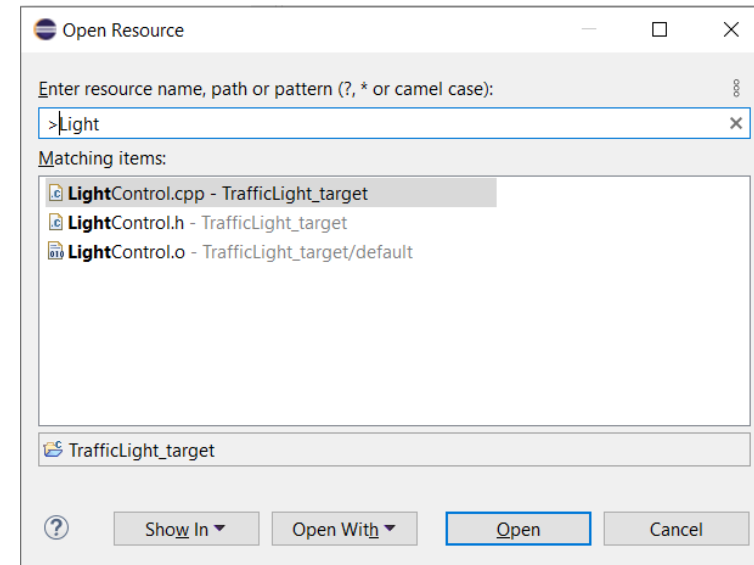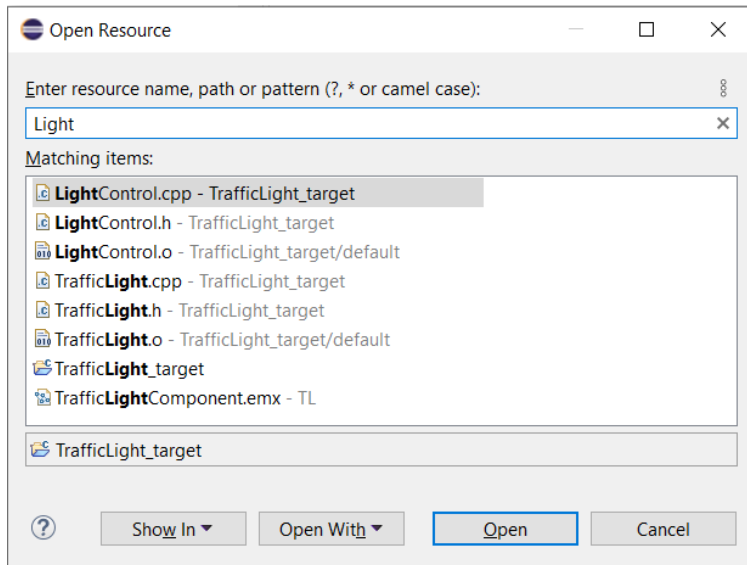
CDT 11.2 (included as part of Eclipse 2023.06)

▶ Several improvements in the C++ parser have been implemented

  ▪ Especially for supporting features of C++ 20 and C++ 23

▶ For more information about CDT improvements see
https://github.com/eclipse-cdt/cdt/blob/main/NewAndNoteworthy/CDT-11.0.md
https://github.com/eclipse-cdt/cdt/blob/main/NewAndNoteworthy/CDT-11.1.md
https://github.com/eclipse-cdt/cdt/blob/main/NewAndNoteworthy/CDT-11.2.md

**HCL SOFTWARE**

Newer EGit Version in the EGit Integration

▸ The EGit integration in Model RealTime has upgraded EGit from 6.2 to 6.6

- This is the recommended and latest version for Eclipse 2023.06

▸ This upgrade provides several new features and bug fixes

- For detailed information about the changes see
  https://wiki.eclipse.org/EGit/New_and_Noteworthy/6.3
  https://wiki.eclipse.org/EGit/New_and_Noteworthy/6.4
  https://wiki.eclipse.org/EGit/New_and_Noteworthy/6.5
  https://wiki.eclipse.org/EGit/New_and_Noteworthy/6.6

▸ Note: In 12.0 the EGit Integration was experimental due to potentially failing merge and rebase operations due to false conflicts. This problem is now solved in 12.0.1 and the EGit integration is no longer experimental.

HCL SOFTWARE

New Location for Info Centers

▶ As part of the product rename, the online documentation was moved to a new location

▶ There are two versions, depending on branding

- HCL: https://model-realtime.hcldoc.com/help/index.jsp

- IBM: https://www.ibm.com/docs/dmrt/12.0

HCL SOFTWARE

Easier to Remove Return Type of Operation

▶ You can now remove the return type of an operation by simply clearing the "Return Type" field in the Properties view
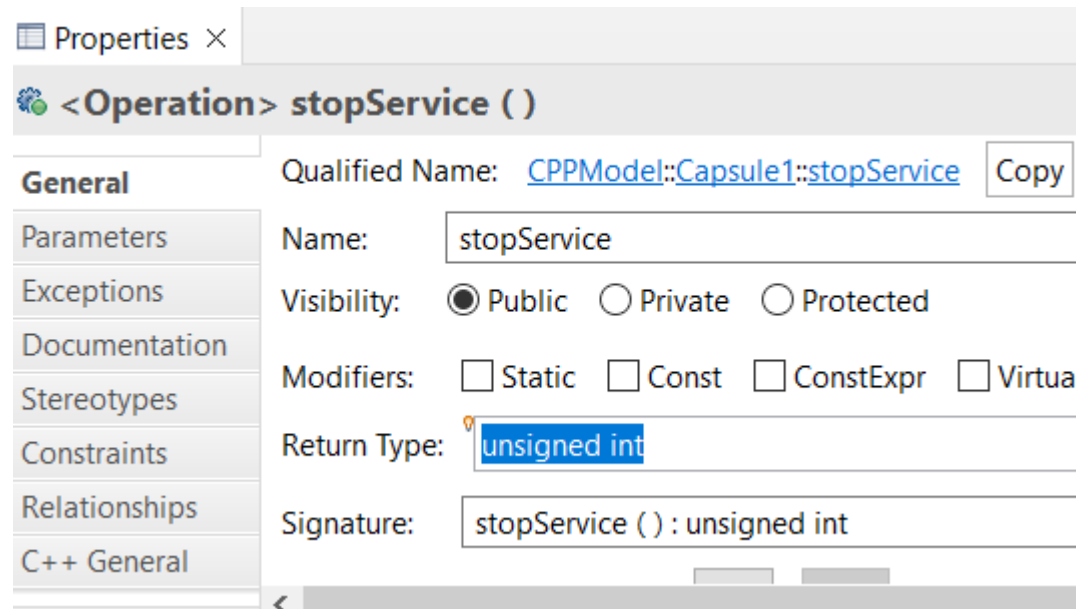
- Previously, this way of clearing the return type kept the return parameter in the model, but with an unspecified type, which caused a warning during code generation

HCL SOFTWARE

Removal of Body for Pure Virtual Operations

▶ If an operation with a body is made pure virtual you now have the option to either keep or remove its body

- This is controlled by a new preference **RealTime Development - Properties View - Remove body when making an operation pure virtual**

▶ This also works for multiple selected operations which makes it easy to convert a class with a "trial implementation" to become an abstract class

Copyright © 2024 HCL Technologies Limited | www.hcltechsw.com
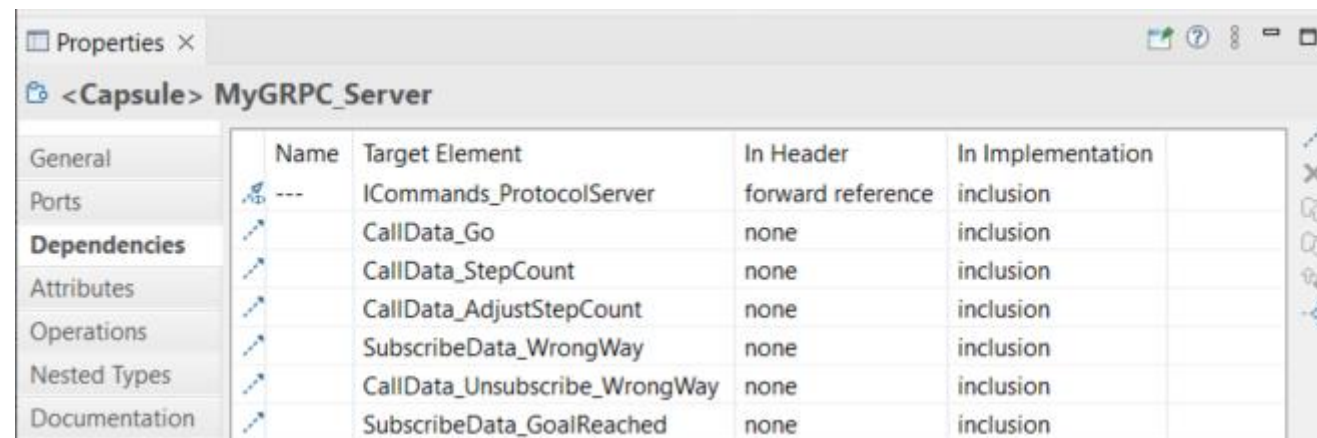
HCL SOFTWARE
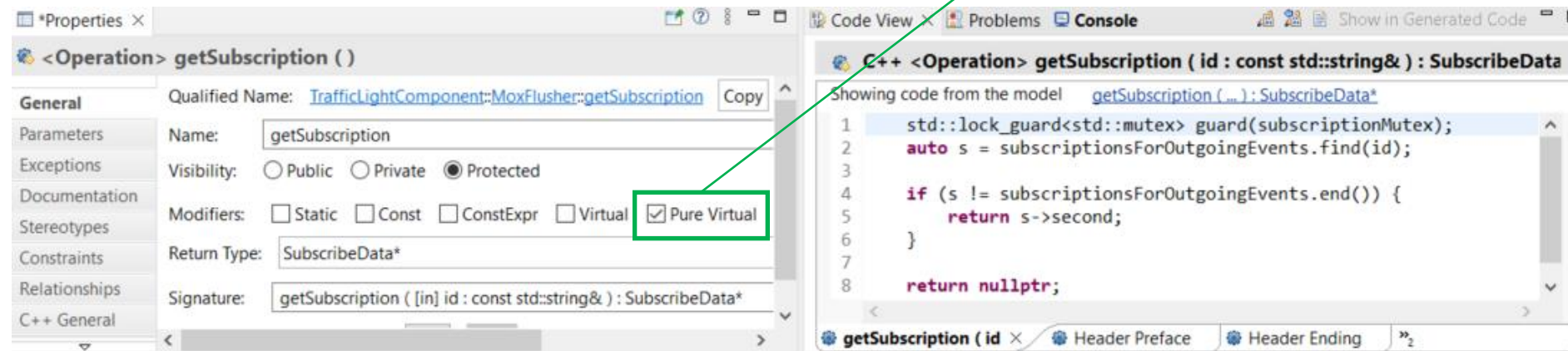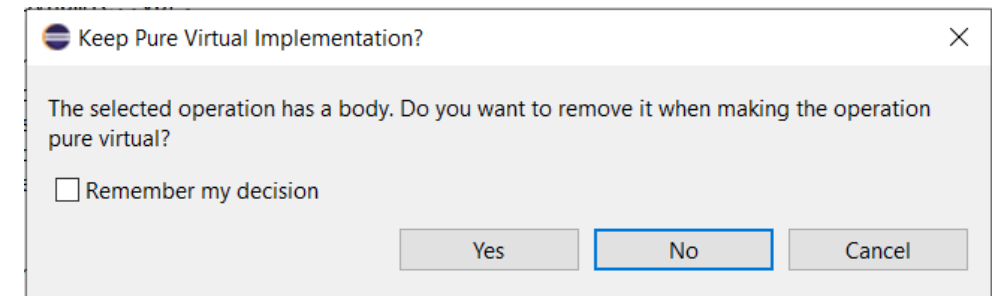
Capsule Destructors

▸ A new command makes it easier to create a destructor for a capsule

- Capsule destructors were already supported by the Model Compiler but the lack of a UI command made it easy to miss this feature.

- The previous workaround to instead create a constructor and then prefix it with a "~" is no longer needed.

HCL SOFTWARE
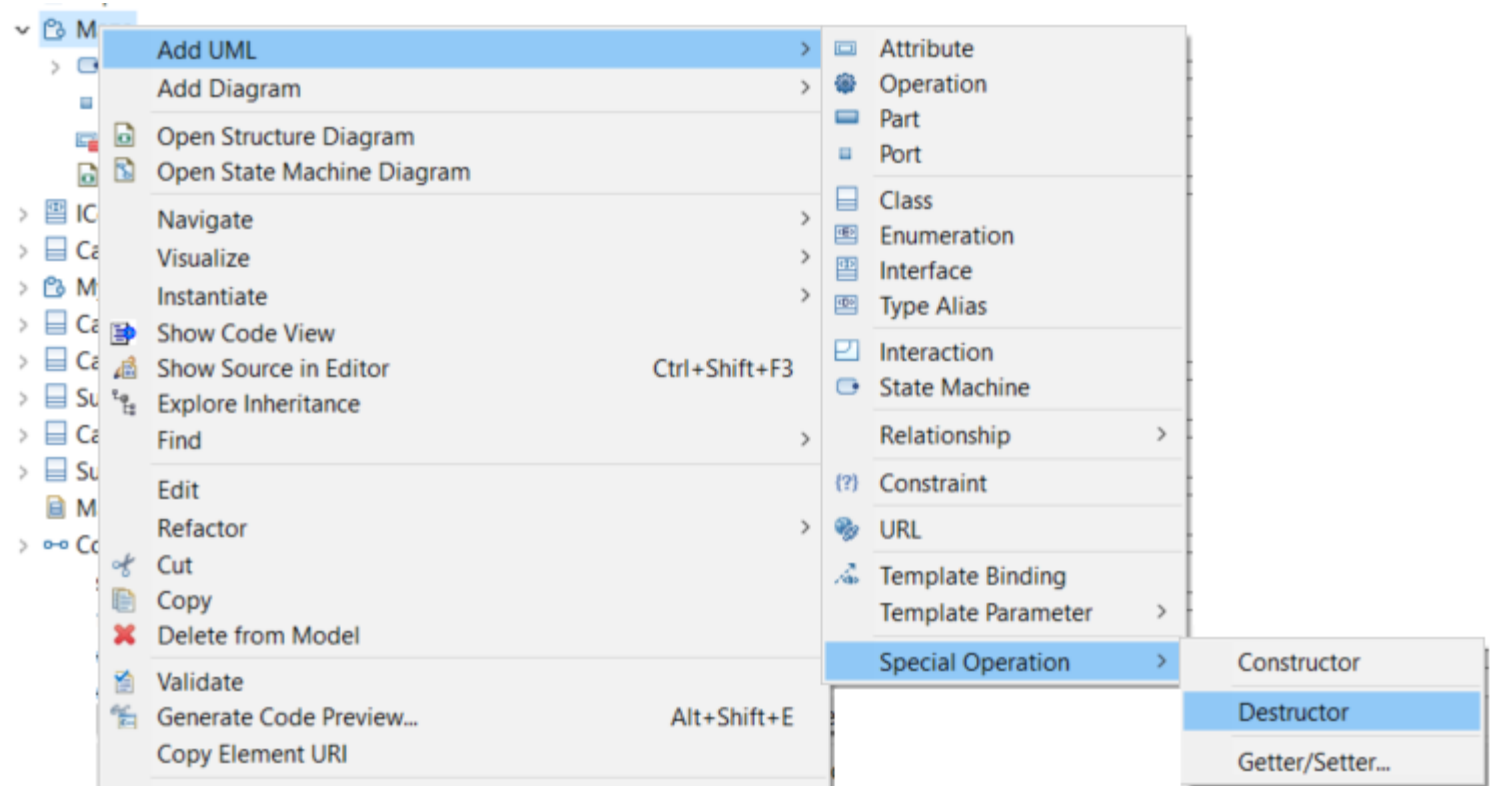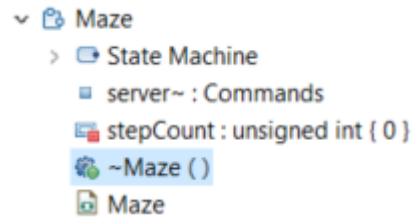
Improved Dependencies Property Page

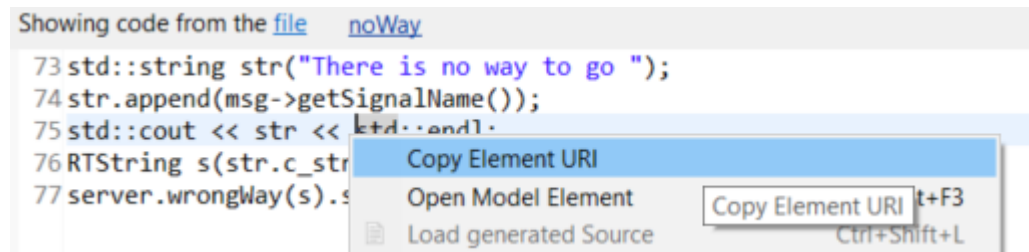▸ The Dependencies property page is now more user-friendly

   ▪ Better default widths of table columns makes it easier to see the "Target Element" of a dependency

   ▪ More compact names for the two last column headers to not waste screen estate

   ▪ Better use of available space when the page is resized

▸ The new behavior avoids most need of manual resizing of columns in this property page

HCL SOFTWARE

Improved Commands for Navigation by URI

▶ The command "Navigate by URI" now supports navigating to a specific line of a code snippet of a certain model element

▶ You can obtain such a URI by means of the "Copy Element URI" command which now is available also in the context menu of the Code Editor and Code View:

   ▪ Previously this command was only available in the context menu of the Project Explorer

▶ Example:

```
platform:/resource/grpc-server/CPPModel.emx#_nAGFwNIxEe6fffW4TiqQOQ|Effect:3
```

line number

code snippet name

HCL SOFTWARE

Better Display of Non-Replicated Ports while Debugging

▶ Ports with single multiplicity are now shown with a single node in the Debug view

- Leads to a cleaner and less cluttered Debug view, that is easier to work with

```
✓ 📇 / [Top : 0x12f5d356e00]
  ✓ 🖵 gRPC_Server [1 instance]
    ✓ 📇 gRPC_Server:0 [MyGRPC_Server : 0x12f5d3a5b30]
      ✓ 🖵 Ports
        ✓ 🖵 commands
          ▪ commands:0
        ✓ 🖵 external
          ▪ external:0
  ✓ 🖵 maze [1 instance]
    ✓ 📇 maze:0 [Maze : 0x12f5d356690]
      ✓ 🖵 Ports
        ✓ 🖵 server
          ▪ server:0
```

```
✓ 📇 / [Top : 0x2e458b276b0]
  ✓ 🖵 gRPC_Server [1 instance]
    ✓ 📇 gRPC_Server:0 [MyGRPC_Server : 0x2e458b78a20]
      ✓ 🖵 Ports
        ▪ commands
        ▪ external
  ✓ 🖵 maze [1 instance]
    ✓ 📇 maze:0 [Maze : 0x2e458b4a5c0]
      ✓ 🖵 Ports
        ▪ server
```

before   *now*

HCL SOFTWARE

Launching the Model Debugger on a TC with Custom JavaScript

▶ The Model Debugger now evaluates the TC that is used for launching a debug session

▶ This makes it possible to also debug TCs which contain JavaScript statements "beyond simple assignments", such as variable declarations

- For these TCs only the Code tab is shown

```
20    let tc = TCF.define(TCF.CPP_TRANSFORM);
21    // Where gRPC sources are located
22 ×» let grpcSourceLocation = 'C:/github/grpc';
23    // Where gRPC tools and libraries are installed
24 ×» JavaScript statement beyond simple assignment
25    tc.sources = [
26        'platform:/resource/grpc-server/CPPModel.emx#_yhYUcNItEe6fffW4TiqQOQ',
27    ];
```
Code

- Previously it was necessary to use a workaround where an extra "dummy" TC was created, for specifying the top capsule to debug. This is no longer required.

HCL SOFTWARE

Isolated Trace Sessions

▶ Previously multiple trace sessions were not always properly isolated from each other

- Settings from one trace session could influence another trace session

▶ This has now been fixed

- A consolidated set of trace settings are now computed from all active trace sessions by a "trace manager"

- The debugged target application uses this "union of trace settings" and the trace manager makes sure to send each trace event from the application to the trace session(s) that should receive it, according to their settings

HCL SOFTWARE
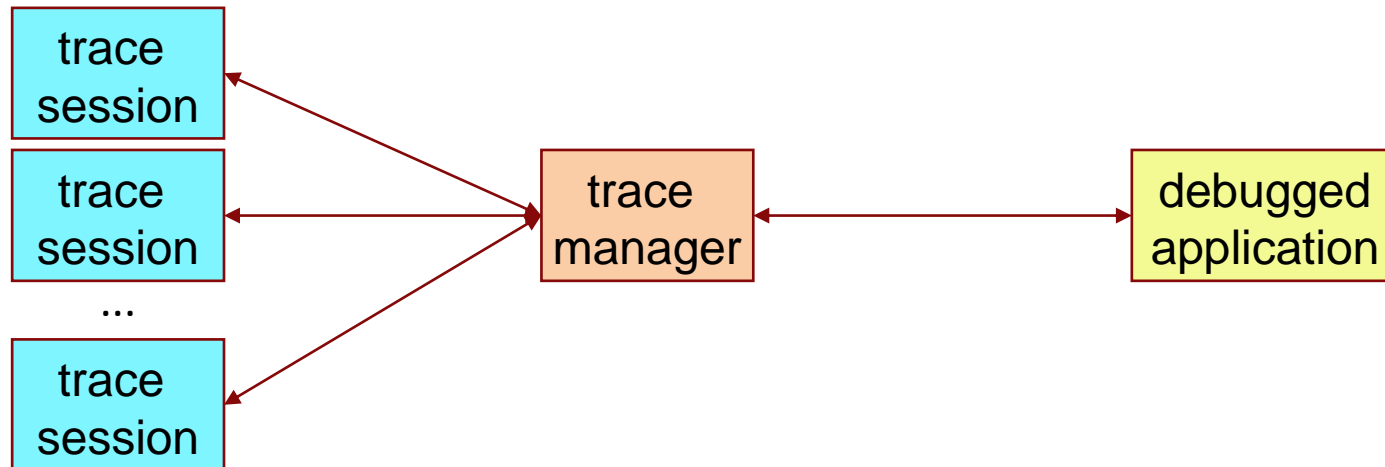
Sorting of TC Properties

▸ TC properties are now sorted alphabetically when saving a TC file

- The "sources", "parents" and "prerequisites" properties will come first, followed by other properties sorted by name

- Makes it easier to find a TC property in the Code tab of the TC editor

- Reduces the risk for merge conflicts when multiple people edit the TC file

- Comments that precede a TC property are moved with that property when sorting the properties

▸ Note 1: Sorting does not happen for TC files that contain JavaScript beyond simple assignments of properties. This is because sorting in such a file could affect the meaning of the TC file.

▸ Note 2: If you have TC properties with values that reference other TC properties, sorting will happen and may affect the meaning of the TC file. Consider turning sorting off in this case.

▸ Sorting can be turned off by means of a new preference
RealTime Development - Transformation Configuration Editor - Auto sort properties on save

```
server.tcjs ×

[C++ Executable] server.tcjs
 1    let tc = TCF.define(TCF.CPP_TRANSFORM);
 2
 3    tc.sources = [
 4       'platform:/resource/grpc-server/CPPModel.emx#_yhYUcNItEe6ff
 5    ];
 6    tc.compilationMakeArguments = '';
 7    tc.compilationMakeType = MakeType.MS_nmake;
 8    tc.compileArguments = '$(DEBUG_TAG)';
 9    tc.createTargetProject = true;
10    tc.inclusionPaths = [
11       grpcSourceLocation + '/include',
12       grpcSourceLocation + '/third_party/abseil-cpp',
13       grpcInstallLocation + '/include',
14       '../../grpc-client/build',
15    ];
16    tc.linkArguments = '/DEBUG';
17    tc.targetConfiguration = 'WinT.x64-VisualC++-17.0';
18    tc.targetProject = 'grpc-server_target';
19    tc.targetServicesLibrary = 'C:/TargetRTSUnzipped/TargetRTS';
20    tc.threads = [
21    {
22       name: 'MainThread',
```

HCL SOFTWARE

Constructor Initializer Ordering

▸ The initializer list for a constructor that is automatically generated is now ordered according to the order of declaration of the initialized members

- ▪ The base class is always initialized first (for a capsule the base class is RTActor, or another capsule class)


- ▪ Constructor initializers are ordered like this both for capsules and classes

▸ This ordering avoids compilation warnings from certain compilers (and errors, if e.g. **-Werror=reorder** is specified for gcc)

HCL SOFTWARE

CDATA Improvements

▶ Storing models with CDATA makes code snippets and documentation easier to read in the XML storage format

▪ It's recommended for example if you do code reviews with GitHub pull requests

▶ The handling of line breaks in code snippets and documentation has been improved

▪ The preference now shows the current platform and the default newline character(s) that will be used (CR+LF on Windows and LF on Linux/MacOs)
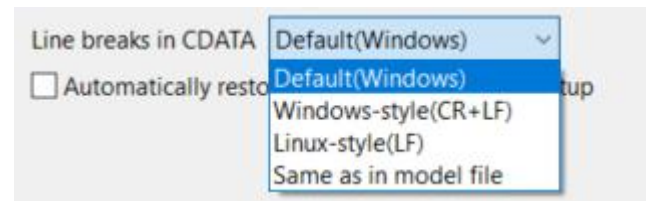
▪ If the option "Same as in model file" is used, Model RealTime tries to determine which kind of line breaks to use by reading the model file to be saved. If this fails (for example in case a new file is saved), it will instead use what is specified by the preference **General - Workspace - New text file line delimiter**.

HCL SOFTWARE

Support for Art Files (Code RealTime Integration)

▶ You can now use Art files developed with Code RealTime in your model

- Makes it possible to start using Code RealTime for developing new parts of an existing Model RealTime application

- Makes it possible to share libraries between Code RealTime and Model RealTime applications

▶ This is currently an EXPERIMENTAL feature and a subset of what an Art file can contain is supported:

- Protocols and events (without or with data of primitive types) can be used just as if they were defined in the model

- User-defined C++ types with type descriptors defined in an Art file can be used from code snippets

▶ Full support for the Art language is planned for a future release
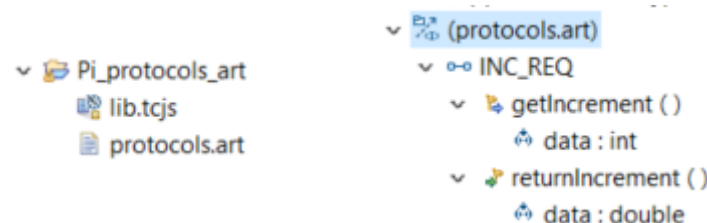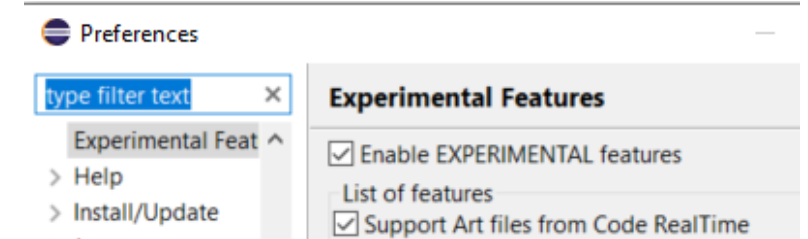
HCL SOFTWARE

Patch Files for Tracking TargetRTS Changes

▸ The TargetRTS now contains patch files for recent changes

- Located in the installation at `rsa_rt/C++/TargetRTS_changelog`

- Simplifies adoption of TargetRTS changes when uplifting to a new version of Model RealTime

- A script `createPath.sh` is also provided that you can use for creating your own patch files, in case you prefer to instead apply your own TargetRTS changes on top of the standard version

▸ For more information and documentation about TargetRTS changes see [this new website](#).

HCL SOFTWARE

Building without "rtperl"

▶ Generated makefiles now allow building without using the "rtperl" utility

▶ You can override the "rtperl" variable using the preference **Run/Debug - String Substitution**
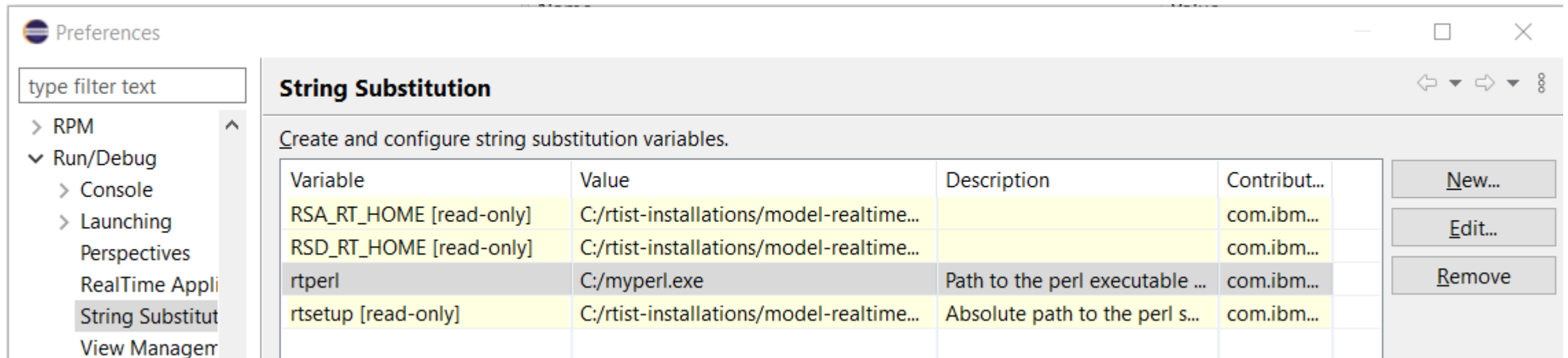
▶ If this variable is not set or has an empty value, the build will use "rtperl" from the installation (for Windows and Linux) or "perl" from the path (for MacOs)

▪ Note: Workspaces created in older versions of Model RealTime may have this variable set to an incorrect path. You should then either remove or update the variable.

HCL SOFTWARE

Improved JSON Support in the TargetRTS

▶ The JSON Encoder was improved to support encoding of more types of data. For example, it now supports encoding of pointers into strings.

```
{"address" : "0xe6ef6fed80"}
```

▶ A new JSON Decoder is now available. It can decode a JSON string produced by the JSON Encoder into a data object. See RTJsonDecoding.

▶ A new JSON Parser is available. It's a general-purpose JSON parser that makes it easier to work with JSON in a realtime application. See RTJsonParser.
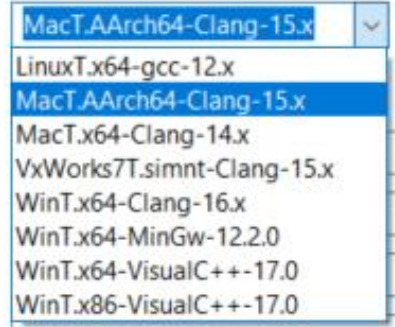
```cpp
RTJsonParser parser;
RTJsonResult result;
bool ok = parser.parseJsonString(result, R"({"field1" : "string", "arr" : [1,true,3.14]})");

std::cout << result["field1"].get_string() << std::endl; // "string"
if (result["arr"].ok()) { // Check if the "arr" key is present
    std::cout << result["arr"].get_size() << std::endl; // 3
ASSERT(result["arr"][2] == 3.14);
ASSERT(result["arr"][1].get_type() == RTJsonResult::RTJSON_BOOL);
```

HCL SOFTWARE

New Target Configuration for MacOS with ARM

▶ A new target configuration for a TargetRTS prebuilt for MacOs with ARM processor ("Apple Silicon") is now available

  ▪ Makes it easier to get started building applications that target this platform

▶ Note that running Model RealTime itself on MacOS with ARM is possible, but has not been fully tested yet. Hence the support for this platform is currently EXPERIMENTAL.

TargetRTS configuration: MacT.AArch64-Clang-15.x

Make type:

LinuxT.x64-gcc-12.x
MacT.AArch64-Clang-15.x
MacT.x64-Clang-14.x

Compile arguments:

VxWorks7T.simnt-Clang-15.x
WinT.x64-Clang-16.x

Compile command:

WinT.x64-MinGw-12.2.0
WinT.x64-VisualC++-17.0

Executable name:

WinT.x86-VisualC++-17.0

HCL SOFTWARE
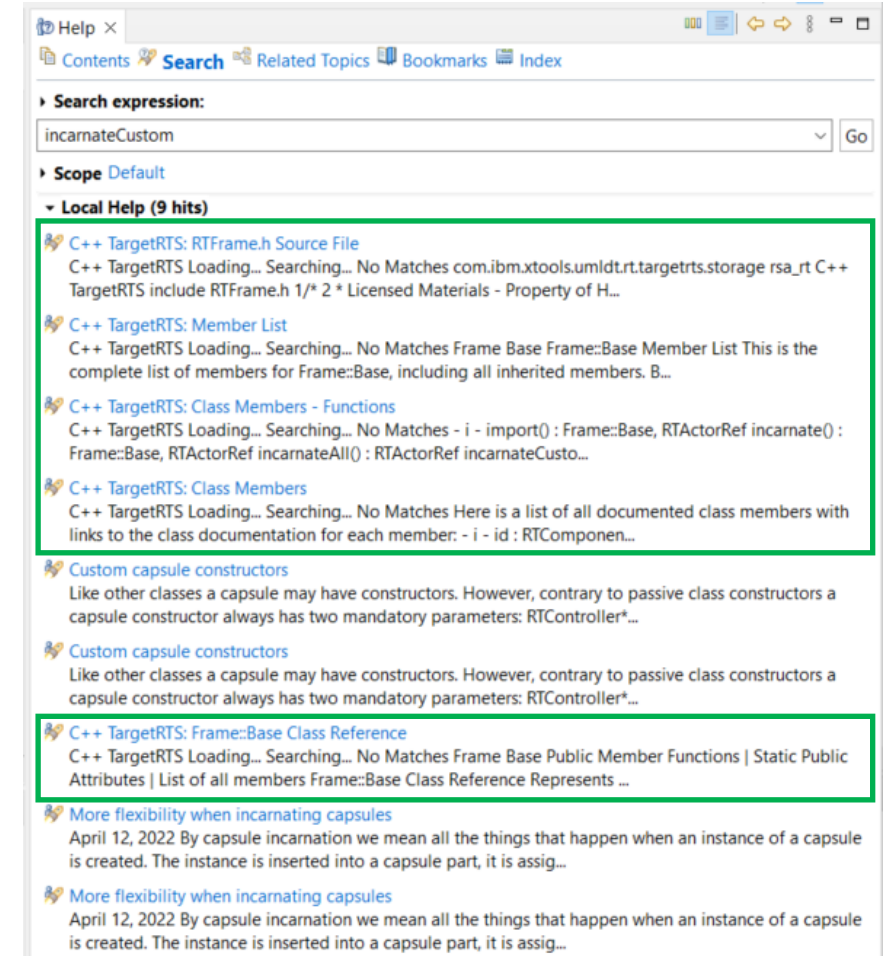
New Configuration Macros for Configuring the Free List

▸ The Free List is the pool where blocks of unused messages are stored while they are not taking place in any message exchange

- Previously the size of the Free List was hard-coded in the TargetRTS implementation

- Now there are 3 new macros for configuring the Free List

  - **MIN_FREE_LIST_SIZE** - minimum size of the Free List

  - **MAX_FREE_LIST_SIZE** - maximum size of the Free List

  - **RTMESSAGE_BLOCK_SIZE** - number of messages in each message block of the Free List

HCL SOFTWARE

Improved TargetRTS API Documentation

▸ The Doxygen generated documentation now covers all TargetRTS APIs

- This includes also "internal" APIs provided by header files
  under `TargetRTS/src/include`

- For conditional parts that depend on TargetRTS configuration macros,
  the defaults as specified in `RTConfig.h` have been used. This ensures
  that the documentation is accurate for the default behavior of
 the TargetRTS.

HCL SOFTWARE

Updated Implementation of TCP Server Library without Poco Usage

▸ The **lib-tcp-server** library no longer depends on the Poco library

- ▪ Now it only depends on the TargetRTS which makes it easier to use the library for different target configurations

▸ Prebuilt libraries for lib-tcp-server are no longer included in the installation

- ▪ To use the library, just add the TC "tcpServerLib" as a prerequisite to your TC, so it will be built as part of building your application

- ▪ To build the library with custom settings, use instead the TC "tcpServerLib_custom" (with your custom changes)

HCL SOFTWARE
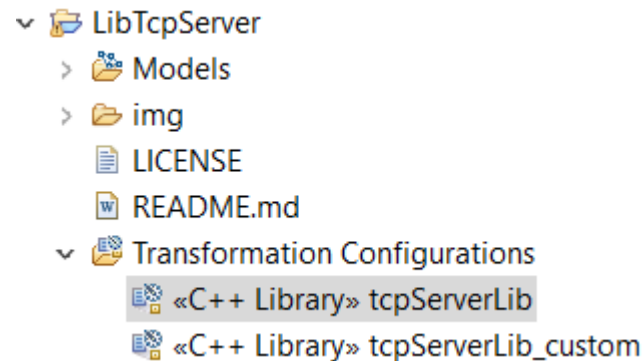
New Sample qt-trafficlight

▶ A new sample is available on GitHub: https://github.com/HCL-TECH-SOFTWARE/qt-traffic-light

▶ Shows how an application created with Model RealTime can be integrated with a user-interface implemented with Qt

   ▪ Qt's concepts of **signals** and **slots** make it easy to update the UI from the realtime application in a thread-safe way

▶ For more information, see this post

HCL SOFTWARE
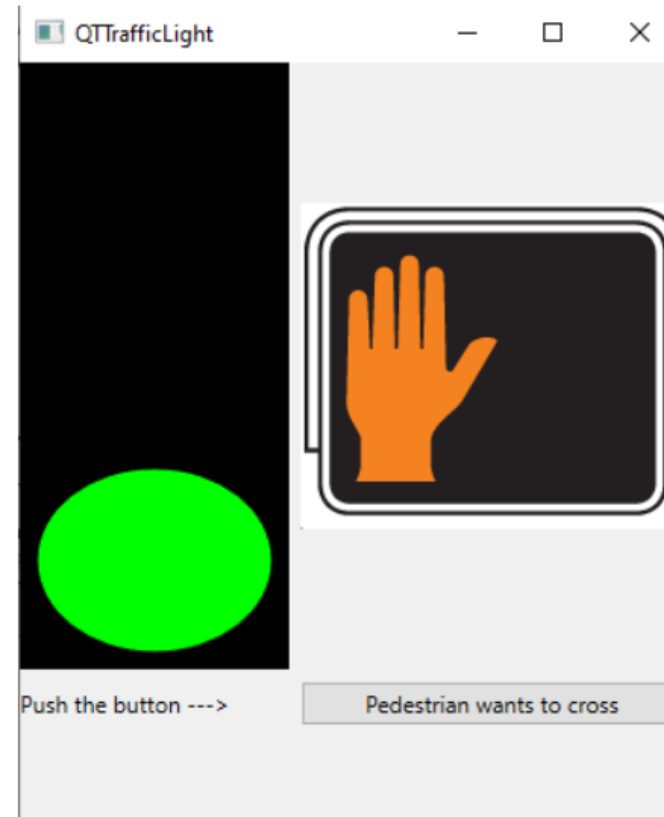
New Sample about Dependency Injection

▸ A new sample is available on GitHub: https://github.com/HCL-TECH-SOFTWARE/dependency-injection

▸ Shows how dependency injection can be combined with Build Variants for building different variants of an application with different implementations of certain capsules

▸ For more information, see this post

HCL SOFTWARE

New gRPC Library and Sample

▸ A new library for using gRPC with Model RealTime applications is available on GitHub:
https://github.com/HCL-TECH-SOFTWARE/lib-grpc-server

 ▪ Also contains client and server sample applications showing how to use the library

▸ With this library a realtime application can act as a gRPC server to let other applications communicate with it by means of gRPC

 ▪ It's similar to the TCP server library, but with some benefits. For example, gRPC uses a binary protocol which is more efficient and it's more "type safe" in C++ than sending JSON strings over TCP. Also, gRPC has more features and flexibility than TCP-based communication, for example message streaming.


```
C:\github\lib-grpc-server\grpc-client\build\Debug>maze_client.exe
Walk the maze. Commands:
east, west, north, south : Take a step in a direction
steps : Report number of steps taken
adjust : Adjust step count by specified number
subscribe : Subscribe to be notified when wrong way taken
unsubscribe : Unsubscribe to be notified when wrong way taken
exit : Exit
>east
>_
```

HCL SOFTWARE
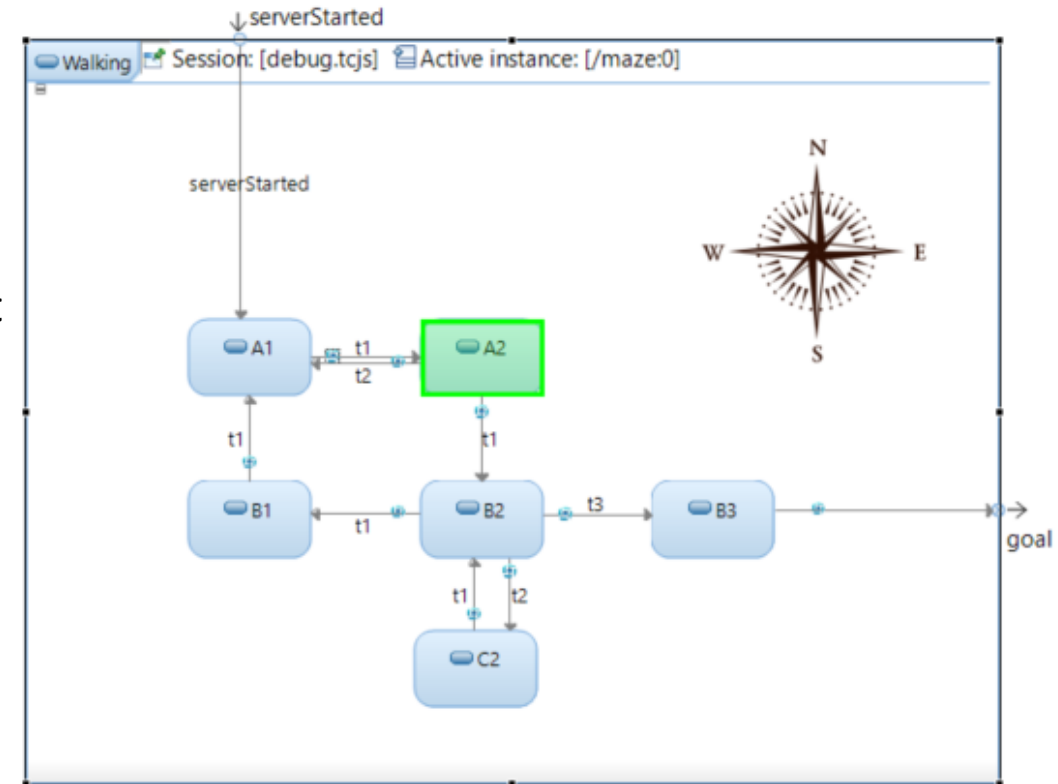
Script for Downgrading Model Files

▸ Model RealTime 12 uses a newer version of GMF that causes model files created with this version to not be fully compatible with older versions

- Diagrams created in Model RealTime 12 may not open correctly in older versions of the tool

▸ A bash script is available on the Info Center for downgrading model files so they can be opened in Model RealTime 11.x and older

```
$ ./versionDowngrade.sh --help

Usage: versionDowngrade.sh [ <folder> ]
The script will search for all model files under <folder> and replace gmf versio
n 1.0.3 with 1.0.2
If argument is not specified, the script will search in current folder
```

- Conversion in the other direction happens automatically when you save an old model with Model RealTime 12

HCL SOFTWARE

HCLSoftware