What’s New in HCL RTist 10.3

updated for sprint 2019.07

PRODUCTS & PLATFORMS

HCL

Overview

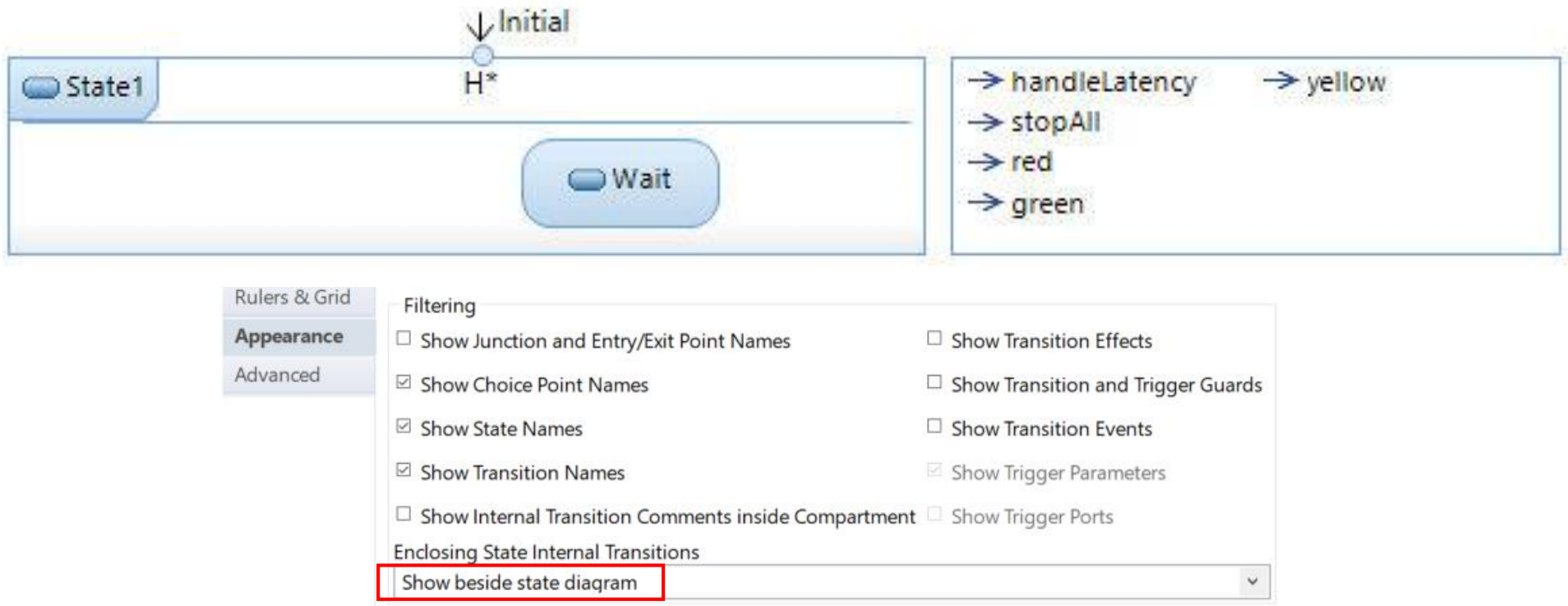
- RTist 10.3 is based on Eclipse Photon (4.8)
- HCL RTist is functionally equivalent and 100% compatible with IBM RSARTE.

PRODUCTS & PLATFORMS

Internal Transitions

- Internal transitions for the enclosing composite state can now be shown in a compartment beside the state chart diagram of the composite state.

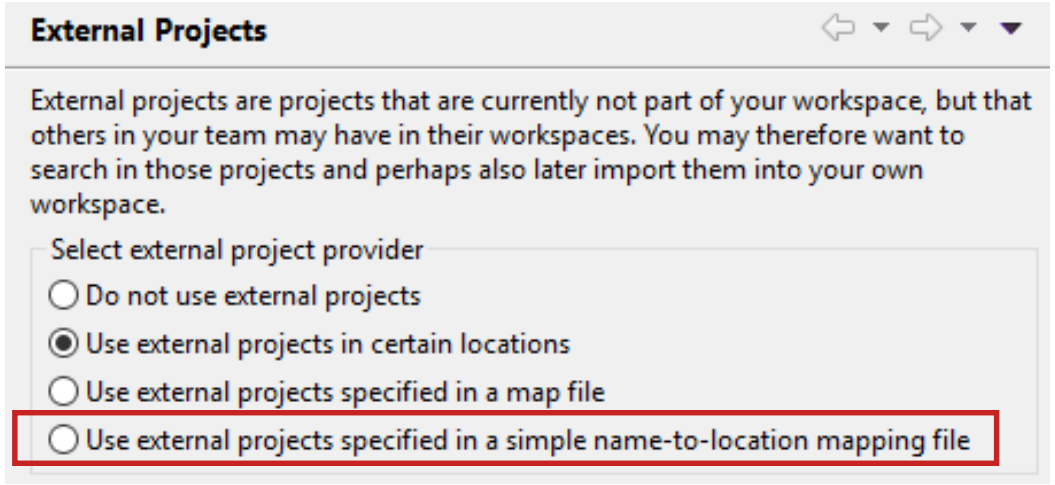
- The transitions can be shown in columns to get a compact presentation

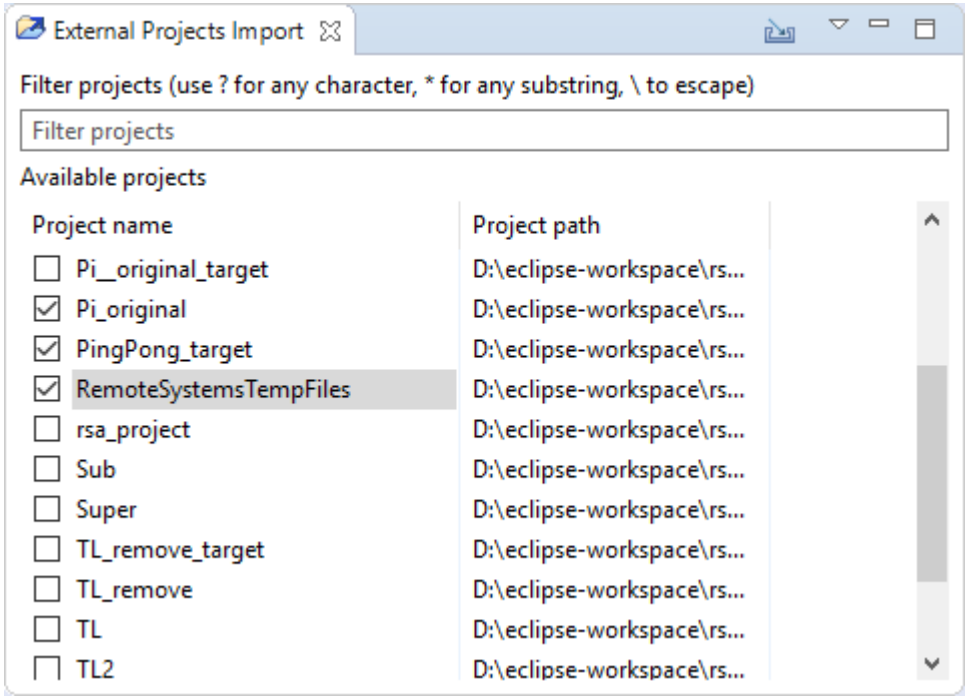


PRODUCTS & PLATFORMS

External Projects Import

- A new view makes it easier to import external projects into the workspace
- Define where to look for external projects and then import found projects easily into the workspace
- Now possible to specify the location of external projects with a file on the same format as is supported by the model compiler

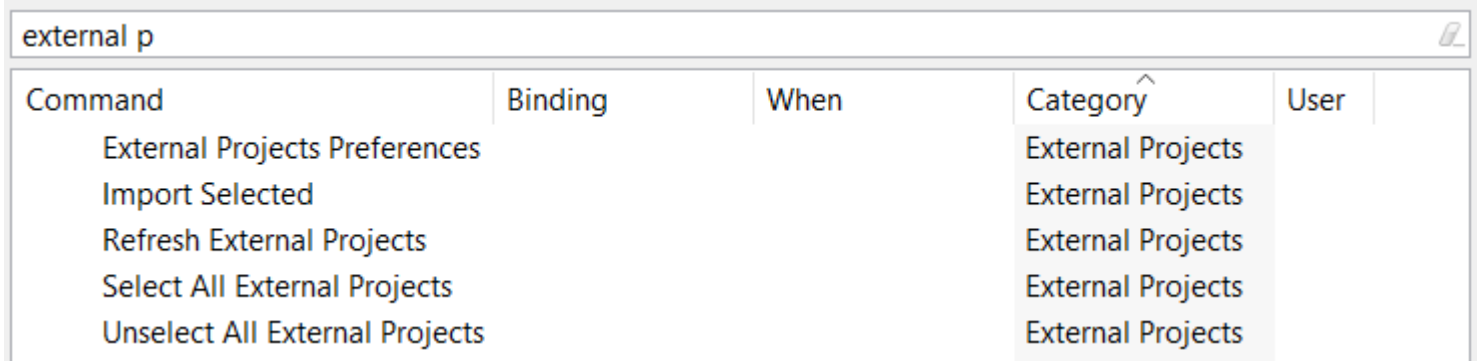




PRODUCTS & PLATFORMS

External Projects Import

- Commands are available for working with the External Projects Import view using keybindings
  - Define your desired keybindings for the commands you want to use (there are no default keybindings for them)

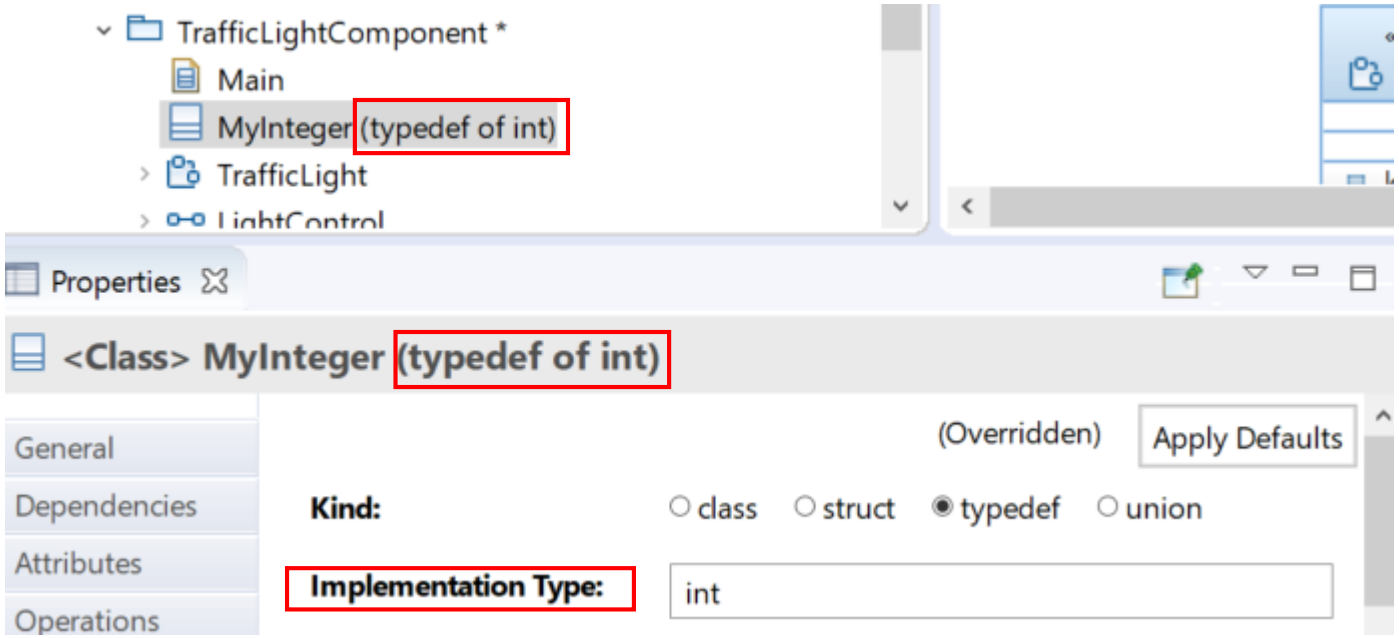


- The External Projects Import view can now be filtered using project paths

- Often the view is automatically updated when external projects change (can for example be caused by switching branch in Git). If not, you can use the Refresh External Projects to update the list of external projects.

Typedef Types

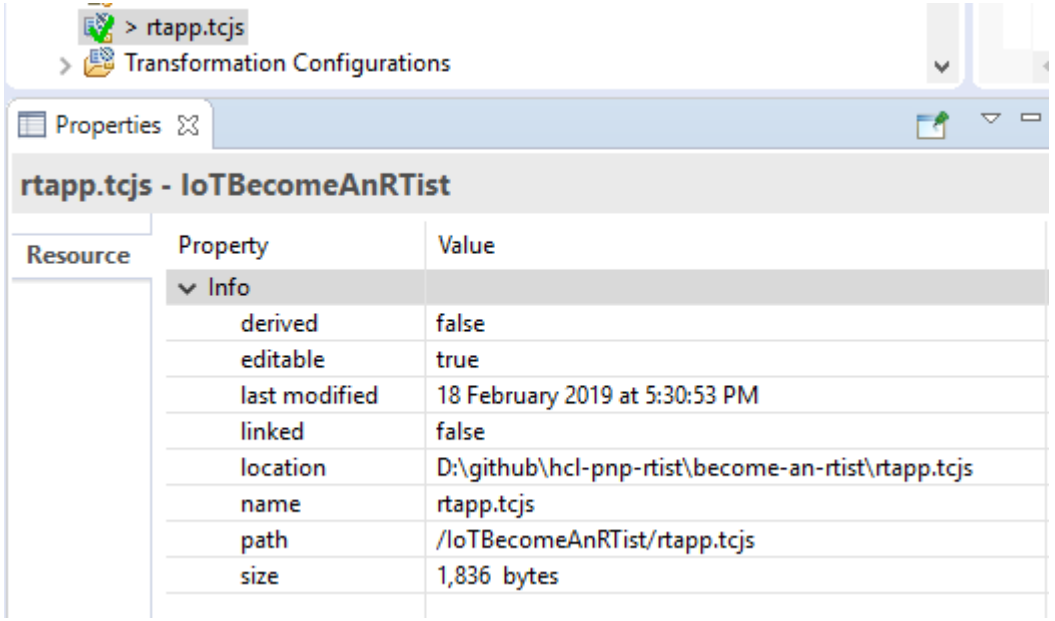
- The typedef implementation type is now shown in the Project Explorer, Properties view etc.
- The “Implementation Type” property is now accessible without scrolling in the Properties view



PRODUCTS & PLATFORMS

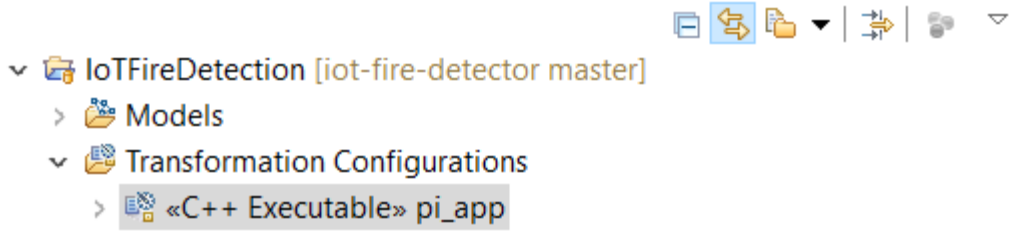
Properties View

- When a TC is selected information about the .tcjs file is now shown in the Properties view



Transformation Configuration Editor

- Now supports the “Link with Editor” button of the Project Explorer



- Improved the command “Navigate to Inherited Value” so that it works in more situations

PRODUCTS & PLATFORMS

Access to Top-Level TC Properties from Prerequisite TCs

- A prerequisite TC can now access properties from the built TC (a.k.a. the “top-level TC”)
- This is useful when creating libraries to be built in many different contexts

For example:

```
let tc = TCF.define(TCF.CPP_TRANSFORM);

let topTC = TCF.getRootTransformationConfiguration();

tc.targetConfiguration = topTC.targetConfiguration || 'WinT.x64-VisualC++-15.0';
```

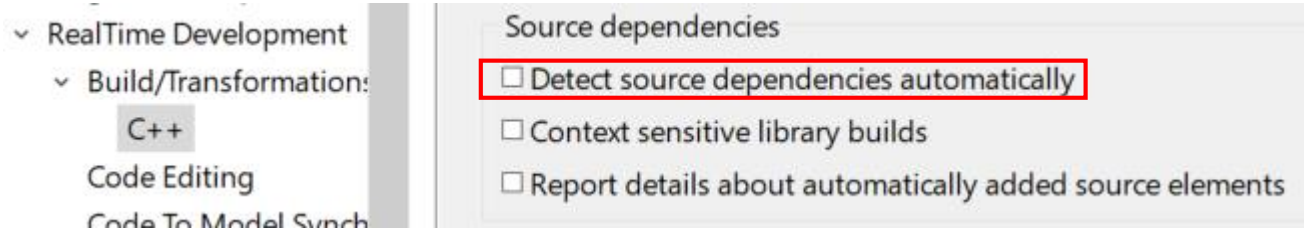

PRODUCTS & PLATFORMS

Enable Support for File Artifacts

- The TC property “Enable Support for file artifacts” has been removed
- File artifacts will be translated to C++ if they are present among the source elements (i.e. they are now handled in the same way as all other kinds of model elements)

Model Compiler

- Support for automatic detection of source dependencies
  - The Sources list of the built TC can be dynamically extended to include all elements that are referenced (directly or indirectly) from built elements

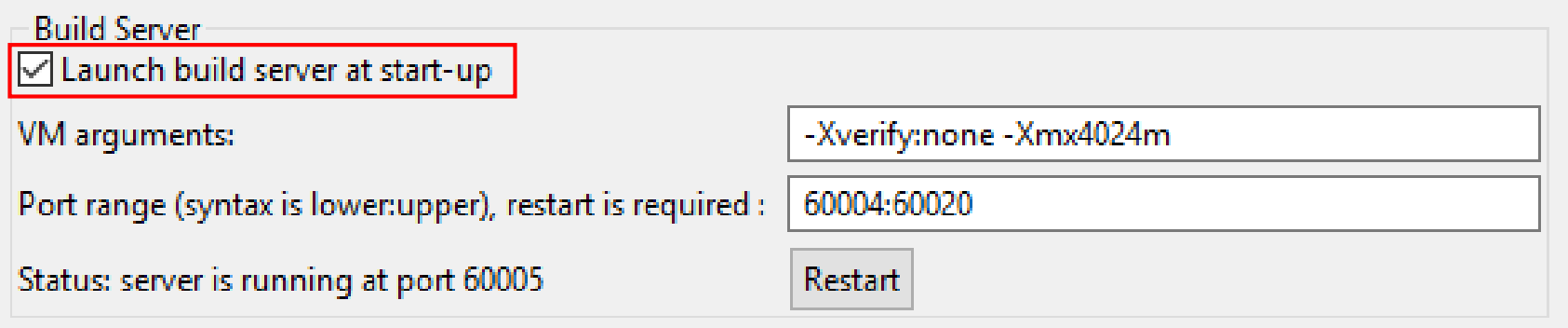


- The location of the model compiler is now shown in the Preference page
  - Makes it easier to run it from the command line

PRODUCTS & PLATFORMS

Model Compiler

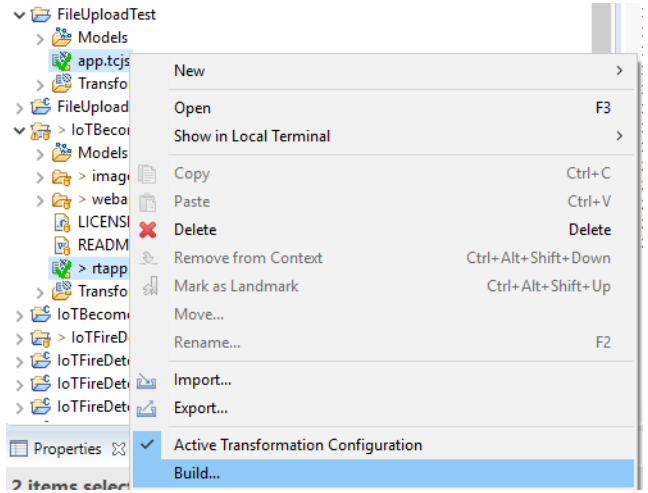
- A new script in <install dir>/rsa_rt/tools/ makes it easier to launch the model compiler
  - No longer necessary to find the location of the containing plugin
- Running the model compiler with no command-line arguments will now show its help
- Make is now always run as part of building (it will detect whether something needs to be built or not)
- The build server (i.e. the a model compiler running in server mode) can now be disabled, so it isn't automatically launched at start-up. A new preference controls this.

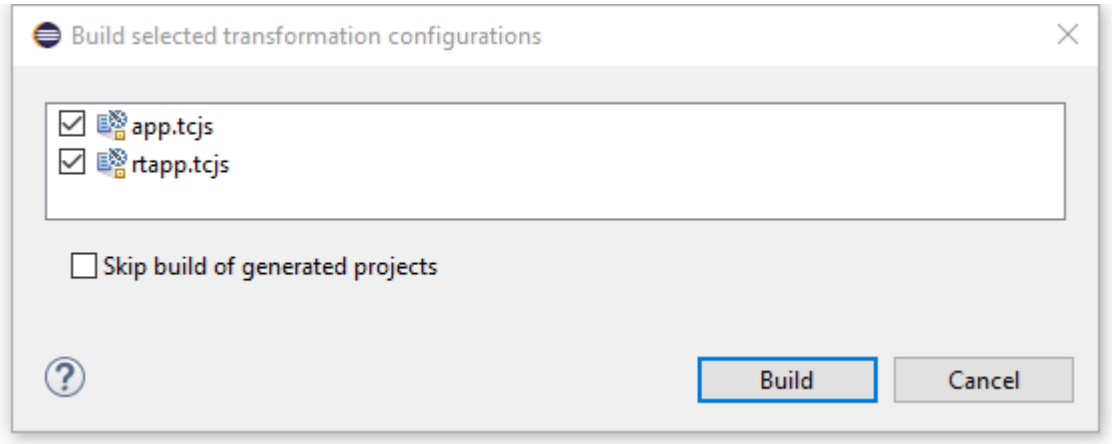


PRODUCTS & PLATFORMS

Model Compiler

- It’s no longer necessary to specify a license when launching the model compiler from command-line
  - The license of the RTist installation will be used
- Multiple TCs can now be built in one go
  - Select all TCs to build in the UI and perform the Build context menu command
  - The selected TCs will be built one by one consecutively

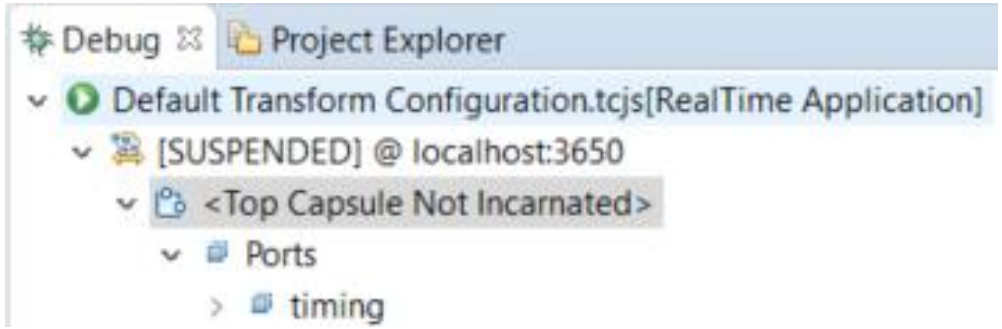




PRODUCTS & PLATFORMS

Model Debugger

- The Debug view now shows more clearly when the top capsule has not yet been incarnated

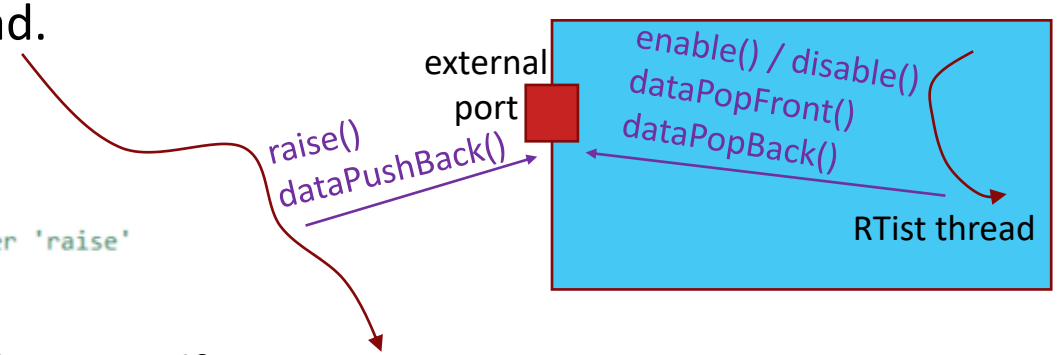


PRODUCTS & PLATFORMS

External Port Data

- The TargetRTS implementation of external ports has been extended to make it easier for code running in an external thread to pass data to an RTist application thread.
  - Data can now be passed when raising an external event

```
// This function may be used only on threads other than the one on
// which the owner capsule executes. Data can optionally be included in the request.
// After calling this the owner capsule is automatically disabled from receiving another 'raise'
// request until 'enable' is called again.
int raise(const void * data = 0, const RTObject_class * type = 0);
```



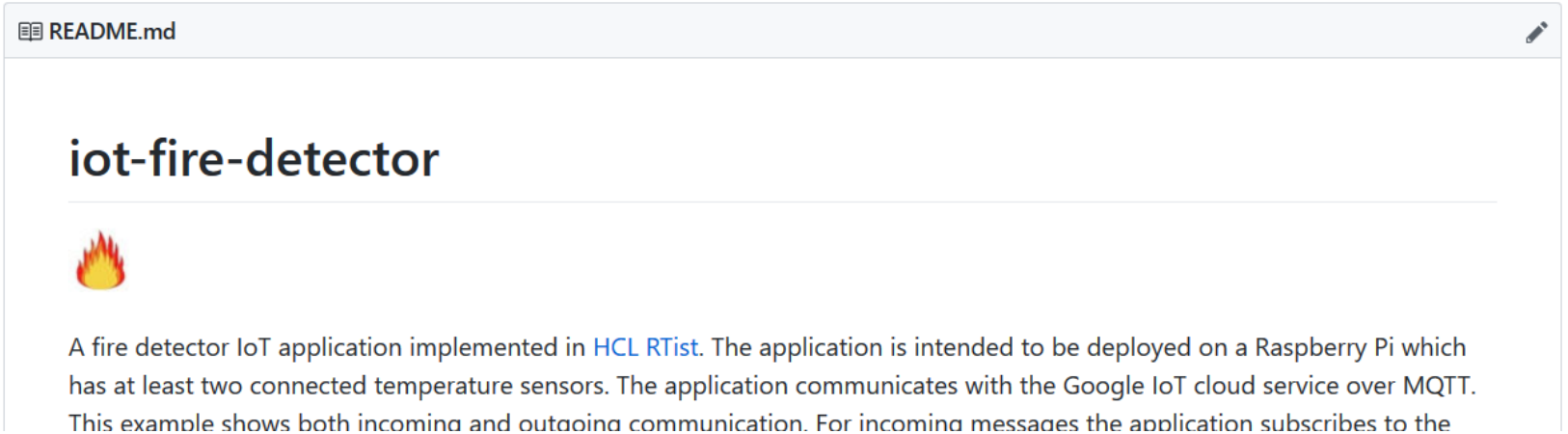
  - A thread-safe shared data area (a list) also exists on the external port itself

```
// Put a data object at the end of the externalData list. The data object should be non-null and dynamically allocated by
// code in the external thread, which calls this function.
void dataPushBack(void*);

// Pops the front data object off the externalData list. Returns the number of remaining data objects in the list
// (the data will be null in case of error, for example when attempting to pop from an empty list).
// This function is usually called by the owner capsule thread, and it should delete the popped data object when done with it.
unsigned int dataPopFront(void**);

// Pops the back data object off the externalData list. Returns the number of remaining data objects in the list
// (the data will be null in case of error, for example when attempting to pop from an empty list).
// This function is usually called by the owner capsule thread, and it should delete the popped data object when done with it.
unsigned int dataPopBack(void**);

// Attempts to find a specific data object in the externalData list. If it is found it will be deleted.
void dataDelete(void*);
```


PRODUCTS & PLATFORMS

GitHub Repositories

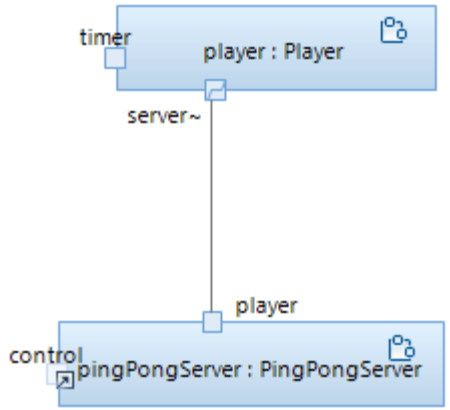
- A number of RTist sample applications have been created on GitHub
  - Focus is to show various ways how to send messages to or from a generated C++ application
- Also used for providing libraries of reusable functionality
- Repositories are located in the https://github.com/hcl-pnp-rtist organization and are MIT licensed
- Pull requests are welcome



PRODUCTS & PLATFORMS

JSON API for External Communication with RTist Applications

- lib-tcp-server is a library that allows an RTist application to communicate with other applications over TCP
  - Provides a JSON-based API for sending and invoking events on certain ports
  - Outgoing events on those ports can be routed to an external application
  - Available on GitHub as an open-source library
- The API can be used for developing distributed applications
  - A light-weight alternative to using Connexis
  - pingpong-distributed is a sample distributed application built this way
- The API can also be used as a way to test applications (black-box testing)

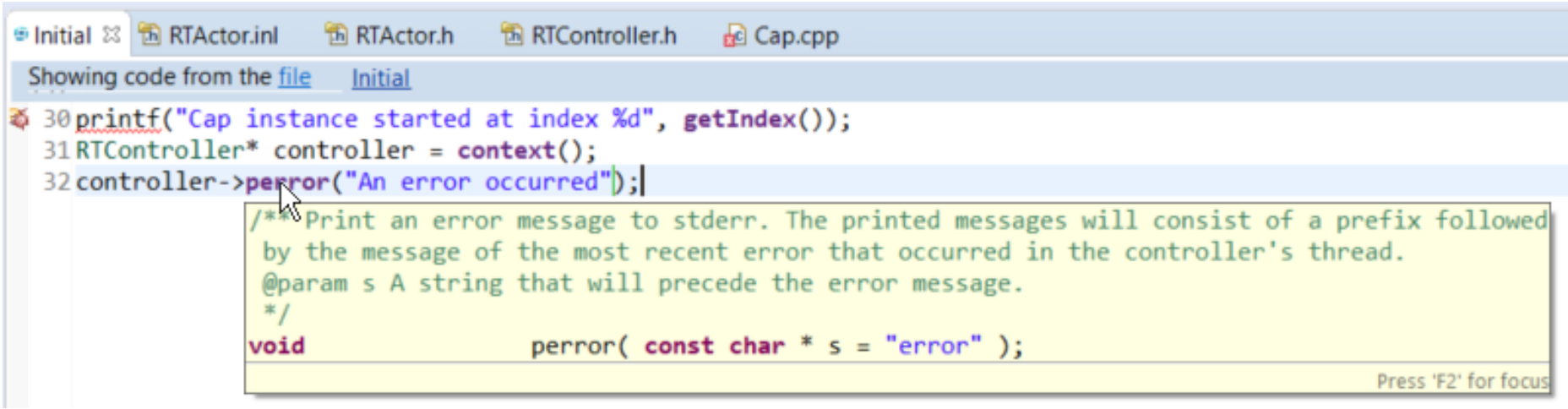



```
{ "command": "sendEvent", "port" : "trafficLight", "event" : "test_int", "data" : "int 15" }
```


PRODUCTS & PLATFORMS

TargetRTS Documentation

- The most commonly used classes and functions in the TargetRTS are now documented using Doxygen comments
- Generated Doxygen documentation is included in the Help
- This documentation is also useful when editing or viewing code in the Code editor

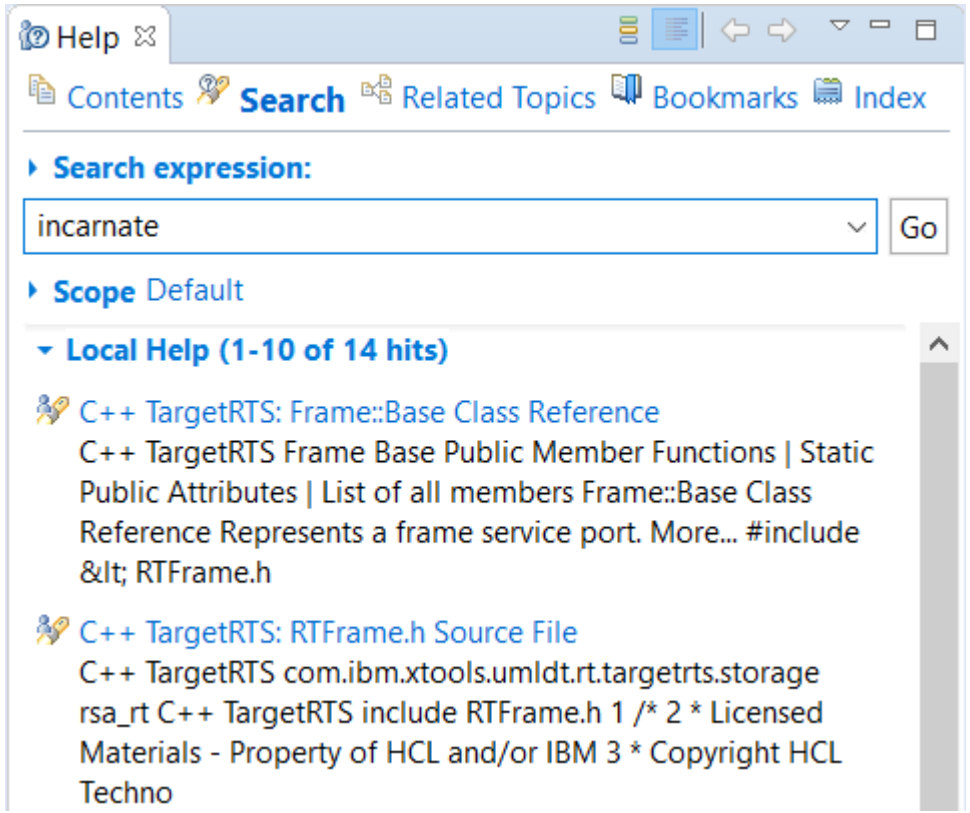


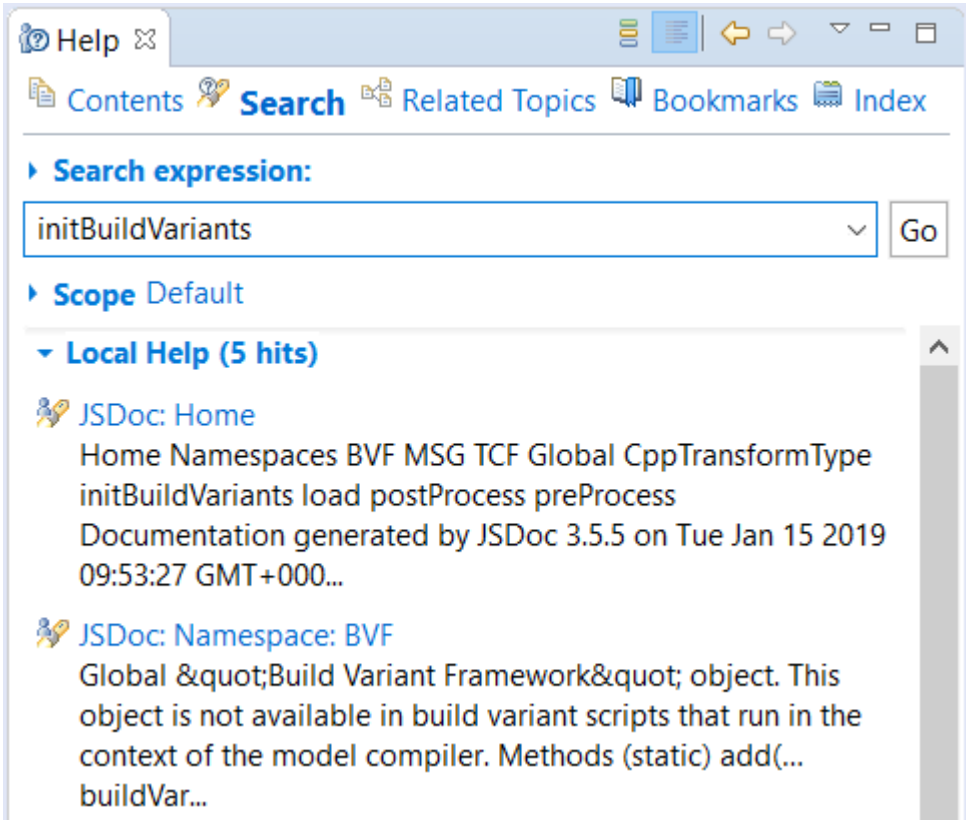
  - Note that the Code view does not support showing these tooltips

PRODUCTS & PLATFORMS

Searchable API Documentation

- Both the C++ TargetRTS APIs and JavaScript APIs are now searchable from the Help

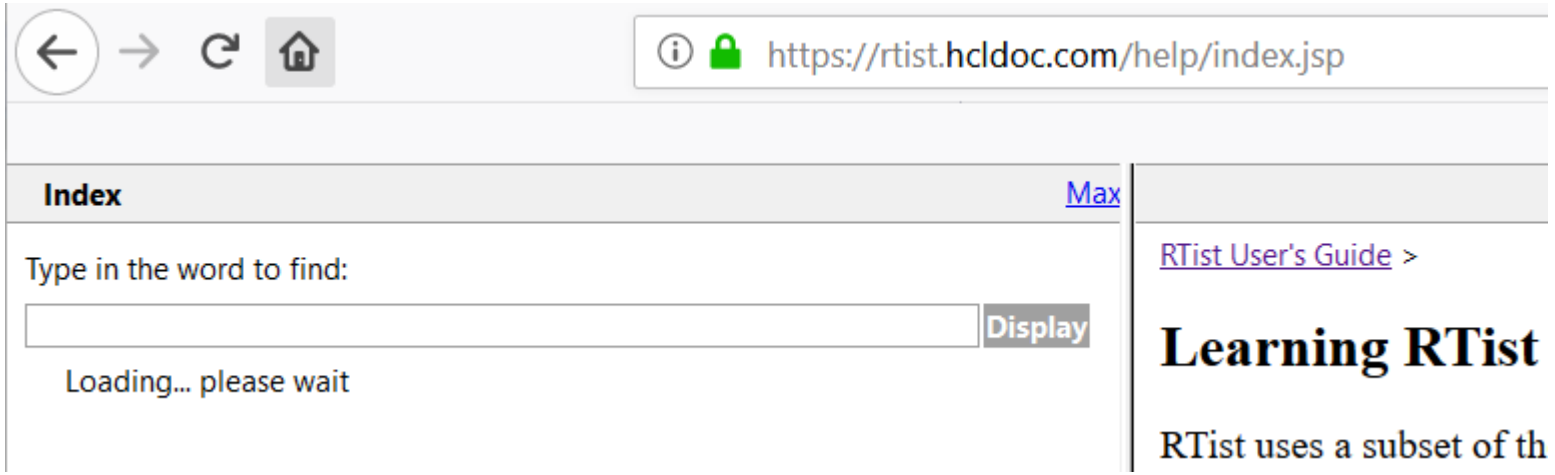




PRODUCTS & PLATFORMS

Online Documentation

- The RTist Eclipse Help documentation is now also available on: https://rtist.hcldoc.com/help/index.jsp
- This documentation is now tagged to enable search engines to index the contents



PRODUCTS & PLATFORMS

HCL

Relationship™

BEYOND THE CONTRACT

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

WATCH THE **FILM**