



Searching in Model RealTime

“seek and you will find”

Updated for DevOps Model RealTime 12

Mattias Mohlin

- ▶ For full Search documentation refer to the Help

The screenshot displays the 'Model RealTime User's Guide' documentation. On the left is a sidebar navigation menu with the following items:

- Model RealTime User's Guide
 - Overview
 - Learning Model RealTime
 - Articles
 - Building
 - CORBA IDL support
 - Comparing and Merging
 - Configuring and customizing
 - Connexis
 - Editing
 - Importing Models from Rose RT
 - Installation
 - Integrations
 - Message logs
 - Modeling realtime applications
 - Performance
 - Role-based learning guide
 - Running and debugging
 - Search and Navigation
 - Link with Editor
 - Navigating Model RealTime models
 - Navigating in the model using bookmarks
 - Search history
 - Searching Model RealTime models**

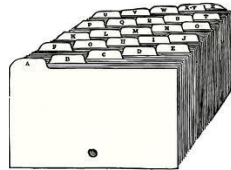
The main content area shows the breadcrumb path: [Model RealTime User's Guide](#) > [Articles](#) > [Search and Navigation](#). The main heading is **Searching Model RealTime models**. Below the heading is a PDF icon and the link [Model RealTime Search](#). The text reads: "This document describes the various search commands that are available in DevOps Model RealTime. The document also provides guidance for when to use the different search commands and hints for how to best find the items you are looking for in your model." It then says: "If you don't have time to read the complete document above, you can get an overview by viewing this [presentation](#) instead." At the bottom, there are two links: [Model search overview](#) and [Filtering search results](#).

Overview

- ▶ Model RealTime provides multiple search commands
 - There are usually more than one way to find the element you are looking for, but depending on situation some commands may be more efficient than others
 - Look upon the multiple search commands as multiple tools in your toolbox (you are more efficient than if you only had one tool for all tasks, but you need to know how to use them!)

- ▶ Three categories of search commands

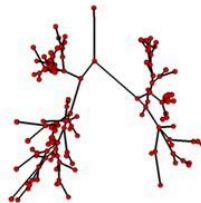
- **Index-based**



Uses a search index. Fast, once the index is up-to-date. Limited search criteria (can only search on the information that is indexed, such as element name). Index is stored in workspace.

Example: *Search Field*

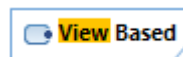
- **Memory-based**



Uses the model loaded into memory. Fast, once the model has been loaded (e.g. using Load UML Models). Custom search criteria can be used.

Example: *Find Triggers*

- **View-based**



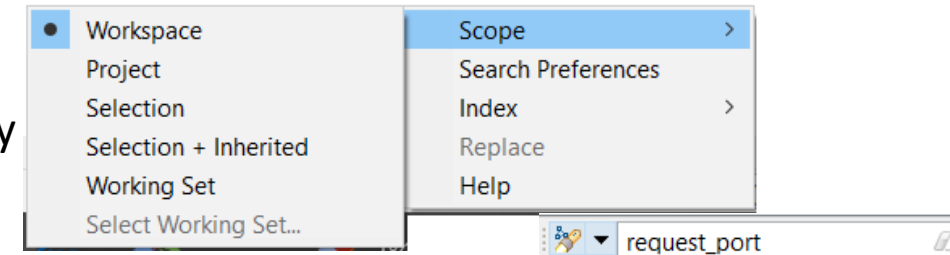
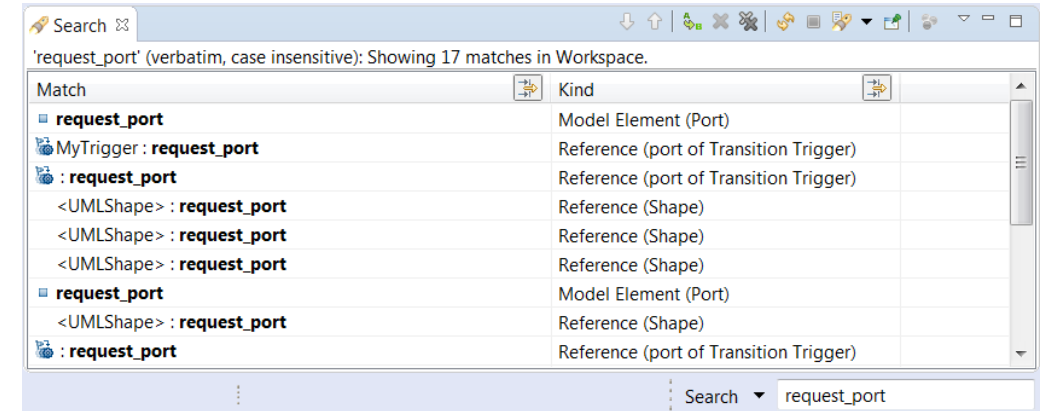
Uses what is shown in an editor.

Example: *Incremental Diagram Find*

Search Field

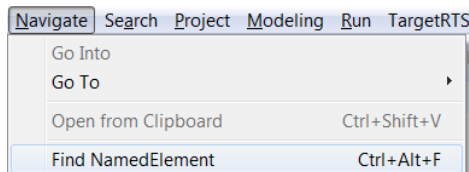
▶ General purpose searching in models

- Index-based search command (index build in background)
- Runs a search without use of modal dialogs
- Automatic proposals popup for common and recently used search terms
- Support for filtering the search result (including negative filters)
- Support for invoking Replace on the search result
- Can search either verbatimly (default) or on multiple words separately
- Set the scope where to look for matches
- The most powerful search command – can search in references, TC settings, and projects external to the workspace.
This command should be your default way of searching for something in Model RealTime!



Find NamedElement

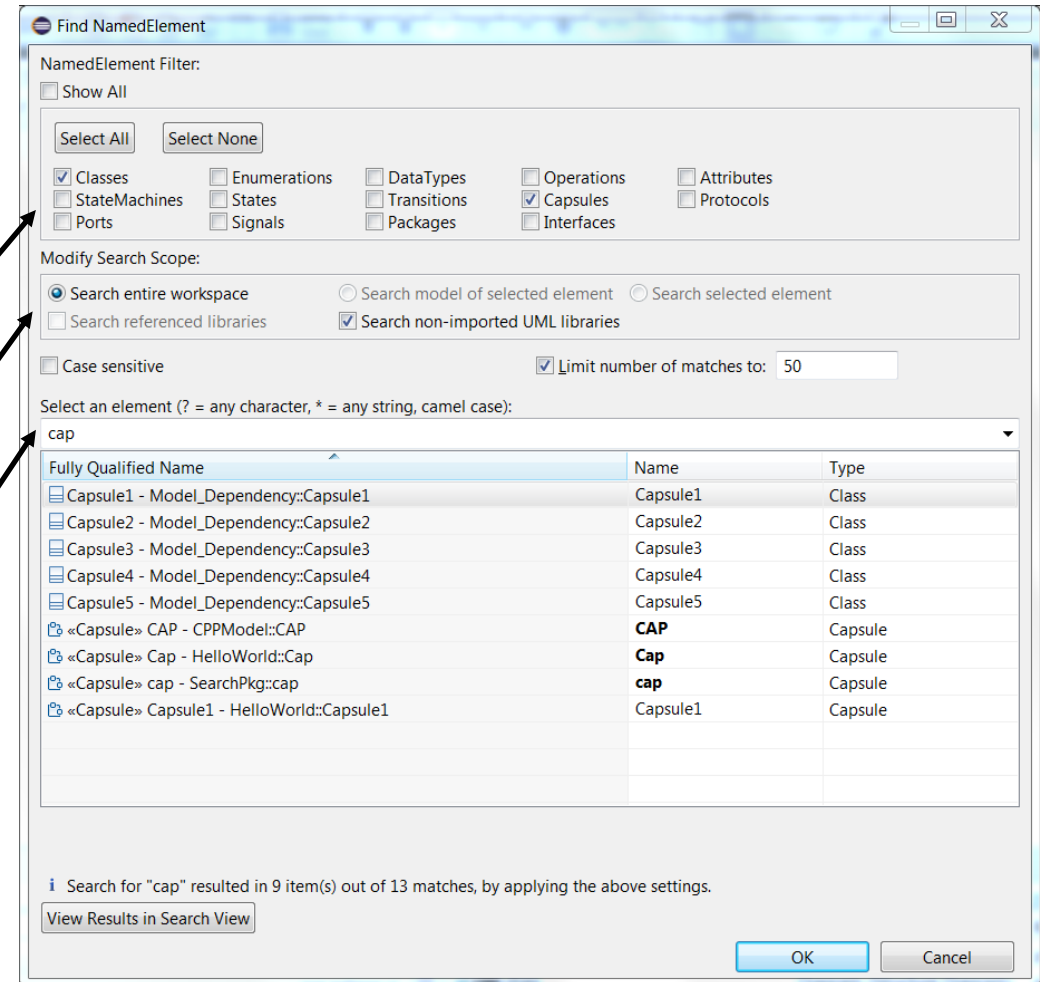
- ▶ Finds an element with a certain name
 - Index-based search command
 - Shows matches in dialog as you type
 - Best to use when you are looking for one particular element in the model



Filters for what is commonly used in RT models

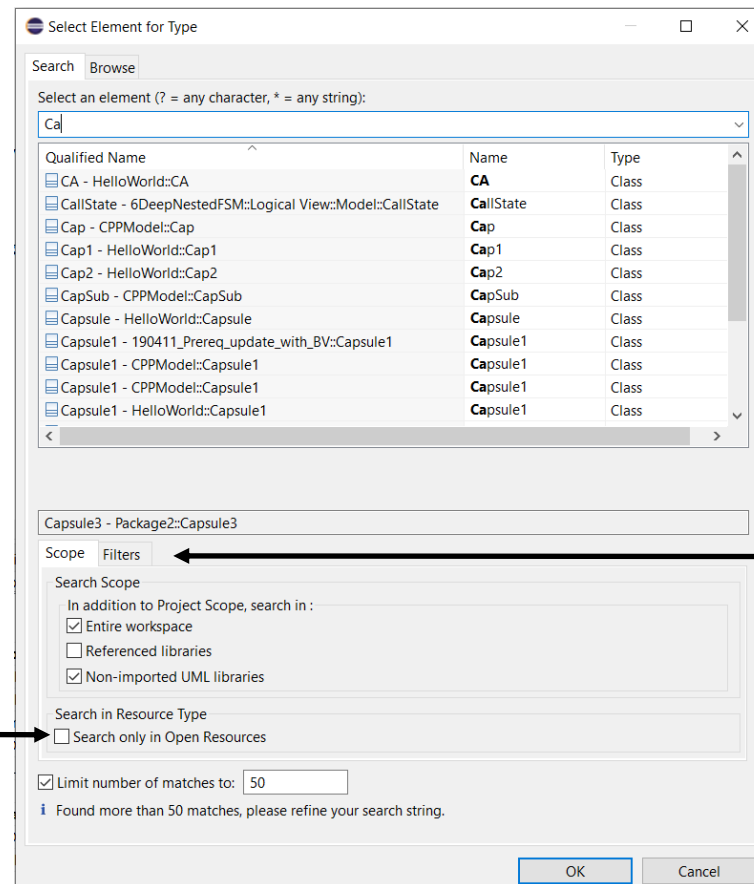
Use as narrow search scope as possible

Search string may use wildcards for pattern searches



Select Element Dialog

- ▶ Finds an element to use in some context other than searching
 - Example: Press the Set button to set the type of an attribute
 - Think about this as a simplified Find NamedElement dialog



Mark to not search in the entire index (skip closed models)

This dialog is not RT specific so there are many more filters here

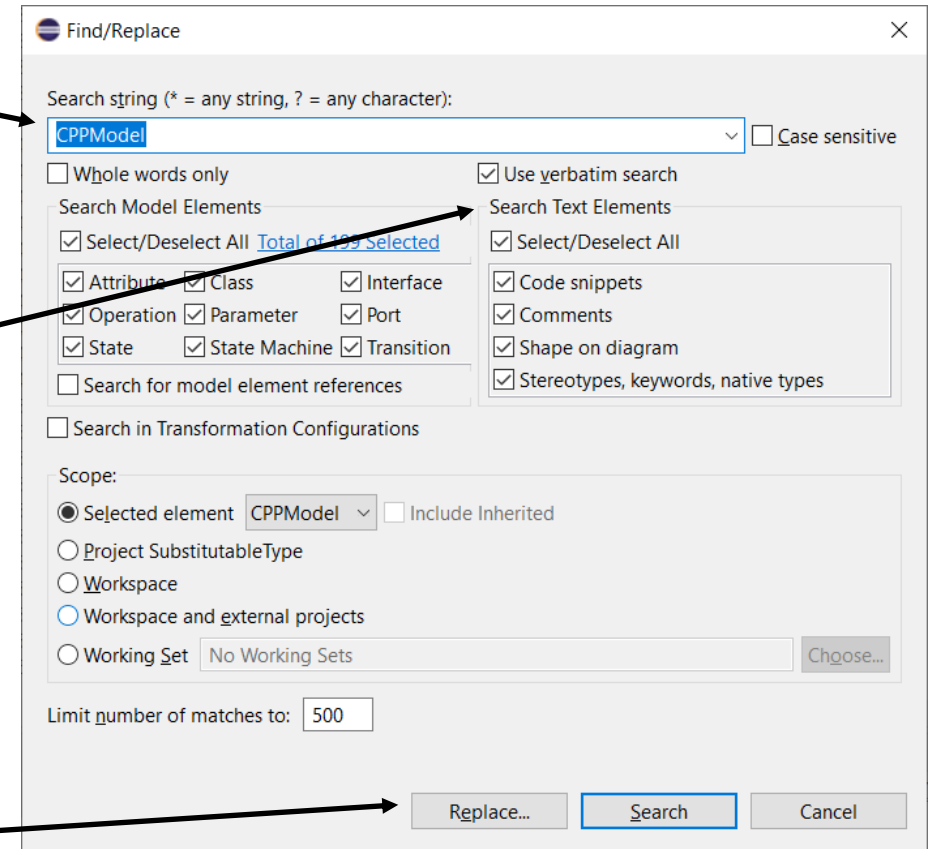
Find/Replace & Model Search Dialogs

- ▶ Finds elements with a certain name
 - Index-based search command
 - Matches are shown in the Search view when the dialog is closed
 - Similar to use of Search field, but requires settings to be made before searching

- ▶ Finds texts in the model
 - Also index based
 - Looks in code snippets, comments etc.
 - Often better to use the Search field instead!

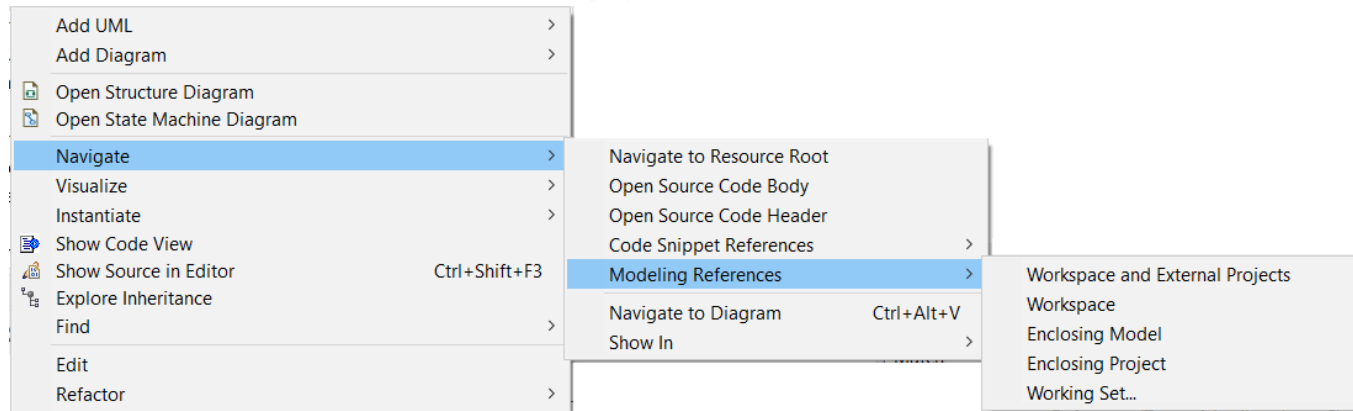
▶ Replacing

- Click “Replace” instead of “Search” to iterate the search result and replace the found name or text with some other string



Modeling References

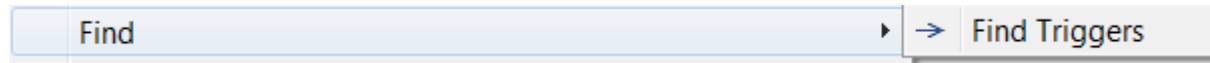
- ▶ Finds model references to an element
 - Index-based search command
 - Available in the context menu of a model element (in the Navigate submenu)
 - Answers the question “Where in the model is this element used?”
 - Note: Only model references are found – not textual references (for example from code snippets). The command Code Snippet References can be used for that. It is based on CDT:s References command (and hence requires presence of generated C++ code).
 - For elements that have a name, you can instead use the Search field to find the references (filter the Kind column to only show reference matches)



Select the scope where to look for elements that reference the selected element.

Find Triggers

- ▶ Finds transition triggers for a selected element
 - Memory-based search command
 - Available in the context menu (in the Find submenu) of a port, protocol event or trigger operation



- **Protocol event**
Finds the transition triggers which trigger on the protocol event.
- **Behavior port**
Finds the transition triggers which trigger when an event arrives at the port.
- **Trigger operation**
Finds the transition triggers which trigger when the trigger operation is called.
- Found triggers are listed in the Search view together with related information.
Navigate to it by double-click (or in Navigate context menu).

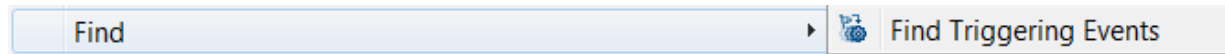
A screenshot of a software interface showing a search result for a transition trigger. The search view displays a table with the following columns: Transition, Port(s), Trigger Guard, Transition Guard, Source, Target, and Owner. The search results show a single match for the event 'InEvent2 (data : int)'. The table contains the following data:

Transition	Port(s)	Trigger Guard	Transition Guard	Source	Target	Owner
t1	pp			State1	State2	cap

A context menu is open over the first row of the table, showing options: Transition 't1', Port 'pp', Source 'State1', Target 'State2', and Owner 'cap'. The 'Navigate' option is highlighted.

Find Triggering Events

- ▶ Finds the events which may trigger a transition
 - Memory-based search command
 - Available in the context menu (in the Find submenu) of a state, state machine or capsule/class



- **State**

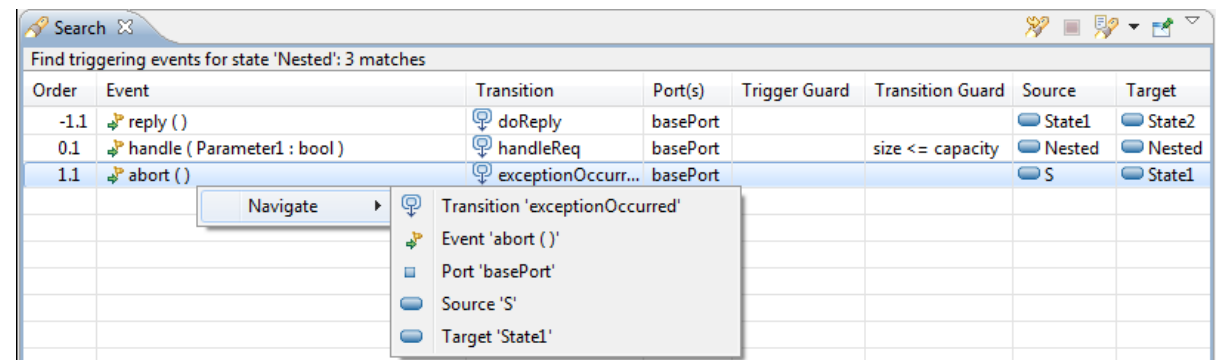
Finds the events which may trigger a transition when the state is active.

- **State machine**

Finds all events which the state machine can handle.

- **Capsule or class with state machine**

Finds all events which the state machine of the capsule/class may handle.



Search X

Find triggering events for state 'Nested': 3 matches

Order	Event	Transition	Port(s)	Trigger Guard	Transition Guard	Source	Target
-1.1	reply ()	doReply	basePort			State1	State2
0.1	handle (Parameter1 : bool)	handleReq	basePort		size <= capacity	Nested	Nested
1.1	abort ()	exceptionOccurr...	basePort			S	State1

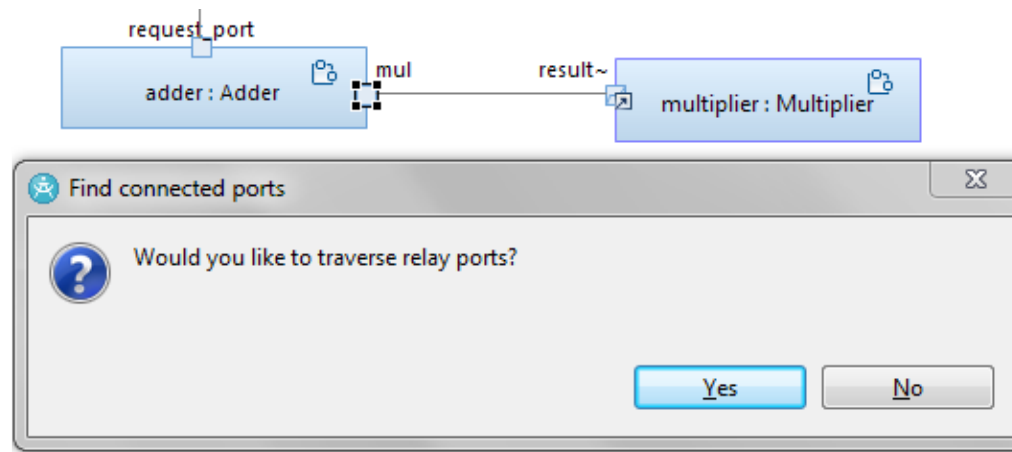
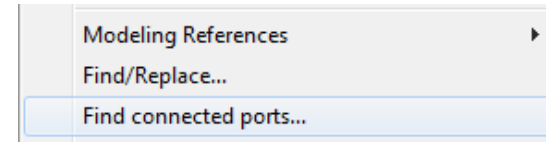
Navigate ▶

- Transition 'exceptionOccurred'
- Event 'abort ()'
- Port 'basePort'
- Source 'S'
- Target 'State1'

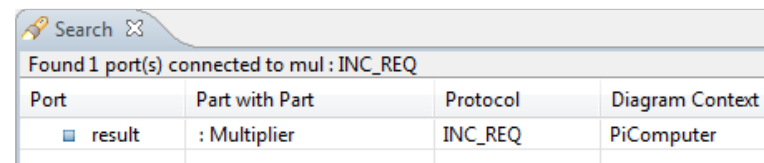
Find Connected Ports

► Finds the ports which a selected port may communicate with

- Memory-based search command
- Available in the context menu of a port (both in diagrams and in Project Explorer)
- You can choose to traverse relay ports



▪ Found ports are listed in the Search view



A screenshot of the Search view showing a table of results. The title bar says 'Search' and the main text says 'Found 1 port(s) connected to mul : INC_REQ'. The table has four columns: Port, Part with Part, Protocol, and Diagram Context.

Port	Part with Part	Protocol	Diagram Context
result	: Multiplier	INC_REQ	PiComputer

Incremental Diagram Search

- ▶ Finds texts shown in a diagram
 - View-based search command
 - Same keyboard accelerators as for text editors (Ctrl+J for forward search and Ctrl+Shift+J for backwards search)
 - In particular useful for finding texts in big and cluttered diagrams
 - Works also in derived diagrams such as browse diagrams

