OPEN**RTOS** APPLICATION NOTE: #34-185-AN-001

INSTALLING AND USING THE WHIS STATE**VIEWER** KERNEL AWARE PLUG-IN FOR ECLIPSE

Issue 2 – July 6 2011

CONTENTS

FIGURES

REFERENCED DOCUMENTS

Ref #	Document	Description
None		

CHAPTER 1

INTRODUCTION

1.1 DESCRIPTION

The STATE**VIEWER** Kernel Aware Plug-in for Eclipse provides a snapshot view of the FreeRTOS/OPEN**RTOS**/SAFE**RTOS** kernel data structures presented in a meaningful tabular format. This allows the current state of tasks, queues and the various forms of semaphore to be inspected directly from within the Eclipse Workbench IDE.

WITTENSTEIN high integrity systems have made the STATE**VIEWER** plug-in available for use by the FreeRTOS.org™ community as well.

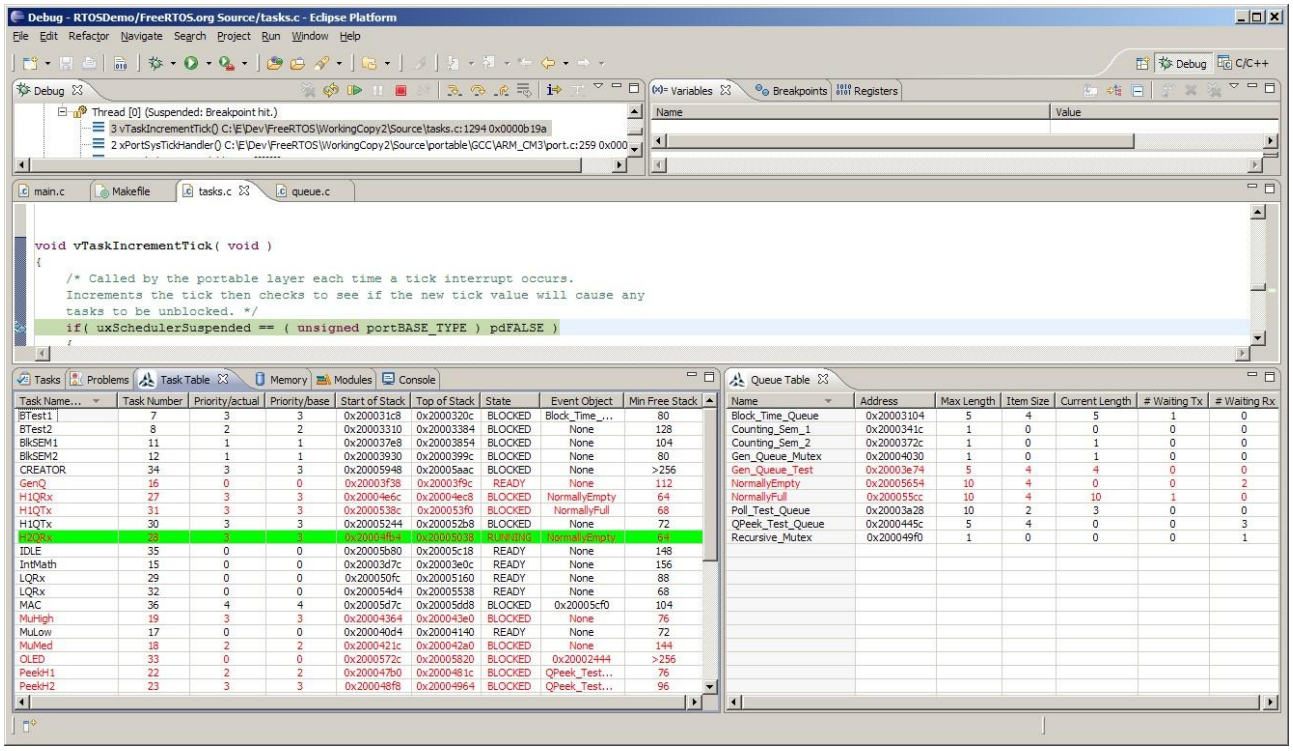


Figure 1-1-1 The Eclipse Workbench IDE with the Task and Queue viewers visible

CHAPTER 2

INSTALLATION & USAGE

2.1 INSTALLING THE PLUG-IN

Step 1. Within Eclipse, from the ‘Help’ menu select ‘Install New Software…’ item.

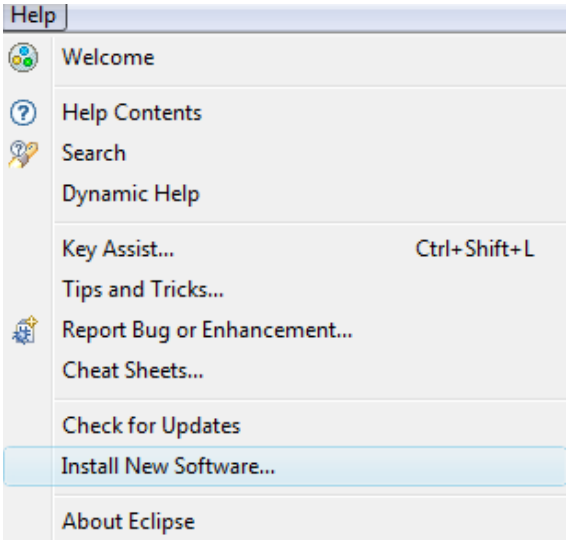


Figure 2-1: 'Install New Software...' menu item.

Step 2. In the ‘Work with:’ category, click the ‘Add…’ button. See Figure 2-3.

Step 3. In the ‘Add Repository’ dialog window, specify plug-in name as ‘StateViewer’ and location path as http://www.highintegritysystems.com/StateViewer/ and click ‘OK’

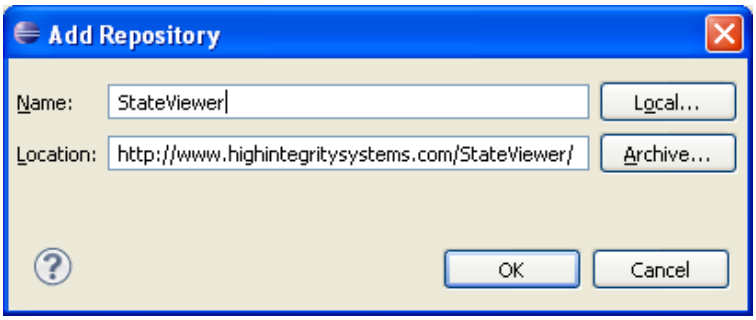


Figure 2-2: 'Add Repository' Dialog.

Step 1. Tick the ‘WHIS StateViewer Plugins’ category and click the ‘Next >’ button.

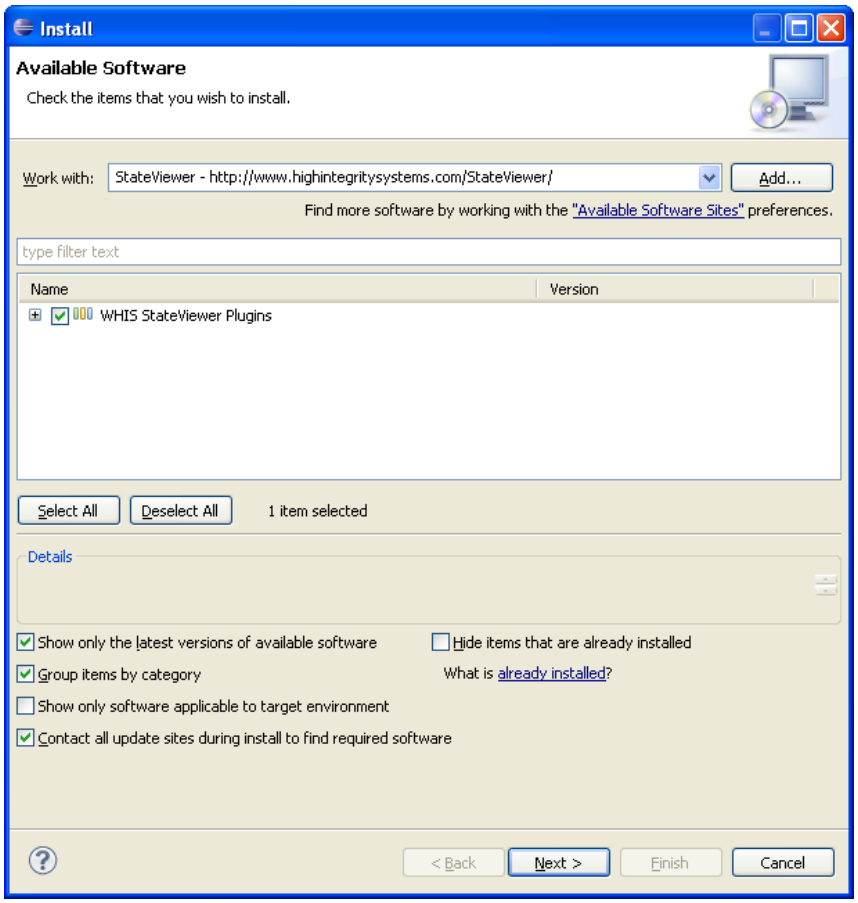


Figure 2-3: ‘Available Software’ window.

Step 2. The ‘Install Details’ window displays the items to be installed. Click ‘Next >’.

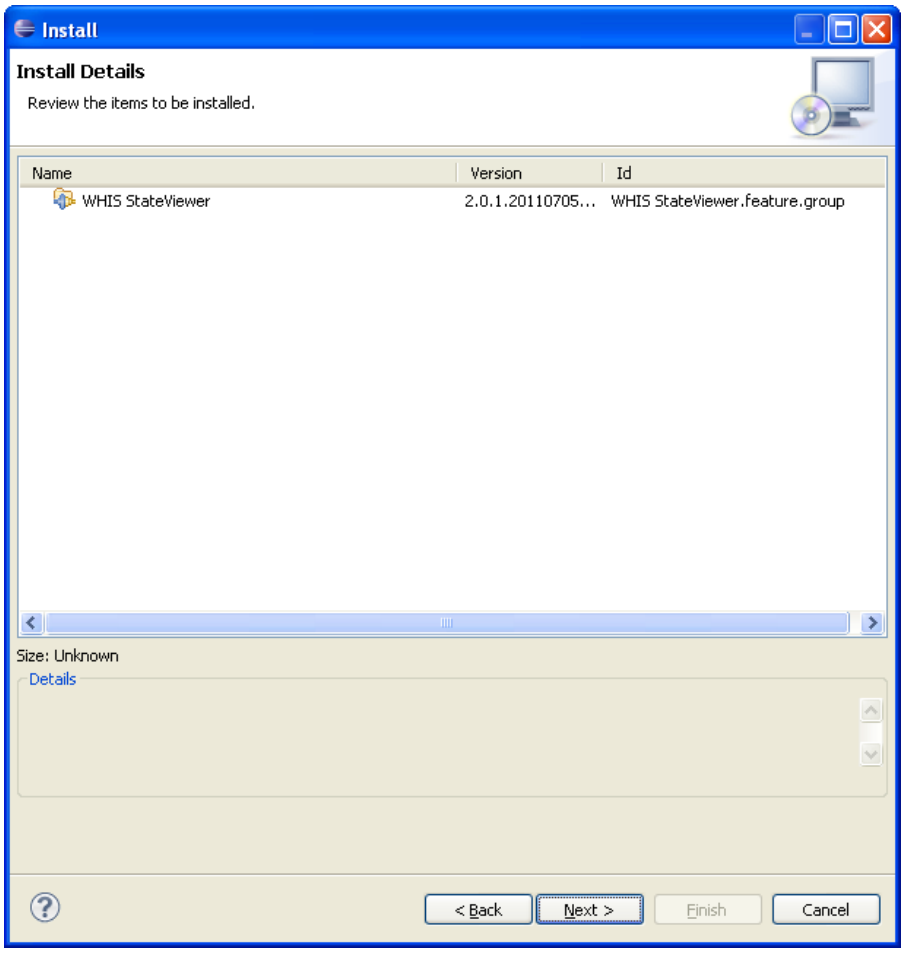


Figure 2-4: 'Install Details' window.

Step 3. The ‘Review Licenses’ window requires accepting the License to use the software to proceed. Read the ‘License Text’ for WHIS StateViewer. If you agree with the terms and conditions, select the ‘I accept the terms of the license agreements’ pushbutton and click ‘Finish’ to install the software.

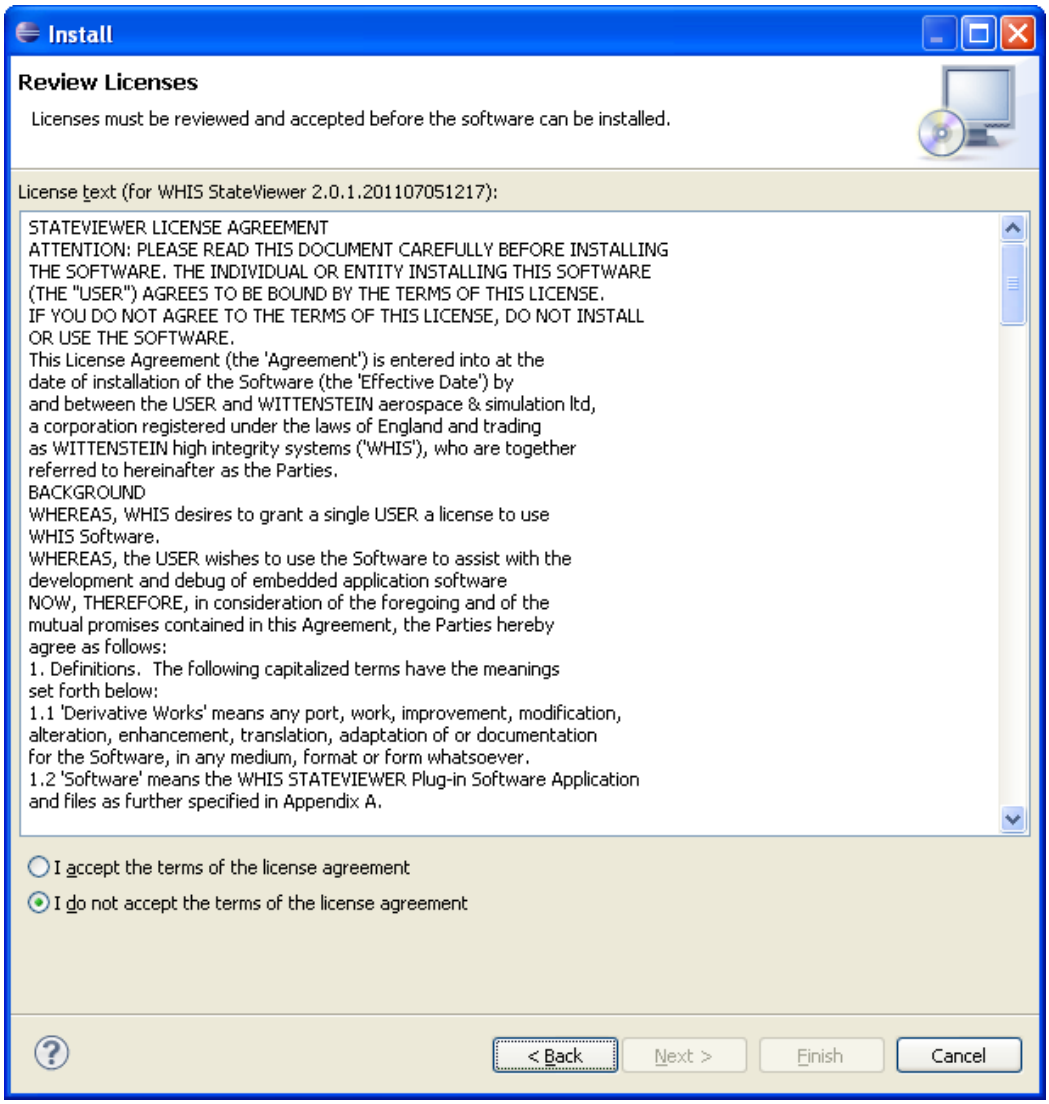


Figure 2-5: 'Review Licenses' window.

Step 4. Check for certificate details as in Figure 2-6 and tick the certificate “WITTENSTEIN High Integrity Systems; High Integrity Systems; WITTENSTEIN aerospace & simulation Ltd.” and then click ‘OK’.

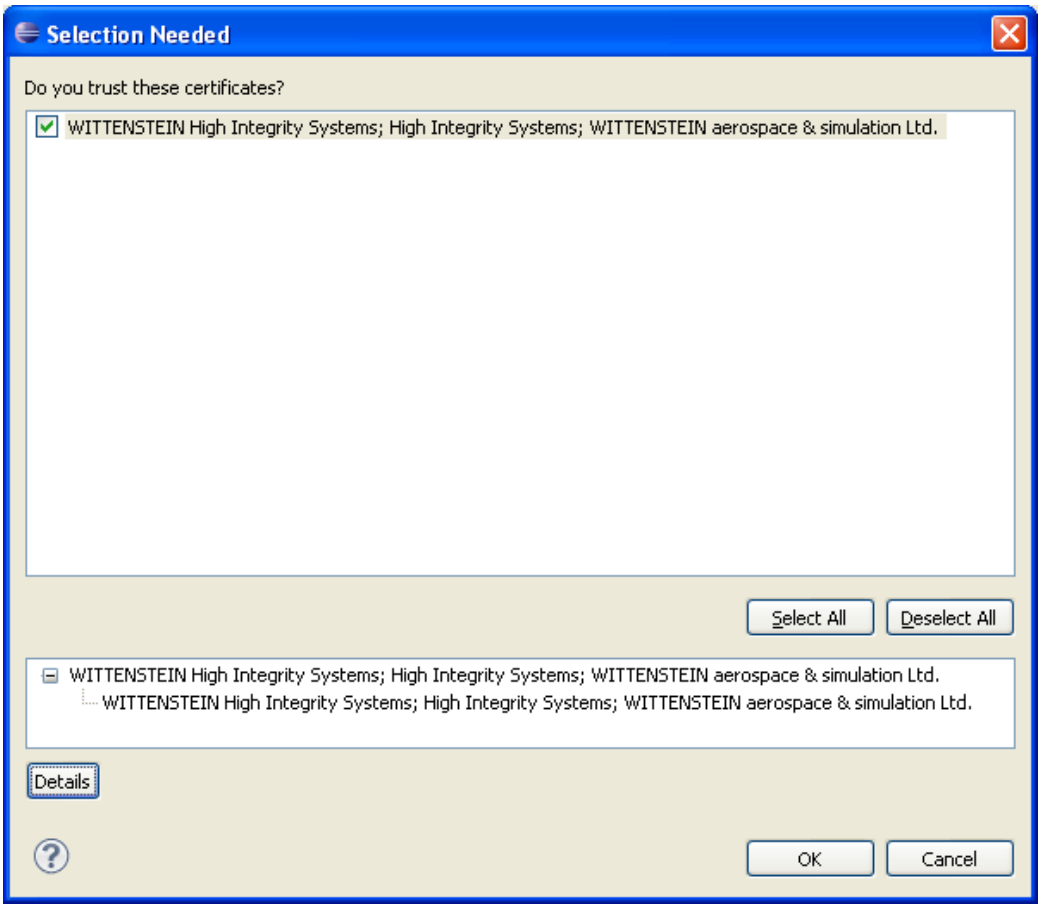


Figure 2-6: Certificate Selection Window

Step 5. When the Installation has finished, Eclipse prompts for restart. Now restart the Eclipse by clicking ‘Restart Now’.

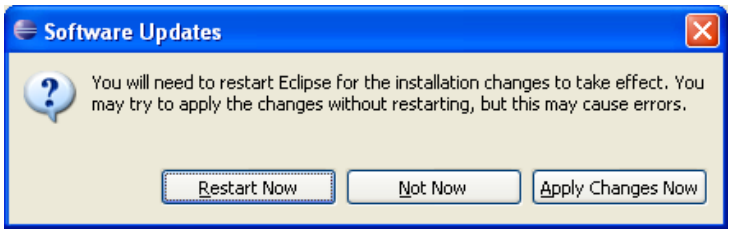


Figure 2-7: Restart Eclipse notice.

Step 6. When WHIS StateViewer plug-in is installed correctly you will see ‘OpenRTOS Viewer’ and ‘SafeRTOS Viewer’ in Eclipse show View window as in Figure 2-8.

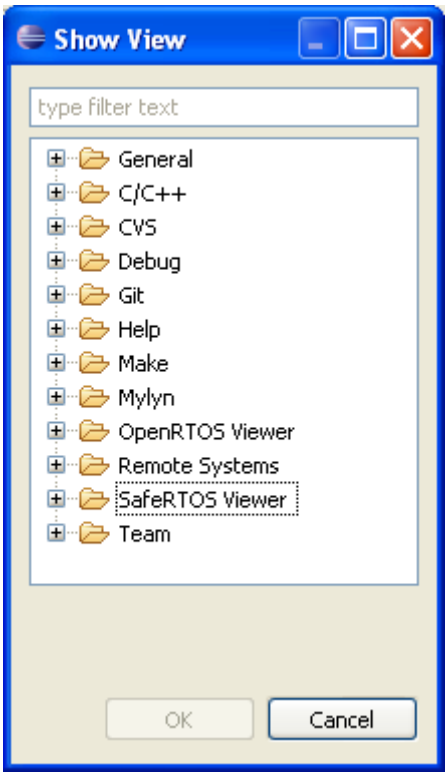


Figure 2-8 Show View window

2.2 OPENING THE RTOS VIEWER TABLES

When you are ready to start a debug session:

Step 1. Start a debug session as normal.

Step 2. Ensure ‘Debug’ is the current Eclipse perspective.

Step 3. From the Eclipse Workbench “Window” menu, select “Show View” → “Other” (Figure 2-9).

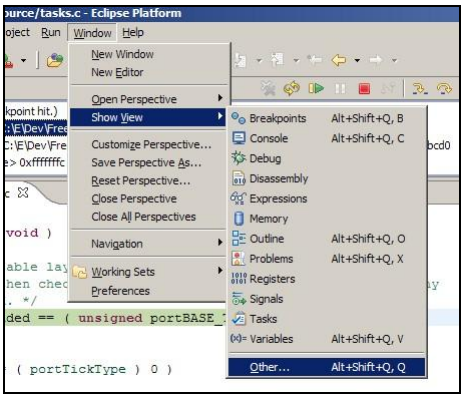


Figure 2-9: Opening up the "Other" view dialogue

Step 4. From the resultant dialogue box, expand the required either OpenRTOS or SafeRTOS Viewer tree (Figure 2-10) and select whichever table is required and click OK to add to the Eclipse perspective.

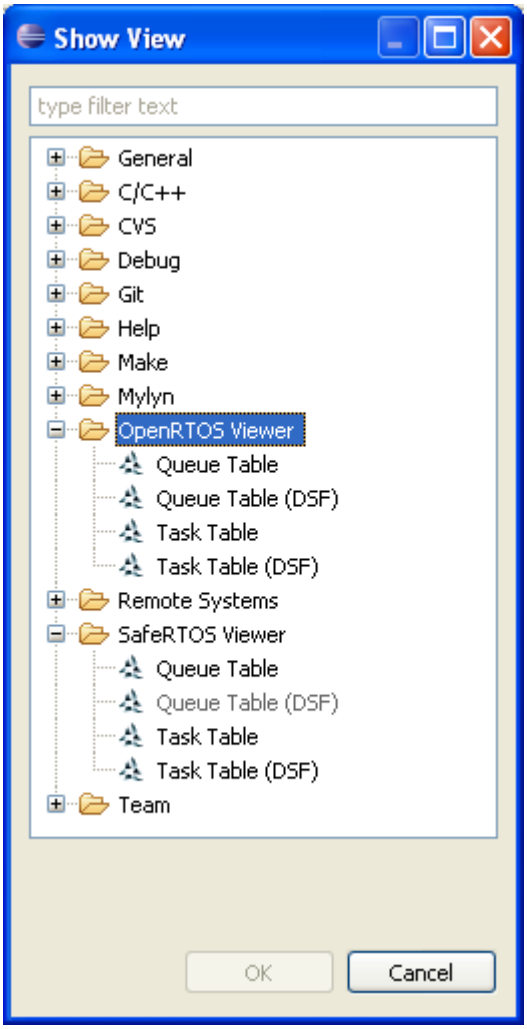
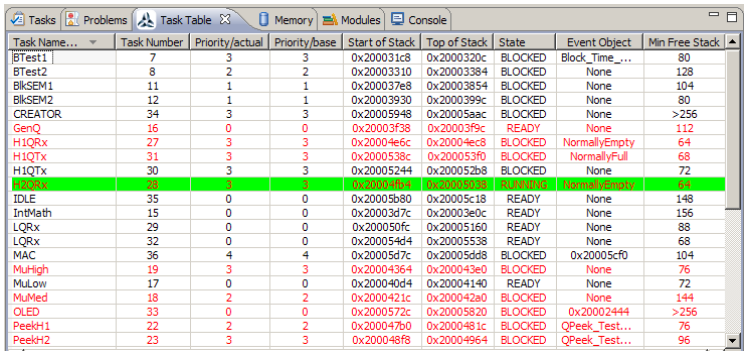


Figure 2-10: StateViewer Tables in the Show View window.

The tables will refresh themselves each time the debugger is manually suspended, hits a break point, or is single stepped.

The tabs that contain the tables can be docked in any position within the Eclipse Workbench IDE simply by dragging and dropping the tabs to your desired location.

2.3 INTERPRETING TASK TABLE DATA



Task Name...	Task Number	Priority/actual	Priority/base	Start of Stack	Top of Stack	State	Event Object	Min Free Stack
BTest1	7	3	3	0x200031c8	0x2000320c	BLOCKED	Block_Time_...	80
BTest2	8	2	2	0x20003310	0x20003384	BLOCKED	None	128
BlkSEM1	11	1	1	0x200037e8	0x20003854	BLOCKED	None	104
BlkSEM2	12	1	1	0x20003930	0x2000399c	BLOCKED	None	80
CREATOR	34	3	3	0x20005948	0x20005aac	BLOCKED	None	>256
GenQ	16	0	0	0x20003f38	0x20003f9c	READY	None	112
H1QRx	27	3	3	0x20004e6c	0x20004ec8	BLOCKED	NormallyEmpty	64
H1QTx	31	3	3	0x2000538c	0x200053f0	BLOCKED	NormallyFull	68
H1QTx	30	3	3	0x20005244	0x200052b8	BLOCKED	None	72
H2QRx	28	3	3	0x20004ffa4	0x20005038	BLOCKED	NormallyEmpty	64
IDLE	35	0	0	0x20005b80	0x20005c18	READY	None	148
IntMath	15	0	0	0x20003d7c	0x20003e0c	READY	None	156
LQRx	29	0	0	0x200050fc	0x20005160	READY	None	88
LQRx	32	0	0	0x200054d4	0x20005538	READY	None	68
MAC	36	4	4	0x20005d7c	0x20005dd8	BLOCKED	0x20005cf0	104
MuHigh	19	3	3	0x20004364	0x200043e0	BLOCKED	None	76
MuLow	17	0	0	0x200040d4	0x20004140	READY	None	72
MuMed	18	2	2	0x2000421c	0x200042a0	BLOCKED	None	144
OLED	33	0	0	0x2000572c	0x20005820	BLOCKED	0x20002444	>256
PeekH1	22	2	2	0x200047b0	0x2000481c	BLOCKED	QPeek_Test...	76
PeekH2	23	3	3	0x200048f8	0x20004964	BLOCKED	QPeek_Test...	96

Figure 2-11: Example task table screen shot

Figure 2-11 provides an example task table screen shot where each table row provides information on a single task. If the row is black then the information it contains has not changed since the debugger was last suspended. Conversely, if the row is red then at least one of the columns in that row has changed since the debugger was last suspended. The green highlight indicates the currently executing task – the task that is in the Running state.

Table 2-1 Task Table Columns

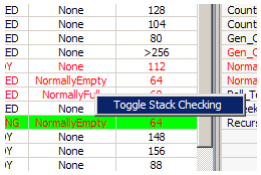
Column Heading	Description
Task Name	The plain text name assigned to the task when the task was created.
Task Number	A unique (assuming no overflows) number assigned to the task by the kernel.
Priority/actual	The priority currently being used by the task. The OPEN**RTOS** priority inheritance mechanism means the actual priority can transiently be higher than the tasks base (assigned) priority.
Priority/base	The priority assigned to the task. A priority is assigned when the task is created and each time the task is the subject of a call to vTaskPrioritySet().
Start of Stack	The address of the beginning of the stack assigned to the task. The beginning of the stack is the extreme of the stack region that gets written into first.
Top of Stack	The address of the current top of stack – that is the position last written to when the context of the stack was last saved. This value has little meaning for the current Running state task.
State	A text description of the task state, as specified in the OPEN**RTOS** user manual.
Event Object	The name or address of the queue or semaphore on which the task is blocked – if any. The name is used if the queue or semaphore has been registered within the queue registry.
Min Free Stack	The stack 'high water mark'. This is the minimum number of bytes there have been between the task stack pointer and the end of the stack assigned to the task. The lower this value the closer the task has come to overflowing its stack. A value of 0 indicates that the task has overflowed its stack at some time. Calculating the stack high water mark takes a noticeable amount of time so it is sometimes desirable to disable the calculation altogether. Right clicking on the task table displays a menu that allows the feature to be toggle on and off. 

Figure 2-12: Right clicking on the Task Table to reveal the "Toggle Stack Checking" pop up menu

2.4 INTERPRETING QUEUE TABLE DATA

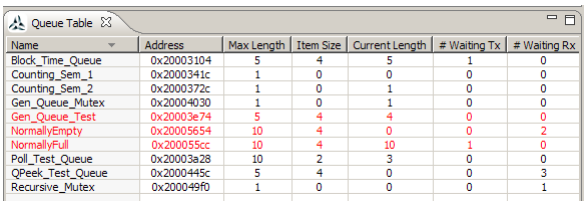


Figure 2-13: Example queue table screen shot

The queue table displays data on all queues and semaphores that have been added to the queue registry by a call to vQueueAddToRegistry() (see the RTOS user manual for more information). The following description refers only to queues, but is equally applicable to binary semaphores, counting semaphores, mutex type semaphores and recursive semaphores.

Figure 2-13 provides an example queue table screen shot where each table row provides information on a single queue or semaphore. If the row is black then the information it contains has not changed since the debugger was last suspended. Conversely, if the row is red then at least one of the columns in that row has changed since the debugger was last suspended.

Table 2-2 Queue Table Columns

Column Heading	Description
Name	The name that was assigned to the queue when the queue was registered.
Address	The address of the queue structure. This is also the queue handle value.
Max Length	The maximum number of items the queue can hold at any one time.
Item Size	The size in bytes that each queued item occupies.
Current Length	The number of items currently in the queue. This can only be less than or equal to the Max Length.
# Waiting Tx	The number of tasks currently blocked waiting for space to become available on the queue. That is – tasks that are blocked waiting to send to the queue.
# Waiting Rx	The number of tasks currently blocked waiting for data to become available from the queue. That is – tasks that are blocked waiting to receive from the queue.

CONTACT INFORMATION

User feedback is essential to the continued maintenance and development of our products. Please provide all software and documentation comments and suggestions to the most convenient contact point listed below.

Address:	WITTENSTEIN high integrity systems Brown's Court, Long Ashton Business Park Yanley Lane, Long Ashton Bristol, BS41 9LB England
Phone:	+44 (0)1275 395 600
Fax:	+44 (0)1275 393 630
Email:	support@HighIntegritySystems.com
Website	www.HighIntegritySystems.com

All Trademarks acknowledged.