

# ***KwikLook***<sup>™</sup> **Fault Finder**

for use with the  
**AMX Multitasking Executive**

## **User's Guide**

**First Printing: December 1, 1995**  
**Last Printing: November 1, 2007**

**Copyright © 1995-2007**

**KADAK Products Ltd.**  
**206 - 1847 West Broadway Avenue**  
**Vancouver, BC, Canada, V6J 1Y5**  
**Phone: (604) 734-2796**  
**Fax: (604) 734-8114**



## TECHNICAL SUPPORT

KADAK Products Ltd. is committed to technical support for its software products. Our programs are designed to be easily incorporated in your systems and every effort has been made to eliminate errors.

Engineering Change Notices (ECNs) are provided periodically to repair faults or to improve performance. You will automatically receive these updates during the product's initial support period. For technical support beyond the initial period, you must purchase a Technical Support Subscription. Contact KADAK for details. Please keep us informed of the primary user in your company to whom update notices and other pertinent information should be directed.

Should you require direct technical assistance in your use of this KADAK software product, engineering support is available by telephone, fax or e-mail. KADAK reserves the right to charge for technical support services which it deems to be beyond the normal scope of technical support.

We would be pleased to receive your comments and suggestions concerning this product and its documentation. Your feedback helps in the continuing product evolution.

KADAK Products Ltd.  
206 - 1847 West Broadway Avenue  
Vancouver, BC, Canada, V6J 1Y5

Phone: (604) 734-2796  
Fax: (604) 734-8114  
e-mail: [amxtech@kadak.com](mailto:amxtech@kadak.com)

**Copyright © 1995-2007 by KADAK Products Ltd.  
All rights reserved.**

No part of this publication may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language or computer language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual or otherwise, without the prior written permission of KADAK Products Ltd., Vancouver, BC, CANADA.

### **DISCLAIMER**

KADAK Products Ltd. makes no representations or warranties with respect to the contents hereof and specifically disclaims any implied warranties of merchantability and fitness for any particular purpose. Further, KADAK Products Ltd. reserves the right to revise this publication and to make changes from time to time in the content hereof without obligation of KADAK Products Ltd. to notify any person of such revision or changes.

### **TRADEMARKS**

AMX in the stylized form and KwikNet are registered trademarks of KADAK Products Ltd. AMX, AMX/FS, InSight, *KwikLook* and *KwikPeg* are trademarks of KADAK Products Ltd. Microsoft, MS-DOS and Windows are registered trademarks of Microsoft Corporation. All other trademarked names are the property of their respective owners.

# **KwikLook User's Guide**

## **Table of Contents**

	<b>Page</b>
<b>1. Introduction to <i>KwikLook</i></b>	<b>1</b>
1.1 General Operation .....	1
1.2 Installed <i>KwikLook</i> Components .....	4
1.3 <i>KwikLook</i> Program Group .....	6
<b>2. <i>KwikLook</i> for Task-Aware Debuggers</b>	<b>7</b>
2.1 Task-Aware Debugging .....	7
2.2 Trouble Shooting Task-Aware Debuggers .....	8
2.3 <i>KwikLook</i> for MetaWare SeeCode .....	9
2.4 <i>KwikLook</i> for Diab-SDS SingleStep .....	11
2.5 <i>KwikLook</i> for Paradigm 16-Bit Debugger .....	15
2.6 <i>KwikLook</i> for Paradigm 32-Bit Debugger .....	19
2.7 <i>KwikLook</i> for Metrowerks CodeWarrior .....	23
2.8 <i>KwikLook</i> for ARM Ltd. RealView .....	27
2.9 <i>KwikLook</i> for IAR Systems C-SPY .....	37

### **Table of Figures**

Figure 1.1-1 <i>KwikLook</i> Display Window .....	2
Figure 1.1-2 <i>KwikLook</i> System Summary .....	3
Figure 2.5-1 Paradigm 16-Bit Environment Options .....	16
Figure 2.6-1 Paradigm 32-Bit Environment Options .....	20
Figure 2.7-1 CodeWarrior Target Debugger Settings .....	24
Figure 2.8-1 RealView Target Connection Control .....	28
Figure 2.8-2 RealView Connection Properties .....	29
Figure 2.8-3 RealView Resource Viewer .....	29
Figure 2.8-4 RealView Plug-In List Control .....	30
Figure 2.8-5 RealView's <i>KwikLook</i> Plug-In Window .....	30
Figure 2.8-6 RealView 3.1 Connect to Target .....	31
Figure 2.8-7 RealView 3.1 Connection Properties .....	32
Figure 2.8-8 RealView 3.1 Resource Viewer .....	33
Figure 2.8-9 RealView 3.1 Plug-In List Control .....	34
Figure 2.8-10 RealView's 3.1 <i>KwikLook</i> Plug-In Window .....	34
Figure 2.9-1 C-SPY Debugger Plugins .....	38

This page left blank intentionally.

# 1. Introduction to *KwikLook*

## 1.1 General Operation

The *KwikLook*<sup>™</sup> Fault Finder is a Windows<sup>®</sup> utility for testing real-time embedded systems developed using KADAK's AMX<sup>™</sup> multitasking kernel. *KwikLook* gives you quick fingertip access to everything controlled by AMX and its managers:

Tasks	Semaphores
Timers (ticks and ms)	Event groups
Mailboxes and message exchanges	Buffer pools
Message contents	Memory pools

All messages queued in mailboxes or message exchanges are visible along with the sender's identification. If no messages are present, *KwikLook* shows the tasks, if any, which are waiting for a message to arrive.

*KwikLook* shows which tasks own resources and which ones are waiting for them. The state of all events in each event group can be viewed complete with a list of the tasks waiting for specific event combinations.

The memory usage display gives a snapshot of current allocation by the Memory Manager. Unexpected fragmentation is readily observable. Similarly, free and used buffers are shown for each buffer pool.

Using *KwikLook*, there are no surprises, no guessing. Disappearing resources can be uncovered. Unexpected task activity is exposed. Even stack overflow and underflow can be observed.

### Method of Use

The *KwikLook* Fault Finder adds true AMX task-awareness to a variety of popular Windows based debuggers. Your target AMX system must be connected to a conventional host PC running Windows. *KwikLook* access to your AMX application is then provided by your task-aware remote debugger, a Windows program running on the host PC.

The *KwikLook* Fault Finder is implemented as a Windows DLL which is attached to your debugger when the debugger is started. *KwikLook* is activated by you using a toolbar icon or pull-down menu item provided by the debugger.

## KwikLook Displays

The *KwikLook* Fault Finder is run from the toolbar icon or pull-down menu item provided by your task-aware debugger. *KwikLook* can only be activated once AMX has been started and your application has been stopped at a debugger breakpoint. Once running, *KwikLook* fetches and displays the state of your AMX application in the display window as illustrated in Figure 1.1-1.

Specific AMX object information is accessed via the radio button selections. The Detail... button can then be used to fetch even more information about the particular AMX object selected from the list. The Summary... button provides access to the System Summary shown in Figure 1.1-2.

At any time, a context sensitive explanation of the display content can be accessed via the Help button. When finished reviewing the state of your AMX application, press the Dismiss button or close the *KwikLook* window to return to the task-aware debugger.

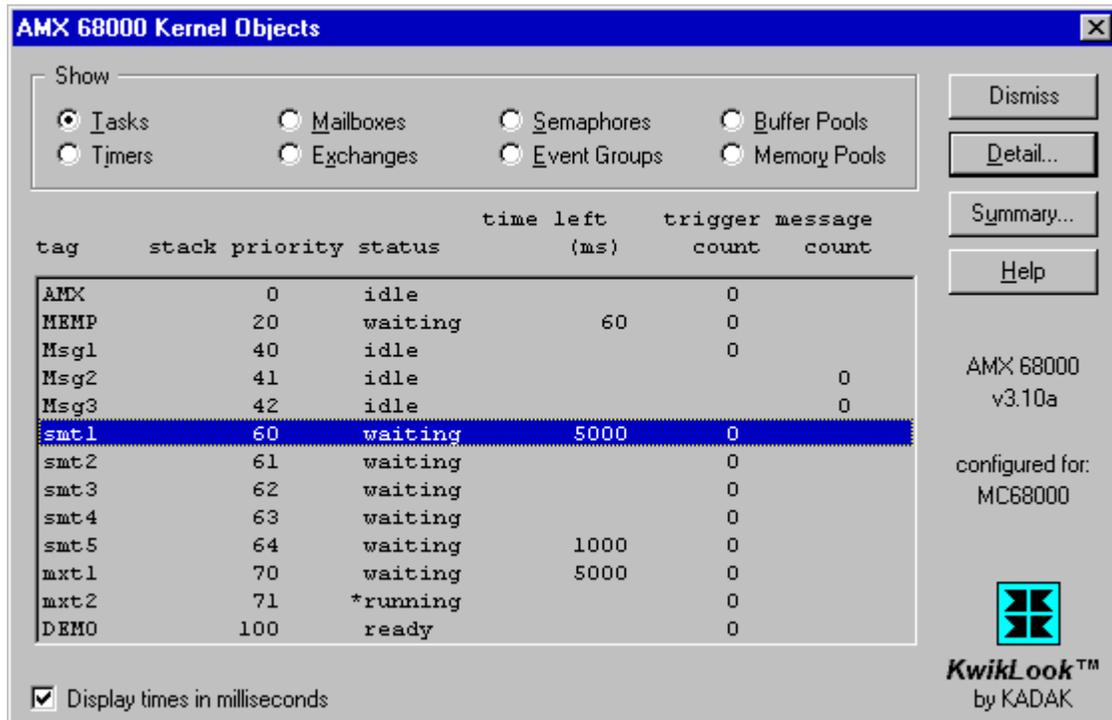


Figure 1.1-1 *KwikLook* Display Window

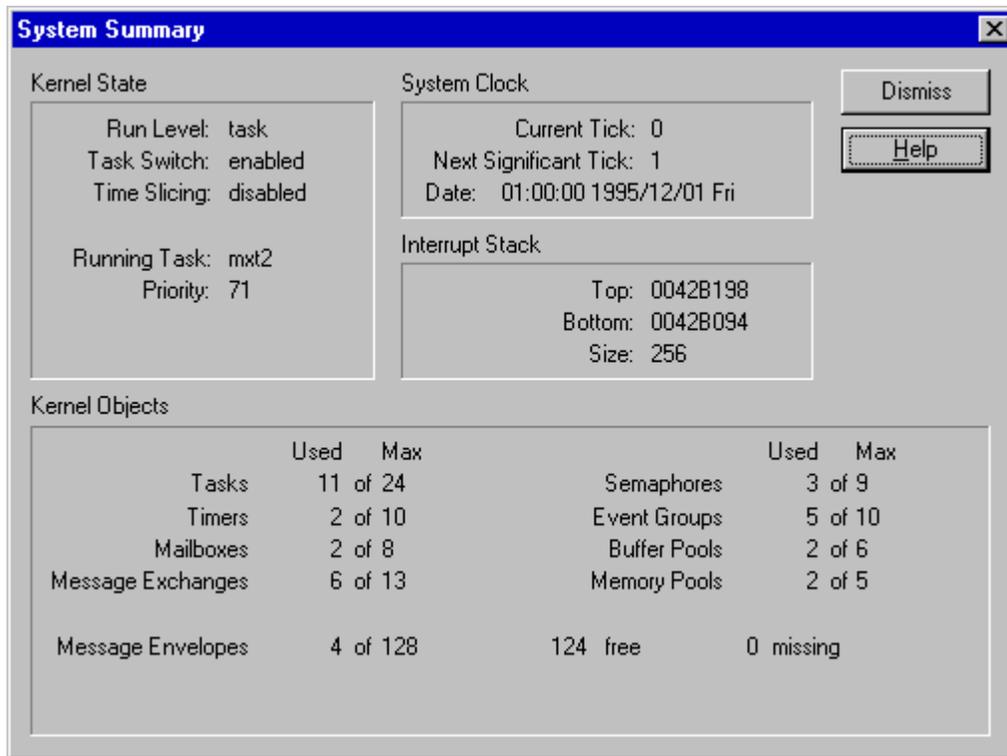


Figure 1.1-2 *KwikLook* System Summary

## 1.2 Installed *KwikLook* Components

*KwikLook* is installed from CD-ROM. The *KwikLook* installer creates a *KwikLook* installation directory into which it copies all *KwikLook* components. *KwikLook* installation directory *KWK302* is used for all 32-bit versions of AMX. *KwikLook* for AMX 86 is installed into directory *KWK840*. Within this manual, any reference to installation directory *KWK302* will apply to *KWK840* unless specifically stated otherwise.

When *KwikLook* is installed for use with a particular task-aware debugger, the following subdirectories will be created in the *KwikLook* installation directory *KWK302*.

<i>HOST</i>	<i>KwikLook</i> DLL and debugger support files
<i>MANUAL</i>	A copy of this manual in Adobe Acrobat <sup>®</sup> PDF format

The *KwikLook* installer will copy a subset of the files from subdirectory *KWK302\HOST* to the task-aware debugger's installation directory if required by the debugger. In some cases, entries will be added to the Windows registry to give the task-aware debugger access to *KwikLook*. In other cases, a debugger specific configuration file may be updated to indicate the presence of *KwikLook*.

The following documentation is always installed. A duplicate of this manual in Adobe Acrobat PDF format is installed in subdirectory *MANUAL*. A separate Windows on-line help manual is installed in subdirectory *HOST*. The help manual provides context sensitive explanations of all of the AMX information presented in the *KwikLook* display window. It also includes guidelines for the proper operation of the particular variant of *KwikLook* being used.

*KwikLook* only supports the task-aware debuggers listed below. When you install *KwikLook*, you will be asked to identify the debugger with which *KwikLook* is to be used.

Each of the debuggers is included with the software development tools from a particular tool vendor. KADAK uses a two character toolset id *xx* to identify these vendors. This is the same mnemonic used to identify the tools with which your AMX application is being developed. The toolset ids which apply to each debugger are listed below.

When *KwikLook* is installed, a *KwikLook* Program Group will be created for quick access via the Windows Start menu to the installed *KwikLook* components. The program group will provide the shortcuts described in Chapter 1.3.

*KwikLook* can be used with the AMX versions and task-aware debuggers listed below.

<b>AMX Part</b>	<b>Target Processor</b>	<b>Toolset Id</b>	<b>Task-Aware Debugger</b>
PN382-1	PowerPC	<i>DA</i>	Diab-SDS SingleStep™
PN382-1	PowerPC	<i>ME</i>	Metrowerks CodeWarrior®
PN382-1	PowerPC	<i>MW</i>	MetaWare SeeCode™
PN402-1	ARM	<i>AR</i>	IAR Systems C-Spy®
PN402-1	ARM	<i>ME</i>	Metrowerks CodeWarrior
PN402-1	ARM	<i>MW</i>	MetaWare SeeCode
PN402-1	ARM	<i>RV</i>	ARM Ltd. RealView®
PN422-1	Thumb	<i>AR</i>	IAR Systems C-Spy
PN422-1	Thumb	<i>ME</i>	Metrowerks CodeWarrior
PN422-1	Thumb	<i>MW</i>	MetaWare SeeCode
PN422-1	Thumb	<i>RV</i>	ARM Ltd. RealView
PN442-1	MIPS32	<i>MW</i>	MetaWare SeeCode
PN512-1	ColdFire	<i>DA</i>	Diab-SDS SingleStep
PN512-1	ColdFire	<i>ME</i>	Metrowerks CodeWarrior
PN532-1	68000	<i>DA</i>	Diab-SDS SingleStep
PN532-1	68000	<i>ME</i>	Metrowerks CodeWarrior
PN722-1	80x86	<i>PD</i>	Paradigm Debugger (protected mode)
PN831-1	80x86/88	<i>PD</i>	Paradigm Debugger (16-bit real mode)
PN831-1	VA-186	<i>PX</i>	Paradigm Debugger (24-bit real mode)

## 1.3 *KwikLook* Program Group

When *KwikLook* is installed, a *KwikLook* Program Group will be created for quick access via the Windows Start menu to the installed *KwikLook* components. The program group will provide the shortcuts described in this chapter.

### Installation Notes

A shortcut called Read Me uses the Windows NotePad utility to open a text file describing the installed *KwikLook* product. This is the same text file which is presented for your viewing at the completion of the installation process.

### Uninstall

A shortcut called Uninstall initiates the removal of the installed *KwikLook* files. Only the files in the *KwikLook* installation directory are deleted. The *KwikLook* files, if any, which are installed in the installation directory of a task-aware debugger are not removed. You can also uninstall *KwikLook* using the Add/Remove Programs service available in the Windows Control Panel.

If you uninstall *KwikLook*, the *KwikLook* installation subdirectory *KWK302* (or *KWK840*) will be deleted, including all files contained therein.

The *KwikLook* files, if any, which are duplicated in the installation directory of your task-aware debugger are not removed. To delete these files so that your debugger installation has no *KwikLook* remnants, follow the instructions in Chapter 2 for the particular debugger that you are using.

### *KwikLook* Help

The *KwikLook* Help shortcut opens the on-line *KwikLook* Help Manual. This is the same manual which is available from within *KwikLook* in a context sensitive manner.

### *KwikLook* User's Guide

This *KwikLook* User's Guide is provided in Adobe PDF format. The *KwikLook* User's Guide shortcut invokes the Adobe Acrobat Reader for on-line viewing of this manual. A copy of the Acrobat Reader is provided with *KwikLook* in case you need it installed.

### Task-Aware Debuggers

When you install *KwikLook* for a task-aware debugger, the program group will usually contain a shortcut to start the debugger for task-aware operation with *KwikLook*. In some cases, additional shortcuts will be created to launch different variants of the debugger for use with a simulator or remote target monitor or with a special BDM or JTAG connection.

## 2. *KwikLook* for Task-Aware Debuggers

### 2.1 Task-Aware Debugging

*KwikLook* shines when used with a source level, task-aware debugger. The host PC is connected to your target system either directly or via an in-circuit emulator using one of the remote links supported by the debugger. The debugger controls your debugging session; *KwikLook* displays the state of your AMX application.

*KwikLook* is available for use with the following task-aware debuggers to test AMX applications developed for the indicated target processor families.

<b>Task-Aware Debugger</b>	<b>Target Processor</b>
ARM Ltd. RealView <sup>®</sup>	ARM, Thumb
IAR Systems C-SPY <sup>®</sup>	ARM, Thumb
Diab-SDS SingleStep <sup>™</sup>	PowerPC, 68000, ColdFire
Metrowerks CodeWarrior <sup>®</sup>	PowerPC, 68000, ColdFire, ARM, Thumb
MetaWare SeeCode <sup>™</sup>	PowerPC, ARM, Thumb, MIPS32
Paradigm 32-Bit Debugger	80x86 in protected mode
Paradigm 16 and 24-Bit Debugger	80x86/88 and VA-186 in real mode

The following in-circuit emulator and debug tool vendors have also created AMX task-aware interfaces to their products based on *KwikLook* technology.

Aisys  
EST  
Lauterbach

The task-aware debugger runs as a Windows application on a host PC. A *KwikLook* Windows DLL is then attached to the debugger when the debugger is started.

The host PC must be connected to your target hardware using any of the methods supported by the debugger. These usually include serial links to a remote target monitor, in-circuit emulator connections and background debug mode (BDM) or JTAG links.

Most debugger vendors also provide a simulator version of their debugger, permitting your AMX application to be tested on the host PC in the absence of real target hardware.

#### Note

*KwikLook* uses services provided by the task-aware debugger to access AMX information via the debugger's connection to the target processor. There is no *KwikLook* component linked with the AMX application being tested.

## 2.2 Trouble Shooting Task-Aware Debuggers

### Debugger Cannot Access *KwikLook*

Most task-aware debuggers must be informed that *KwikLook* is to be used. The debuggers use command line switches, custom initialization files or Windows registry entries to derive this information. If the debugger is started without the relevant directives, *KwikLook* will most likely be inaccessible from the debugger. The debugger toolbar icon or menu item used to start *KwikLook* may be missing or inactive or it may generate an error message. Specific instructions for starting each task-aware debugger are provided in the chapters which follow.

Most task-aware debuggers require the *KwikLook* DLL file, the *KwikLook* help file and, in some cases, other debugger dependent files to be installed in a specific location. If the files are not present in the expected location, the debugger will not be able to provide access to *KwikLook*. A message is usually generated by the debugger indicating that the *KwikLook* files are missing or that *KwikLook* is unavailable. Review the installation requirements in the subsequent chapter describing the task-aware debugger of interest.

### *KwikLook* Cannot Access AMX Information

*KwikLook* requires access via the task-aware debugger to symbolic debug information in your AMX System Configuration Module. You must compile your System Configuration Module with debug information in order for *KwikLook* to find the AMX private data for display. Use the C compiler's debug command line switch as described in the AMX Tool Guide for the toolset which you are using.

If *KwikLook* cannot find the symbolic information it requires, it will generate an error dialogue and await your acknowledgement before returning to the debugger. Be sure that you have actually provided the debugger with an AMX load module to test. Some debuggers may require that you run as far as the *main()* entry point before symbolic information is available.

Also note that until AMX has been launched, there is no AMX application running for *KwikLook* to interrogate. Hence, until AMX is running and all Restart Procedures have been executed, *KwikLook* may have little information to provide to you.

## 2.3 *KwikLook* for MetaWare SeeCode

The MetaWare SeeCode™ Debugger runs as a Windows application on a host PC. The host PC must be connected to your target hardware using any of the methods supported by MetaWare. These include in-circuit emulator connections and links to background debug mode (BDM) or JTAG connectors on the target hardware.

The SeeCode Debugger can also simulate the target processor, permitting your AMX application to be tested in the absence of real target hardware.

*KwikLook* supports SeeCode for testing AMX applications on the following target processors. Unless otherwise directed by you during the installation process, SeeCode and *KwikLook* will be installed in the directories as indicated below.

<b>SeeCode Directory</b>	<b><i>KwikLook</i> Directory</b>	<b>Target Processor</b>
<i>C:\HCPPC</i>	<i>C:\KADAK\KWK302</i>	PowerPC family
<i>C:\HCARM</i>	<i>C:\KADAK\KWK302</i>	ARM v4, v5 (ARM mode)
<i>C:\HCARM</i>	<i>C:\KADAK\KWK302</i>	ARM v4T, v5T (ARM/Thumb mode)
<i>C:\HCMIPS</i>	<i>C:\KADAK\KWK302</i>	MIPS32 family

The following subdirectories will be created in the *KwikLook* installation directory.

<i>HOST</i>	<i>KwikLook</i> DLL and SeeCode support files
<i>MANUAL</i>	A copy of this manual in Adobe Acrobat® PDF format

### ***KwikLook* Integration with SeeCode**

Some of the files from *KwikLook* subdirectory *HOST* must be copied to a subdirectory named *KADAK* in the SeeCode installation directory. The copied files must then be renamed. When *KwikLook* is installed from CD-ROM, the installer will automatically copy and rename these files unless you specifically prohibit integration with SeeCode.

If you do not allow the automatic integration of *KwikLook* with SeeCode, you will have to create the SeeCode *KADAK* subdirectory and manually transfer the files to it for *KwikLook* to operate properly with SeeCode. Copy all files *BJ302.\** from *KwikLook* subdirectory *KWK302\HOST* to your SeeCode *KADAK* subdirectory and rename the files *AMXSI.\**.

### **Starting the SeeCode Debugger**

The installation process will create a shortcut to launch the SeeCode Debugger with the command switch needed to enable task-aware debugging with *KwikLook*. If you wish to create a custom shortcut to start SeeCode, you must start the debugger using one of the following Run commands. Additional command switches may be required to meet your particular needs.

<i>C:\HCPPC\BIN\SCPPC.EXE -kwiklook</i>	PowerPC
<i>C:\HCARM\BIN\SCARM.EXE -kwiklook</i>	ARM or Thumb
<i>C:\HCMIPS\BIN\SCMIPS.EXE -kwiklook</i>	MIPS32

## Debugging with SeeCode

The SeeCode task-aware debugger must hit a breakpoint in your AMX application before you will be able to use *KwikLook* to view the state of your program.

In the SeeCode Source window, find a function in one of your AMX tasks which you know will be encountered shortly after AMX is launched. Double click on the first statement within this function to set a breakpoint there. Then click on the Run button in the SeeCode Command window to launch AMX. SeeCode will break on entry to your function.

Once SeeCode has stopped at the breakpoint, you can use *KwikLook* to explore the state of your system from the AMX perspective. Click on the Tasks button in the Command window to see a list of all available tasks.

To invoke *KwikLook*, click on the AMX menu button in the Command window and select Examine... from the menu. The *KwikLook* window will appear. Use the *KwikLook* buttons to view all of the AMX objects.

Once *KwikLook* is active, you can use the *KwikLook* Help facility for complete explanations of all fields on every *KwikLook* window. Press the F1 key to get help for the current *KwikLook* display.

Close the main *KwikLook* window or click on its Dismiss button to return to the debugger.

To set a breakpoint on some other function, find the function in the SeeCode Source window and double click on the first statement within the function. Then click on the Run button in the SeeCode Command window to resume execution of your AMX application.

You should always run with a breakpoint on AMX procedure *cjksfatal* so that you can detect fatal conditions that have forced AMX to stop execution. You can also place a breakpoint on AMX procedure *cjksbreak* to stop on any AMX procedure which is about to return an error indication to the caller.

## Uninstalling *KwikLook*

*KwikLook* can be uninstalled as described in Chapter 1.3. However, when *KwikLook* is removed, the *KwikLook* files copied into the debugger's installation directory are not removed. To delete those files so that your debugger installation has no *KwikLook* remnants, proceed as follows.

Go to the SeeCode installation directory and delete subdirectory *KADAK* and all of the files which it contains.

## 2.4 *KwikLook* for Diab-SDS SingleStep

The Diab-SDS SingleStep™ Debugger runs as a Windows application on a host PC. The host PC must be connected to your target hardware using any of the methods supported by Diab-SDS. These include serial links to the SDS remote monitor, in-circuit emulator connections and Motorola background debug mode (BDM) links.

The SingleStep Simulator/Debugger can also simulate the target processor, permitting your AMX application to be tested in the absence of real target hardware.

*KwikLook* supports SingleStep for testing AMX applications on the following target processors. Unless otherwise directed by you during the installation process, SingleStep and *KwikLook* will be installed in the directories as indicated below.

<b>SingleStep Directory</b>	<b><i>KwikLook</i> Directory</b>	<b>Target Processor</b>
C:\SDS75	C:\KADAK\KWK302	PowerPC family
C:\SDS762	C:\KADAK\KWK302	ColdFire family
C:\SDS762	C:\KADAK\KWK302	68000 family

The following subdirectories will be created in the *KwikLook* installation directory.

<i>HOST</i>	<i>KwikLook</i> DLL and SingleStep support files
<i>MANUAL</i>	A copy of this manual in Adobe Acrobat® PDF format

### ***KwikLook* Integration with SingleStep**

Some of the files from *KwikLook* subdirectory *HOST* must be copied to the *CMD* subdirectory in the SingleStep installation directory. The copied files must then be renamed. When *KwikLook* is installed from CD-ROM, the installer will automatically copy these files unless you specifically prohibit integration with SingleStep.

If you do not allow the automatic integration of *KwikLook* with SingleStep, you will have to manually transfer the files to subdirectory *CMD*. The copied files must also be renamed as indicated below for *KwikLook* to operate properly with SingleStep.

<b><i>KwikLook</i> Name</b>	<b>SingleStep Name</b>	<b>Target Processor</b>
<i>BJ302.DLL</i>	<i>BJ390.DLL</i>	PowerPC family
<i>BJ302.HLP</i>	<i>BJ390.HLP</i>	
<i>BJ302.DLL</i>	<i>BJ520.DLL</i>	ColdFire family
<i>BJ302.HLP</i>	<i>BJ520.HLP</i>	
<i>BJ302.DLL</i>	<i>BJ540.DLL</i>	68000 family
<i>BJ302.HLP</i>	<i>BJ540.HLP</i>	

## Starting the SingleStep Debugger

The installation process will create shortcuts to launch each of the available SingleStep Debuggers with the command switch needed to enable task-aware debugging with *KwikLook*. The *KwikLook* installer looks for each of the SingleStep debuggers (target monitor, simulator or BDM) in the SingleStep command directory. For each one which it finds, it adds a shortcut to allow that debugger to be started with *KwikLook* capability.

If you wish to create a custom shortcut to start SingleStep, you must start the debugger using one of the following Run commands. Additional command switches may be required to meet your particular needs.

<code>C:\SDS75\CMD\MONPPC.EXE -k BJ390</code>	PowerPC Target Monitor
<code>C:\SDS75\CMD\SIMPPC.EXE -k BJ390</code>	PowerPC Simulator
<code>C:\SDS75\CMD\BDM403.EXE -k BJ390</code>	PPC4xx BDM
<code>C:\SDS75\CMD\MONP58.EXE -k BJ390</code>	MPC5xx BDM
<code>C:\SDS75\CMD\MONP58.EXE -A bdm8xx -k BJ390</code>	MPC8xx BDM
<code>C:\SDS762\CMD\MON68K.EXE -A MONMCF -k BJ520</code>	ColdFire Target Monitor
<code>C:\SDS762\CMD\BDMMCF.EXE -k BJ520</code>	ColdFire BDM
<code>C:\SDS762\CMD\MON68K.EXE -k BJ540</code>	68xxx Target Monitor
<code>C:\SDS762\CMD\SIM68K.EXE -k BJ540</code>	68xxx Simulator
<code>C:\SDS762\CMD\BDM68K.EXE -k BJ540</code>	68xxx BDM

## Debugging with SingleStep

The SingleStep task-aware debugger must hit a breakpoint in your AMX application before you will be able to use *KwikLook* to view the state of your program.

Click on the SingleStep toolbar FUNCTIONS button. In the FUNCTIONS window, find the name of a function in one of your AMX tasks which you know will be encountered shortly after AMX is launched. Select the function to set a breakpoint there. Then click on the GO UNTIL button to launch AMX. SingleStep will break on entry to your function.

The status bar shows the task in which you are stopped. The TASKS window will show all available tasks. To determine where a preempted task is stopped, click on the task's name in the TASKS window and then click the SHOW button.

Once SingleStep has stopped at the breakpoint, you can use *KwikLook* to explore the state of your system from the AMX perspective. Start *KwikLook* by clicking on the KERNEL OBJECTS menu item under the main SingleStep DATA menu. The *KwikLook* window will appear. Use the *KwikLook* buttons to view all of the AMX objects.

Once *KwikLook* is active, you can use the *KwikLook* Help facility for complete explanations of all fields on every *KwikLook* window. Press the F1 key to get help for the current *KwikLook* display.

Close the main *KwikLook* window or click on its Dismiss button to return to the debugger.

To set a breakpoint on some other function, click on the toolbar FUNCTIONS button, select the function and click on the BREAK button. Then allow SingleStep to resume execution of your AMX application.

You should always run with a breakpoint on AMX procedure *cjksfatal* so that you can detect fatal conditions that have forced AMX to stop execution. You can also place a breakpoint on AMX procedure *cjksbreak* to stop on any AMX procedure which is about to return an error indication to the caller.

## Uninstalling *KwikLook*

*KwikLook* can be uninstalled as described in Chapter 1.3. However, when *KwikLook* is removed, the *KwikLook* files copied into the debugger's installation directory are not removed. To delete those files so that your debugger installation has no *KwikLook* remnants, proceed as follows.

Go to subdirectory *CMD* in the SingleStep installation directory.

Delete files *BJ390.\** to remove *KwikLook* support for the PowerPC family.

Delete files *BJ520.\** to remove *KwikLook* support for the ColdFire family.

Delete files *BJ540.\** to remove *KwikLook* support for the 68000 family.

This page left blank intentionally.

## 2.5 *KwikLook* for Paradigm 16-Bit Debugger

The Paradigm Debugger is a component of the Paradigm Integrated Development Environment (IDE), a Windows application running on a host PC. The host PC must be connected to your target hardware using any of the methods supported by Paradigm. These include serial links to the Paradigm remote target monitor and in-circuit emulator connections to the target hardware.

*KwikLook* supports the Paradigm Debugger for testing AMX 86 applications on the following target processors. Unless otherwise directed by you during the installation process, the Paradigm Debugger and *KwikLook* will be installed in the directories as indicated below.

<b>Paradigm Directory</b>	<b><i>KwikLook</i> Directory</b>	<b>Target Processor</b>
<i>C:\PARADIGM</i>	<i>C:\KADAK\KWK840</i>	Intel 16-bit 80x86/88 family or VAutomation 24-bit Turbo86 family (operating in real mode)

The following subdirectories will be created in the *KwikLook* installation directory.

<i>HOST</i>	<i>KwikLook</i> DLL and Paradigm Debugger support files
<i>MANUAL</i>	A copy of this manual in Adobe Acrobat® PDF format

### ***KwikLook* Integration with the Paradigm Debugger**

When *KwikLook* is installed from CD-ROM, the installer automatically finds the Paradigm Debugger from its entry in the Windows registry. Information is then recorded in the Paradigm registry entry to allow the Paradigm Debugger to directly access the files in the *KwikLook* subdirectory *HOST*.

If the *KwikLook* installer cannot update the Paradigm registry entry, you will have to use the Windows registry editor *REGEDIT.EXE* to add the necessary information to allow *KwikLook* to operate properly with the Paradigm Debugger.

Open the following registry entry:

*HKEY\_LOCAL\_MACHINE\Software\Paradigm\Paradigm C++\6.0\RTOS*

Add the following key to the "*RTOS*" list of RTOS names:

*KADAK AMX 86*

Add the following value name and data to the "*KADAK AMX 86*" description:

*RTOSd11 "C:\KADAK\KWK840\HOST\BJ840.DLL"*

The text string "*KADAK AMX 86*" will be used by the Paradigm IDE to identify AMX 86 as an RTOS for which task-awareness is available. The registry value named *RTOSd11* provides the Paradigm Debugger with the path and filename of the *KwikLook* DLL. Be sure to use the full path to the directory in which you installed *KwikLook* for AMX 86

## Starting the Paradigm Debugger

The Paradigm Debugger must be made AMX task-aware. From the Options menu, select Environment... and choose Debugger | Third Party RTOS as shown in Figure 2.5-1. Select KADAK AMX 86 from the available list and press OK.

Since the Paradigm Debugger operates as a component of the Paradigm IDE, it must be started from within the IDE. It cannot be started using a Windows shortcut.

You can add a speed button to the Paradigm toolbar to provide easy access to *KwikLook*. From the Options menu, select Environment... and choose SpeedBar | Customize. From the list of Available buttons, select the View Third Party RTOS button to add it to the list of Active buttons. The button will appear on the toolbar but will be dimmed unless the Paradigm Debugger is active.

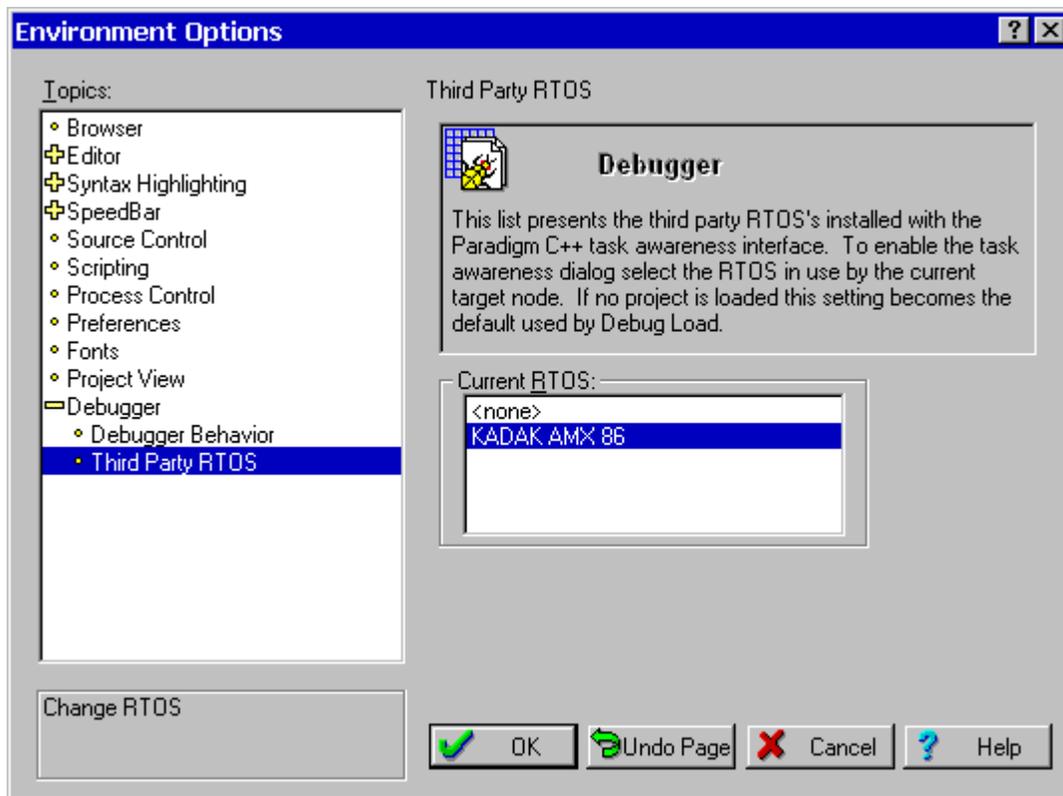


Figure 2.5-1 Paradigm 16-Bit Environment Options

## Debugging with the Paradigm Debugger

The Paradigm task-aware debugger must hit a breakpoint in your AMX application before you will be able to use *KwikLook* to view the state of your program.

In the Paradigm CPU window, find the name of a function in one of your AMX tasks which you know will be encountered shortly after AMX is launched. Click on the first statement within this function. Click on the Toggle Breakpoint button on the toolbar to set a breakpoint on the function. Then click on the Run button to launch AMX. The Paradigm Debugger will break on entry to your function.

Once the Paradigm Debugger has stopped at the breakpoint, you can use *KwikLook* to explore the state of your system from the AMX perspective. If you have not already done so, add an RTOS speed button to the debugger's toolbar as described previously in this chapter. Start *KwikLook* by clicking on the RTOS speed button. The *KwikLook* window will appear. Use the *KwikLook* buttons to view all of the AMX objects.

Once *KwikLook* is active, you can use the *KwikLook* Help facility for complete explanations of all fields on every *KwikLook* window. Press the F1 key to get help for the current *KwikLook* display.

Close the main *KwikLook* window or click on its Dismiss button to return to the debugger.

To set a breakpoint on some other function, go to the Paradigm Debugger CPU window and find the function. Then click on the Toggle Breakpoint button on the toolbar to set a breakpoint on the function. Then allow the debugger to resume execution of your AMX application.

You should always run with a breakpoint on AMX procedure *cjksfatal* so that you can detect fatal conditions that have forced AMX to stop execution. You can also place a breakpoint on AMX procedure *cjksbreak* to stop on any AMX procedure which is about to return an error indication to the caller.

## Uninstalling *KwikLook*

*KwikLook* can be uninstalled as described in Chapter 1.3. However, when *KwikLook* is removed, the *KwikLook* information is not deleted from the Paradigm Debugger's entry in the Windows registry. To edit the registry entry so that your debugger installation has no *KwikLook* remnants, proceed as follows.

Using the Windows registry editor *REGEDIT.EXE*, open the following registry entry:

`HKEY_LOCAL_MACHINE\Software\Paradigm\Paradigm C++\6.0\RTOS`

Delete the following key from the "RTOS" list of RTOS names:

`KADAK AMX 86`

This page left blank intentionally.

## 2.6 *KwikLook* for Paradigm 32-Bit Debugger

The Paradigm Debugger is a component of the Paradigm Integrated Development Environment (IDE), a Windows application running on a host PC. The host PC must be connected to your target hardware using any of the methods supported by Paradigm. These include serial links to the Paradigm remote target monitor and in-circuit emulator connections to the target hardware.

*KwikLook* supports the Paradigm Debugger for testing AMX 386/ET applications on the following target processors. Unless otherwise directed by you during the installation process, the Paradigm Debugger and *KwikLook* will be installed in the directories as indicated below.

<b>Paradigm Directory</b>	<b><i>KwikLook</i> Directory</b>	<b>Target Processor</b>
<i>C:\PARADIGM</i>	<i>C:\KADAK\KWK302</i>	Intel 80x86 family (operating in protected mode)

The following subdirectories will be created in the *KwikLook* installation directory.

<i>HOST</i>	<i>KwikLook</i> DLL and Paradigm Debugger support files
<i>MANUAL</i>	A copy of this manual in Adobe Acrobat® PDF format

### ***KwikLook* Integration with the Paradigm Debugger**

When *KwikLook* is installed from CD-ROM, the installer automatically finds the Paradigm Debugger from its entry in the Windows registry. Information is then recorded in the Paradigm registry entry to allow the Paradigm Debugger to directly access the files in the *KwikLook* subdirectory *HOST*.

If the *KwikLook* installer cannot update the Paradigm registry entry, you will have to use the Windows registry editor *REGEDIT.EXE* to add the necessary information to allow *KwikLook* to operate properly with the Paradigm Debugger.

Open the following registry entry:

*HKEY\_LOCAL\_MACHINE\Software\Paradigm\Paradigm C++\6.0\RTOS*

Add the following key to the "*RTOS*" list of RTOS names:

*KADAK AMX 386ET*

Add the following value name and data to the "*KADAK AMX 386ET*" description:

*RTOSd11 "C:\KADAK\KWK302\HOST\BJ302.DLL"*

The text string "*KADAK AMX 386ET*" will be used by the Paradigm IDE to identify AMX 386/ET as an RTOS for which task-awareness is available. The registry value named *RTOSd11* provides the Paradigm Debugger with the path and filename of the *KwikLook* DLL. Be sure to use the full path to the directory in which you installed *KwikLook*.

## Starting the Paradigm Debugger

The Paradigm Debugger must be made AMX task-aware. From the Options menu, select Environment... and choose Debugger | Third Party RTOS as shown in Figure 2.6-1. Select KADAK AMX 386ET from the available list and press OK.

Since the Paradigm Debugger operates as a component of the Paradigm IDE, it must be started from within the IDE. It cannot be started using a Windows shortcut.

You can add a speed button to the Paradigm toolbar to provide easy access to *KwikLook*. From the Options menu, select Environment... and choose SpeedBar | Customize. From the list of Available buttons, select the View Third Party RTOS button to add it to the list of Active buttons. The button will appear on the toolbar but will be dimmed unless the Paradigm Debugger is active.

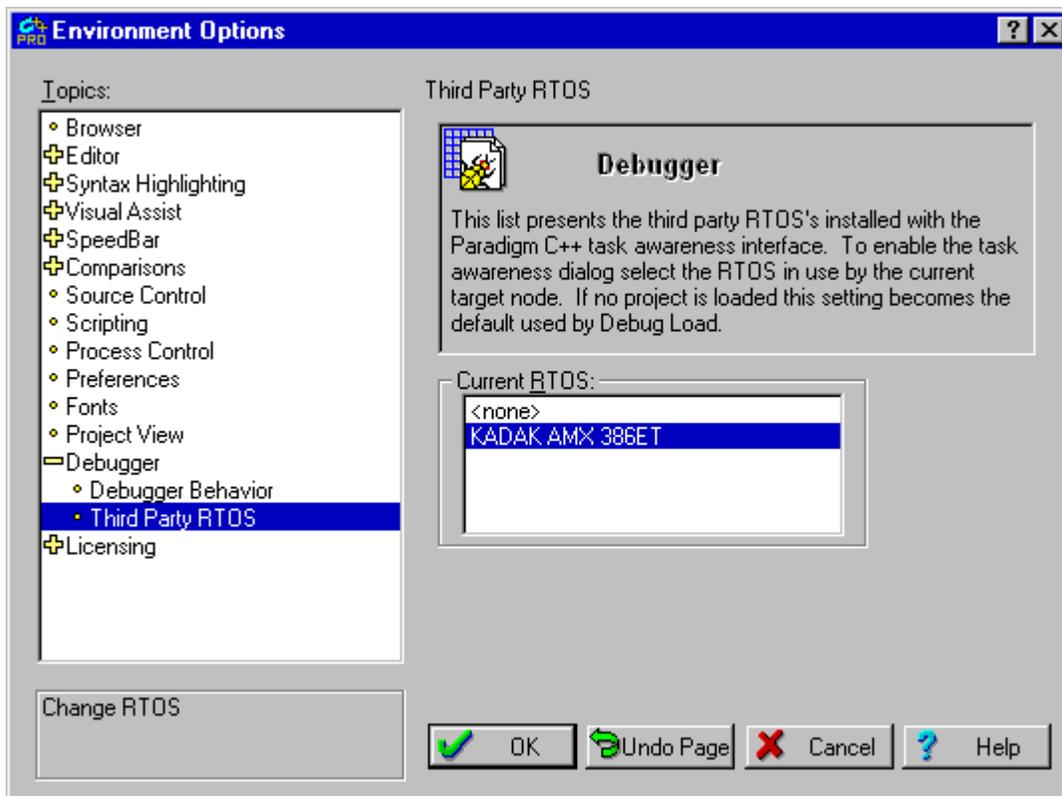


Figure 2.6-1 Paradigm 32-Bit Environment Options

## Debugging with the Paradigm Debugger

The Paradigm task-aware debugger must hit a breakpoint in your AMX application before you will be able to use *KwikLook* to view the state of your program.

In the Paradigm CPU window, find the name of a function in one of your AMX tasks which you know will be encountered shortly after AMX is launched. Click on the first statement within this function. Click on the Toggle Breakpoint button on the toolbar to set a breakpoint on the function. Then click on the Run button to launch AMX. The Paradigm Debugger will break on entry to your function.

Once the Paradigm Debugger has stopped at the breakpoint, you can use *KwikLook* to explore the state of your system from the AMX perspective. If you have not already done so, add an RTOS speed button to the debugger's toolbar as described previously in this chapter. Start *KwikLook* by clicking on the RTOS speed button. The *KwikLook* window will appear. Use the *KwikLook* buttons to view all of the AMX objects.

Once *KwikLook* is active, you can use the *KwikLook* Help facility for complete explanations of all fields on every *KwikLook* window. Press the F1 key to get help for the current *KwikLook* display.

Close the main *KwikLook* window or click on its Dismiss button to return to the debugger.

To set a breakpoint on some other function, go to the Paradigm Debugger CPU window and find the function. Then click on the Toggle Breakpoint button on the toolbar to set a breakpoint on the function. Then allow the debugger to resume execution of your AMX application.

You should always run with a breakpoint on AMX procedure *cjksfatal* so that you can detect fatal conditions that have forced AMX to stop execution. You can also place a breakpoint on AMX procedure *cjksbreak* to stop on any AMX procedure which is about to return an error indication to the caller.

## Uninstalling *KwikLook*

*KwikLook* can be uninstalled as described in Chapter 1.3. However, when *KwikLook* is removed, the *KwikLook* information is not deleted from the Paradigm Debugger's entry in the Windows registry. To edit the registry entry so that your debugger installation has no *KwikLook* remnants, proceed as follows.

Using the Windows registry editor *REGEDIT.EXE*, open the following registry entry:

`HKEY_LOCAL_MACHINE\Software\Paradigm\Paradigm C++\6.0\RTOS`

Delete the following key from the "RTOS" list of RTOS names:

`KADAK AMX 386ET`

This page left blank intentionally.

## 2.7 KwikLook for Metrowerks CodeWarrior

The Metrowerks CodeWarrior® Debugger runs as a Windows application on a host PC. The host PC must be connected to your target hardware using any of the methods supported by Metrowerks. These include in-circuit emulator connections and links to background debug mode (BDM) or JTAG connectors on the target hardware.

KwikLook supports CodeWarrior for testing AMX applications on the following target processors. It is assumed that during the installation process, you indicated that CodeWarrior was to be installed in the directories listed below. Also, unless otherwise directed by you during the installation process, KwikLook will be installed in the directories indicated below.

CodeWarrior Directory	KwikLook Directory	Target Processor
C:\CW_PPC	C:\KADAK\KWK302	PowerPC family
C:\CW_CF	C:\KADAK\KWK302	ColdFire family
C:\CW_68K	C:\KADAK\KWK302	68000 family
C:\CW_ARM	C:\KADAK\KWK302	ARM and Thumb families

The following subdirectories will be created in the KwikLook installation directory.

HOST	KwikLook DLL and CodeWarrior support files
MANUAL	A copy of this manual in Adobe Acrobat® PDF format

### KwikLook Integration with CodeWarrior

Some of the files from KwikLook subdirectory *HOST* must be copied to the subdirectory named *Bin\plugins\debugger\RTOS* in the CodeWarrior installation directory. The copied files must then be renamed. When KwikLook is installed from CD-ROM, the installer will automatically copy and rename these files unless you specifically prohibit integration with CodeWarrior.

If you do not allow the automatic integration of KwikLook with CodeWarrior, you will have to manually transfer the files to subdirectory *Bin\plugins\debugger\RTOS*. The copied files must also be renamed as indicated below for KwikLook to operate properly with CodeWarrior. Note that the name *E68Krtos\_AMX* is used for the 68000 processor family and also for versions of CodeWarrior for ColdFire prior to release 4.

KwikLook Name	CodeWarrior Name	Target Processor
BJ302.DLL	EPPCrtos_AMX.dll	PowerPC family (CodeWarrior v8.5 and prior)
BJ302.HLP	EPPCrtos_AMX.hlp	
BJ302.DLL	CFrtos_AMX.dll	ColdFire family (CodeWarrior v4.0, v5.1, v6.1)
BJ302.HLP	CFrtos_AMX.hlp	
BJ302.DLL	E68Krtos_AMX.dll	ColdFire family (CodeWarrior prior to v4.0)
BJ302.HLP	E68Krtos_AMX.hlp	
BJ302.DLL	E68Krtos_AMX.dll	68000 family (CodeWarrior v3.2)
BJ302.HLP	E68Krtos_AMX.hlp	
BJ302.DLL	ARMrtos_AMX.dll	ARM and Thumb families (CodeWarrior v2.1.1)
BJ302.HLP	ARMrtos_AMX.hlp	

## Starting the CodeWarrior Debugger

The CodeWarrior Debugger must be made AMX task-aware for each AMX project. Open an AMX project and select the target configuration to be debugged. From the Edit menu, select CfgName Settings... where CfgName is your target configuration name. In the illustration below, the configuration name is Default Project.

From the leftmost pane of Target Settings Panels, choose Debugger | Exxx Debugger Settings as shown in Figure 2.7-1. The string Exxx Debugger identifies the variant of the CodeWarrior Debugger being used. Exxx is EPPC for the PowerPC family, E68K for the 68000 family, CF for the ColdFire family, or ARM for the ARM and Thumb families.

Finally, in the Target OS drop-down list, select AMX and press OK to make CodeWarrior AMX aware.

Since the CodeWarrior Debugger operates as a component of the CodeWarrior IDE, it must be started from within the IDE. It cannot be started using a Windows shortcut.

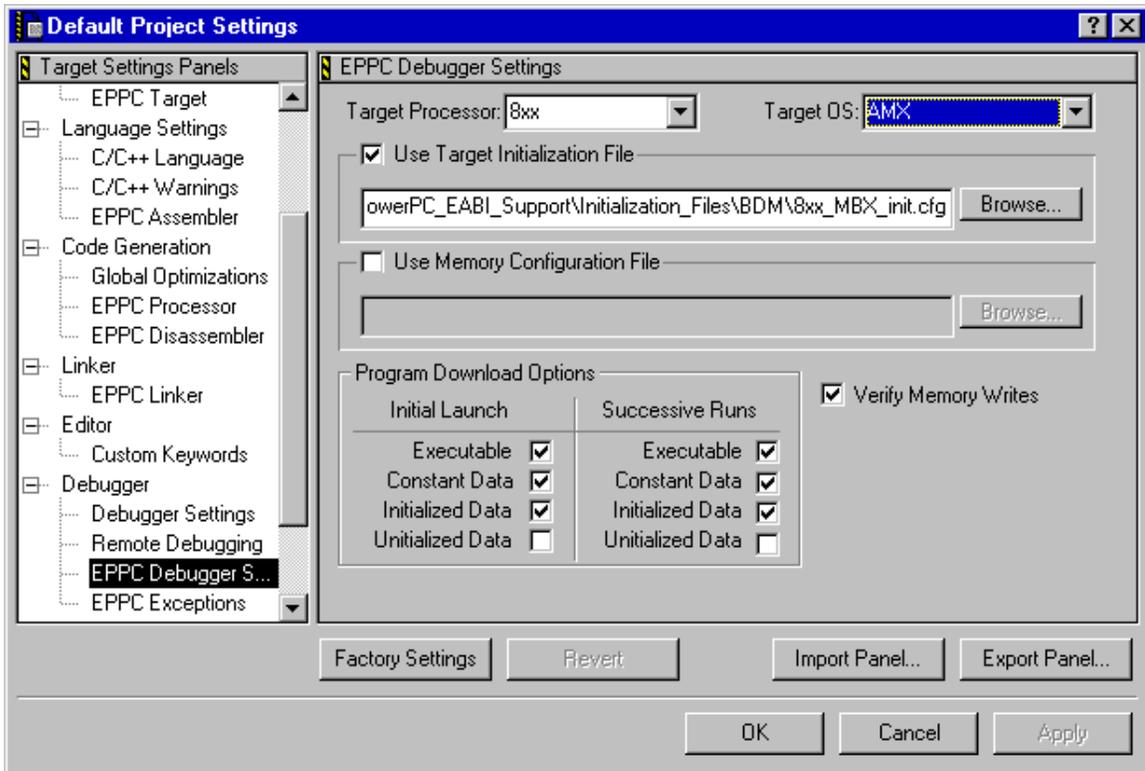


Figure 2.7-1 CodeWarrior Target Debugger Settings

## Debugging with CodeWarrior

The CodeWarrior task-aware debugger must hit a breakpoint in your AMX application before you will be able to use *KwikLook* to view the state of your program.

In CodeWarrior, open the source file which contains a function in one of your AMX tasks which you know will be encountered shortly after AMX is launched. Click on the dash in the Breakpoint Column next to the first statement within this function to set a breakpoint there. Then click on the Run button on the CodeWarrior toolbar to launch AMX. CodeWarrior will break on entry to your function.

Once CodeWarrior has stopped at the breakpoint, you can use *KwikLook* to explore the state of your system from the AMX perspective. Click on the CodeWarrior View menu button and select Processes from the menu to see a list of all available tasks.

To invoke *KwikLook*, click on the AMX menu button and select Examine... from the menu. The *KwikLook* window will appear. Use the *KwikLook* buttons to view all of the AMX objects.

Once *KwikLook* is active, you can use the *KwikLook* Help facility for complete explanations of all fields on every *KwikLook* window. Press the F1 key to get help for the current *KwikLook* display.

Close the main *KwikLook* window or click on its Dismiss button to return to the debugger.

To set a breakpoint on some other function, find the function in its source window and click on the dash in the Breakpoint Column next to the body of the function. Then click on the Run button on the CodeWarrior toolbar to resume execution of your AMX application.

You should always run with a breakpoint on AMX procedure *cjksfatal* so that you can detect fatal conditions that have forced AMX to stop execution. You can also place a breakpoint on AMX procedure *cjksbreak* to stop on any AMX procedure which is about to return an error indication to the caller.

## Uninstalling *KwikLook*

*KwikLook* can be uninstalled as described in Chapter 1.3. However, when *KwikLook* is removed, the *KwikLook* files copied into the debugger's installation directory are not removed. To delete those files so that your debugger installation has no *KwikLook* remnants, proceed as follows.

Go to subdirectory *Bin\plugins\debugger\RTOS* in the CodeWarrior installation directory.

Delete files *EPPCrtos\_AMX.\** to remove *KwikLook* support for the PowerPC family.

Delete files *CFrtos\_AMX.\** to remove *KwikLook* support for the ColdFire family.

Delete files *E68Krtos\_AMX.\** to remove *KwikLook* support for versions of CodeWarrior for ColdFire prior to release 4.

Delete files *E68Krtos\_AMX.\** to remove *KwikLook* support for the 68000 family.

Delete files *ARMrtos\_AMX.\** to remove *KwikLook* support for the ARM and Thumb families.

Note that the CodeWarrior name *E68Krtos\_AMX* is used for the 68000 family and for versions of CodeWarrior for ColdFire prior to release 4. Hence, do not delete these files unless it is your intention to remove support for both of these processor families.

This page left blank intentionally.

## 2.8 *KwikLook* for ARM Ltd. RealView

The ARM RealView<sup>®</sup> Debugger runs as a Windows application on a host PC. The host PC must be connected to your target hardware using any of the methods supported by ARM Ltd. These include in-circuit emulator connections and links to JTAG connectors on the target hardware.

*KwikLook* supports RealView for testing AMX applications on the following target processors. It is assumed that during the installation process, you indicated that RealView was to be installed in the directories listed below. Also, unless otherwise directed by you during the installation process, *KwikLook* will be installed in the directories indicated below.

<b>RealView Directory</b>	<b><i>KwikLook</i> Directory</b>	<b>Target Processor</b>
<i>C:\Program Files\ARM</i>	<i>C:\KADAK\KWK302</i>	ARM or Thumb family

The following subdirectories will be created in the *KwikLook* installation directory.

<i>HOST</i>	<i>KwikLook</i> DLL and RealView support files
<i>MANUAL</i>	A copy of this manual in Adobe Acrobat <sup>®</sup> PDF format

### ***KwikLook* Integration with RealView**

Some of the files from *KwikLook* subdirectory *HOST* must be copied to subdirectories in the RealView Debugger installation subdirectory. The RealView Debugger installation subdirectory name is version dependent and of the following form:

*RVD\Core\<version>\<build>\win\_32-pentium..*

<b>RealView Version</b>	<b>RealView Installation Directory</b>
1.6.1	<i>C:\RealView\RVD\Core\1.6.1\95\win_32-pentium</i>
1.7	<i>C:\RealView\RVD\Core\1.7\150\win_32-pentium</i>
1.8	<i>C:\RealView\RVD\Core\1.8\734\win_32-pentium</i>
3.1	<i>C:\RealView\RVD\Core\3.1\881\win_32-pentium</i>

Once copied, the *KwikLook* files must be renamed. When *KwikLook* is installed from CD-ROM, the installer will automatically copy and rename these files unless you specifically prohibit integration with RealView.

If you do not allow the automatic integration of *KwikLook* with RealView, you will have to manually transfer the files to the RealView Debugger subdirectories. The copied files must also be renamed as indicated below for *KwikLook* to operate properly with RealView.

<b>KwikLook Name</b>	<b>RealView Name</b>	<b>RealView Subdirectory</b>
<i>BJ302.DLL</i>	<i>o2_amx00.dll</i>	<i>lib</i>
<i>BJ302.DLL</i>	<i>e1_amxkl.dll</i>	<i>lib\custom</i>
<i>BJ302.HLP</i>	<i>e1_amxkl.hlp</i>	<i>lib\custom</i>
<i>BJ302.BCD</i>	<i>amx_rtos.bcd</i>	<i>etc</i>

For RealView Debugger v3.1 and higher, it may be necessary to copy RealView file *w1\_ARMEX.dll* from one directory to another. Check RealView subdirectory *lib\custom* for the presence of file *w1\_ARMEX.dll*. If the file is not present, copy *w1\_ARMEX.dll* from subdirectory *lib\extensions* to *lib\custom*.

### Starting the RealView Debugger (versions prior to 3.1)

The RealView Debugger must be made AMX task-aware for each target connection. Start the RealView Debugger and open the Connection Control window. Select the target connection that is to be made AMX aware and right click that connection. From the pop-up menu, select the Connection Properties... item. Figure 2.8-1 illustrates this process for the ARMulator connection.

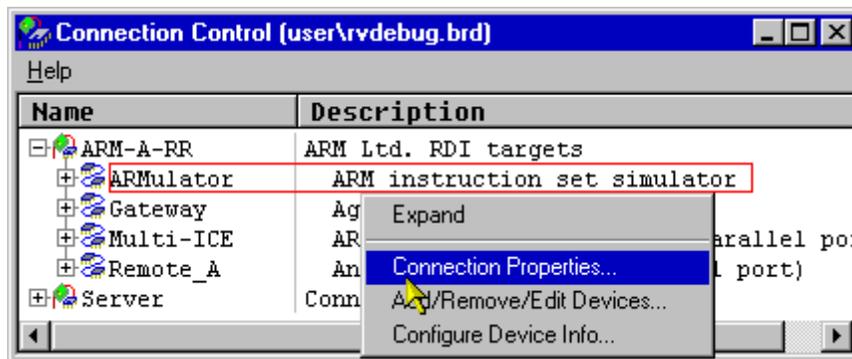


Figure 2.8-1 RealView Target Connection Control

In the Connection Properties window, add a BoardChip name and give it the value *AMX\_rtos*, as shown in Figure 2.8-2. Close the Connection Properties window, making sure that you save the new configuration.

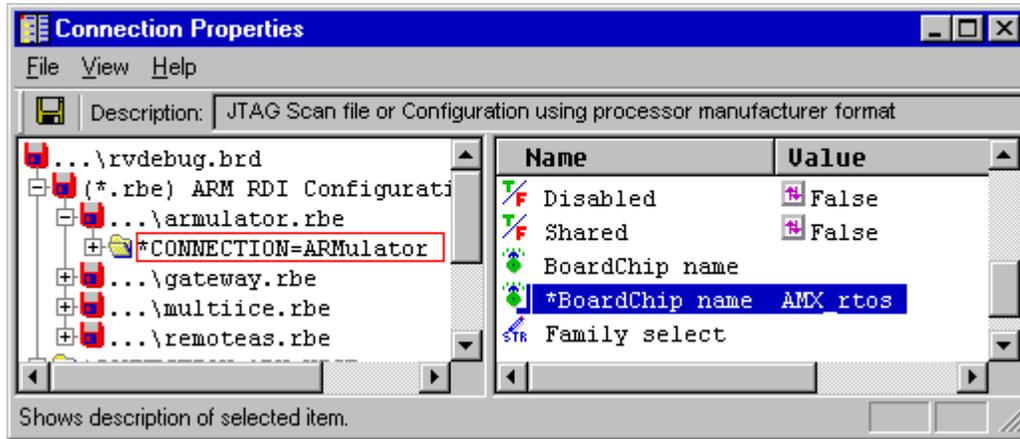


Figure 2.8-2 RealView Connection Properties

Finally, open the AMX aware target connection from the Connection Control window. You can then use the RealView Resource Viewer window to view the AMX resource lists as shown in Figure 2.8-3. You can find more detailed instructions for controlling RealView's RTOS support in Chapter 3 of the RealView Debugger Extensions Guide.

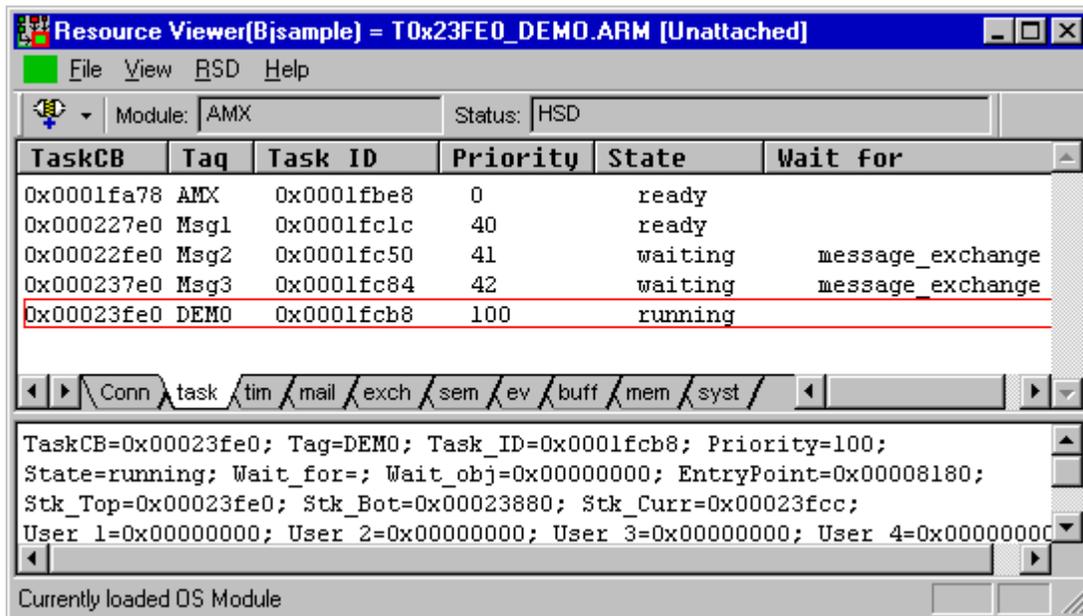


Figure 2.8-3 RealView Resource Viewer

To enable the *KwikLook* for RealView plug-in, activate the RealView Custom Plug-In List window and select the *KwikLook* plug-in, *e1\_amxkl.dll*, as shown in Figure 2.8-4. Click the Start button to activate the plug-in. Then click the Add Connection button to select the target connections which the *KwikLook* plug-in can access. You should then see the *KwikLook* for RealView plug-in window similar to the one shown in Figure 2.8-5.

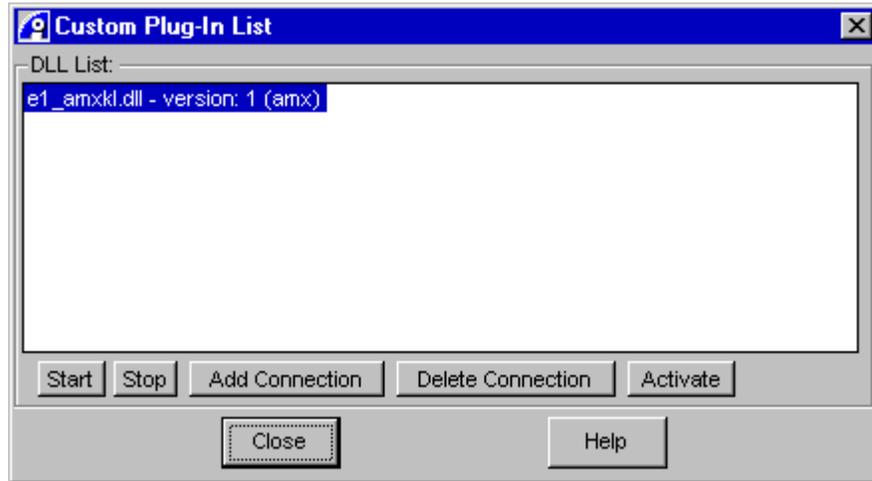


Figure 2.8-4 RealView Plug-In List Control



Figure 2.8-5 RealView's *KwikLook* Plug-In Window

### Starting the RealView Debugger (version 3.1 and higher)

The RealView Debugger v3.1 must be made AMX task-aware for each target connection. Start the RealView Debugger and click Connect to Target... from the Target menu to open the Connect to Target window. Select the target connection that is to be made AMX aware and right click that connection. From the pop-up menu, select the Properties item. Figure 2.8-6 illustrates this process for the RealView Instruction Set Simulator (RVISS) connection.

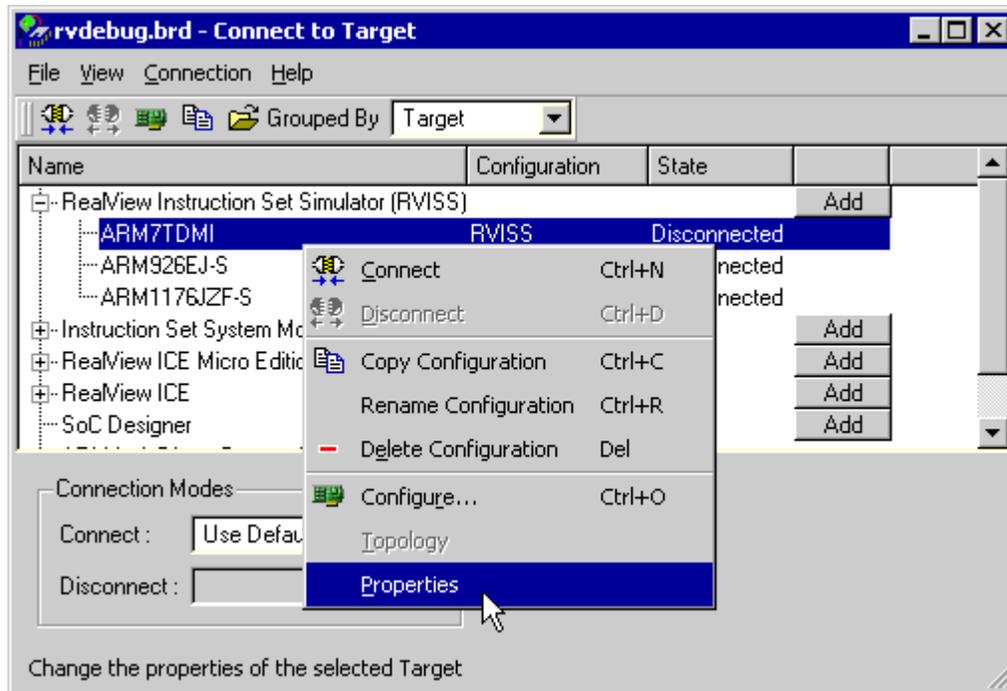


Figure 2.8-6 RealView 3.1 Connect to Target

In the Connection Properties window, add a BoardChip name and give it the value *AMX\_rtos*, as shown in Figure 2.8-7. Close the Connection Properties window, making sure that you save the new configuration.

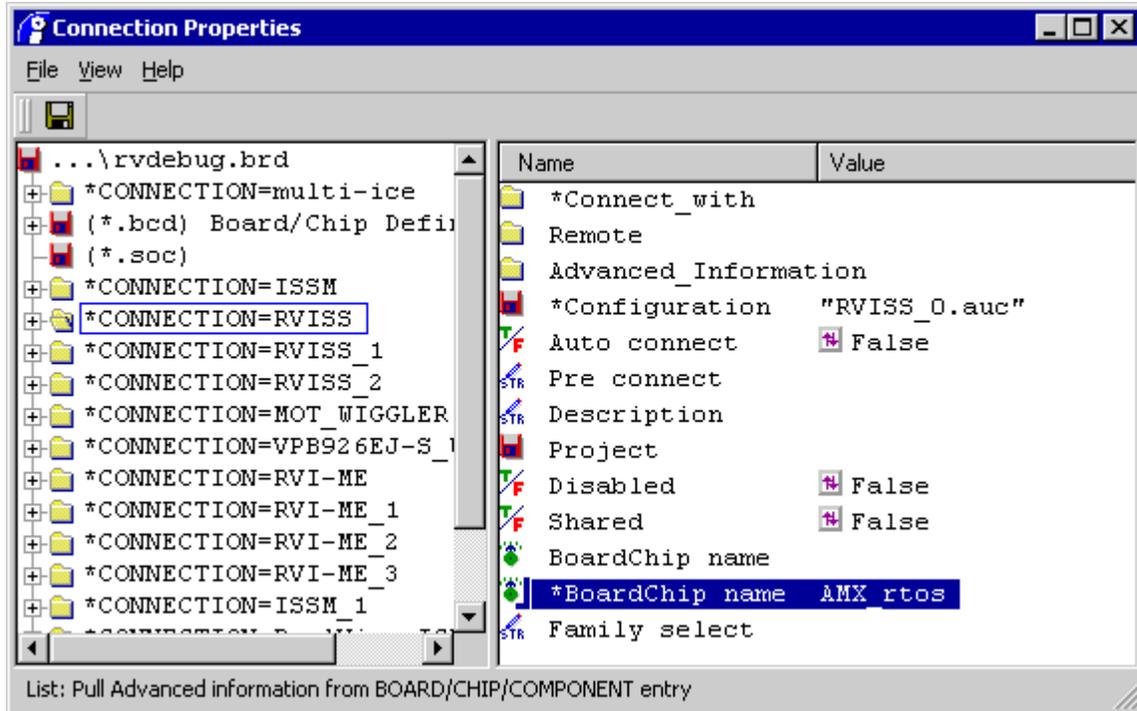


Figure 2.8-7 RealView 3.1 Connection Properties

Finally, click Connect from the Connection menu of the Connect to Target window to open the AMX aware target connection. You can then use the RealView Resource Viewer window to view the AMX resource lists as shown in Figure 2.8-8. You can find more detailed instructions for controlling RealView's RTOS support in Chapter 3 of the RealView Debugger Extensions Guide.

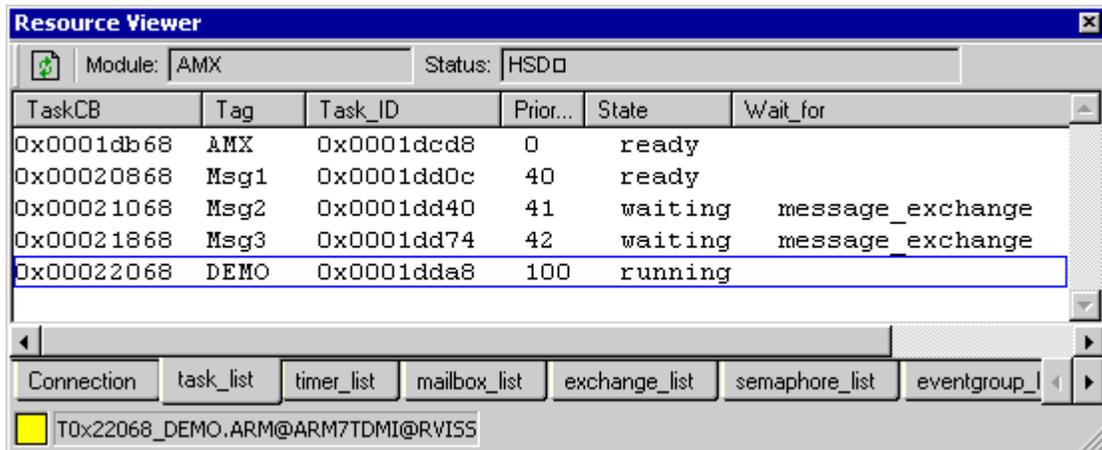


Figure 2.8-8 RealView 3.1 Resource Viewer

To enable the *KwikLook* for RealView plug-in, activate the RealView Custom Plug-In List window and select the *KwikLook* plug-in, *e1\_amxkl.dll*, as shown in Figure 2.8-9. If the RealView Custom Plug-In List window does not open automatically when the RealView debugger is started, check that there are no files missing from the RealView *lib\custom* subdirectory (see integration description at beginning of this chapter).

Click the Start button to activate the plug-in. Then click the Add Connection button to select the target connections which the *KwikLook* plug-in can access. You should then see the *KwikLook* for RealView plug-in window similar to the one shown in Figure 2.8-10.

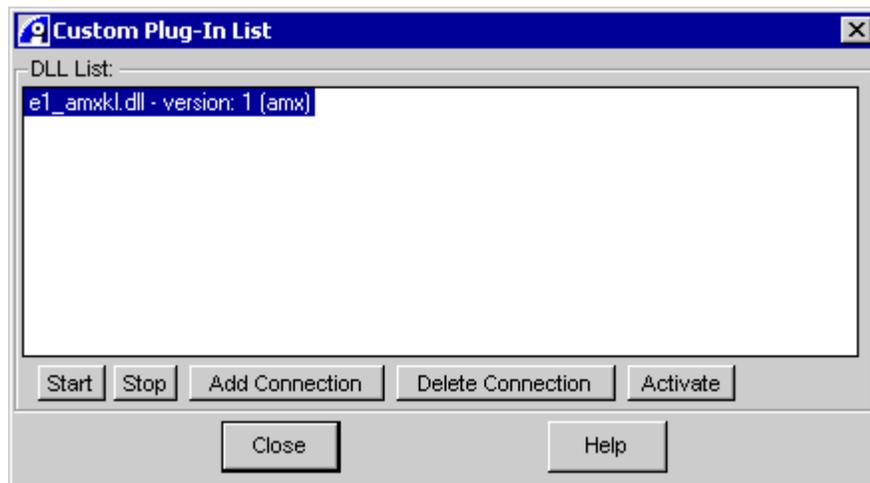


Figure 2.8-9 RealView 3.1 Plug-In List Control



Figure 2.8-10 RealView 3.1 *KwikLook* Plug-In Window

## Debugging with RealView

The RealView task-aware debugger must hit a breakpoint in your AMX application before you will be able to use *KwikLook* to view the state of your program.

In RealView, open the source file which contains a function in one of your AMX tasks which you know will be encountered shortly after AMX is launched. Double click on the the Breakpoint Column next to the first statement within this function to set a breakpoint there. Then click on the Go button on the RealView toolbar to launch AMX. RealView will break on entry to your function.

Once RealView has stopped at the breakpoint, you can use *KwikLook* to explore the state of your system from the AMX perspective. Click on the RealView View menu button and select Resource Viewer Window from the menu to see lists of AMX resources.

To invoke *KwikLook*, click on the View AMX... button on the KwikLook for RealView plug-in window. The *KwikLook* window will appear. Use the *KwikLook* buttons to view all of the AMX objects.

Once *KwikLook* is active, you can use the *KwikLook* Help facility for complete explanations of all fields on every *KwikLook* window. Press the F1 key to get help for the current *KwikLook* display.

Close the main *KwikLook* window or click on its Dismiss button to return to the debugger.

To set a breakpoint on some other function, find the function in its source window and double click in the Breakpoint Column next to the body of the function. Then click on the Go button on the RealView toolbar to resume execution of your AMX application.

You should always run with a breakpoint on AMX procedure *cjksfatal* so that you can detect fatal conditions that have forced AMX to stop execution. You can also place a breakpoint on AMX procedure *cjksbreak* to stop on any AMX procedure which is about to return an error indication to the caller.

## Uninstalling *KwikLook*

*KwikLook* can be uninstalled as described in Chapter 1.3. However, when *KwikLook* is removed, the *KwikLook* files copied into the debugger's installation directory are not removed. To delete those files so that your debugger installation has no *KwikLook* remnants, proceed as follows.

Go to the RealView installation directory.

Delete file *lib\o2\_amx.dll*

Delete files *lib\custom\e1\_amxkl.\**

Delete file *etc\amx\_rtos.bcd*

This page left blank intentionally.

## 2.9 KwikLook for IAR Systems C-SPY

The IAR Systems C-SPY<sup>®</sup> Debugger runs as a Windows application on a host PC. The host PC must be connected to your target hardware using any of the methods supported by IAR Systems. These include in-circuit emulator connections and links to background debug mode (BDM) or JTAG connectors on the target hardware.

*KwikLook* supports C-SPY for testing AMX applications on the following target processors. It is assumed that during the installation process, you indicated that C-SPY was to be installed in the directories listed below. Also, unless otherwise directed by you during the installation process, *KwikLook* will be installed in the directories indicated below.

C-SPY Directory	<i>KwikLook</i> Directory	Target Processor
C:\IAR\ARM	C:\KADAK\KWK302	ARM or Thumb family

The following subdirectories will be created in the *KwikLook* installation directory.

<i>HOST</i>	<i>KwikLook</i> DLL and C-SPY support files
<i>MANUAL</i>	A copy of this manual in Adobe Acrobat <sup>®</sup> PDF format

### *KwikLook* Integration with C-SPY

Some of the files from *KwikLook* subdirectory *HOST* must be copied to a subdirectory named *PLUGINS\RTOS\AMX* in the C-SPY installation directory. The copied files must then be renamed. When *KwikLook* is installed from CD-ROM, the installer will automatically copy and rename these files unless you specifically prohibit integration with C-SPY.

If you do not allow the automatic integration of *KwikLook* with C-SPY, you will have to create the directory *AMX* in the C-SPY *PLUGINS\RTOS* subdirectory and manually transfer the files to the *PLUGINS\RTOS\AMX* directory. The copied files must also be renamed as indicated below for *KwikLook* to operate properly with C-SPY.

<i>KwikLook</i> Name	C-SPY Name	Target Processor
<i>BJ302AR.EWP</i>	<i>BJ302AR.EWPLUGIN</i>	<i>KwikLook</i> plugin information for C-SPY
<i>BJ302AR.DLL</i>	<i>BJ302AR.DLL</i>	C-SPY / <i>KwikLook</i> interface
<i>BJ302.DLL</i>	<i>BJ410.DLL</i>	Use with AMX 4-ARM (PN402-1)
<i>BJ302.HLP</i>	<i>BJ410.HLP</i>	
<i>BJ302.DLL</i>	<i>BJ430.DLL</i>	Use with AMX 4-Thumb (PN422-1)
<i>BJ302.HLP</i>	<i>BJ430.HLP</i>	

## Starting the C-SPY Debugger

The C-SPY Debugger operates as a component of the IAR Embedded Workbench IDE which can be started from the Windows Programs folder. For example, if you are using an ARM processor, locate the "IAR Systems" program folder, choose the "IAR Embedded Workbench for ARM" and select the "IAR Embedded Workbench".

When using the IAR Embedded Workbench IDE to create and manage your AMX application, use the following procedure to allow the C-SPY Debugger to be AMX task-aware. Open your application project and select Options... from the Project menu. A window similar to that shown in Figure 2.9-1 will appear. In this example, the project name is AMX Project. From the leftmost pane titled Category:, choose Debugger. From the rightmost pane select the Plugins tab and click the checkbox for KwikLook for AMX. Click the OK button to accept the change to the project. To save the project settings click Save All from the File menu.

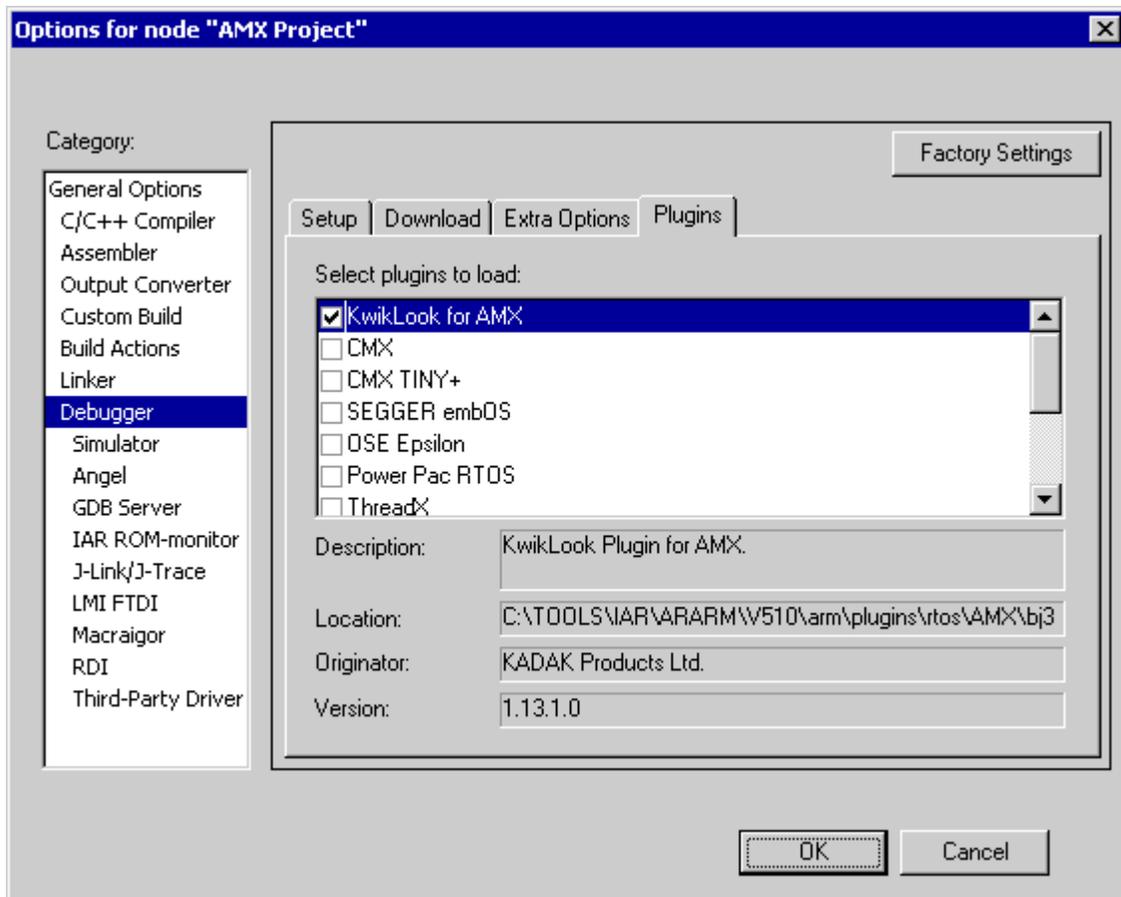


Figure 2.9-1 C-SPY Debugger Plugins

Alternatively, you can run the IAR Embedded Workbench IDE from the Windows command line and start the C-SPY Debugger to test your application. The form of the command line will be similar to that shown below. If you are using an ARM processor, directions will be found in the IAR document file *CSARM.HTM* with title "Release notes for ARM IAR C-SPY Debugger". Note that *KwikLook* is enabled using the *--plugin* command line switch to specify the full path to the *KwikLook* plugin named *BJ302AR.DLL*. Additional command line switches (shown as "... " in the example below) will be required to meet your particular needs as described in the IAR release notes.

```
IarIdePm.exe --DBG ARM ...  
C:\YOURPATH\YOURAPP.OUT ...  
--plugin C:\IAR\plugins\rtos\AMX\BJ302AR.DLL ...
```

#### Note

When the C-SPY Debugger runs from within the IDE, it will generate a batch file which can be used to execute the IAR utility program *CSPYBAT.EXE* to launch the C-SPY Debugger with the command line switches established according to the settings of your AMX project. When the debugger is started this way, *KwikLook* will not be available.

## Debugging with C-SPY

The C-SPY task-aware debugger must be active and stopped at a breakpoint in your AMX application before you will be able to use *KwikLook* to view the state of your program. To activate the C-SPY debugger from within the IAR Embedded Workbench IDE, click Debug from the Project menu.

In the C-SPY source file view, set the cursor on a function in one of your AMX tasks which you know will be encountered shortly after AMX is launched. Click Toggle Breakpoint from the Edit menu. Then click Go from the Debug menu to launch AMX. C-SPY will break on entry to your function.

Once C-SPY has stopped at the breakpoint, you can use *KwikLook* to explore the state of your system from the AMX perspective. To invoke *KwikLook*, select *KwikLook...* from either the View menu or the Tools menu. Alternatively, click on the button labelled **AMX** on the IDE toolbar. The *KwikLook* window will appear. Use the *KwikLook* buttons to view all of the AMX objects.

Once *KwikLook* is active, you can use the *KwikLook* Help facility for complete explanations of all fields on every *KwikLook* window. Press the F1 key to get help for the current *KwikLook* display.

Close the main *KwikLook* window or click on its Dismiss button to return to the debugger.

To set a breakpoint on some other function, find the function in the C-SPY source file view, set the cursor on the source statement of interest and click Toggle Breakpoint from the Edit menu. Then click Go from the Debug menu to resume execution of your AMX application.

You should always run with a breakpoint on AMX procedure *cjksfatal* so that you can detect fatal conditions that have forced AMX to stop execution. You can also place a breakpoint on AMX procedure *cjksbreak* to stop on any AMX procedure which is about to return an error indication to the caller.

## Uninstalling *KwikLook*

*KwikLook* can be uninstalled as described in Chapter 1.3. However, when *KwikLook* is removed, the *KwikLook* files copied into the debugger's installation directory are not removed. To delete those files so that your debugger installation has no *KwikLook* remnants, proceed as follows.

Go to the C-SPY installation directory and locate the C-SPY directory *PLUGINS\RTOS*. Within that directory, delete subdirectory *AMX* and all of the files which it contains.