



## **AMX™ 68000 Tool Guide**

**First Printing: April 15, 1994**

**Last Printing: November 1, 2007**

**Copyright © 1994 - 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 © 1994-2007 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, B.C., 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.

**AMX 68000 TOOL GUIDE**  
**Table of Contents**

	<b>Page</b>
<b>1. Selecting a Tool Set</b>	<b>1-1</b>
<b>2. Diab-SDS (DA) Tool Guide</b>	<b>2-1</b>
<b>3. Microtec (MR) Tool Guide</b>	<b>3-1</b>
<b>4. Tasking (IM) Tool Guide</b>	<b>4-1</b>
<b>5. Metrowerks (ME) Tool Guide</b>	<b>5-1</b>

This page left blank intentionally.

# 1. Selecting a Tool Set

## Available Toolsets

AMX™ 68000 and the *KwikLook™* Fault Finder have been developed on a PC with Microsoft® Windows® using the software development tools described in this guide.

To simplify the selection process, KADAK has prepared this Tool Guide. This chapter introduces the tools and defines the subsets which KADAK has used with success. Subsequent chapters provide specific guidelines for using each of the supported toolset combinations with AMX 68000.

Note that AMX 68000 is delivered to you ready to use with each of the supported toolsets. Should you wish to rebuild the AMX 68000 Library for any reason, follow the construction guidelines provided in Appendix D of the AMX User's Guide.

To construct your embedded application, you will require a C or C++ compiler, an assembler, a librarian (optional), a linker and/or locator and a remote debugger. The vendors listed below provide these tools. The tool name listed is the vendor's product name or the name of the executable program used to run the tool. The tool name listed will be used throughout this manual to reference the specific tool from a particular vendor.

<b>Vendor</b>	<b>C/C++</b>	<b>Assembler</b>	<b>Librarian</b>	<b>Linker</b>	<b>Locator</b>	<b>Debugger</b>
Diab-SDS	<i>DCC</i>	<i>DCC</i>	<i>DAR</i>	<i>DCC</i>	<i>DDUMP</i>	<i>SingleStep</i>
Microtec	<i>MCC68K</i>	<i>ASM68K</i>	<i>LIB68K</i>	<i>LNK68K</i>		<i>XRAY</i>
TASKING	<i>C68000</i>	<i>ASM68000</i>	<i>LIBR</i>	<i>LLINK</i>	<i>FORM</i>	<i>XDB,</i> <i>PassKey,</i> <i>CrossView</i>
Metrowerks	<i>MWCCE68K</i>	<i>MWASME68K</i>	<i>MWLDE68K</i>	<i>MWLDE68K</i>		<i>CodeWarrior</i>

## Supported Toolsets

Unfortunately you cannot arbitrarily use any combination of the listed tools. Of all the tools listed, KADAK has identified several combinations which can be used with AMX 68000. The supported toolsets are divided into major classes according to the C/C++ compiler vendor and then, if necessary, into sub-classes, one for each locator and/or debugger.

Each supported toolset is given a three character mnemonic called a **toolset id** which is used by KADAK to identify the toolset combination. The first two characters of the mnemonic identify the compiler vendor. The third character, if needed, identifies the locator and/or debugger used.

### Compiler

<i>DA</i>	Diab-SDS, Inc. C/C++
<i>MR</i>	Mentor Graphics' Microtec Division C/C++
<i>IM</i>	TASKING, Inc. (formerly Intermetrics) C/C++
<i>ME</i>	Metrowerks Inc. C/C++

The following toolset combinations are supported by KADAK.

Toolset id:	<b>DA</b>	<b>MR</b>	<b>IM</b>	<b>ME</b>
Vendor:	Diab-SDS	Microtec	TASKING	Metrowerks
C/C++	<i>DCC</i>	<i>MCC68K</i>	<i>C68000</i>	<i>MWCCE68K</i>
Assembler	<i>DCC</i>	<i>ASM68K</i>	<i>ASM68000</i>	<i>MWASME68K</i>
Librarian	<i>DAR</i>	<i>LIB68K</i>	<i>LIBR</i>	<i>MWLDE68K</i>
Linker/	<i>DCC</i>	<i>LNK68K</i>	<i>LLINK</i>	<i>MWLDE68K</i>
Locator	<i>DDUMP</i>		<i>FORM</i>	
Debugger	<i>SingleStep</i>	<i>XRAY</i>	<i>XDB</i> <i>PassKey</i> <i>CrossView</i>	<i>CodeWarrior</i>

## 2. Diab-SDS (DA) Tool Guide

AMX™ 68000 has been developed on a PC with Windows® NT v4.0 using the Diab-SDS tools listed below. The AMX libraries and object modules on the product disks have been generated using the most recent tools listed. If you are not using this toolset, you may have to rebuild the AMX libraries in order to use your out-of-date tools.

<b>Diab-SDS Tools</b>		<b><u>v4.2</u></b>	<b><u>v4.3</u></b>	<b><u>v4.4</u></b>	<b><u>v5.0</u></b>	<b><u>v5.1</u></b>	<b><u>v5.2</u></b>
<i>DCC</i>	68000 C/C++ compiler	4.2	4.3	4.4	5.0a	5.1.1	5.2.1
<i>DCC</i>	68000 assembler	4.2	4.3	4.4	5.0a	5.1.1	5.2.1
<i>DAR</i>	68000 librarian	4.2	4.3	4.4	5.0a	5.1.1	5.2.1
<i>DLD</i>	68000 linker	4.2	4.3	4.4	5.0a	5.1.1	5.2.1
<i>DDUMP</i>	68000 locator	4.2	4.3	4.4	5.0a	5.1.1	5.2.1
	SingleStep 68000 Debugger	7.4	7.4	7.6	7.6	7.6	7.6
	SingleStep Target Monitor						

AMX 68000 and *KwikLook* have been tested on the following platforms.

Motorola M68EC040 Integrated Development Platform  
 Motorola M68332EVK Evaluation Kit  
 Motorola MVME133 VMEmodule (MC68020 processor)  
 Motorola M68360QUADS Application Development System  
 GreenSpring Platform332™ Single Board Computer

### Environment Variables

Set the following environment variables to provide access to all AMX and Diab-SDS tools, header files, object files and libraries.

<i>CJPATH</i>	Path to AMX installation directory (. . . \AMX532)
<i>PATH</i>	Path to AMX and Diab-SDS executable programs
<i>TMPDIR</i>	Path to a temporary directory for use by Diab-SDS tools
<i>DIABLIB</i>	Path to Diab-SDS installation directory
<i>DTARGET</i>	Target 68000 processor
<i>DOBJECT=F</i>	Generate ELF object format
<i>DFP=S</i>	Assume software floating point emulation
or <i>DFP=H</i>	Assume hardware floating point
<i>DENVIRON=cross</i>	Compiler cross-compiles on PC for 68000

The AMX libraries have been constructed using the following Diab-SDS parameters. The resulting AMX 68000 libraries are ready for use with all 68000 implementations.

<i>DOBJECT=F</i>	AMX 68000 is generated in ELF object format
<i>DTARGET=MC68000</i>	AMX 68000 target processor is MC68000
<i>DFP=S</i>	Assume software floating point emulation

## Object Formats

The AMX 68000 libraries and object modules are provided in ELF format. Your object modules and the AMX and Diab-SDS libraries and object modules, all in ELF format, can be combined to create an executable module in ELF format suitable for use with the Diab-SDS SingleStep Debugger.

## Parameter Passing Conventions

AMX 68000 uses the stack based parameter passing convention common to all toolsets supported by KADAK.

## Register Usage

The Diab-SDS version of AMX makes the following C interface register assumptions. Registers *D0*, *D1*, *A0* and *A1* can always be altered by C procedures. Integers and pointers are returned from C procedures in register *D0*. No register is dedicated for global data access. You must NOT use any C compilation switch which changes these register assumptions.

## Using the Diab-SDS C Compiler

All AMX header files *CJ532xxx.H* and the generic AMX include file *CJZZZ.H* must be present in the current directory together with your source file being compiled.

Use the following compilation switches when you are compiling modules for use in the AMX environment.

by default	; no stack checking
-o	; output object module <i>FILENAME.OBJ</i>
-@E=	; redirect C error messages to <i>FILENAME.ERR</i>
-c	; compile only
-XO	; full optimize for speed
-g	; (optional) generate debug information

The compilation command line is therefore of the form:

```
DCC -c -XO -o FILENAME.OBJ FILENAME.C -@E=FILENAME.ERR
```

## Compiling the AMX System Configuration Module

Your AMX System Configuration Module *SYSCFG.C* is compiled as follows. All AMX header files *CJ532xxx.H* and the generic AMX include file *CJZZZ.H* must be present in the current directory together with file *SYSCFG.C*.

```
DCC -c -XO -o SYSCFG.OBJ SYSCFG.C -@E=SYSCFG.ERR
```

## Assembling the AMX Target Configuration Module

Your AMX Target Configuration Module *HDWCFG.ASM* is assembled as follows. The generic AMX header file *CJZZZK.DEF* must be present in the current directory together with file *HDWCFG.ASM*.

The Diab-SDS C command line driver is used to invoke the assembler. Some of the command line switches match those used for C. Others are as follows.

by default	; assemble with case sensitivity
-o	; output object module <i>HDWCFG.OBJ</i>
-@E=	; redirect assembler error messages to <i>HDWCFG.ERR</i>
-Wa, -x	; omit local symbols from object module
-Wa.ASM	; use <i>ASM</i> as extension for assembly language files
-c	; assemble only

```
DCC -c -Wa.ASM -Wa, -x -o HDWCFG.OBJ HDWCFG.ASM -@E=HDWCFG.ERR
```

## Making Libraries

To make a library from a collection of object modules, create a library specification file *YOURLIB.LBM*. Use the Diab-SDS version of the AMX library specification file *CJ532.LBM* as a guide.

Use the following command line switches when using the Diab-SDS librarian.

<i>YOURLIB.LIB</i>	; output library module <i>YOURLIB.LIB</i>
> <i>YOURLIB.LBE</i>	; redirect librarian error messages to <i>YOURLIB.LBE</i>
-qc	; create a new library; use quick append mode

Make your library as follows.

```
DAR -qc YOURLIB.LIB -@YOURLIB.LBM >YOURLIB.LBE
```

## Linking with the Diab-SDS Linker

When used with Diab-SDS C, the modules which form your AMX system must be linked in the following order.

Your *MAIN* module

Other application modules

<i>SYSCFG.OBJ</i>	; AMX System Configuration Module
<i>HDWCFG.OBJ</i>	; AMX Target Configuration Module
<i>CHxxxxxT.OBJ</i>	; AMX chip-specific clock driver or your equivalent
<i>CJ532UF.OBJ</i>	; Launch and leave AMX (may be customized)
<i>CJ532RAC.OBJ</i>	; AMX ROM Access Module (customized) ; (only if AMX placed in a separate ROM) ; (see Appendix C in AMX 68000 Target Guide)
<i>CJ532CV.LIB</i>	; AMX 68000 vc Conversion Library ; (only if converting an AMX 86 v3, AMX 386 v1 or ; AMX 68000 v2 application)
<i>CJ532.LIB</i>	; AMX 68000 Library

Diab-SDS C Runtime Libraries for target hardware

Create a link specification file *YOURLINK.LKS*. Use the Diab-SDS version of the AMX Sample Program link specification file *CJSAMPLE.LKS* as a guide.

Start with the sample link specification file for the board which most closely resembles your hardware configuration.

Note

If you decide to omit any of the link and locate commands from the sample specification, you may encounter link errors or run-time faults.

Link and locate with the Diab-SDS linker and locator using the following command line switches.

```
-m                ; create section summary
-Wm              ; no default link command file
-o              ; direct link output to file YOURLINK.OUT
-@E=           ; direct link error messages to file YOURLINK.LKE
>YOURLINK.MAP   ; direct section summary to file YOURLINK.MAP

-t              ; create summary of symbol values
-v              ; inhibit output of .bss section to minimize
                ; the size of the resulting HEX file.
-R              ; generate Motorola S-record format
                ; other formats can be generated
                ; (see Diab-SDS manual)
-o              ; direct locate output to file YOURLINK.HEX
>YOURLINK.SYM  ; direct symbol summary to file YOURLINK.SYM
```

The link and locate command lines are therefore of the form:

```
DCC -m -Wm -o YOURLINK.OUT YOURLINK.LKS -@E=YOURLINK.LKE >YOURLINK.MAP
DDUMP -t -v -R -o YOURLINK.HEX YOURLINK.OUT >YOURLINK.SYM
```

The resulting load module *YOURLINK.OUT* is suitable for use with the Diab-SDS SingleStep 68000 debugger.

The resulting load module *YOURLINK.HEX* is ready for burning into EPROM.

## Linking a Separate AMX ROM

AMX can be committed to a separate ROM as described in Appendix C of the AMX Target Guide. Use the AMX Configuration Manager to edit your Target Parameter File *HDWCFG.UP* to define your ROM option parameters. Then use the Manager to generate your ROM Option Module *CJ532ROP.ASM*, ROM Access Module *CJ532RAC.ASM* and ROM Option link specification file *CJ532ROP.LKS*.

The ROM Option and ROM Access source modules are assembled as follows.

```
DCC -c -Wa.ASM -Wa,-x -o CJ532ROP.OBJ CJ532ROP.ASM -@E=CJ532ROP.ERR
DCC -c -Wa.ASM -Wa,-x -o CJ532RAC.OBJ CJ532RAC.ASM -@E=CJ532RAC.ERR
```

The AMX ROM is linked using link specification file *CJ532ROP.LKS* as follows.

```
DCC -m -Wm -Ws -Wc -o AMXROM.OUT CJ532ROP.LKS
                                     -@E=AMXROM.LKE >AMXROM.MAP
DDUMP -t -v -R -o AMXROM.HEX AMXROM.OUT >AMXROM.SYM
```

This example generates file *AMXROM.HEX* in Motorola S-record format suitable for transfer to ROM. Other formats supported by Diab-SDS can be selected with the appropriate command switch.

Note that command line switch *-Ws* is used to prevent loading of the default C startup module. Command line switch *-Wc* is used to prevent loading of the default C runtime library.

When you link your AMX application, be sure to include your customized AMX ROM Access Module *CJ532RAC.OBJ* (created above) in your system link specification file.

## Using the AMX Configuration Generator

If you cannot use the AMX Configuration Manager, you may still be able to use the stand-alone AMX Configuration Generator to generate the ROM Option Module *CJ532ROP.ASM*, ROM Access Module *CJ532RAC.ASM* and ROM Option link specification file *CJ532ROP.LKS*.

Copy the ROM Option and ROM Access template files *CJ532ROP.CT* and *CJ532RAC.CT* to the current directory. Also copy the ROM Option Link Specification Template file *CJ532ROP.LKT* to the current directory.

Use the AMX Configuration Generator to generate the ROM option source modules as follows.

```
CJ532CG HDWCFG.UP CJ532ROP.CT CJ532ROP.ASM
CJ532CG HDWCFG.UP CJ532RAC.CT CJ532RAC.ASM
CJ532CG HDWCFG.UP CJ532ROP.LKT CJ532ROP.LKS
```

Once the ROM option source modules have been created, you can proceed to build your AMX ROM image and your AMX application as described above.

## **Diab-SDS SingleStep Debugger**

The Diab-SDS SingleStep™ 68000 Debugger supports source level debugging of your AMX 68000 system.

The SingleStep Debugger can operate by simulating an M68000 or by using a BDM or JTAG connection to the M68000 processor.

SingleStep can also operate using a serial (or other) connection to the target M68000 system under test. When used in this fashion, you must install the SingleStep Target Monitor in your target hardware. Instructions for doing so are provided in the SingleStep Reference Manual. Your version of the SingleStep Target Monitor must provide a device driver for the serial (or other) device used for communication with the SingleStep Debugger. It is recommended that your driver use polled I/O so that the SingleStep Target Monitor can operate with interrupts disabled.

## **Using the *KwikLook* Fault Finder**

The *KwikLook*™ Fault Finder is compatible with the SingleStep Debugger providing full screen, source level, task-aware debugging from within the Microsoft Windows® environment. *KwikLook* can be invoked directly from the debugger while at breakpoints giving you finger tip access to your application from the AMX perspective. Note that *KwikLook* and SingleStep share a common link to the target system.

### 3. Microtec (MR) Tool Guide

AMX™ 68000 has been developed on a PC with Windows® NT v4.0 using the Mentor Graphics' Microtec Division tools listed below. The AMX libraries and object modules on the product disks have been generated using the most recent tools listed. If you are not using this toolset, you may have to rebuild the AMX libraries in order to use your out-of-date tools.

<b>Microtec Tools</b>	<b><u>v4.4</u></b>	<b><u>v4.5</u></b>	<b><u>v6C</u></b>	<b><u>no ver</u></b>
<i>MCC68K</i> C Compiler	v4.4	v4.5G	v5.0J	v5.1
<i>ASM68K</i> Assembler	v7.0	v7.1C	v7.3C	v7.4
<i>LIB68K</i> Object Module Librarian	v10.2	v10.7	v10.11A	v10.11A
<i>LNK68K</i> Linker	v7.0	v7.1C	v7.3C	v7.4
<i>XHM68K</i> XRAY Debugger	v2.3C	v3.6M	D4.3	D4.4
<i>XDM68K</i> XRAY Monitor				

AMX 68000 and *KwikLook* have been tested on the following platforms.

Motorola M68EC040 Integrated Development Platform  
Motorola M68332EVK Evaluation Kit  
Motorola MVME133 VME module (MC68020 processor)  
Motorola M68360QUADS Application Development System  
GreenSpring Platform332™ Single Board Computer

#### Environment Variables

Set the following environment variables to provide access to all AMX and Microtec tools, header files, object files and libraries.

<i>CJPATH</i>	Path to AMX installation directory (... \AMX532)
<i>PATH</i>	Path to AMX and Microtec executable programs
<i>MRI_68K_BIN</i>	Path to Microtec executable programs
<i>MRI_68K_INC</i>	Path to all Microtec include header files
<i>MRI_68K_LIB</i>	Path to all Microtec object files and libraries (For use in batch files only: -L%LIB%)
<i>TMP</i>	Path to a temporary directory for use by tools

## Object Formats

The AMX 68000 libraries and object modules are provided in IEEE-695 format. Your object modules and the AMX and Microtec libraries and object modules, all in IEEE-695 format, can be combined to create an executable module in IEEE-695 format suitable for use with the Microtec XRAY Debugger.

## Parameter Passing Conventions

AMX 68000 uses the stack based parameter passing convention common to all toolsets supported by KADAK.

## Register Usage

The Microtec version of AMX makes the following C interface register assumptions. Registers *D0*, *D1*, *A0* and *A1* can always be altered by C procedures. Integers and pointers are returned from C procedures in register *D0*. No register is dedicated for global data access. You must NOT use any C compilation switch which changes these register assumptions.

## Using the Microtec C Compiler

All AMX header files *CJ532xxx.H* and the generic AMX include file *CJZZZ.H* must be present in the current directory together with your source file being compiled.

Use the following compilation switches when you are compiling modules for use in the AMX environment.

	by default	; pass all parameters on the stack
	by default	; add leading underscores to symbols
	by default	; no stack checking
	-c	; compile but do not link
	-Za4	; quad-align structure members larger than short
or	-Z4	; (for releases prior to v4.5G)
	-nQ	; do not suppress warning messages
	-o	; output object module <i>FILENAME.OBJ</i>
	-g	; (optional) generate debug information for XRAY

The compilation command line is therefore of the form:

```
MCC68K -c -Za4 -nQ -o FILENAME.OBJ FILENAME.C
```

## Compiling the AMX System Configuration Module

Your AMX System Configuration Module *SYSCFG.C* is compiled as follows. All AMX header files *CJ532xxx.H* and the generic AMX include file *CJZZZ.H* must be present in the current directory together with file *SYSCFG.C*.

```
MCC68K -c -Za4 -nQ -o SYSCFG.OBJ SYSCFG.C
```

## Assembling the AMX Target Configuration Module

Your AMX Target Configuration Module *HDWCFG.ASM* is assembled as follows. The generic AMX header file *CJZZZK.DEF* must be present in the current directory together with file *HDWCFG.ASM*.

The Microtec command line to invoke the assembler is as follows.

```
ASM68K -o HDWCFG.OBJ HDWCFG.ASM
```

## Making Libraries

To make a library from a collection of object modules, create a library specification file *YOURLIB.LBM*. Use the Microtec version of the AMX library specification file *CJ532.LBM* as a guide.

Use the following command line switches when using the Microtec librarian.

```
< ; use Library Specification File YOURLIB.LBM
```

Make your library as follows.

```
LIB68K <YOURLIB.LBM
```

## Microtec Startup Code

The C startup code is provided by Microtec in file *CSYS.C*. This source file must be compiled to create object module *CSYS.OBJ*. You may wish to tailor *CSYS.C* to meet your unique environmental requirements. To force initialized data to be copied from ROM to RAM at startup, compile the module as follows and copy the resulting object module *CSYS.OBJ* to the library directory specified by environment variable *MRI\_68K\_LIB*.

```
MCC68K -c -D_INITDATA -o CSYS.OBJ CSYS.C {v4.2 tools}  
MCC68K -c -o CSYS.OBJ CSYS.C {v4.3 and later tools}
```

## Linking with the Microtec Linker

When used with Microtec C, the modules which form your AMX system must be linked in the following order.

<i>CSYS.O</i>	; Microtec C startup module
Your <i>MAIN</i> module	
Other application modules	
<i>SYSCFG.OBJ</i>	; AMX System Configuration Module
<i>HDWCFG.OBJ</i>	; AMX Target Configuration Module
<i>CHxxxxxxT.OBJ</i>	; AMX chip-specific clock driver or your equivalent
<i>CJ532UF.OBJ</i>	; Launch and leave AMX (may be customized)
<i>CJ532RAC.OBJ</i>	; AMX ROM Access Module (customized) ; (only if AMX placed in a separate ROM) ; (see Appendix C in AMX 68000 Target Guide)
<i>CJ532CV.LIB</i>	; AMX 68000 vc Conversion Library ; (only if converting an AMX 86 v3, AMX 386 v1 or ; AMX 68000 v2 application)
<i>CJ532.LIB</i>	; AMX 68000 Library
Microtec C Runtime Libraries for target hardware	

Create a link specification file *YOURLINK.LKS*. Use the Microtec version of the AMX Sample Program link specification file *CJSAMPLE.LKS* as a guide.

Start with the sample link specification file for the board which most closely resembles your hardware configuration.

Note

If you decide to omit any of the link and locate commands from the sample specification file, you may encounter link errors or run-time faults.

Link and locate with the Microtec linker using the following command line switches.

<i>-M</i>	; generate map file <i>YOURLINK.MAP</i>
by default	; generate IEEE Std. 695 records for XRAY use
<i>-F s</i>	; generate Motorola S-records
	; other formats can be generated
	; (see Microtec manual)
<i>-C CASE</i>	; symbols are case sensitive

The link and locate command lines are therefore of the form:

```
LNK68K -M -C CASE -o YOURLINK.ABS -c YOURLINK.LKS
```

The resulting load module *YOURLINK.ABS* is suitable for use with the Microtec XRAY 68000 debugger.

You can use the *-F s* switch to create a load module *YOURLINK.HEX* in Motorola S-record format suitable for transfer to ROM. Other formats supported by Microtec can be selected with the appropriate *-F* command line switch.

## Linking a Separate AMX ROM

AMX can be committed to a separate ROM as described in Appendix C of the AMX Target Guide. Use the AMX Configuration Manager to edit your Target Parameter File *HDWCFG.UP* to define your ROM option parameters. Then use the Manager to generate your ROM Option Module *CJ532ROP.ASM*, ROM Access Module *CJ532RAC.ASM* and ROM Option link specification file *CJ532ROP.LKS*.

The ROM Option and ROM Access source modules are assembled as follows.

```
ASM68K -o CJ532ROP.OBJ CJ532ROP.ASM
```

```
ASM68K -o CJ532RAC.OBJ CJ532RAC.ASM
```

The AMX ROM is linked using link specification file *CJ532ROP.LKS* as follows.

```
LNK68K -M -F S -C CASE -o AMXROM.HEX CJ532ROP.LKS
```

This example generates file *AMXROM.HEX* in Motorola S-record format suitable for transfer to ROM. Other formats supported by Microtec can be selected with the appropriate *-F* command line switch.

When you link your AMX application, be sure to include your customized AMX ROM Access Module *CJ532RAC.OBJ* (created above) in your system link specification file.

## Using the AMX Configuration Generator

If you cannot use the AMX Configuration Manager, you may still be able to use the stand-alone AMX Configuration Generator to generate the ROM Option Module *CJ532ROP.ASM*, ROM Access Module *CJ532RAC.ASM* and ROM Option link specification file *CJ532ROP.LKS*.

Copy the ROM Option and ROM Access template files *CJ532ROP.CT* and *CJ532RAC.CT* to the current directory. Also copy the ROM Option Link Specification Template file *CJ532ROP.LKT* to the current directory.

Use the AMX Configuration Generator to generate the ROM option source modules as follows.

```
CJ532CG HDWCFG.UP CJ532ROP.CT CJ532ROP.ASM
```

```
CJ532CG HDWCFG.UP CJ532RAC.CT CJ532RAC.ASM
```

```
CJ532CG HDWCFG.UP CJ532ROP.LKT CJ532ROP.LKS
```

Once the ROM option source modules have been created, you can proceed to build your AMX ROM image and your AMX application as described above.

## Microtec XRAY Debugger

The Microtec MS-DOS<sup>®</sup> and Windows<sup>®</sup> based XRAY Debuggers support source level debugging of your AMX 68000 system.

The Microtec XRAY Debugger can operate using a serial (or other) connection to the target M68000 system under test. When used in this fashion, you must install the Microtec *XDM68K* XRAY Monitor in your target hardware. Instructions for doing so are provided in the XRAY68K Reference Manual. Your version of the monitor must provide a device driver for the serial (or other) device used for communication with the debugger. It is recommended that your driver use polled I/O so that the monitor can operate with interrupts disabled.

## Using the *KwikLook* Fault Finder

The *KwikLook*<sup>™</sup> Fault Finder does not provide support for the XRAY Debugger. Hence, although XRAY can be used to test your AMX application, XRAY is not AMX-aware.

### Note

The AMX-aware Diab-SDS SingleStep<sup>™</sup> Debugger can be used with *KwikLook* to debug AMX applications generated using Microtec tools.

## 4. TASKING (IM) Tool Guide

AMX™ 68000 has been developed on a PC with Windows® NT v4.0 using the TASKING Inc. (formerly Intermetrics Microsystems Software Inc.) tools listed below. The AMX libraries and object modules on the product disks have been generated using the most recent tools listed. If you are not using this toolset, you may have to rebuild the AMX libraries in order to use your out-of-date tools.

<b>TASKING Tools</b>		<b><u>v8.3</u></b>	<b><u>v8.4.2</u></b>	<b><u>v9.03</u></b>	<b><u>v9.2</u></b>
<i>C68000</i>	C Compiler	v8.3	v8.4.2	v9.03	v9.2
<i>ASM68000</i>	Assembler	v8.3	v8.4.2	v9.03	v9.2
<i>LIBR</i>	Object Module Librarian	v8.3	v8.4.2	v9.03	v9.2
<i>LLINK</i>	Linker	v8.3	v8.4.2	v9.03	v9.2
<i>FORM</i>	Object Module Formatter	v8.3	v8.4.2	v9.03	v9.2
<i>GSMAP</i>	Global Symbol Mapper	v8.3	v8.4.2	v9.03	v9.2
	PassKey Debugger	v1.1.1	v2.03		
	CrossView Pro Debugger			v3.0.1	v9.2
	SmartMON ROM Monitor				

AMX 68000 and *KwikLook* have been tested on the following platforms.

Motorola M68EC040 Integrated Development Platform  
 Motorola M68332EVK Evaluation Kit  
 Motorola MVME133 VME module (MC68020 processor)  
 Motorola M68360QUADS Application Development System  
 GreenSpring Platform332™ Single Board Computer

### Environment Variables

Set the following environment variables to provide access to all AMX and TASKING tools, header files, object files and libraries.

<i>CJPATH</i>	Path to AMX installation directory (. . . \AMX532)
<i>PATH</i>	Path to AMX and TASKING executable programs
<i>INCLUDE</i>	Path to all TASKING include header files
<i>LIB</i>	Path to all TASKING object files and libraries
<i>TMP</i>	Path to a temporary directory for use by tools

## Object Formats

The AMX 68000 libraries and object modules are provided in TASKING absolute format. Your object modules and the AMX and TASKING libraries and object modules, all in absolute format, can be combined to create an executable module in absolute format suitable for use with the TASKING XDB, PassKey or CrossView Debuggers.

## Parameter Passing Conventions

AMX 68000 uses the stack based parameter passing convention common to all toolsets supported by KADAK.

## Register Usage

The TASKING version of AMX makes the following C interface register assumptions. Registers *D0*, *D1*, *A0* and *A1* (or *A4*) can always be altered by C procedures. Integers are returned from C procedures in register *D0*; pointers are returned in register *A0*. Register *A5* is dedicated for global data access. You must NOT use any C compilation switch, such as *-ar*, which changes these register assumptions.

## Using the TASKING C Compiler

All AMX header files *CJ532xxx.H* and the generic AMX include file *CJZZZ.H* must be present in the current directory together with your source file being compiled.

Use the following compilation switches when you are compiling modules for use in the AMX environment.

by default	; pass all parameters on the stack
by default	; add leading underscores to symbols
by default	; no stack checking
-L	; <i>int</i> is 4 bytes; <i>short</i> is 2 bytes
-ia	; use in-line assembler
-sd	; non stack data is <i>separate</i>
-j	; use short branches if possible
-pack 4	; use fullword data alignment for fullword data
-vv	; assume global variables are volatile
	; (only for v8.3 of the C compiler)
-o	; output object module <i>FILENAME.OBJ</i>
-d	; (optional) generate debug information
-do	; (optional) no optimization if debugging

The compilation command line is therefore of the form:

```
C68000 FILENAME.C -L -sd -j -pack 4 -vv -o FILENAME.OBJ
```

It is recommended that you also use the "-cs" and "-ss *idseg udseg*" switches to force the compiler to place constants, initialized data and uninitialized data into single memory sections. By so doing, the number of memory sections present in the final load module is greatly reduced. This reduction is essential if you intend to use the Diab-SDS SingleStep Debugger to test AMX applications generated using TASKING tools.

## Compiling the AMX System Configuration Module

Your AMX System Configuration Module *SYSCFG.C* is compiled as follows. All AMX header files *CJ532xxx.H* and the generic AMX include file *CJZZZ.H* must be present in the current directory together with file *SYSCFG.C*.

```
C68000 SYSCFG.C -L -sd -j -pack 4 -vv -o SYSCFG.OBJ
```

## Assembling the AMX Target Configuration Module

Your AMX Target Configuration Module *HDWCFG.ASM* is assembled as follows. The generic AMX header file *CJZZZK.DEF* must be present in the current directory together with file *HDWCFG.ASM*.

The TASKING command line to invoke the assembler is as follows.

```
ASM68000 HDWCFG.ASM -o HDWCFG.OBJ
```

## Making Libraries

To make a library from a collection of object modules, create a library specification file *YOURLIB.LBM*. Use the TASKING version of the AMX library specification file *CJ532.LBM* as a guide.

Use the following command line switches when using the TASKING librarian.

```
-L                ; output library module YOURLIB.LIB  
-rf              ; create a new library  
YOURLIB.LBM     ; use Library Specification File YOURLIB.LBM
```

Make your library as follows.

```
LIBR -L YOURLIB.LIB -rf YOURLIB.LBM
```

## TASKING Startup Code

KADAK has modified the C startup code provided by TASKING. Source file *CJ532SU.ASM* (object module *CJ532SU.OBJ*) is the modified file suitable for use with AMX in embedded systems. You may wish to tailor *CJ532SU.ASM* to meet your unique environmental requirements. Assemble the module (see Assembling the AMX Target Configuration Module) and copy the resulting object module *CJ532SU.OBJ* to the AMX library directory *TOOLIM\LIB*.

## Linking with the TASKING Linker

When used with TASKING C, the modules which form your AMX system must be linked in the following order.

*CJ532SU.OBJ* ; Modified TASKING C startup module

Your *MAIN* module  
Other application modules

*SYSCFG.OBJ* ; AMX System Configuration Module  
*HDWCFG.OBJ* ; AMX Target Configuration Module

*CHxxxxxxT.OBJ* ; AMX chip-specific clock driver or your equivalent

*CJ532UF.OBJ* ; Launch and leave AMX (may be customized)

*CJ532RAC.OBJ* ; AMX ROM Access Module (customized)  
; (only if AMX placed in a separate ROM)  
; (see Appendix C in AMX 68000 Target Guide)

The following libraries must be specified in a library specification file.

*CJ532CV.LIB* ; AMX 68000 vc Conversion Library  
; (only if converting an AMX 86 v3, AMX 386 v1 or  
; AMX 68000 v2 application)

*CJ532.LIB* ; AMX 68000 Library  
TASKING C Runtime Libraries for target hardware

Create a link specification file *YOURLINK.LKS*. Use the TASKING version of the AMX Sample Program link specification file *CJSAMPLE.LKS* as a guide.

Create a memory specification file *YOURMEM.LOC*. Use the TASKING version of the AMX Sample Program memory specification file *CJSAMPLE.LOC* as a guide.

Create a library specification file *YOURLIB.LBS*. Use the TASKING version of the AMX Sample Program library specification file *CJSAMPLE.LBS* as a guide. If you are converting an existing AMX 68000 v2 application, be sure to add the AMX Conversion Library file *CJ532CV.LIB* to the head of the list in your library specification file *YOURLIB.LBS*.

Start with the sample link, memory and library specification files for the board which most closely resembles your hardware configuration.

Note

If you decide to omit any of the link and locate commands from the sample specification files, you may encounter link errors or run-time faults.

Link and locate with the TASKING linking locator, object module formatter and global symbol mapper using the following command line switches.

<i>-i</i>	; link specification file <i>YOURLINK.LNK</i>
<i>-c</i>	; memory specification file <i>YOURMEM.LOC</i> ; (locator commands)
<i>-il</i>	; library specification file <i>YOURLIB.LBS</i>
<i>-o</i>	; output file <i>YOURLINK.xxx</i>
<i>-f pm</i>	; generate Motorola S records ; other formats can be generated ; (see TASKING manual)
<i>-n</i>	; sort symbols numerically
<i>-x</i>	; (optional) link for debugging

The link and locate command lines are therefore of the form:

```
LLINK -x -i YOURLINK.LKS -c YOURMEM.LOC -il YOURLIB.LBS
                                     -o YOURLINK.AB
FORM YOURLINK.AB -x -f pm -o YOURLINK.HEX
GSMAP YOURLINK.AB -n -o YOURLINK.RPT
```

If you are using the CrossView Debugger with a ROM Monitor, do not use the *-x* switch. Instead, add the 32-bit module *END.LLN* (not the 16-bit module *END.LN*) to your link specification file.

The resulting load module *YOURLINK.AB* is suitable for use with the TASKING PassKey or CrossView 68000 debugger.

File *YOURLINK.HEX* is in Motorola S-record format suitable for transfer to ROM. Other formats supported by TASKING can be selected with the appropriate *-f* command switch.

## Linking a Separate AMX ROM

AMX can be committed to a separate ROM as described in Appendix C of the AMX Target Guide. Use the AMX Configuration Manager to edit your Target Parameter File *HDWCFG.UP* to define your ROM option parameters. Then use the Manager to generate your ROM Option Module *CJ532ROP.ASM*, ROM Access Module *CJ532RAC.ASM* and ROM Option memory specification file *CJ532ROP.LOC*.

The AMX Configuration Manager must have access to the ROM Option Memory Specification Template file *CJ532ROP.LCT*. If you have installed AMX for multiple toolsets, the Manager may not be referencing the TASKING toolset directory *TOOLIM* for its template files. Go to the **File, Templates...** menu and, from the list of selectors, choose the selector for the ROM Option Link/Locate File. Adjust the configuration template by browsing for the file *TOOLIM\CFG\CJ532ROP.LCT*.

The ROM Option and ROM Access source modules are assembled as follows.

```
ASM68000 CJ532ROP.ASM -o CJ532ROP.OBJ  
  
ASM68000 CJ532RAC.ASM -o CJ532RAC.OBJ
```

The AMX ROM is linked using memory specification file *CJ532ROP.LOC*, link specification file *CJ532ROP.LKS* and library specification file *CJ532ROP.LBS* as follows.

```
LLINK -i CJ532ROP.LKS -c CJ532ROP.LOC -il CJ532ROP.LBS -o AMXROM.AB  
FORM AMXROM.AB -f pm -o AMXROM.HEX  
GSMAP AMXROM.AB -n -o AMXROM.RPT
```

This example generates file *AMXROM.HEX* in Motorola S-record format suitable for transfer to ROM. Other formats supported by TASKING can be selected with the appropriate *-f* command line switch.

When you link your AMX application, be sure to include your customized AMX ROM Access Module *CJ532RAC.OBJ* (created above) in your system link specification file.

## Using the AMX Configuration Generator

If you cannot use the AMX Configuration Manager, you may still be able to use the stand-alone AMX Configuration Generator to generate the ROM Option Module *CJ532ROP.ASM*, ROM Access Module *CJ532RAC.ASM* and ROM Option memory specification file *CJ532ROP.LOC*.

Copy the ROM Option and ROM Access template files *CJ532ROP.CT* and *CJ532RAC.CT* to the current directory. Also copy the ROM Option Memory Specification Template file *CJ532ROP.LCT* to the current directory.

Use the AMX Configuration Generator to generate the ROM option source modules as follows.

```
CJ532CG HDWCFG.UP CJ532ROP.CT CJ532ROP.ASM
CJ532CG HDWCFG.UP CJ532RAC.CT CJ532RAC.ASM
CJ532CG HDWCFG.UP CJ532ROP.LCT CJ532ROP.LOC
```

Once the ROM option source modules have been created, you can proceed to build your AMX ROM image and your AMX application as previously described.

## TASKING XDB, PassKey and CrossView Debuggers

The TASKING MS-DOS<sup>®</sup> based XDB and Windows<sup>®</sup> based PassKey and CrossView Debuggers support source level debugging of your AMX 68000 system.

The TASKING XDB, PassKey and CrossView Debuggers can operate using a serial (or other) connection to the target M68000 system under test. When used in this fashion, you must install the TASKING *XDBrom* or *SmartMON* ROM Monitor in your target hardware. Instructions for doing so are provided in the *XDBrom* and *SmartMON* ROM Monitor User's Guides. Your version of the monitor must provide a device driver for the serial (or other) device used for communication with the debugger. It is recommended that your driver use polled I/O so that the monitor can operate with interrupts disabled.

### Using the *KwikLook* Fault Finder

The *KwikLook*<sup>™</sup> Fault Finder does not provide support for the XDB, PassKey or CrossView Debuggers. Hence, although these debuggers can be used to test your AMX application, the debuggers are not AMX-aware.

#### Note

The AMX-aware Diab-SDS SingleStep<sup>™</sup> Debugger can be used with *KwikLook* to debug AMX applications generated using TASKING tools.

## 5. Metrowerks (ME) Tool Guide

AMX™ 68000 has been developed on a PC with Windows® NT v4.0 using the Metrowerks tools listed below. The AMX libraries and object modules on the product disks have been generated using the most recent tools listed. If you are not using this toolset, you may have to rebuild the AMX libraries in order to use your out-of-date tools.

<b>Metrowerks Tools</b>	<b><u>v2.0</u></b>	<b><u>v2.5</u></b>	<b><u>v3.2</u></b>
<i>MWCCE68K</i> 68000 C/C++ compiler	2.0	2.5	3.2
<i>MWASME68K</i> 68000 assembler	2.0	2.5	3.2
<i>MWLDE68K</i> 68000 linker/librarian	2.0	2.5	3.2
CodeWarrior IDE			
CodeWarrior 68000 Debugger			
<i>MetroTRK</i> Target Resident Kernel			

AMX 68000 and *KwikLook* have been tested on the following platforms.

Motorola M68EC040 Integrated Development Platform  
Motorola M68332EVK Evaluation Kit  
Motorola MVME133 VME module (MC68020 processor)  
Motorola M68360QUADS Application Development System  
GreenSpring Platform332™ Single Board Computer

### Environment Variables

Set the following environment variables to provide access to all AMX and Metrowerks tools, header files, object files and libraries.

<i>CJPATH</i>	Path to AMX installation directory (... \AMX532)
<i>PATH</i>	Path to AMX and Metrowerks executable programs
<i>TMPDIR</i>	Path to a temporary directory for use by Metrowerks tools
<i>CWFolder</i>	Path to Metrowerks installation directory
<i>MWCIncludes</i>	Path to Metrowerks include directories
<i>MWLibraries</i>	Path to Metrowerks library directories
<i>MWLibraryFiles</i>	File names of Metrowerks C libraries to be searched

## Command Line Tools

The Metrowerks CodeWarrior Integrated Development Environment (IDE) provides a software development environment within which you can readily create a project which incorporates AMX. However, the AMX library construction process is independent of the CodeWarrior IDE.

To make the AMX libraries and to construct an AMX application as described in this Tool Guide, you must use the Metrowerks command line tools. It is assumed that the following Metrowerks tools have been copied from the Metrowerks installation directory to the Metrowerks *BIN* directory and renamed as follows.

<code>... \E68K_Tools\Command_Line_Tools\mwcc68k.exe</code>	to <code>MWCC.EXE</code>
<code>... \E68K_Tools\Command_Line_Tools\mwasm68k.exe</code>	to <code>MWASM.EXE</code>
<code>... \E68K_Tools\Command_Line_Tools\mwld68k.exe</code>	to <code>MWLD.EXE</code>

## Object Formats

The AMX 68000 libraries and object modules are provided in ELF format. Your object modules and the AMX and Metrowerks libraries and object modules, all in ELF format, can be combined to create an executable module in ELF format suitable for use with the Metrowerks CodeWarrior Debugger.

## Parameter Passing Conventions

AMX 68000 uses the stack based parameter passing convention common to all toolsets supported by KADAK.

## Register Usage

The Metrowerks version of AMX makes the following C interface register assumptions. Registers *D0*, *D1*, *D2*, *A0* and *A1* can always be altered by C procedures. Integers and pointers are returned from C procedures in register *D0*. Register *A5* is dedicated for global data access. You must NOT use any C compilation switch which changes these register assumptions.

## Using the Metrowerks C Compiler

All AMX header files *CJ532xxx.H* and the generic AMX include file *CJZZZ.H* must be present in the current directory together with your source file being compiled.

Use the following compilation switches when you are compiling modules for use in the AMX environment.

<i>by default</i>	; no stack checking
<i>by default</i>	; target processor is big endian
<i>-c</i>	; compile only
<i>-proc 68000</i>	; generic 68xxx target
<i>-sdata 0</i>	; small mutable data section is empty
<i>-nopic -nopid</i>	; no position independent code or data
<i>-intsize 4</i>	; use 32-bit integers
<i>-model far</i>	; use 32-bit addressing
<i>-Cpp_exceptions off</i>	; disable C++ exceptions
<i>-o</i>	; output object module <i>FILENAME.O</i>
<i>&gt;FILENAME.ERR</i>	; redirect C error messages to <i>FILENAME.ERR</i>
<i>-Op</i>	; optimize for speed
<i>-g</i>	; (optional) generate debug information

The compilation command line is therefore of the form:

```
MWCC -c -proc 68000 -sdata 0 -nopic -nopid
      -intsize 4 -model far -Cpp_exceptions off -Op
      -o FILENAME.O FILENAME.C >FILENAME.ERR
```

If the command line becomes too long, use a command file. For example, create a text file *CCSW.CMD* which contains the following command string.

```
-c -proc 68000 -sdata 0 -nopic -nopid
-intsize 4 -model far -Cpp_exceptions off -Op
```

The compilation command line can then reduce to the form:

```
MWCC @CCSW.CMD -o FILENAME.O FILENAME.C >FILENAME.ERR
```

## Compiling the AMX System Configuration Module

Your AMX System Configuration Module *SYSCFG.C* is compiled as follows. All AMX header files *CJ532xxx.H* and the generic AMX include file *CJZZZ.H* must be present in the current directory together with file *SYSCFG.C*.

```
MWCC -c -proc 68000 -sdata 0 -nopic -nopid
      -intsize 4 -model far -Cpp_exceptions off -Op
      -o SYSCFG.O SYSCFG.C >SYSCFG.ERR
```

## Assembling the AMX Target Configuration Module

Your AMX Target Configuration Module *HDWCFG.S* is assembled as follows. The generic AMX header file *CJZZZK.DEF* must be present in the current directory together with file *HDWCFG.S*.

The Metrowerks assembler requires the following command line switches.

```
by default      ; assemble with case sensitivity
by default      ; target processor is big endian
by default      ; assemble only with generic 68000 code
-o              ; output object module HDWCFG.O
>HDWCFG.ERR     ; redirect assembler error messages to HDWCFG.ERR
```

The Metrowerks command to invoke the assembler is as follows.

```
MWASM -o HDWCFG.O HDWCFG.S >HDWCFG.ERR
```

## Making Libraries

To make a library from a collection of object modules, create a library specification file *YOURLIB.LBM*. Use the Metrowerks version of the AMX library specification file *CJ532.LBM* as a guide.

Use the following command line switches when using the Metrowerks linker/librarian.

```
-library        ; create a library
-o YOURLIB.A    ; output library module YOURLIB.A
>YOURLIB.LBE    ; redirect librarian error messages to YOURLIB.LBE
```

Make your library as follows.

```
MWLD -library -o YOURLIB.A @YOURLIB.LBM >YOURLIB.LBE
```

## Linking with the Metrowerks Linker

When used with Metrowerks C, the modules which form your AMX system must be linked in the following order.

Your *MAIN* module

Other application modules

<i>SYSCFG.O</i>	; AMX System Configuration Module
<i>HDWCFG.O</i>	; AMX Target Configuration Module
<i>CHxxxxxT.O</i>	; AMX chip-specific clock driver or your equivalent
<i>CJ532UC.O</i>	; AMX minimal C replacement library ; (used to eliminate <i>sprintf</i> I/O support from C library)
<i>CJ532UF.O</i>	; Launch and leave AMX (may be customized)
<i>CJ532RAC.O</i>	; AMX ROM Access Module (customized) ; (only if AMX placed in a separate ROM) ; (see Appendix C in AMX 68000 Target Guide)
<i>CJ532CV.A</i>	; AMX 68000 vc Conversion Library ; (only if converting an AMX 86 v3, AMX 386 v1 or ; AMX 68000 v2 application)
<i>CJ532.A</i>	; AMX 68000 Library ; Metrowerks C Runtime Libraries are automatically loaded ; per environment variable <i>MWLibraryFiles</i>

Create a link specification file *YOURLINK.LKS*. Use the Metrowerks version of the AMX Sample Program link specification file *CJSAMPLE.LKS* as a guide.

Create a linker command file *YOURLINK.LCF*. Use the Metrowerks version of the AMX Sample Program linker command file *CJSAMPLE.LCF* as a guide.

Start with the sample link specification file and linker command file for the board which most closely resembles your hardware configuration.

Note

If you decide to omit any of the link and locate commands from the sample specification, you may encounter link errors or run-time faults.

Link and locate with the Metrowerks linker and locator using the following command line switches.

by default	; target processor is big endian
by default	; main entry point is at symbol <i>__start</i>
<i>-proc 68000</i>	; generic 68xxx target
<i>-nopic -nopid</i>	; no position independent code or data
<i>-map unused</i>	; direct section and symbol summary to ; file <i>YOURLINK.OUT.XMAP</i>
	; include unused symbols in map file output
<i>-g</i>	; (optional) add debug information to the output to file
<i>-srec</i>	; direct <i>HEX</i> output to file <i>YOURLINK.OUT.S19</i> ; generate Motorola S-record format
<i>-o</i>	; direct link output to file <i>YOURLINK.OUT</i> ; use linker command file <i>YOURLINK.LCF</i>
@	; use link specification file <i>YOURLINK.LKS</i>
>	; direct link error messages to file <i>YOURLINK.LKE</i>

To avoid an overlength command line, create a command file *LDSW.CMD*, a text file containing the command line switches which you require.

```
-proc 68000 -nopic -nopid -map unused -srec
```

The link and locate command line is then of the form:

```
MWLD @LDSW.CMD -o YOURLINK.OUT  
YOURLINK.LCF @YOURLINK.LKS >YOURLINK.LKE
```

The resulting load module *YOURLINK.OUT* is suitable for use with the Metrowerks CodeWarrior 68000 debugger.

The resulting load module *YOURLINK.OUT.S19* is ready for burning into EPROM.

## Linking a Separate AMX ROM

AMX can be committed to a separate ROM as described in Appendix C of the AMX Target Guide. Use the AMX Configuration Manager to edit your Target Parameter File *HDWCFG.UP* to define your ROM option parameters. Then use the Manager to generate your ROM Option Module *CJ532ROP.S*, ROM Access Module *CJ532RAC.S* and ROM Option linker command file *CJ532ROP.LCF*.

The AMX Configuration Manager must have access to the ROM Option Linker Command Template file *CJ532ROP.LCT*. If you have installed AMX for multiple toolsets, the Manager may not be referencing the Metrowerks toolset directory *TOOLME* for its template files. Go to the **F**ile, **T**emplates... menu and, from the list of selectors, choose the selector for the ROM Option Link/Locate File. Adjust the configuration template by browsing for the file *TOOLME\CFG\CJ532ROP.LCT*.

The ROM Option and ROM Access source modules are assembled as follows.

```
MWASM -o CJ532ROP.O CJ532ROP.S >CJ532ROP.ERR
```

```
MWASM -o CJ532RAC.O CJ532RAC.S >CJ532RAC.ERR
```

The AMX ROM is linked using linker command file *CJ532ROP.LCF* and link specification file *CJ532ROP.LKS* as follows.

```
MWLD -proc 68000 -nopic -nopicid -map unused -srec  
-nostdlib -main "_cjksender" -o AMXROM.OUT  
CJ532ROP.LCF @CJ532ROP.LKS >AMXROM.LKE
```

This example generates file *AMXROM.OUT.S19* in Motorola S-record format suitable for transfer to ROM.

Note that command line switch *-main "\_cjksender"* is used to prevent loading of the default C startup module. Command line switch *-nostdlib* is used to prevent loading of the default C runtime library.

When you link your AMX application, be sure to include your customized AMX ROM Access Module *CJ532RAC.O* (created above) in your system link specification file.

## Using the AMX Configuration Generator

If you cannot use the AMX Configuration Manager, you may still be able to use the stand-alone AMX Configuration Generator to generate the ROM Option Module *CJ532ROP.S*, ROM Access Module *CJ532RAC.S* and ROM Option linker command file *CJ532ROP.LCF*.

Copy the ROM Option and ROM Access template files *CJ532ROP.CT* and *CJ532RAC.CT* to the current directory. Also copy the ROM Option Linker Command Template file *CJ532ROP.LCT* to the current directory.

Use the AMX Configuration Generator to generate the ROM option source modules as follows.

```
CJ532CG HDWCFG.UP CJ532ROP.CT CJ532ROP.S
CJ532CG HDWCFG.UP CJ532RAC.CT CJ532RAC.S
CJ532CG HDWCFG.UP CJ532ROP.LCT CJ532ROP.LCF
```

Once the ROM option source modules have been created, you can proceed to build your AMX ROM image and your AMX application as previously described.

## Metrowerks CodeWarrior Debugger

The Metrowerks CodeWarrior<sup>®</sup> 68000 Debugger supports source level debugging of your AMX 68000 system.

The CodeWarrior Debugger can operate by using a BDM connection to the 68000 target.

The CodeWarrior Debugger can also operate using a serial (or other) connection to the target 68000 system under test. When used in this fashion, you must install the CodeWarrior *MetroTRK* Target Resident Kernel in your target hardware. Instructions for doing so are provided in the CodeWarrior Reference Manual. Your version of the Target Resident Kernel must provide a device driver for the serial (or other) device used for communication with the CodeWarrior Debugger. It is recommended that your driver use polled I/O so that the Target Resident Kernel can operate with interrupts disabled.

## Using the *KwikLook* Fault Finder

The *KwikLook*<sup>™</sup> Fault Finder is compatible with the CodeWarrior Debugger providing full screen, source level, task-aware debugging from within the Microsoft Windows<sup>®</sup> environment. *KwikLook* can be invoked directly from the debugger while at breakpoints giving you finger tip access to your application from the AMX perspective. Note that *KwikLook* and CodeWarrior share a common link to the target system.

This page left blank intentionally.