



AMX™ 386/ET Tool Guide

First Printing: October 15, 2001

Last Printing: March 1, 2005

Copyright © 1994 - 2005

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

**Copyright © 1994-2005 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 386/ET TOOL GUIDE
Table of Contents

	Page
1. Selecting a Tool Set	1-1
2. Paradigm (PD) Tool Guide	2-1

This page left blank intentionally.

1. Selecting a Tool Set

Available Toolsets

AMX™ 386/ET 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 386/ET.

Note that AMX 386/ET is delivered to you ready to use with each of the supported toolsets. Should you wish to rebuild the AMX 386/ET 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.

Vendor	C/C++	Assembler	Librarian	Linker	Locator	Debugger
Paradigm C++	<i>PCC32</i>	<i>PASM32</i>	<i>PLIB</i>	<i>PLINK32</i>	<i>LOCATE32</i>	<i>via IDE</i>

Supported Toolsets

Unfortunately you cannot arbitrarily use any combination of the listed tools. Of all the tools listed, KADAK has identified combinations which can be used with AMX 386/ET. 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 two character mnemonic called a **toolset id** which is used by KADAK to identify the toolset combination. The two characters of the mnemonic identify the compiler vendor.

Compiler

PD Paradigm Systems 32-Bit C++

The following toolset combinations are supported by KADAK.

Toolset id:	PD
Vendor:	Paradigm
C++	<i>PCC32</i>
Assembler	<i>PASM32</i>
Librarian	<i>PLIB</i>
Linker	<i>PLINK32</i>
Locator	<i>LOCATE32</i>
Debugger	<i>via IDE</i>

2. Paradigm (PD) Tool Guide

AMX™ 386/ET has been developed on a PC with Windows® NT v4.0 using the Paradigm Systems software development 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.

Tools		<u>v6.0</u>
<i>PCC32</i>	Paradigm C++ compiler	v6.0
<i>PASM32</i>	Paradigm assembler (Note 1)	v5.0
<i>PLIB</i>	Paradigm Librarian	v6.0
<i>PLINK32</i>	Paradigm Linker	v6.0
<i>LOCATE32</i>	Paradigm Locator	v6.0
<i>via IDE</i>	Paradigm Debugger	v6.0
	Paradigm <i>PDREMOTE/ROM</i> Target Monitor	

Note 1: The assembler provided with Paradigm v5.0 and v6.0 tools identifies itself as v5.0.

AMX 386/ET and *KwikLook* have been tested on the following platforms.

80486 PC/AT
Pentium PC/AT

Environment Variables

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

<i>CJPATH</i>	Path to AMX installation directory (. . . \AMX722)
<i>PATH</i>	Path to AMX and Paradigm executable programs
<i>INCLUDE</i>	Path to all Paradigm include header files
<i>LIB</i>	Path to all Paradigm object files and libraries
<i>TMP</i>	Path to a temporary directory for use by tools

Object Formats

Paradigm generates the OMF object format. The AMX 386/ET libraries and object modules are provided in OMF format. Your object modules and the AMX and Paradigm libraries and object modules, all in OMF format, can be combined to create an executable module in Portable Executable (PE) format suitable for use with the Paradigm IDE debugger.

Parameter Passing Conventions

AMX 386/ET follows the stack based parameter passing convention common to all toolsets supported by KADAK.

Register Usage

The Paradigm version of AMX 386/ET makes the following C interface register assumptions. Registers *EAX*, *EBX*, *ECX* and *EDX* can always be altered by C procedures. Registers *ESI*, *EDI*, *EBP*, *ESP* and all selector registers are preserved by AMX 386/ET according to the Paradigm rules for C procedures. Integers and pointers are returned from C procedures in register *EAX*. The *DS* register is dedicated for access to the global data segment. All code which calls AMX or which is called by AMX must reside in the common code segment referenced by *CS*. You must NOT use any C compilation switch which changes these register assumptions.

Segmentation Assumptions

AMX 386/ET assembler modules do NOT use *ER* and *RW* segment attributes since the Paradigm assembler does not require them. Data segments are given the class name '*DATA*' and are grouped into group *DGROUP*. Code segments are given the class name '*CODE*' and are NOT grouped into group *CGROUP*.

IDE or Command Line

The Paradigm C++ Integrated Development Environment (IDE) provides a Windows® environment in which the Paradigm tools can be used to create and test your AMX application. The tools can also be invoked directly from the Windows command prompt. This tool guide describes the use of the command line tools, identifying the particular command line switches required for construction of your AMX application.

Most developers will prefer to use the Paradigm IDE in which command line switches are replaced by IDE option settings. Browse the HTML manual *AMX_PD.HTM* in installation directory *AMX722\MANUALS\TOOLPD\IDE*. It describes how the Paradigm project *CJSAMPLE.IDE* in directory *AMX722\TOOLPD\IDE* was created and then used to build the AMX 386/ET Sample Program.

Compiling the AMX System Configuration Module

Your AMX System Configuration Module *SYSCFG.C* is compiled as follows. All AMX header files *CJ722xxx.H* and the generic AMX include file *CJZZZ.H* must be present in the current directory together with file *SYSCFG.C*.

```
PCC32 -c -f- -O2 -w-aus -w-par -w-pro 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*.

```
PASM32 /ML /N /T HDWCFG.ASM
```

Making Libraries

To make a library from a collection of object modules, create a library specification file *YOURLIB.LBM*. Use the Paradigm version of the AMX library specification file *CJ722.LBM* as a guide.

Make your library as follows.

```
PLIB @YOURLIB.LBM
```

Using the Paradigm C++ Compiler

All AMX header files *CJ722xxx.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 stack
by default	; output object module <i>FILENAME.OBJ</i>
by default	; use flat model
<i>-c</i>	; compile but do not link
<i>-f-</i>	; (optional) no floating point
<i>-O2</i>	; (optional) use default optimization
<i>-v</i>	; (optional) generate debug information

The compilation command line is therefore of the form:

```
PCC32 -c -f- -O2 FILENAME.C
```

The following compilation switches are also used when compiling AMX modules to suppress unavoidable warning messages generated by the Paradigm compiler.

<i>-w-aus</i>	; suppress warning for unused variables
<i>-w-par</i>	; suppress warning for unused function parameters
<i>-w-pro</i>	; suppress warning for function pointer without prototype

Linking with the Paradigm Linker

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

C0X32.OBJ ; Paradigm C++ startup module
Your *MAIN* module
Other application modules

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

AMX PC/AT or Intel386EX Evaluation Board clock driver:
AT386BRD.OBJ ; AMX PC/AT board support module
CH8253T.OBJ ; AMX 8253 clock driver or your equivalent

CJ722UF.OBJ ; Launch and leave AMX (may be customized)

CJ722RAC.OBJ ; AMX ROM Access Module (customized)
; (only if AMX placed in a separate ROM)
; (see Appendix C in AMX 386/ET Target Guide)

CJ722CV.LIB ; AMX 386/ET vc Conversion Library
; (only if converting an AMX 86 v3, AMX 386 v1 or
; AMX 68000 v2 application)

CJ722.LIB ; AMX 386/ET Library
RTTHEAP.LIB ; Paradigm C++ Runtime Libraries for target hardware
NOEH32.LIB
CW32.LIB
EMBED32.LIB

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

Create a locate specification file *YOURLINK.CFG*. Use the Paradigm version of the AMX Sample Program locate specification file *CJSAMPLE.CFG* as a guide.

Start with the sample link and locate 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, you may encounter link errors or run-time faults.

Link with the Paradigm linker using the following linker options on the command line or in the link specification file.

<i>-Tpe</i>	; generate intermediate PE file <i>YOURLINK.EXE</i>
<i>-c</i>	; symbols are case sensitive
<i>-m</i>	; create a segment and symbol map file <i>YOURLINK.MAP</i>
<i>-v</i>	; (optional) add debug information for IDE debugger use

The link command line for toolset *PD* is therefore of the form:

```
PLINK32 -Tpe -c -m @YOURLINK.LKS
```

The resulting load module *YOURLINK.EXE* must be converted to a form suitable for use with the Paradigm IDE debugger. Locate with the Paradigm *LOCATE32* tool using the following command line options.

by default	; create a segment and symbol map file <i>YOURLINK.LOC</i>
<i>YOURLINK</i>	; use configuration file <i>YOURLINK.CFG</i>
by default	; generate output files <i>YOURLINK.AXE</i> and <i>YOURLINK.RTB</i>
	; generate image file <i>YOURLINK.HEX</i> if specified by
	; configuration file <i>YOURLINK.CFG</i>

The locate command line for toolset *PD* is therefore of the form:

```
LOCATE32 YOURLINK
```

The resulting executable modules *YOURLINK.AXE* and *YOURLINK.RTB* are ready for use with the Paradigm IDE debugger. The resulting load module *YOURLINK.HEX* is ready for burning into ROM.

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 *CJ722ROP.ASM*, ROM Access Module *CJ722RAC.ASM* and ROM Option locate specification file *CJ722ROP.CFG*.

The AMX Configuration Manager must have access to the ROM Option Locate Template file *CJ722ROP.CFT*. If you have installed AMX for multiple toolsets, the Manager may not be referencing the Paradigm toolset directory *TOOLPD* 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 *TOOLPD\CFG\CJ722ROP.CFT*.

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

```
PASM32 /ML /N /T CJ722ROP.ASM
```

```
PASM32 /ML /N /T CJ722RAC.ASM
```

The AMX ROM is linked and located with the Paradigm linker and locator using the AMX ROM Option link specification file *CJ722ROP.LKS* and locate specification file *CJ722ROP.CFG* as follows.

```
PLINK32 -Tpe -c -m @CJ722ROP.LKS  
LOCATE32 CJ722ROP
```

This example generates files *CJ722ROP.AXE* and *CJ722ROP.RTB* in the file format supported by the Paradigm IDE debugger. The resulting load module *CJ722ROP.HEX* is ready for burning into ROM.

When you link your AMX application, be sure to include your customized AMX ROM Access Module *CJ722RAC.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 *CJ722ROP.ASM*, ROM Access Module *CJ722RAC.ASM* and ROM Option locate specification file *CJ722ROP.CFG*.

Copy the ROM Option and ROM Access template files *CJ722ROP.CT* and *CJ722RAC.CT* to the current directory. Also copy the ROM Option Locate Specification Template file *CJ722ROP.CFT* to the current directory.

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

```
CJ722CG HDWCFG.UP CJ722ROP.CT CJ722ROP.ASM
CJ722CG HDWCFG.UP CJ722RAC.CT CJ722RAC.ASM
CJ722CG HDWCFG.UP CJ722ROP.CFT CJ722ROP.CFG
```

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.

Paradigm IDE Debugger

The Paradigm IDE debugger is a Windows® based debugger which supports source level debugging of your AMX 386/ET system.

The Paradigm debugger can operate using an in-circuit emulator connected to your 80x86 target hardware. Check with Paradigm to determine which emulators are supported.

When used with hardware assisted breakpointing, your target processor is effectively halted while at breakpoints.

The Paradigm IDE debugger can also operate using a serial (or other) connection to the system under test. When used in this fashion, you must install the Paradigm *PDREMOTE/ROM* Target Monitor in your target hardware. Instructions for doing so are provided by Paradigm. Your version of the target monitor must provide a device driver for the serial (or other) device used for communication with the host debugger. It is recommended that your driver use polled I/O so that the target monitor can operate with interrupts disabled.

Using the *KwikLook* Fault Finder

The *KwikLook*™ Fault Finder is compatible with the Paradigm IDE 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 the Paradigm debugger share a common link to the target system.

This page left blank intentionally.