Microcross GNU X-Tools™

Installation Guide

Version 1.0

Open Source, Ready-to-Run™ …

ISBN: N/A

Microcross, Inc.
151 Osigian Blvd., Suite 154, Warner Robins, GA 31088

GNU X-Tools Installation Guide
Supports Product Part Numbers: MC-XT-100-1/-2/-3/-4/-5/-6, November 2000

Contents

Introduction to GNU X-Tools

GNU X-Tools is a pre-built version of the GNU GCC C/C++ compiler and supporting development tools for embedded software development within a Unix (or GNU/Linux) host environment. GNU X-Tools provides a quick start environment for developers desiring to use or evaluate the GCC compiler tools suite as a cross-development system, by eliminating the build from source, debug, and installation process. GNU X-Tools provides complete edit, compile, build, and source debug environment for most popular 32-bit embedded targets. Each of the 21 toolchains with support tools operate in the Unix shell environment, which is the most mature and widely understood computing user-interface used by professional software developers.

The GNU X-Tools Cross Development suite is available for three supported host environments:

a) Linux 2.x – Installs directly onto any Linux host environment running Linux 2.x kernel, glibc-6, and having native software development tools preinstalled (i.e., gcc, g++, libraries, headers, etc.).

b) Cygwin – Cygwin is a Unix/POSIX compatibility layer for Microsoft Windows operating systems, which creates an i386 POSIX API compatibility layer over Win32, allowing versions of Unix/POSIX tools to be built and run on Windows 9x, NT, and Win-2k hosts. This unique host environment, while not as robust as true Unix/Linux, provides the advantage of supporting software development in a Unix/POSIX-like development environment while simultaneously running tools and applications that require the Windows operating environment. In a sense, this operating environment provides the best of both Unix and Windows worlds at the expense of some loss of fidelity within the POSIX API. This is generally not a severe limitation for embedded development, because the target code often uses only a small subset of (if any) of the host API. For example, many embedded applications require no file system services.

c) Solaris – Installs directly onto preinstalled versions of Solaris 8 or higher.

There are currently two versions of the Cygwin compatibility layer available. Cygwin B20 (pronounced beta 20) is the most recent unrestricted version and is available for download from the Red Hat/Cygnus ftp site. Because of its unrestricted nature, a complete image of the B20 version is provided on the GNU X-Tools 1.0 (Cygwin B20) CD-ROM and is installed by the Microcross GNU X-Tools Installation Manager.

The second Cygwin version, 1.0, is a commercial release in which the compatibility layer windows dll has restricted rights. This product, available from Red Hat and its resellers for approximately $100, has improved functionality over the B20 version and includes many additional useful tools and utilities such as Xemacs, the popular GNU Swiss-Army knife of text editors and Cygnus Insight, a source level native debugger. Unfortunately, Red Hat made significant changes to the internal API layer and as a result executable programs for the two Cygwin versions are not directly compatible. For this reason, a separate GNU X-Tool version has been produced for each Cygwin version. Because of the commercial status of the Cygwin 1.0 version, the GNU X-Tools 1.0 for Win32 (Cygwin 1.0) requires purchase and installation of Cygwin 1.0 as a prerequisite to installation of the cross toolchains.

The Cygwin Unix/Linux environment is fully documented in the Cygwin 1.0 *Getting Started* Manual, which is included with the Cygwin 1.0 commercial distribution or on the web at http://www.sourceware.cygnus.com/cygwin.

The Cygwin operating environment is the GNU bash shell and GNU utilities, so using the Cygwin shell environment will be very familiar if you have experience with Unix, Linux, or other GNU environment (precisely the reason for using it). If you are new to the Unix environment, a Unix text such as reference 15,18 (bibliography in the GNU X-Tools User Guide) will accelerate the familiarization process. The Cygwin Reference Manual accompanies the Cygwin 1.0 release and is another good reason to purchase this product.

Solaris
A GNU X-Tools release in Solaris 8 hosted configuration is planned. This release will extend the unified host and target development environment to the Solaris workplace.

Target Microprocessors

The array of target microprocessors supported by GNU X-Tools is based on the current options available in the current GNU tools source tree for the member tools. The objective of GNU X-Tools Release 1.0 is to provide a baselined controlled tools suite for each target platform and each host. From this baseline, new hosts, targets, and target enhancements will be added in future releases. Because of incomplete or missing options, the capabilities of each target suite are not always uniform. For example, some toolchains do not include a CPU simulator as part of the debugger. Table 1 enumerates each of the targets supported in GNU X-Tools and notes the extent of partial functionality for target chains that are incomplete.

Table 1. Target Microprocessors Supported by GNU X-Tools

No.	Alias	Target Manufacturer and Description	Notes
1	arc-elf	ARC – Argonaut RISC Cores 32-bit RISC Core	4
2	arm-coff	ARM – Advanced RISC Machines 32-bit Processor Family	
3	d10v-elf	Mitsubishi D10V 32-bit Multimedia Processor	2
4	h8300-hms	Hitachi H8/300, H8S, H8/300H 8/16/32-bit CISC Processor	5
5	i386-coff	Intel x86 standalone 32-bit Flat Model CISC Processor	4, 6
6	i960-coff	Intel i960 32-bit RISC Processor	
7	m32r-elf	Mitsubishi M32R DSP 32-bit Multimedia Processor	6
8	m6811-elf	Motorola m6811 8/16-bit Microcontroller	3, 5
9	m68k-coff	Motorola m68k, CPU32, Coldfire 32-bit CISC/RISC Processor	4, 6
10	m88k-coff	Motorola m88k 32-bit RISC Processor	4
11	mips-elf	MIPS 32-bit RISC Processor Family	
12	mn10200-elf	Matsushita MN10200 Series (Panasonic) 16-bit CISC Processor	5
13	mn10300-elf	Matsushita MN10300 Series (Panasonic) 32-bit CISC Processor	
14	ns32k-aout	National Semiconductor NS32FX200 Series 32000 32-bit RISC Processor	4
15	ppc-elf	Motorola Power PC 32-bit RISC Processor	
16	sh-hms	Hitachi SH/2, SH/3 32-bit RISC Processor	6
17	sparc-elf	SPARC 32-bit RISC Processor Family	6
18	tic30-aout	Texas Instruments TMS 320 C30 32-bit Floating Point DSP Processor	1
19	v850-elf	NEC V85x 32-bit RISC Processor	
20	w65-coff	Rockwell/WDC (W65C816, W65C02) 8/16-bit Microcontroller	1
21	z8k-coff	Zilog Z8000 16/32-bit Processor	2

Notes
1. Assembly tools only.
2. Assembly tools, debugger, and simulator only.
3. No C++ compiler or libraries.
4. No simulator.
5. 16-bit and no floating point support.
6. Contains remote debugging stub template.

System Requirements and Prerequisites

GNU X-Tools provides a GNU Unix/Linux like development environment (on Win32 hosts, too) and presumes that the user has a working knowledge of Unix command line utilities and software tools. It is ideally suited for developers who have prior experience with the GNU C/C++ compiler as a native development system. If you need additional familiarization with the Unix environment, a wide variety of publications are available, several of which are listed in the bibliography. The O'Reilly text, *Programming with GNU Software* (Loukides and Oram), is included with the GNU X-Tools and is an excellent resource for the programmer.

System Requirements (Windows Version)

1) Intel architecture (i586) PC running one of the following Microsoft operating systems:
   - Windows 2000 Professional (or higher)
   - Windows NT 4.0 Workstation, with Service Pack 4 or higher
   - Windows 98, Second Edition (or higher)
   - Windows versions not meeting Y2k compliance are not recommended or supported.

2) CPU Clock Rate:
   - 133 MHz or higher

3) Minimum System RAM:
   1. 128 MB (Windows NT/2000)
   2. 64 MB (Windows 98)

4) Free Disk Space:
   - 70 MB (for installation of Cygwin B20)
   - 130 MB (for installation of Cygwin 1.0)
   - 60 MB average per installed target (see target Appendix for specific requirements)
   - 1.2 GB for all targets
   - 2-2.5 GB if building from sources

Note: If installing the Cygwin 1.0 version of GNU X-Tools, you must purchase and install Cygwin 1.0 prior to installing GNU X-Tools. The Cygwin B20 version of GNU X-Tools includes Cygwin B20 and can be installed through the GNU X-Tools Installation Manager.

System Requirements (Linux Version)

1) Intel architecture (i586) PC running a Linux 2.x kernel and glibc ver5.

2) CPU Clock Rate:
   - 133MHz or higher

4) Minimum System RAM:
   - 64 MB

5) Free Disk Space:
   - 50 MB average per installed target
   - 1.2 GB for all targets
   - 2-2.5 GB if building from sources

See the appropriate target appendix for specific requirements by target. The GNU X-Tools suite has been installed, built and tested on the following Linux distributions:

- Red Hat Linux Ver. 6.0
- Red Hat Linux Ver. 6.1
- Mandrake Linux Ver. 6.2

INSTALLATION PROCEDURES

GNU X-Tools for Win32 (Cygwin 1.0) Installation Procedures

GNU X-Tools for Win32 (Cygwin 1.0) version requires the user to purchase (from Red Hat or its resellers) and install Cygwin 1.0. GNU X-Tools for Cygwin 1.0 CD-ROM comes complete with a custom installer that gives the user an easy means of installing software. Important: Cygwin 1.0 must be installed prior to installation of GNU X-Tools.

Steps

1) Insure that the system requirements have been met, especially RAM and free disk space. On Windows NT/2000 hosts, insure that you are logged in as administrator, or have full permissions to the root directory of the destination disk volume.

2) Insert the Microcross GNU X-Tools CD into your PC CD-ROM. Run GNU X-Tools Installation Manager (setup.exe); click on *Install* and read the license agreement; then, click on *Accept* and *Next*. Now begin the installation process by clicking on *Install Resource Files* (Step 1).

3) Next, click on *Create Shell Shortcut* (puts a shortcut on the desktop). If you desire to change the icon to the GNU X-Tools icon, click on the MS-DOS icon (upper left corner of shell window) and select properties. Click on *Change Icon* and browse to the GNU X-Tools Cygwin root directory and select *Xtools.ico* (e.g., <drive>:\Cygwin\Xtools.ico) and click on *Open* and *OK* and *OK* once more. You should have the GNU X-Tools icon now on the desktop.

4) This step is optional if you do not already have Acrobat 4.0 Reader installed. If you have an older version of Acrobat Reader, uninstall it first before installing the newer version. You may install Acrobat Reader later if you prefer.

5) Next, click on *Install Docs* to install the documentation for GNU X-Tools, VIDE, and GNU utilities. If the user desires to install only a subset of the documentation, browse the GNU X-Tools CD *docs* directory and copy the desired *pdf* files to your hard drive.

6) Next, click on *Install VIDE* and follow the on-screen directions. Recommend installing VIDE into the default directory: *c:\Program Files\vide.*

7) Next, install toolchains. Click on the drop-down combo-box to select the toolchain that you want to install and then click on *Install Toolchain*. Repeat this step for each desired target toolchain. You will need approximately 1.2 GB of disk space to install all 21-target toolchains.

8) Exit GNU X-Tools Installation Manager and setup the environment for GNU X-Tools. For VIDE and GNU X-Tools to work properly, you *must* have your Windows PATH and Make Mode environment variables set correctly. Follow the instructions below to set the PATH on your host platform.

Windows 98
To add the VIDE and GNU X-Tools Path to Windows 98, edit the file "<drive>:\autoexec.bat" (normally this file is located in "C" drive). Simply add the compiler's binary directory (i.e., <drive>:\Cygwin\bin) and VIDE directory to the PATH command, and add the Make mode variable setting and **re-boot** the computer. The GNU make, version 3, has a mode that allows the GNU make to be compatible with Microsoft Windows. The make mode variable controls the compatibility mode; therefore, it must be set on the Windows host operating system before building software using GNU X-Tools.

Example:
```
SET MAKE_MODE=Unix
SET PATH=c:\cygwin\bin;c:\"Program Files"\Vide; -- add other paths after these
```

Note that the quotes around Program Files are needed because the space between the words is not recognized in Windows 98 (DOS).

Windows NT/2000
On NT/2000, right click on the My Computer icon on the desktop to get to the *System Properties* Tab; click on *Environment* tab and then click on *Path* to change the PATH in the environment. Enter the *Path* settings.

Example of Path Settings
Enter "c:\Program Files\Vide; c:\cygwin\bin;"

In addition to path settings the user needs to insert an environment variable on the same *System Properties* Tab; click on *Environment* tab and enter the following variable and value:

Example
Variable: MAKE_MODE
Value: Unix

All path settings in VIDE and the GNU X-Tools command line make environment will use forward "/" slashes as opposed to back "\" slashes.

Important Notes
Do not delete any of the other path settings – use ";" to separate path settings. In Windows NT you do not need to re-boot the computer; however, if you are in VIDE, you must exit and startup again.

Once installed and setup (and rebooted if necessary), the user may go to the desktop and double click on the GNU X-Tools icon created during the installation to get Bash Shell operating. You must login as Administrator if you are installing for a multi-user environment.

9) Last, go to setup VIDE and your first project.

Note: The GNU X-Tools Installation Manager has an Uninstaller utility that can be run at any time from the CD-ROM, which will allow the user to uninstall one or more toolchains. You should not uninstall Cygwin 1.0 through the windows utility until you have uninstalled all of the toolchains that are installed.

GNU X-Tools for Win32 (Cygwin B20) Installation Procedures

GNU X-Tools for Win32 (Cygwin B20) comes with the Cygwin B20 application layer for Windows 98/NT/2000. The product comes complete with a custom installer that gives the user an easy means of installing/uninstalling the software. Below are the installation steps.

Steps

1) Insure that system requirements have been met, especially RAM and free disk space. On Windows NT/2000 hosts, insure that you are logged in as administrator, or have full permissions to the root directory of the destination disk volume.

2) Insert the Microcross GNU X-Tools CD into your PC CD-ROM. Run GNU X-Tools Installation Manager (setup.exe); click on *Install* and read the license agreement; then, click on *Accept* and *Next*. Now begin the installation process by clicking on *Install Cygwin B20* (Step 1) and follow the on-screen directions. The user will be prompted to insert a drive letter for the Cygwin directory. Select a drive that has enough disk space to meet installation requirements.

3) Next, click on *Create Shell Shortcut* (puts a shortcut on the desktop). If you desire to change the icon to the GNU X-Tools icon, click on the MS-DOS icon (upper left corner of shell window) and select properties. Click on *Change Icon* and browse to the GNU X-Tools Cygwin root directory and select *Xtools.ico* (e.g., <drive>:\Cygwin\Xtools.ico) and click on *Open* and *OK* and *OK* once more. You should have the GNU X-Tools icon now on the desktop.

4) This step is optional if you do not already have Acrobat 4.0 Reader installed. If you have an older version of Acrobat Reader, uninstall it first before installing the newer version. You may install Acrobat Reader later if you prefer.

5) Next, click on *Install Docs* to install the documentation for GNU X-Tools, VIDE, and GNU utilities. If the user desires to install only a subset of the documentation, browse the GNU X-Tools CD *docs* directory and copy the desired *pdf* files to your hard drive.

6) Next, click on *Install VIDE* and follow the on-screen directions. We recommend installing VIDE into the default directory: *c:\Program Files\vide.*

7) Next, install toolchains. Click on the drop-down combo-box to select the toolchain that you want to install and then click on *Install Toolchain*. Repeat this step for each desired target toolchains. You will need approximately 1.2 GB of disk space to install all 21-target toolchains.

8) Exit GNU X-Tools Installation Manager and setup the environment for GNU X-Tools. For VIDE and GNU X-Tools to work properly, you *must* have your Windows PATH and Make Mode environment variables set correctly. Follow the instructions below to set the PATH on your host platform.

Windows 98
To add the VIDE and GNU X-Tools Path to Windows 98, edit the file "<drive>:\autoexec.bat" (normally this file is located in "C" drive). Simply add the compiler's binary directory (i.e., <drive>:\Cygwin\bin) and VIDE directory to the PATH command, and add the Make mode variable setting and **re-boot** the computer. The GNU make, version 3, has a mode that alows the GNU make to be compatible with Microsoft Windows. The make mode variable controls the compatibility mode; therefore, it must be set on the Windows host operating system before building software using GNU X-Tools.

Example:
```
SET MAKE_MODE=Unix
```

SET PATH=c:\cygwin\bin;c:\"Program Files"\Vide; -- add all other paths after these

Note that the quotes around Program Files are needed because the space between the words is not recognized in Windows 98 (DOS).

Windows NT/2000
On NT/2000, right click on the My Computer icon on the desktop to get to the *System Properties* Tab; click on *Environment* tab and then click on *Path* to change the PATH in the environment. Enter the *Path* settings.Example of Path Settings
Enter "c:\Program Files\Vide; c:\cygwin\bin;"

In addition to path settings the user needs to insert an environment variable on the same *System Properties* Tab; click on Environment tab and enter the following variable and value:
Example
Variable: MAKE_MODE
Value: Unix

All path settings in VIDE and the GNU X-Tools™ command line make environment will use forward "/" slashes as opposed to back "\" slashes.

Important Notes
Do not delete any of the other path settings – use ";" to separate path settings. In Windows NT you do not need to re-boot the computer; however, if you are in VIDE, you must exit and startup again.

Once installed and setup (and rebooted if necessary), the user may go to the desktop and double click on the GNU X-Tools icon created during the installation to get Bash Shell operating. You must login as Administrator if you are installing for a multi-user environment.

9) Last, go to Section setup VIDE and your first project.

Note: The GNU X-Tools Installation Manager has an Uninstaller utility that can be run at any time from the CD-ROM, which will allow the user to uninstall one or more toolchains or the entire Cygwin B20 and toolchain installation (remove all function). If you do not uninstall the Cygwin B20 in this manner, you will not be able to reinstall Cygwin due to the registry entries being preserved from a previous installation. Please uninstall Cygwin B20 from the CD-ROM utility provided with GNU X-Tools and you'll save yourself problems later on.

Manual Cygwin B20 Installation Procedures

The following steps will guide the user through installation of GNU X-Tools™ for Win32 (Cygwin B20) in directory "\Cygwin" and mount it as the pseudo root ("/") for the Cygwin shell.

Steps

1) Mount CD-ROM on CD drive; for example, *d:* -- most commonly assigned drive letter.

2) Create the Cygwin root directory on the desired installation drive (e.g., *mkdir c:\Cygwin* – Cygwin is the required path name).

3) Change directory to the Cygwin root (e.g., *cd c:\Cygwin*).

4) Unzip the Cygwin, *cyg-b20.zip* archive (e.g., *d:\win32\unzip d:\win32\cyg-b20.zip*).

5) Change to the bin directory and execute bash (e.g., *cd bin; ./bash –login*).

6) Remount the Cygwin root using two commands (e.g., *./umount /; ./mount –fc c:\\Cygwin /*).

7) Exit the Bash shell (e.g., exit).

8) Restart Bash shell clicking on *Start* | *Run* | *Open* and entering *c:\Cygwin\bin\bash –login* (if c: is your drive location, otherwise insert the proper drive letter). The Bash shell should restart, presenting the Microcross GNU X-Tools banner.

9) Create the desktop icon for the shell using same as normal procedure and enter *c:Cygwin\bin\bash.exe –login* into the shortcut Target Property.

GNU X-Tools for Linux Installation Procedures

The Linux installation procedure for GNU X-Tools is a manual procedure, but is simple and straightforward. Basically you must manually install the *rc* files into the */usr/bin* directory, and then all subsequent install/remove operations for the toolchains are executed using the *xtools* command tool from within the command shell.

Steps

1) Make sure that the host system prerequisites have been met, paying particular attention to the free disk space requirements (see Section 1.5.2), and the shared libraries version. You can check for presence of the required libraries by executing the command */sbin/ldconfig -p* | *less* and browsing the output, which describes all of the installed libraries and versions, and locations on your system. The required shared libraries are listed in Table 2.1.

2) Login as root (or use su root). You will want to do this each time you run an xtools install/remove because you need write access to the */usr/bin* and */usr/lib* directories where xtools is installed. At this point you are running a bash shell.

3) Mount the GNU X-Tools distribution CD on the CD-ROM drive using the mount command, Example:

```
mount -t iso9660 /dev/cdrom /mnt/cdrom
```

The system will announce that it has mounted the CD-ROM as read-only.

4) Install the GNU X-Tools command tool, and shell rc file by using the following commands:

```
cd /
tar –xvzf   /mnt/cdrom/bin/rcfiles.tgz
```

At this point you should be able to execute the GNU X-Tools command tool by its name. The command, *xtools (cr),* should invoke the abbreviated tool help screen that shows the basic operating commands (see Chapter 3.2 for a complete description of the command tool).

5) The xtools command, *xtools install,* will untar an individual toolchain, installing each of its components into the proper */usr/bin*, */usr/lib*, and */usr/<tgt-alias>* directories (see GNU X-Tools User Guide). The install command assumes the default archive source path is */mnt/cdrom/bin* using */mnt/cdrom* as the CD-ROM mount point. If your default mount point is different, you can change the default value by editing */usr/bin/xtools*, line 38.

6) To install an individual toolchain, simply execute the command *xtools install <tgt-alias>*, where target alias is one of the toolchain names listed in Table 1.1 . The command tool will extract the toolchain tarball, installing all files in their appropriate locations. If you would like to override the

default path to the distribution medium you may enter the command as *xtools install <tgt-alias> arc-path*. The arc path should always end in "bin" as the tar archives are in the bin directory on the CD-ROM.

7) At any later time you wish to remove a toolchain, you may do so by using the *xtools remove <tgt-alias>* command (remember to login as root or su to root). All toolchains may be removed by invoking the command *xtools remove-all go*.

8) Refer to Chapter 3 of the GNU X-Tools User Guide for familiarization with the GNU X-Tools.

Tables 2 shows the required shared object libraries used by GNU X-Tools (Linux 2.x version).

Table 2. Required Shared Object Libraries

libc.so.6	all
ld-linux.so.2	all
libm.so.6	gdb
libgpm.so.1	gdb
libncurses.so.5	gdb
libnsl.so.1	run (simulator)

Table 3 shows the GNU X-Tools tested configurations.

Table 3. Linux Test Configurations:

Mandrake	Linux Ver 6.1
RedHat	Linux Ver 6.0
RedHat	Linux Ver 6.1

Installing the Documentation Files

Many of the Manuals documented in the bibliography are provided in Adobe Acrobat format and are located in the */docs* directory of the distribution CD-ROM. They may be accessed directly from the CD-ROM using the Adobe Acrobat Reader (a copy is provided on the CD-ROM). Most Linux distributions have the Acrobat Reader included and it. The manuals may also be copied to an online file system, if desired.

Installing Example Programs and Sources

The example programs and toolchain source tarsets (tarball) are included in the src directory of the distribution CD-ROM. They may be extracted using the tar command as follows:

```
cd <srcroot>
tar -xvzf /mnt/cdrom/src/<srcname.tgz>
```

The *srcroot* is the pathname of the desired source root directory. The *srcname.tgz* is the name of the source archive. All of the archive pathnames are prefixed with a directory name, so each package will be segregated into a subdirectory of its own.

Installing VIDE, The “V” Integrated Development Environment

Procedures for installing VIDE: We recommend installing VIDE under */usr/*, which when extracted will be under */usr/vide*.

```
cd /usr/
tar –xvzf /mnt/cdrom/bin/vide-linux-118.tgz
```

To execute VIDE from the command line, type the path name on the command line (e.g., /usr/vide/vide). If you are using a desktop manager, create an icon and point the path to the executable.

Installation of Source Code

Installation of source code is a simple procedure. Location of installed source files is described in Appendix 1, Release Notes, para. 4 of the GNU X-Tools User Guide. All sources should be installed under a common source tree. You have the option of installing one or more source files if you desire. Procedures to untar the desired source files follows:

From within a bash/xtools shell, execute these commands:

```
cd /usr/src
tar –xvzf <CD-ROM path>/src/<file name>
```

Where <CD-ROM path> = /mnt/cdrom/ (Linux) or //<drive>/ (Windows/Cygwin), and <file name> = any of the source files in the src directory.

VIDE Setup

VIDE comes with the GNU X-Tools distribution. It works with the GNU gcc/g++ compiler, Borland’s BCC 5.5 compiler and Sun's JDK for Java. VIDE is currently at release 1.18 and is available in executable form for MS-Windows and Intel Linux platforms. The VIDE executable is totally freeware, and the source is available under the GNU GPL as part of the “V” GUI Framework. VIDE was developed by Dr. Bruce E. Wampler and associate developers. Before beginning the VIDE example setup and test, install the arm-coff (for verifying the example setup) or another toolchain through the Installation Manager or by manual procedure.

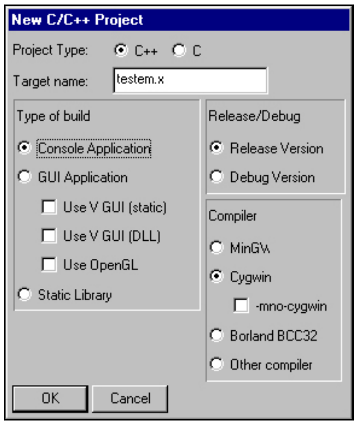
VIDE for Win32 (Cygwin B20/1.0) Example Setup

Setup of the VIDE environment for GNU X-Tools is a rather simple task. Follow the steps below and you should be compiling and debugging code in a short time.

Step

1. After installing VIDE, startup VIDE from the *Start* | *Programs* menu in Windows.

2. Click on *Project* | *New C++ Project*. Figure 1 shows the dialog window that is started. For this example, enter the information exactly as shown in the dialog and click *OK*.



New C/C++ Project

Project Type: [✓ C++] C

Target name: testem.x

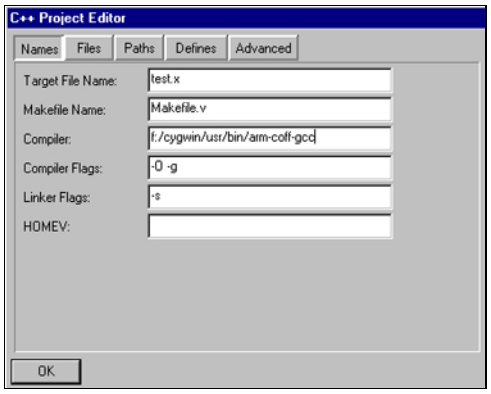
Type of build: [✓ Console Application] / GUI Application / Use V GUI (static) / Use V GUI (DLL) / Use OpenGL / Static Library

Release/Debug: [✓ Release Version] / Debug Version

Compiler: MinGW / [✓ Cygwin] / -mno-cygwin / Borland BCC32 / Other compiler

Figure 1. New C/C++ Project Dialog

3. Next, enter the information shown in the project editor, except substitute for the <drive> letter for the compiler directory – observe the forward "/" slashes. If you set up the environment correctly during the installation, then the Unix style paths should work. Compiler and Linker flags should be entered as shown for the arm-coff example; however, other targets may require different settings – see Compiler Options in the Appendices of this User Guide for the specific target toolchain you are working with.

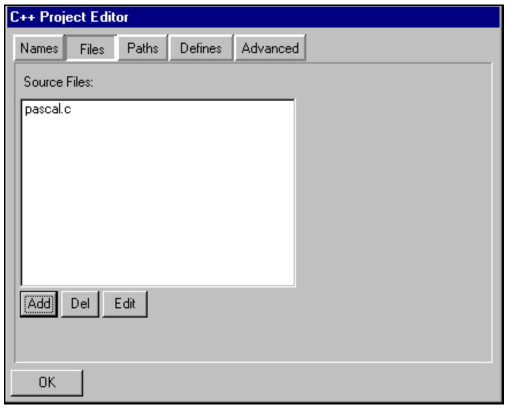


C++ Project Editor

Target File Name:	test.x
Makefile Name:	Makefile.v
Compiler:	f:/cygwin/usr/bin/arm-coff-gcc
Compiler Flags:	-O -g
Linker Flags:	-s
HOMEV:	

Figure 2. C++ Project Editor | Names Tabs Dialog

4. Next, click on *Files* tab. For this example, click on *Add* and browse to the *<drive>:\Cygwin\home\test* directory and select *pascal.c* as the test source file.



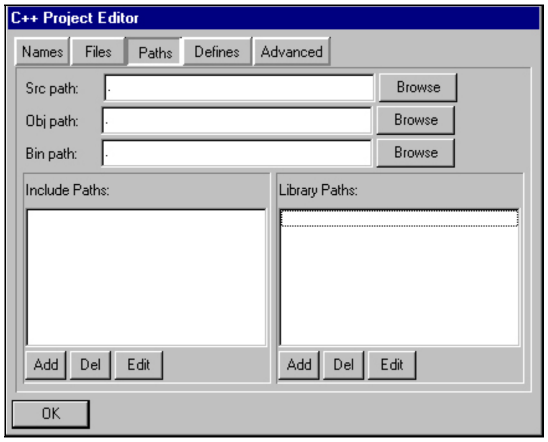
C++ Project Editor

Source Files:

pascal.c

Figure 3. C++ Project Editor | Files Tab Dialog

5. Click on *Path* tab. No entries are necessary for this example. Read the VIDE User Guide to get acquainted with these features (later).



Src path: .
Obj path: .
Bin path: .

Figure 4. C++ Project Editor | Paths Tab Dialog

6. Click on *Defines* tab. No entries are necessary for this example. Read the VIDE User Guide to get acquainted with these features (later).

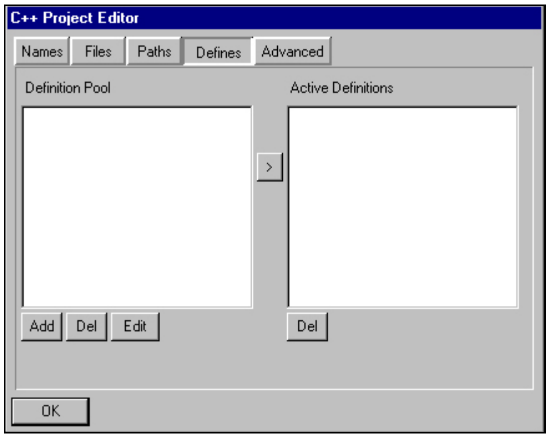
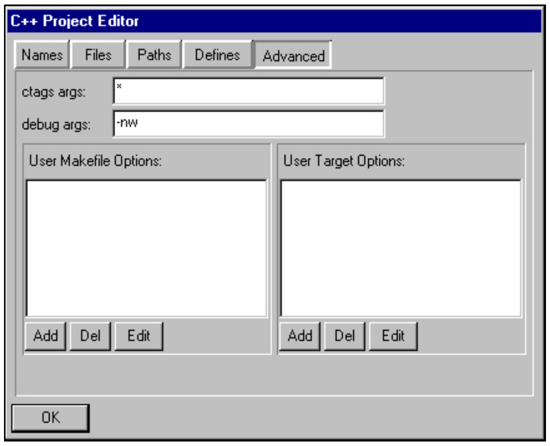


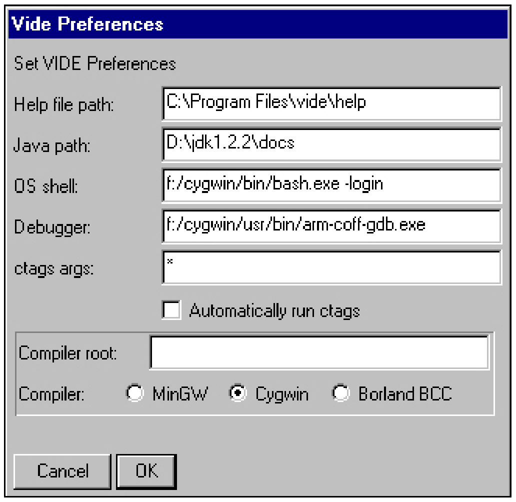
Figure 5. C++ Project Editor | Defines Tab Dialog

7. Click on *Advanced* tab. Again, no entries are necessary for this example. Click *OK* to finish and exit the *C++ Project Editor*. You may go back and edit this dialog at any time by clicking on *Project* | *Edit.*



ctags args: *
debug args: -nw

Figure 6. C++ Project Editor | Advanced Tab Dialog



Vide Preferences

Set VIDE Preferences

Help file path:	C:\Program Files\vide\help
Java path:	D:\jdk1.2.2\docs
OS shell:	f:/cygwin/bin/bash.exe -login
Debugger:	f:/cygwin/usr/bin/arm-coff-gdb.exe
ctags args:	*

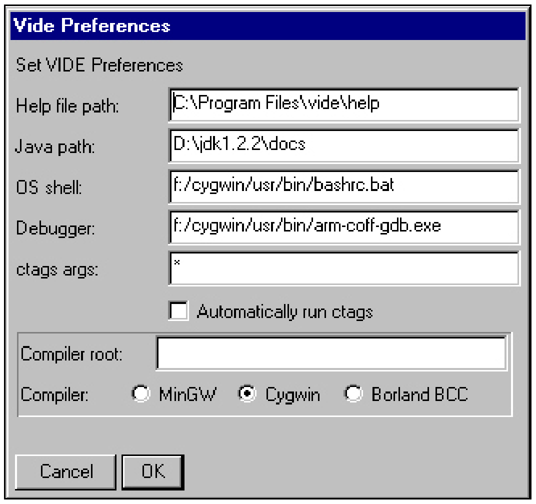
☐ Automatically run ctags

Compiler root:

Compiler: ○ MinGW [✓ Cygwin] ○ Borland BCC

Cancel OK

Figure 7. Windows 98 Setup of Vide Preferences



Vide Preferences

Set VIDE Preferences

Help file path:	C:\Program Files\vide\help
Java path:	D:\jdk1.2.2\docs
OS shell:	f:/cygwin/usr/bin/bashrc.bat
Debugger:	f:/cygwin/usr/bin/arm-coff-gdb.exe
ctags args:	*

☐ Automatically run ctags

Compiler root:

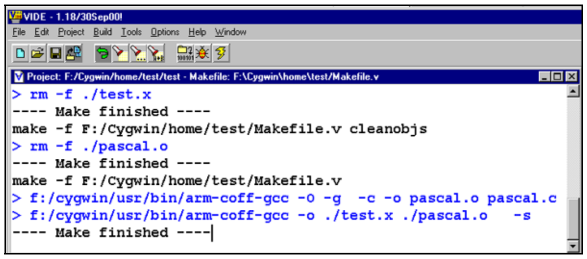
Compiler: ○ MinGW [✓ Cygwin] ○ Borland BCC

Cancel OK

Figure 8. Windows NT/2000 Setup of Vide Preferences

8. Next, click on *Options* | *VIDE* from the main menu. Depending on your Win32 installation, use either Figure 7 (Windows 98) or Figure 8 (Windows NT/2000) as a guide for setting up the VIDE Preferences. Remember to substitute your <drive> letter for the path. Click *Ok* to return to the main window. As a short test, click on *Tools* | *Run OS Shell* and you should get the GNU X-Tools shell startup. If you do not get the GNU X-Tools shell, then there is a path problem or installation problem that you need to troubleshoot.

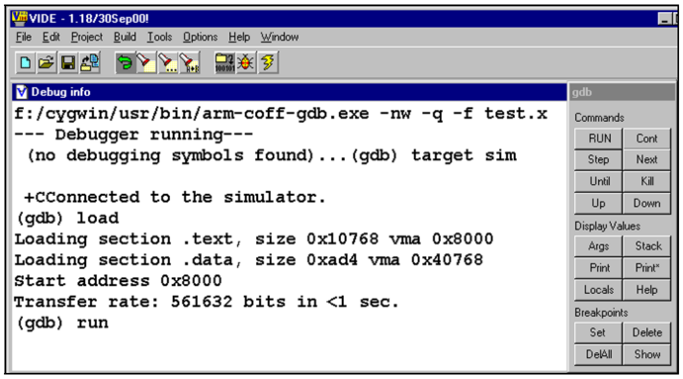
9. Next, click on *Build* | *Compile…* or press F7 to build the target executable. You should get something like what's displayed in Figure 9. If you get an error, recheck your path settings and environment setup.


VIDE - 1.18/30Sep00!
File Edit Project Build Tools Options Help Window
Project: F:/Cygwin/home/test/test - Makefile: F:\Cygwin\home\test\Makefile.v

```
> rm -f ./test.x
---- Make finished ----
make -f F:/Cygwin/home/test/Makefile.v cleanobjs
> rm -f ./pascal.o
---- Make finished ----
make -f F:/Cygwin/home/test/Makefile.v
> f:/cygwin/usr/bin/arm-coff-gcc -O -g  -c -o pascal.o pascal.c
> f:/cygwin/usr/bin/arm-coff-gcc -o ./test.x ./pascal.o   -s
---- Make finished ----
```

Figure 9. VIDE Makefile Run Example

10. Click on the toolbar icon that looks like a bug, which will start your debugger running. Type in <u>target sim</u> at the GDB command line window; type in <u>load</u> at the next line; and last, type in <u>run</u>. A full description on GDB can be found in docs.

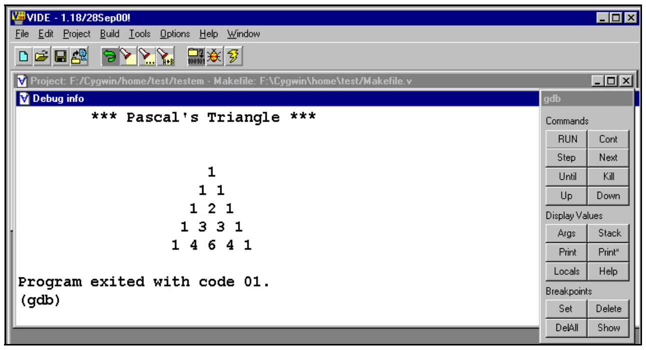


```
f:/cygwin/usr/bin/arm-coff-gdb.exe -nw -q -f test.x
--- Debugger running---
 (no debugging symbols found)...(gdb) target sim

 +CConnected to the simulator.
(gdb) load
Loading section .text, size 0x10768 vma 0x8000
Loading section .data, size 0xad4 vma 0x40768
Start address 0x8000
Transfer rate: 561632 bits in <1 sec.
(gdb) run
```

Figure 10. GDB Debugger Operation

11. Figure 11 shows the program executing pascal.c, the test file we selected earlier.



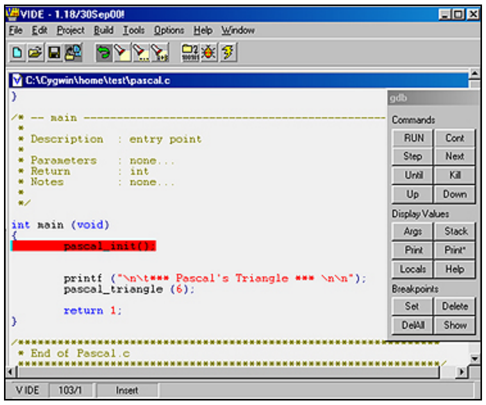
```
        *** Pascal's Triangle ***


                     1
                    1 1
                   1 2 1
                  1 3 3 1
                 1 4 6 4 1

Program exited with code 01.
(gdb)
```

Figure 11. Running a Simulator in the Debugger

12. Figure 12 shows the transition to source debug. That concludes a brief demonstration of using VIDE. The VIDE User Guide is provided with the Docs, and updates can be gotten from www.objectcentral.com.



```
}

/* -- main -------------------------------------------
 *
 * Description   : entry point
 *
 * Parameters    : none...
 * Return        : int
 * Notes         : none...
 *
 */

int main (void)
{
        pascal_init();


        printf ("\n\t*** Pascal's Triangle *** \n\n");
        pascal_triangle (6);

        return 1;
}

/*****************************************************************
 * End of Pascal.c
 *****************************************************************/
```

Figure 12. Debug with Break Points

VIDE for Linux Example Setup

The assumption at this point is that you have GNU X-Tools setup for Linux and the target arm-coff installed – if not installed, follow the steps in the installation section of this User Guide for Linux. If ready, follow the steps below to setup, run, and debug a simple project.

Step

1. After installing VIDE by command line on a Linux host, you are ready to begin. Startup VIDE from your command line (in X-Windows session if running one) or through rxvt or equivalent shell (e.g., execute $/usr/vide/vide at the command prompt starts VIDE from my computer because that is the path to VIDE).

2. Click on *Project* | *New C++ Project*. Figure 13 shows the dialog window that is started. For this example, browse to the directory, /user/src/test, and enter the information exactly as shown at the bottom in the dialog and click *OK*. Another short dialog box will pop up, enter target name (i.e., test2.x) and leave GUI selections unselected so that we default to a console application for cross development. Exit this dialog and go to the next step.

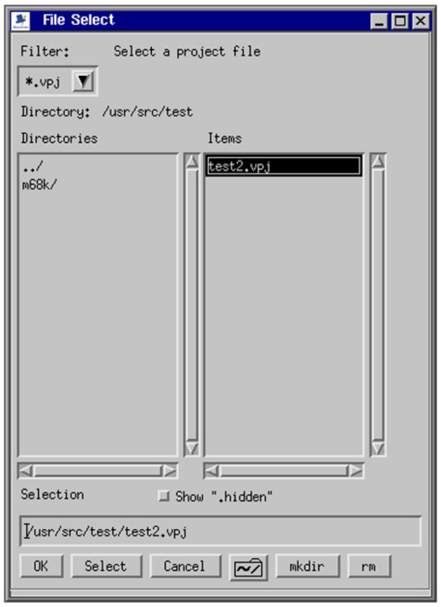
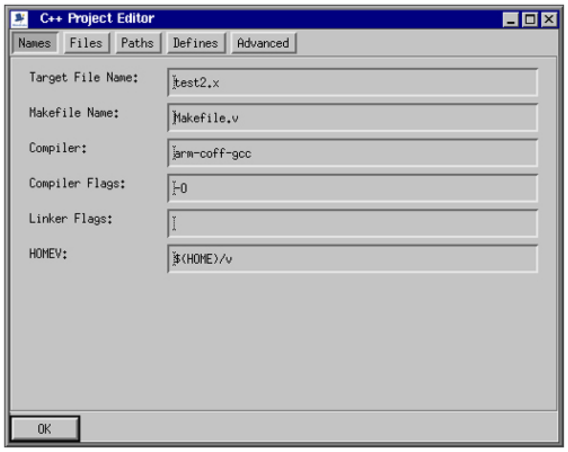


Figure 13. Creating a Project on Linux Host

3. Enter the information shown in the project editor. Compiler and Linker flags are not required for the arm-coff example; however, other targets may require them – see Compiler Options in the Appendices of this User Guide for the specific target toolchain you are working with.



Target File Name:	test2.x
Makefile Name:	Makefile.v
Compiler:	arm-coff-gcc
Compiler Flags:	-O
Linker Flags:	
HOMEV:	$(HOME)/v

Figure 14. C++ Project Editor | Names Tab

4. Next, click on *Files* tab. For this example, click on *Add* and browse to the */usr/src/test* directory and select *dtor.c* as the test source file. This file is the only C++ test file in this directory. Later on if you wish to test with C++ files, go to Chapter 4 and read up on using Bench++.

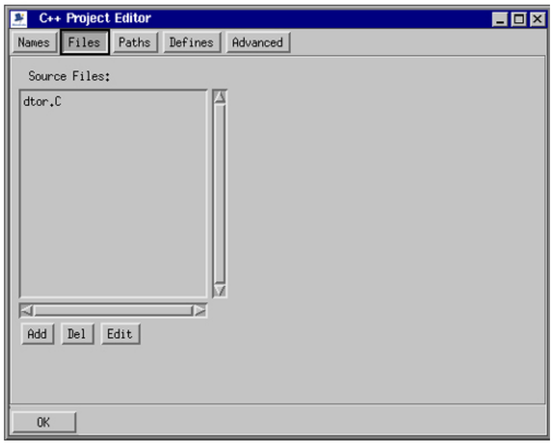


Figure 15. C++ Project Editor | Files Tab

5. Click on *Path* tab. No entries are necessary for this example. Read the VIDE User Guide to get acquainted these features (later).

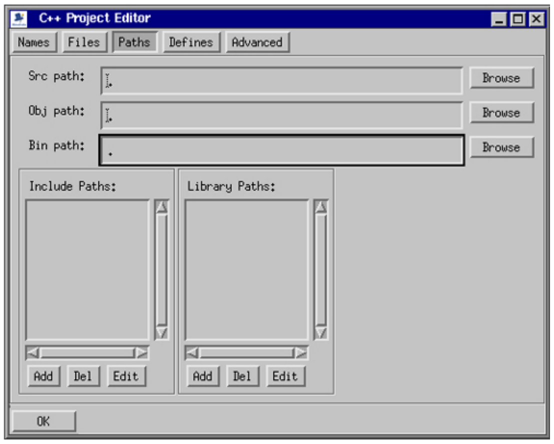


Figure 16. C++ Project Editor | Paths Tab

6. Click on *Defines* tab. No entries are necessary for this example. Read the VIDE User Guide to get acquainted these features (later).

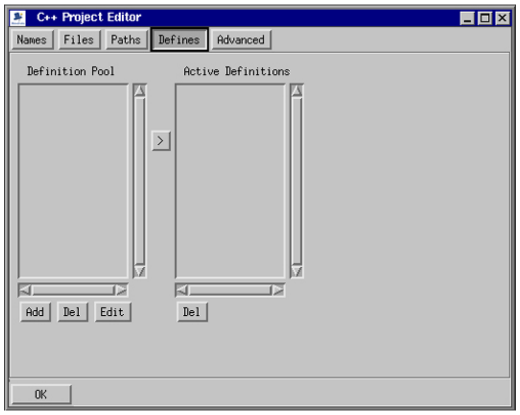


Figure 17. C++ Project Editor | Defines Tab

7. Click on *Advanced* tab. Again, no entries are necessary for this example. Click *OK* to finish and exit the *C++ Project Editor*. You may go back and edit this dialog at any time by clicking on *Project* | *Edit.*

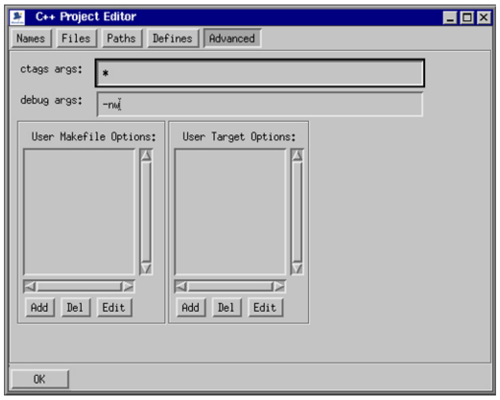
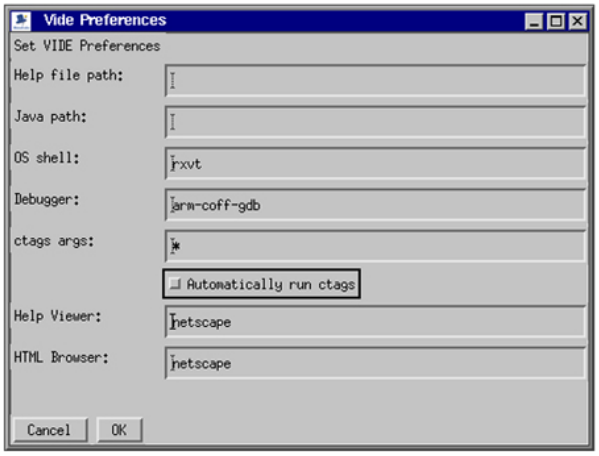


Figure 18. C++ Project Editor | Advanced Tab

8. Next, click on *Options* | *VIDE* from the main menu. Enter the information as in the example in Figure 2.18. Click *Ok* to return to the main window. As a short test, click on *Tools* | *Run OS Shell* and you should get the GNU X-Tools shell startup. If you do not get the GNU X-Tools shell, then there is a path problem or installation problem that you need to troubleshoot.

9. Go back to the previous section and follow the same steps (beginning at Step 9) to build the executable and start the debugger. The nice part about these tools is they work about 95 percent the same way on all host operating systems that Microcross supports: Windows, Linux, and Solaris.



Set VIDE Preferences

Help file path:	
Java path:	
OS shell:	rxvt
Debugger:	arm-coff-gdb
ctags args:	*
	☐ Automatically run ctags
Help Viewer:	netscape
HTML Browser:	netscape

Figure 19. VIDE Preferences

For more information on how to use the tools, consult the either the GNU X-Tools User Guide or the PDF documentation on the distribution CD.

Index