BUILDING IMAKE, CPP, and MAKEDEPEND

Michael Litzkow

Computer Sciences Department
University of Wisconsin - Madison
mike@cs.wisc.edu

1. INTRODUCTION

Imake, **cpp**, and **makedepend** are not part of **condor**, but you will need correctly working copies of all three to build **condor**. **Imake** is a program for building ‘‘Makefiles’’ which are customized for a particular set of hardware, software, and local environment. **Imake** accomplishes much of this task by using **cpp** to expand macros. Since **imake** makes certain assumptions about the operation of **cpp** which will not be true in all installations, it may be necessary to build a customized **cpp** for **imake’s** use. **Makedepend** is a program for generating dependencies in ‘‘Makefiles’’ for C programs which include header files. Since some versions of **makedepend** do not parse certain **cpp** directives correctly, you may need the version included here even if you already have **makedepend**.

All three of these programs are included in the **X11** distribution, and are available from MIT. If you already have X11, you should already have these programs available. They are included here for your convenience in the event you do not have X11. Instructions for building these programs are included in the X11 distribution, but those instructions assume you want to build the whole distribution. These instructions are intended to be a ‘‘direct path’’ to building just these three programs. The **condor** developers assume no responsibility for their completeness, correctness, etc.

2. BUILDING IMAKE

(1) cd to "imake"

(2) To build correct Makefiles, **imake** must determine which platform, (both hardware and variant of UNIX), it is running on. It depends on certain unique strings being defined by your local version of **cpp** to do this. If your local **cpp** does not define the strings which **imake** expects, you can build a custom version of **imake** which will pass those strings to **cpp** by using "-D<string>" on the command line. Some hardware platforms are unique enough that the unix variation is implicit. In other cases two strings are used. Determine whether your local **cpp** defines the necessary strings to identify your platform. The expected strings are as follows:

String(s)	Platform
"RiscAIX"	IBM R6000 running AIX3.1
"mips" & "ultrix"	DecStation running ULTRIX 4.0
"sun" & "mc68020"	SUN 3 running SunOS 4.1+
"sun" & "sparc"	SUN 4 running SunOS 4.1+
"mc68000" & "hp300"	HP Bobcat running BSD 4.3
"vax" & "ultrix"	Vax running Ultrix 3.2
"i386" & "sequent"	Sequent Symmetry running Dynix
"ibm032"	IBM RT running BSD 4.3

(3) If you need to customize your **imake** to pass such strings to **cpp**:

a. To "Makefile.ini" add "-D<PLATFORM_ID>" to the CFLAGS macro, where <PLATFORM_ID> is replaced by any reasonable name for your platform.

b. To ‘‘imakemdep.h’’ in the ‘‘cpp_argv’’ definition, add

```
#ifdef <PLATFORM_ID>
        "-D<PlatformId>",
#endif
```

where <PLATFORM_ID> matches the definition you added to ‘‘Makefile.ini’’ and <PlatformId> matches the appropriate string(s) listed in the above table. Note, if two identifiers are listed above, you will have to add two lines here.

(4) Compile **imake** by running "make -f Makefile.ini".

3. BUILDING CPP

(1) Some versions of **cpp** will not accept "/**/" as a zero length separator. This will cause constructs like

```
/**/# Makefile comment line
```

to be flagged as errors. If this is the case with your **cpp**, you will need to build a special version to use with **imake**. Just try running such a line through your local version of **cpp** to determine whether you have the problem.

(2) If you do need the special **cpp**, go into the **cpp** directory and build it. Hopefully no customization will be needed.

4. MAKEDEPEND

(1) You will also need a correctly working version of "makedepend". This program reads a set of source files along with the flags which will be given to the C compiler when those sources are compiled and generates a set of dependence lines to be added to the local Makefile.

There are some versions of this program around which incorrectly process lines of the form:

#if <expression>

Since condor uses this construct extensively to include the correct header files for different platforms, you will need a correctly working version. A shell script version called "mdepend.sh" is included here. It is much slower that the compiled version, but should work correctly.