# Building epics 3.14.8.2 on win32-x86

J. Sebek

November 8, 2007

**Abstract**

I document the steps I used to build epics 3.14.8.2 on the win32-x86 platform using the standard tools for the WinXP platform.

## 1 Obtain the necessary software

- Tools
  - Microsoft Visual C/C++: This is the native compiler for XP. I used the current version, Visual Studio 8, part of the distribution of Microsoft Visual Studio 2005. The compiler sets up its environment variables for the proper Path, INCLUDE, and LIB directories from a file
    vcvars32.bat
    located in the Visual C default directory
    C:\Program Files\Microsoft Visual Studio 8\VC\bin.
    The only function of this file is to call vsvars32.bat located in the default directory
    C:\Program Files\Microsoft Visual Studio 8\Common7\Tools.
    The Visual Studio installation should define an environment variable
    VS80COMNTOOLS
    to have this directory as its value, which can be verified by typing, at the command prompt
    echo %VS880COMNTOOLS%
    After opening a command window which one will use for building the epics package, one runs the batch file
    "%VS80COMNTOOLS%vsvars32.bat"
    Note that the quotation marks are necessary to have the spaces in the directory names of the various paths correctly parsed by all of the tools in the compilation and build process.
  - gmake: The EPICS build uses the gnu tools make. I downloaded an executable of version 3.81 from the cygwin distribution which I obtained from http://www.cygwin.com. I installed the executables in the default directory
    c:\cygwin\bin
    which I placed in the system path, using
    Start -> Settings -> Control Panel -> System -> Advanced -> Environment Variables -> system variables -> Path
  - perl: The EPICS build also uses the perl language. The version that I used was Version 5.8.8

which I downloaded from

`http://www.activestate.com/`
as an `.msi` file. I ran this file to install `perl` in the default directory
`c:\Perl`
I also added this directory to the path.

- EPICS source: This is $R3.14.8.2$ downloaded from

  `http://www.aps.anl.gov/epics/download/base/index.php`

  I used `winzip` to extract the `tar.gz` file.

## 2 Build the EPICS package

- Notes from the distribution related to compiling for `WindowsXP` are in the file
  `C:\epics\base\documentation\README.MS_Windows`
- Copy the extracted EPICS into a `base` (sub)directory. For my particular installation, I extracted the files into
  `c:\epics\base`
- Add to your path the name of the directory into which EPICS will install its binaries
  `<EPICS>base/bin/win32-x86`
- Insure that the `Path`, `INCLUDE`, and `LIB` environment variables will also find the `make` and `perl` executables created above as well as the Visual Studio files. In my installation, I explicitly entered the paths of `make` and `perl` using
  `Settings->Control Panel->System->Advanced->Environment Variables`
  and, after opening a `command` window, I modified the environment in that window for Visual Studio by entering the command
  `"%VS80COMNTOOLS%vsvars32.bat"`
  as discussed above.
- Insure that a `TMP` environment variable exists and points to a valid directory. This variable was defined for me in my `Environment Variables`.
- Create a new environment variable `HOST_ARCH=win32-x86` to be used by `make`. This can either be created in `Environment Variables` or it can be set in the current `command` window with the command
  `set HOST_ARCH=win32-x86`
  Note that setting `HOST_ARCH=WIN32`, the original value for the `WindowsXP` opearting system, currently also works. But the modern name of this variable is `win32-x86`, and one assumes that this value will exist after `WIN32` becomes obsolete. Currently these two names are aliases of each other.
- In `base` directory, run

  `make`

  to generate the EPICS build.

# 3 Compiling `labca` for MATLAB

The detailed instructions for compiling `labca` are found in the document "*labCA* – An EPICS Channel Access Interface for *scilab* and *matlab*" by Till Straumann in the `documentation` subdirectory in the `labca` distribution. This current note applies to the build of `labca_3_1` for MATLAB on Windows XP. As of October 31, 2007, the build for Windows XP still has some problems. Although we worked around these problems to complete the build, the workarounds were not incorporated into the package, since we did not want to risk breaking the builds for the other operating systems by modifying the build rules.

We first followed the instructions in the `labca` reference document. As stated there, only two files need be edited. In `CONFIG` I used the default values for the `MAKEFOR` and `CONFIG_USE_CTRL_C` lines. These lines are

```
MAKEFOR=MATLAB
CONFIG_USE_CTRL_C=YES
```

respectively.

In `RELEASE` I needed to change three lines

```
EPICS_BASE=c:/epics/base
MATLABDIR=c:/matlab/R2007b
MATLIB_SUBDIR=win32/microsoft
```

Note that the syntax that I used for directory names is neither Windows XP nor `linux/unix`. If I used straight Windows XP syntax, the `EPICS_BASE` directory was interpreted as

```
c:\epicsbase
```

although the separations between names of subdirectories of `EPICS_BASE` were properly interpreted. For reasons that are similar to the ones that made this build difficult, it is strongly recommended that the location of `MATLABDIR` have no spaces in any of its directory names. This environment variable is used in standard EPICS make files and we were not able to find an easy solution to make these files work with spaces in their path names. We now install MATLAB in the non-default directory structure in the directory location

```
c:\matlab
```

The R2007a and R2007b versions of MATLAB install all Microsoft libraries in one directory, independent of any Visual Studio version. Previous versions of MATLAB had separate subdirectories for different versions of the Visual Studio compilers.

# 4 Trouble found in this build

The current thinking is that the problems with the build are probably due to the different ways that Windows XP and `linux/unix` treat environment variables and proper directory names. Windows XP uses a ":" to identify a drive; `linux/unix` uses the ":" as a separator in environment variables such as `PATH`. Windows XP uses "\" to separate directory names and allows spaces in these names; `linux/unix` uses "/" to separate the directory names and does not allow spaces in the names. These differences in syntax rules cause problems with some of the statements created during the `make` process.

The main problem that we had was in trying to execute the file

```
LABCA_BASE/ezca/O.win32-x86/MakefileInclude
```

During the `make`, this file reported the error on line 95

MakefileInclude:95:  *** target pattern contains no '%'.  Stop.
The contents of this line are
$(LIB_PREFIX)ezcamt$(LIB_SUFFIX):$(ezcamt_DEPLIBS)
The make tools could not properly parse the environment variables to correctly interpret this statement.

MakefileInclude is generated by a perl script from the standard epics package
EPICS_BASE/configure/tools/makeMakefileInclude.pl
which automatically generates files for the various programs that need to be made. After running make once and failing, we edited MakefileInclude to remove lines 95 and 97, the lines with the offending string ezcamt_DEPLIBS, and then reran make. Then labca compiled successfully.

The make had one other problem, however. It could not create its documentation because I did not have the program
latex2html
installed on my computer. We worked around this problem by commenting out the line that asks the documentation to be produced
DIRS += documentation
in the top level Makefile located in the LABCA_BASE directory. Placing a "#" at the front of the line comments out this command.

# 5 Other issues in the build

Labca must be built with the proper tools compatible with the EPICS build. We used the Cygwin tools for both make and perl in the successful build, although the non-Cygwin ActiveState perl would also probably have worked, since it was used and worked well for the EPICS build. After we ran into the trouble documented above, we tried a version of make from GnuWin32, which is built differently than is the Cygwin make. We found that GnuWin32 make gave us other problems, so we abandoned it and went back to the Cygwin make.

Note that in the Cygwin environment, the location
C:\
is mapped to
/cygdrive/c/
and
C:\cygwin\bin
is mapped to
/usr/bin

We worked with a minimum configuration, modeled after what we did to build EPICS. We opened a command window and created a minimum path with the commands
PATH=%systemroot%\system32;%systemroot%;%systemroot%\system32\wbem;
PATH=%PATH%C:\cygwin\bin;
set EPICS_HOST_ARCH=win32-x86
"%VS80COMNTOOLS%vsvars32.bat"
Note that both make and perl were in the \cygwin\bin directory. The last line sets up the proper

environment variables for the Visual Studio 2005 compiler.

Once we had entered this data in the command window, we opened a bash window and ran the make from there. The make may also have worked from within the command environment, but we found out that the "%VS80COMNTOOLS%vsvars32.bat" command was not properly interpreted in the bash environment.

# 6 Packaging

We performed one additional step to package the executables. We copied the three EPICS files
ca.dll
caRepeater.exe
Com.dll
from the EPICS directory
EPICS_BASE\bin\win32-x86
into the directory
LABCA_BASE\bin\win32-x86
so that only this directory needs to be placed in the system path in order for labca to be used within MATLAB. Of course the directory
LABCA_BASE\bin\win32-x86\labca
must be added to the MATLAB path from within MATLAB for it to access the various labca functions.

# 7 Deployment to other Windows machines

The compiled code needs to link to compatible Microsoft dynamic link libraries (dll's) in order to execute. For the Visual C version 8.0, which is the C compiler in Visual Studio 2005, these dll's are
msvcm80.dll
msvcp80.dll
msvcr80.dll
In order to keep track of which files are needed, a .manifest file is created during the compilation process with, among other things, the file names and their version numbers. During execution, the compiled files search for these versions of the files, so the files must be deployed along with the labca application.

Microsoft recommends several ways to deploy these files. Since we do not compile labca from within the Visual Studio IDE, we choose a manual deployment method that still allows Microsoft to know where these dll files will be if and when they need to be patched. Microsoft has created a directory
C:\WINNT\WinSxS
in which all necessary versions of the dll's are kept. Each version is kept in a separate subdirectory that is identified by its version number.

The compiler comes with an executable, vcredist_x86.exe that is to be run on the deployed machine. This program recognizes the operating system on the deployed machine, extracts the proper dll's and places them in the appropriate directory for that operating system. In the default installation, this file is located in

C:\Program Files\Microsoft Visual Studio 8\...
...\SDK\v2.0\BootStrapper\Packages\vcredist_x86

Therefore vcredist_x86.exe must be packaged with the compiled labca package and this file must be executed on the deployed machine in order for labca to execute.