Porting QT5 Applications to OS/2

Gregg Young

What is QT?

- Full development framework
- Also called a widget toolkit
- Cross platform
- Pronounced “cute”
- Open source

OS/2 Port and QT5 Applications

- Bitwise Works
- Dmitriy Kuminov
- Community funded
- Elbert Pol
- Gregg Young
- Paul Smedley

Getting What You Need

- Everything is available from the netlabs-rel RPM repository
- In ANPM set the platform to Pentium 4
- If you aren't using yum or the Arca Noae Package Manager (ANPM) install ANPM
  - https://www.arcanoae.com/resources/downloadables/arca-noae-package-manager/
- ANPM installs the UNIXROOT structure to the disk you select.
  - Since most of the tools you need are from the UNIX world these structures are needed.

Install the Tool Chain
Must Haves

- Compilers and linker
    - GCC 9.2 (Netlabs)
        - cpp gcc gcc-c++
        - gcc-wlink gcc-wrc
- Make Utility
    - kbuild-make

Install the Tool Chain Must Haves 2

- os2tk45
- rpm-build
  - binutils diffutils findutils
  - file patch tar xz
- Configuration Utility
  - cmake
    - cmake-data
    - cmake-filesystem
    - cmake-rpm-macros

Additional Tools Needed for Some Projects

- kbuild
- os2-base-unixtools-path
- bison – general purpose parser generator
- nasm – x86 assembler
- autoconfig
  - m4 perl-Data-Dumper
- GCC 10 and 11 versions
  - Paul Smedley
- Not a comprehensive list

Needed Libraries

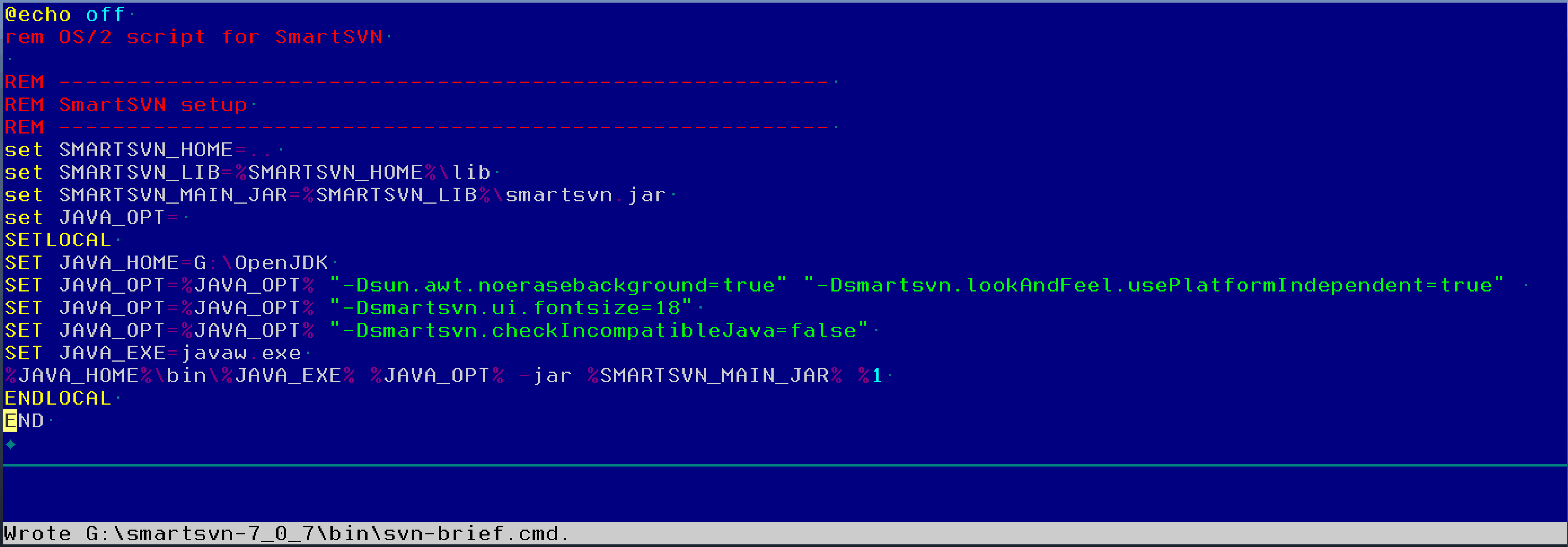
- libc-devel
- libcx-devel
    - exceptq-devel
- libstdc++-devel

Editors

- Something with syntax highlighting
    - NEPM
        - Not multiprocessor safe (markexe, exeheader)
    - EFTE/2
- E can be used for small changes but really is inadequate for this type of thing
- If you have something multilingual QE can be useful

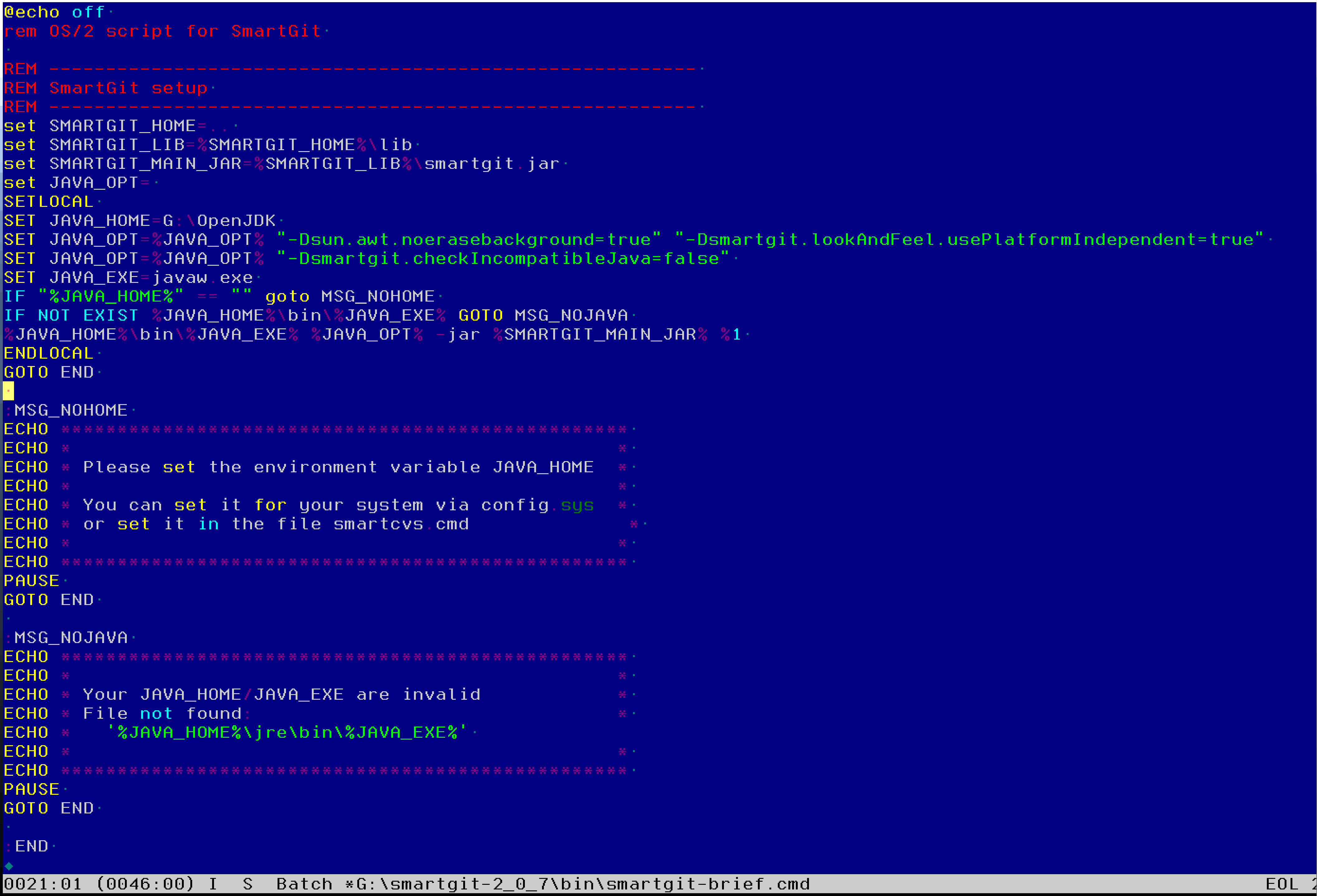
Versioning

- git
    - git-core git-core-doc
    - perl-error perl-git
- subversion
- SmartSVN 7.07
- SmartGit 2.07



```
@echo off
rem OS/2 script for SmartSVN

REM -----------------------------------------------------------
REM SmartSVN setup
REM -----------------------------------------------------------
set SMARTSVN_HOME=..
set SMARTSVN_LIB=%SMARTSVN_HOME%\lib
set SMARTSVN_MAIN_JAR=%SMARTSVN_LIB%\smartsvn.jar
set JAVA_OPT=
SETLOCAL
SET JAVA_HOME=G:\OpenJDK
SET JAVA_OPT=%JAVA_OPT% "-Dsun.awt.noerasebackground=true" "-Dsmartsvn.lookAndFeel.usePlatformIndependent=true"
SET JAVA_OPT=%JAVA_OPT% "-Dsmartsvn.ui.fontsize=18"
SET JAVA_OPT=%JAVA_OPT% "-Dsmartsvn.checkIncompatibleJava=false"
SET JAVA_EXE=javaw.exe
%JAVA_HOME%\bin\%JAVA_EXE% %JAVA_OPT% -jar %SMARTSVN_MAIN_JAR% %1
ENDLOCAL
END
```

```
Wrote G:\smartsvn-7_0_7\bin\svn-brief.cmd.
```



```
@echo off
rem OS/2 script for SmartGit

REM ----------------------------------------------------------
REM SmartGit setup
REM ----------------------------------------------------------
set SMARTGIT_HOME=..
set SMARTGIT_LIB=%SMARTGIT_HOME%\lib
set SMARTGIT_MAIN_JAR=%SMARTGIT_LIB%\smartgit.jar
set JAVA_OPT=
SETLOCAL
SET JAVA_HOME=G:\OpenJDK
SET JAVA_OPT=%JAVA_OPT% "-Dsun.awt.noerasebackground=true" "-Dsmartgit.lookAndFeel.usePlatformIndependent=true"
SET JAVA_OPT=%JAVA_OPT% "-Dsmartgit.checkIncompatibleJava=false"
SET JAVA_EXE=javaw.exe
IF "%JAVA_HOME%" == "" goto MSG_NOHOME
IF NOT EXIST %JAVA_HOME%\bin\%JAVA_EXE% GOTO MSG_NOJAVA
%JAVA_HOME%\bin\%JAVA_EXE% %JAVA_OPT% -jar %SMARTGIT_MAIN_JAR% %1
ENDLOCAL
GOTO END

:MSG_NOHOME
ECHO **************************************************
ECHO *                                                *
ECHO * Please set the environment variable JAVA_HOME  *
ECHO *                                                *
ECHO * You can set it for your system via config.sys  *
ECHO * or set it in the file smartcvs.cmd              *
ECHO *                                                *
ECHO **************************************************
PAUSE
GOTO END

:MSG_NOJAVA
ECHO **************************************************
ECHO *                                                *
ECHO * Your JAVA_HOME/JAVA_EXE are invalid            *
ECHO * File not found:                                *
ECHO *   '%JAVA_HOME%\jre\bin\%JAVA_EXE%'
ECHO *                                                *
ECHO **************************************************
PAUSE
GOTO END

:END

0021:01 (0046:00) I   S  Batch *G:\smartgit-2_0_7\bin\smartgit-brief.cmd                                              EOL 2
```

Versioning Best Practices

- If the code is in a repository get it from the repository as opposed to a zip of the source
- If you only have access to a zip create a local git repository
    - git init
    - git add files you plan to change and commit
- If you have already made changes to unversioned source code unzip the the original in a separate directory create a git repo git add then copy your changed files in.

QT5 Components

- qt5 qt5-devel
- qt5-qtbase-common qt5-qtbase-devel qt5-qtbase-gui qt5-qtbase-private-devel qt5-qtbase-static qt5-qtbase-debuginfo
- qt5-qtdeclarative qt5-qtdeclarative-devel qt5-qtdeclarative-static qt5-qtdeclarative-debuginfo
- qt5-multimedia qt5-multimedia-devel qt5-multimedia-debuginfo
- qt5-qtsvg qt5-qtsvg-devel qt5-qtsvg-debuginfo

QT5 Components 2

- qt5-qttools qt5-qttools-common qt5-qttools-devel qt5-qttools-libs-designer qt5-qttools-libs-designercomponets qt5-qttools-libs-help qt5-qttools-static qt5-qttools-debuginfo
- qt5-qtwebchannel qt5-qtwebchannel-devel qt5-qtwebchannel-debuginfo
- qt5-qtwebengine qt5-qtwebengine-devel qt5-qtwebengine-debuginfo
- qt5-qtwebsockets qt5-qtwebsockets-devel qt5-qtwebengine-debuginfo

Other QT5 Tools

- qt5-designer
- qt5-doctools
- qt5-linguist
- qt5-rpm-macros
- qmake-qt5.exe is in qt5-qtbase-devel
- While you won't need all these for many ports it is easier to install them all then it is figure out what is missing later.

QT5 Dependencies

- cups-devel
- dbus-libs
- expat-devel
- ffmpeg-libs
- fontconfig-devel
- freetype-devel
- gmp-c++
- gmp-devel
- gnutils-c++
- gnutils-devel
- lame-libs
- libevent
- libmpc
- libogg
- libpng-devel

QT5 Dependencies

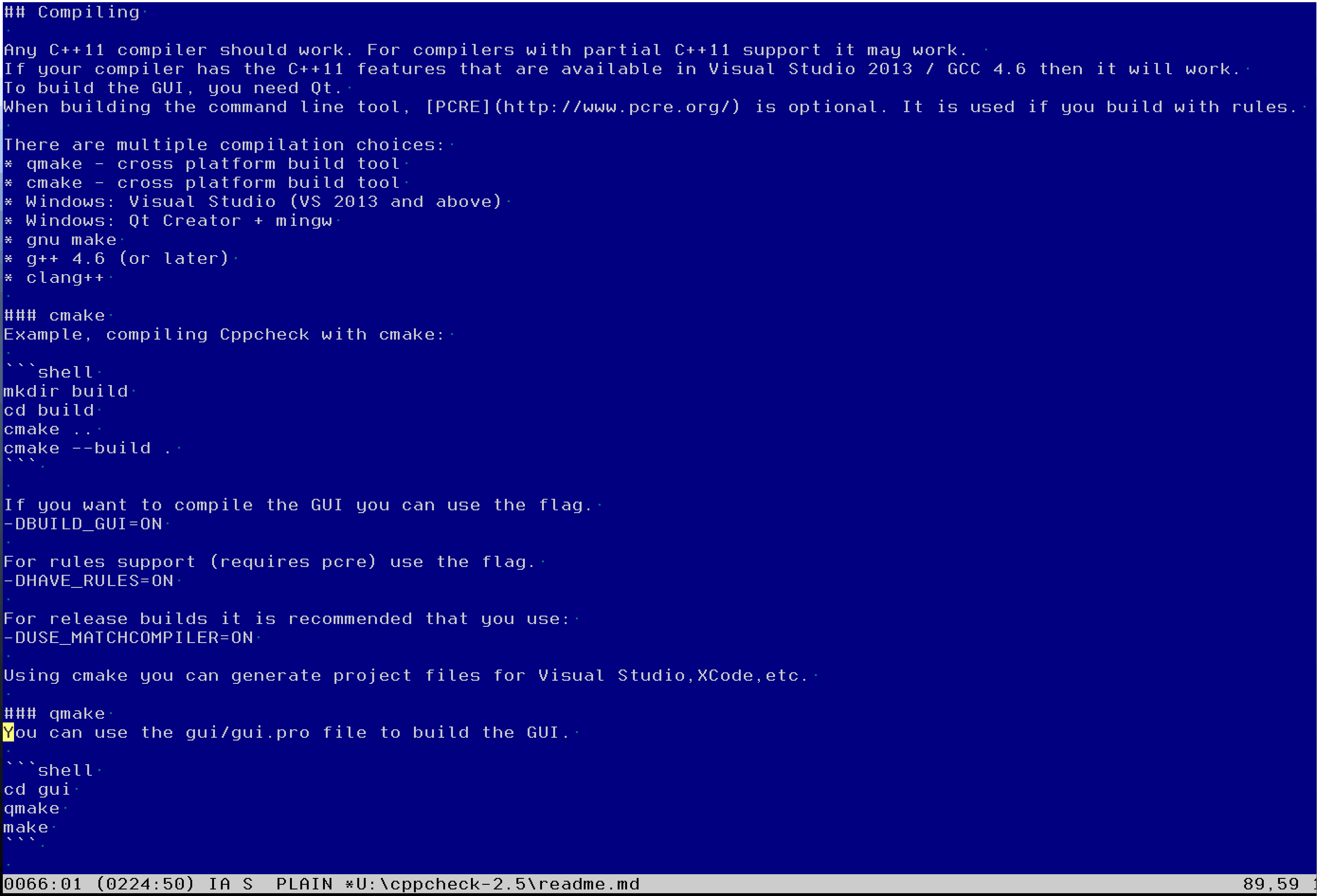
- libtasn1-devel
- libtasn1-tools
- libteora
- libvorbis
- libvpx
- libwebp
- libxslt
- minizip
- nettle-devel
- opus
- p11-kit-devel
- pcre2-syntax
- pcre2-utf16
- x254-libs
- xvidcore
- zlib-devel

QT5 Dependencies (Tools)

- cmake
- cmake-data
- cmake-filesystem
- cmake-rpm-macros
- cpp
- gcc
- gcc-cpp
- gettext
- libstdc++-devel
- mpfr

Building CPPCheck

- Readme.md or sometimes a wiki or web site will contain the build instructions
- While CPPCheck's instructions in readme.md were fine some are incomplete/out of date



~~~
## Compiling

Any C++11 compiler should work. For compilers with partial C++11 support it may work.
If your compiler has the C++11 features that are available in Visual Studio 2013 / GCC 4.6 then it will work.
To build the GUI, you need Qt.
When building the command line tool, [PCRE](http://www.pcre.org/) is optional. It is used if you build with rules.

There are multiple compilation choices:
* qmake - cross platform build tool
* cmake - cross platform build tool
* Windows: Visual Studio (VS 2013 and above)
* Windows: Qt Creator + mingw
* gnu make
* g++ 4.6 (or later)
* clang++

### cmake
Example, compiling Cppcheck with cmake:

```shell
mkdir build
cd build
cmake ..
cmake --build .
~~~

If you want to compile the GUI you can use the flag.
-DBUILD_GUI=ON

For rules support (requires pcre) use the flag.
-DHAVE_RULES=ON

For release builds it is recommended that you use:
-DUSE_MATCHCOMPILER=ON

Using cmake you can generate project files for Visual Studio,XCode,etc.

qmake
You can use the gui/gui.pro file to build the GUI.

```shell
cd gui
qmake
make
```

0066:01 (0224:50) IA S PLAIN *U:\cppcheck-2.5\readme.md   89,59
```
```

Needed and Optional Libs

- You need pcre for CPPCheck if you want to build with rules
  - Optional packages are common
  - Helpful to us since we don't necessarily have a given package
- If a package is needed and not at Netlabs try Hobbes and Paul Smedley's site
- Kalendar (discussed later) requires sqlite-devel
  - Failing these you can try to build the package or ask someone else to build it

Defines

- -DBUILD_GUI=ON is a define that indicates you want to build the GUI for CPPCheck
- There are generally additional configuration options in the configuration files
- Accept the default configuration for the first try
- The exceptions are when you don't have an optional package which is used by default or you want to build an optional component (the GUI in this case)

Unix Paths

- Most of the time paths are setup by either Qt or the C lib which means they will be correct for the platform
- In some apps the paths are by default UNIX style but it will usually have code for DOS style paths #if defined for Windows (__WIN__ or __WIN32__)
    - The fix is to add __OS2__ to that #if def

Preparing to Run Make

- Install all the needed packages
- Review the build instructions

Invoking Make

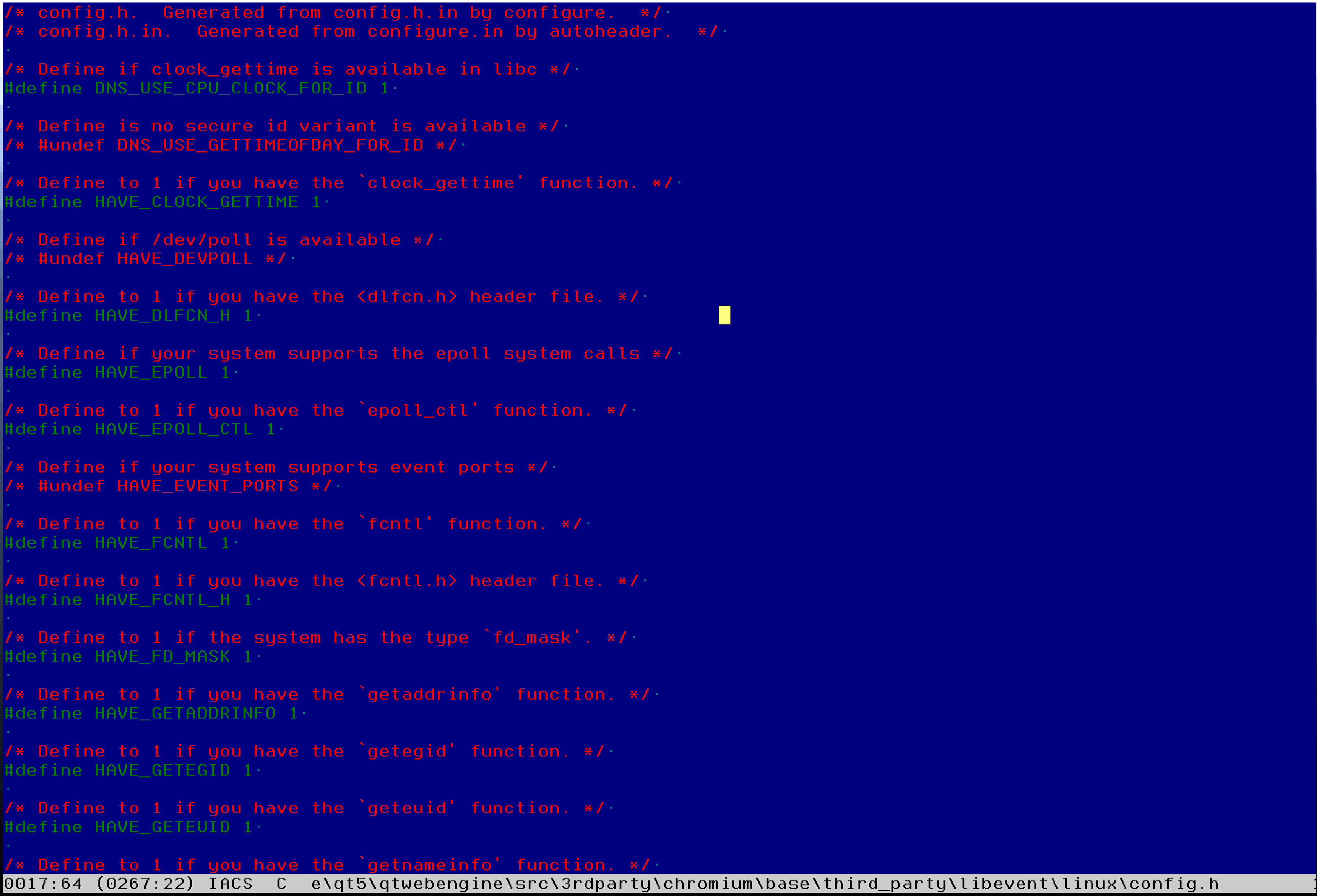
- Always use verbose make output V=1
- Always redirect the output to a file so you can review what has happened and search.
- Keep several generations so you can compare
    - Can be automated
- After an error restart the build from the directory where the error occurred if possible
- Try running the failing command from the command line if possible

Fixing Errors

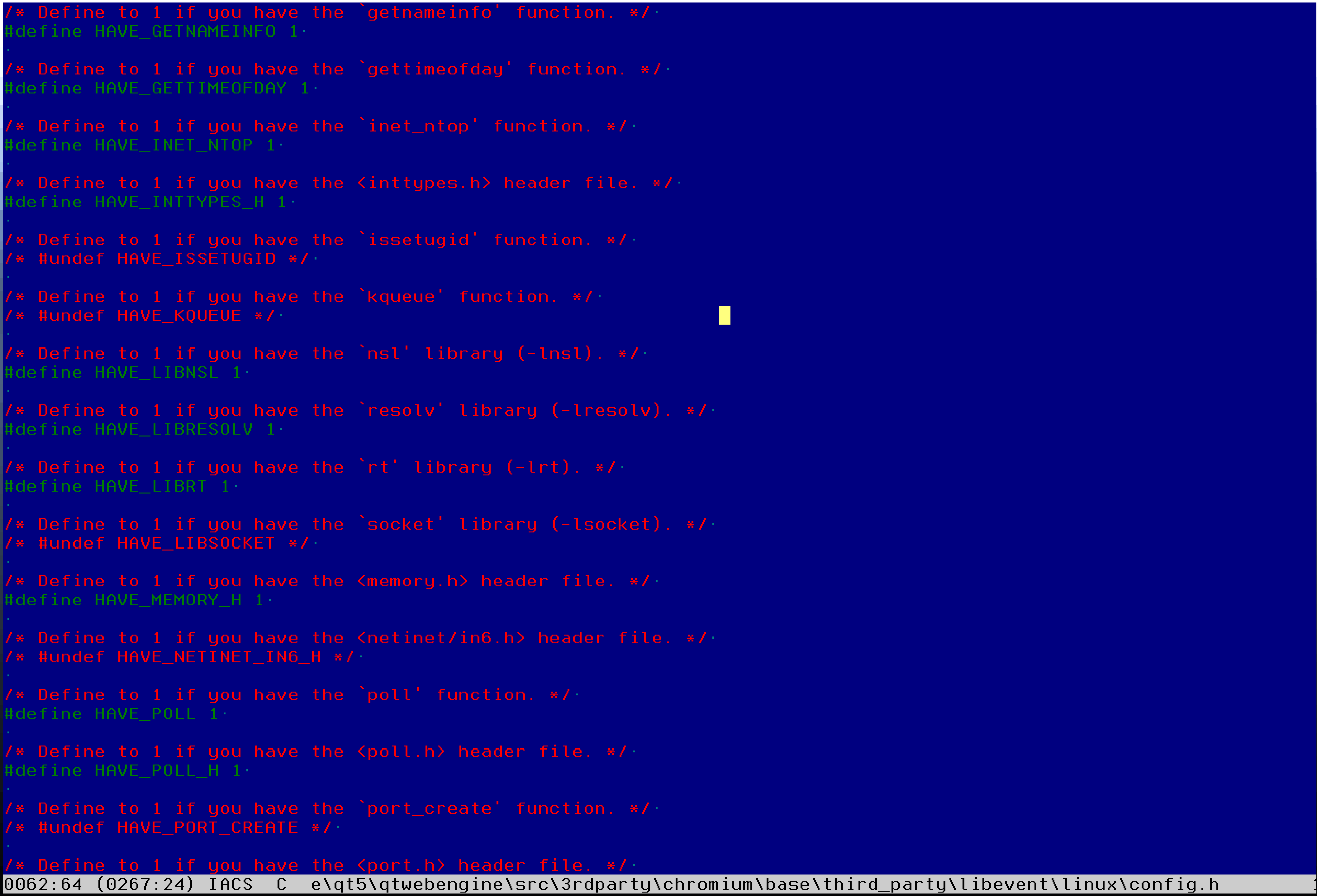
- Missing header – search the rpm packages for the header name if found install the package
  - If not found search the web for the name. In one case I had to create the file based on a forum post
- Missing lib (unresolved symbols) – find the header that contains the symbol and determine the lib associated with that header add that lib to the LIBS = or to the linker command in the makefile. Also check the lib name particularly the suffix

Config.h

- Some program have platform specific config.h files. These specify what the platform has available. Naturally there won't be one for OS2. I start with the Linux one and then undefine (rem out) anything the compiler complains about. Always undefine anything that sets stuff as 64bit.



```
/* config.h.  Generated from config.h.in by configure.  */
/* config.h.in.  Generated from configure.in by autoheader.  */

/* Define if clock_gettime is available in libc */
#define DNS_USE_CPU_CLOCK_FOR_ID 1

/* Define is no secure id variant is available */
/* #undef DNS_USE_GETTIMEOFDAY_FOR_ID */

/* Define to 1 if you have the `clock_gettime' function. */
#define HAVE_CLOCK_GETTIME 1

/* Define if /dev/poll is available */
/* #undef HAVE_DEVPOLL */

/* Define to 1 if you have the <dlfcn.h> header file. */
#define HAVE_DLFCN_H 1

/* Define if your system supports the epoll system calls */
#define HAVE_EPOLL 1

/* Define to 1 if you have the `epoll_ctl' function. */
#define HAVE_EPOLL_CTL 1

/* Define if your system supports event ports */
/* #undef HAVE_EVENT_PORTS */

/* Define to 1 if you have the `fcntl' function. */
#define HAVE_FCNTL 1

/* Define to 1 if you have the <fcntl.h> header file. */
#define HAVE_FCNTL_H 1

/* Define to 1 if the system has the type `fd_mask'. */
#define HAVE_FD_MASK 1

/* Define to 1 if you have the `getaddrinfo' function. */
#define HAVE_GETADDRINFO 1

/* Define to 1 if you have the `getegid' function. */
#define HAVE_GETEGID 1

/* Define to 1 if you have the `geteuid' function. */
#define HAVE_GETEUID 1

/* Define to 1 if you have the `getnameinfo' function. */
0017:64 (0267:22) IACS  C  e\qt5\qtwebengine\src\3rdparty\chromium\base\third_party\libevent\linux\config.h
```



```
/* Define to 1 if you have the `getnameinfo' function. */
#define HAVE_GETNAMEINFO 1

/* Define to 1 if you have the `gettimeofday' function. */
#define HAVE_GETTIMEOFDAY 1

/* Define to 1 if you have the `inet_ntop' function. */
#define HAVE_INET_NTOP 1

/* Define to 1 if you have the <inttypes.h> header file. */
#define HAVE_INTTYPES_H 1

/* Define to 1 if you have the `issetugid' function. */
/* #undef HAVE_ISSETUGID */

/* Define to 1 if you have the `kqueue' function. */
/* #undef HAVE_KQUEUE */

/* Define to 1 if you have the `nsl' library (-lnsl). */
#define HAVE_LIBNSL 1

/* Define to 1 if you have the `resolv' library (-lresolv). */
#define HAVE_LIBRESOLV 1

/* Define to 1 if you have the `rt' library (-lrt). */
#define HAVE_LIBRT 1

/* Define to 1 if you have the `socket' library (-lsocket). */
/* #undef HAVE_LIBSOCKET */

/* Define to 1 if you have the <memory.h> header file. */
#define HAVE_MEMORY_H 1

/* Define to 1 if you have the <netinet/in6.h> header file. */
/* #undef HAVE_NETINET_IN6_H */

/* Define to 1 if you have the `poll' function. */
#define HAVE_POLL 1

/* Define to 1 if you have the <poll.h> header file. */
#define HAVE_POLL_H 1

/* Define to 1 if you have the `port_create' function. */
/* #undef HAVE_PORT_CREATE */

/* Define to 1 if you have the <port.h> header file. */
0062:64 (0267:24) IACS  C  e\qt5\qtwebengine\src\3rdparty\chromium\base\third_party\libevent\linux\config.h
```

Missing Tools (file not found)

- Search the rpms for the package containing it.
- If not found try Hobbes and Paul Smedley's site
- Is the name wrong – qmake.exe is qmake-qt5.exe in our qt5 python2.exe is python2.7.exe
- See if there is another option to do the job in the configuration
- See if what it is doing is part of something that is optional
- As a last resort build it

CPPCheck Results

- Compiled without error using the cmake instructions
- Command line version worked as expected
- GUI trapped on startup
  - Traced problem to memory mapping code
  - However this was really a configuration issue -lcx had not been added to LIBS =
  - Added it to the makefile rebuilt and the GUI ran as expected

Kalendar Build Issue

- Configuration issue -- the debug build failed unable to find Qt5Widgetsd.lib
  - The failure is caused because the *d.lib and *d.dll are not in the RPMs
  - They do get built. Not in spec file

Kalendar Modifications

- Kalendar used some experimental c++ headers those headers were no longer experimental as of gcc 9.2 but require -std=c++17 be set
- Kalendar uses 64bit for its date calculations. This can be set to 32bit but doing this led to failures. Leaving it 64bit worked. This only effects to “date” that signifies that something is a todo item instead of appointment (year 2038 problem)

Kalendar Password Issue

- Kalendar expected that there would be password protection on the file system which didn't work for OS2.
  - /* Open the database (will be created if it doesn't exist) */
  - #ifndef __OS2__
  - this->db_folder = string(getpwuid(getuid())->pw_dir) + string("/" FOLDER_NAME "/");
  - #else
  - this->db_folder = string(getenv("HOME")) + string("/" FOLDER_NAME "/");
  - #endif

Kalendar Scripts

- Kalendar has the capacity to open scripts but defaulted to either SH or BAT scripts. Changed this to CMD or BTM scripts.

```
#ifdef __OS2__
        if ((p.extension() == ".cmd") || (p.extension() == ".btm"))
#else
        if ((p.extension() == ".sh") || (p.extension() == ".bat"))
#endif
```

Thank You

- Questions
- ygk@qwest.net