HUMIQ

Advanced Software

ICT Automatisering

Cygwin Tcl/Tk: Windows or UNIX?

Jan Nijtmans

july 7, 2012

Summary

- What is Cygwin
- Build environment
- Cygwin support for Tcl/Tk
- Stubs
- Event loop
- Conclusion
- Demo

What is Cygwin

- cygwin1.dll
  - licence GPLv3 (with exception) !
- mount()
- fork()
- rebase

```
#### C:\Tcl\etc\fstab    ####

C:/Users /Users fat32 binary 0 0
C:/Tcl /Tcl fat32 binary 0 0
```

- X11 (cygx11-6.dll, cygxss-1.dll)

Build environment

- For Tcl:
  - gcc-core (3.4.4-999) or gcc4-core (4.5.3-3)
  - make (3.82.90-1)
  - mingw64-i686-gcc-core (4.5.3-6)
- For Tk (X11)
  - libXss-devel (1.2.0-1)
  - xinit (1.3.2-1)
- Optional
  - zlib-devel (1.2.7-1) (Optional, If you don't want zlib to be built into Tcl)
  - mingw64-x86_64-gcc-core (4.5.3-6) (Optional, only for win64)

How to compile Cygwin/win32 Tcl/Tk 8.6

- cd win
- configure --prefix=/Tcl (--enable-64bit)
- make

Tcl + Tk

- cd ../unix
- configure --prefix=/Tcl
- make

Tcl + Tk

- make install
- cd ../win
- make install

Tcl + Tk

win32 configure

-
  ```
  #if !defined(_WIN32)
      if test –enable-64bit = true
          CC = x86_64-w64-mingw32-gcc
      else
          CC = i686-w64-mingw32-gcc
  #endif
  ```
- TEA modified as well (Thread, Itcl, tdbc)

Stubs/TEA

- Febr 2012: Cygwin switched from win32 to unix
  - file paths   C:\Tcl   ->   /cygdrive/c/Tcl
  - stub tables

```
typedef struct TclPlatStubs {
    ......
#if defined(__WIN32__) || defined(__CYGWIN__) /* WIN */
    TCHAR * (*tcl_WinUtfToTChar) (......); /* 0 */
    char * (*tcl_WinTCharToUtf) (......); /* 1 */
#endif /* WIN */
#ifdef MAC_OSX_TCL /* MACOSX */
    int (*tcl_MacOSXOpenBundleResources) (......); /* 0 */
    int (*tcl_MacOSXOpenVersionedBundleResources) (.....) /* 1 */
#endif /* MACOSX */
} TclPlatStubs;
```

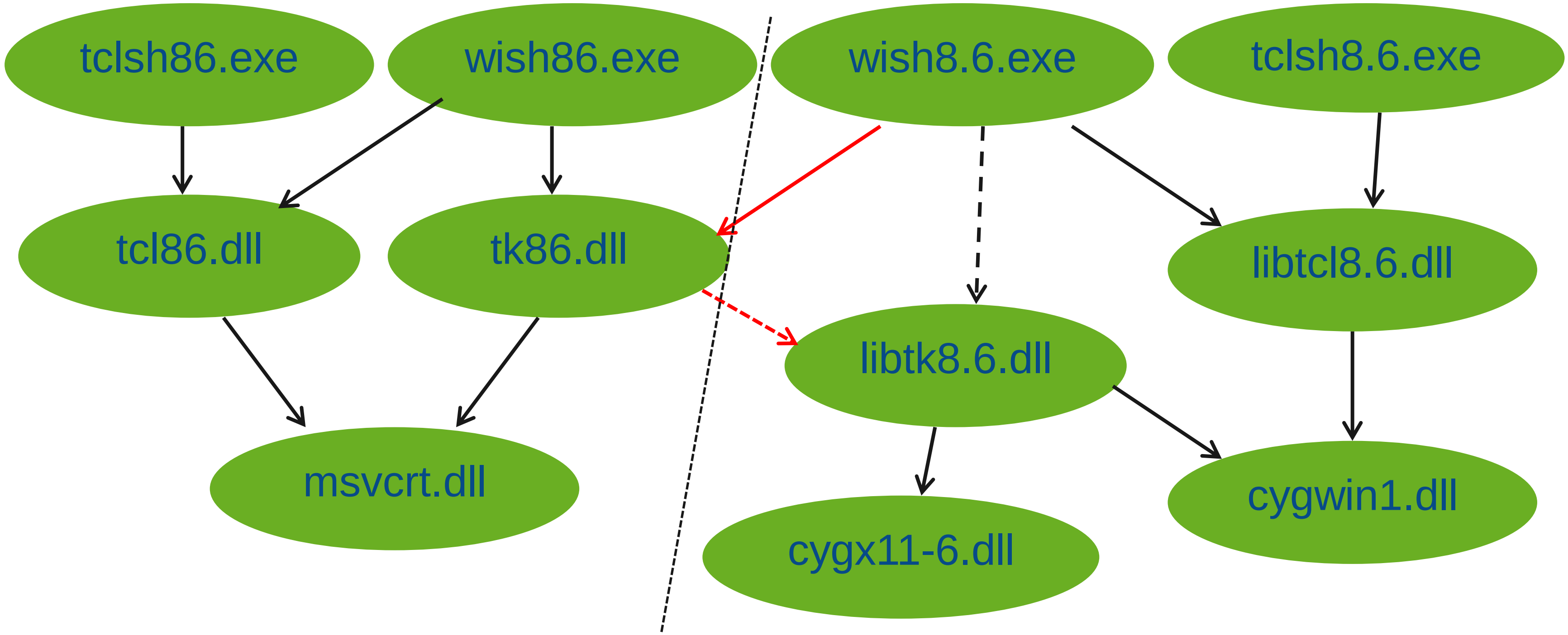
  - struct Tcl_StatBuf

Load dde.dll/reg.dll in cygwin tclsh:

- Stub table adaptation

```
typedef struct TclIntPlatStubs {
     …..
#if !defined(__WIN32__) && !defined(__CYGWIN__) /* UNIX */
    …..
    TclFile (*tclpCreateTempFile) (…..); /* 9 */
    ….
#endif /* UNIX */
#if defined(__WIN32__) || defined(__CYGWIN__) /* WIN */
    …..
     int (*tclWinGetPlatformId) _ANSI_ARGS_(…..); /* 9 */
    …..
endif /* WIN */
    …..
} TclIntPlatStubs;
```

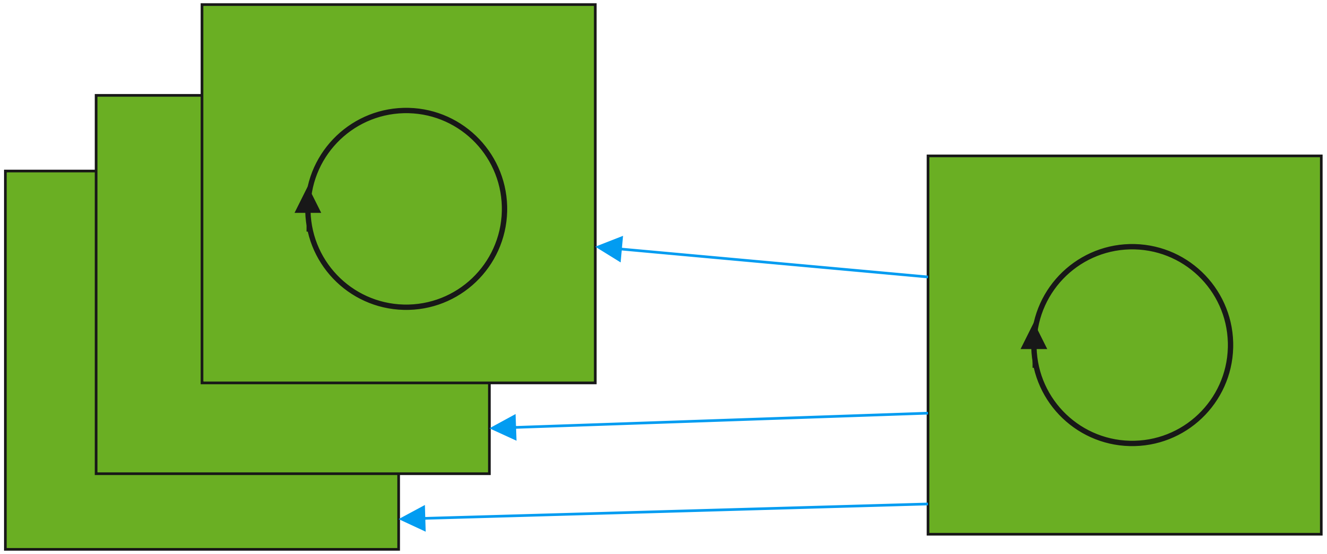
Load tk86.dll in tclsh8.6.exe



- If all stub entries are compatible this should be possible
- Wrong: tclsh8.6 doesn’t know anything about win32 events.

Event loop

- Notifier
  - Threaded or not



Notifier implementation

- UNIX:
  - select()
  - Tcl_ConditionNotify/Tcl_ConditionWait()
- Windows:
  - MsgWaitForMultipleObjects()
  - Tcl_ConditionNotify/Tcl_ConditionWait()
- Cygwin
  - select()
  - PostMessage/MsgWaitForMultipleObjects()

Macro’s vs functions

- win32:
  - #define Xfree(data) {if (data) ckfree(data)}
- X11:
  - extern int Xfree(void *data);

~~Macro~~

Conclusion

- Cygwin Tcl/Tk: UNIX but with win32 additions
  - registry/dde
  - [tk windowingsystem] = “x11” + “win32”
- Cygwin as build environment (gcc/MinGW-w64)
- All stub entries equal between win32/Cygwin
- Event loop + win32 events
- Rebase: give each dll a fixed base

- Questions?

HUMIQ

Advanced Software

ICT Automatisering