Ruby master - Bug #4141

Tk extension is not accepting any type of parameter combination

12/10/2010 05:01 AM - luislavena (Luis Lavena)

Status:	Closed		
Priority:	Normal		
Assignee:	nagai (Hidetoshi Nagai)		
Target version:	2.0.0		
ruby -v:	ruby 1.9.3dev (2010-12-09 trunk 30151) [i386-mingw32]	**Backport:**	

Description

=begin
Hello,

As part of RubyInstaller team, we want to include tk extension in the next release. For that purpose, we built both tcl and tk 8.5.9 and made bin, includes and lib directories available using the standard GCC variables (PATH, CPATH and LIBRARY_PATH)

All the existing Ruby extensions (openssl, curses, etc) work with this and properly detect the available headers, except tk extension.

Normally, this is the output:

compiling tk
check functions.....
check struct members..
Use ActiveTcl libraries (if available).
Search tclConfig.sh and tkConfig.sh.....
Fail to find [tclConfig.sh, tkConfig.sh]
Search Tcl library
Warning:: cannot find Tcl library. tcltklib will not be compiled (tcltklib is disabled on your Ruby
== Ruby/Tk will not work). Please check configure options.
Can't find proper Tcl/Tk libraries. So, can't make tcltklib.so which is required by Ruby/Tk.
compiling tk/tkutil

To workaround this, I've tried manually provide parameters to Ruby configure process (like --with-tcl-dir, --with-tk-dir, --with-tcllib and --with-tklib), all been marked as warning and no effect in the extension itself.

Manually trying to build the extension and provide these parameters:

mkdir build
cd build
ruby ..\extconf.rb --with-tcl-dir=../../tcl --with-tk-dir=../../tk --with-tcllib=tcl85s --with-tklib=tk85s

Since tk extension checks for tcl85g, not the static version (as build for RubyInstaller)

Generates the following output:

Configure options for Ruby/Tk may be updated.
So, delete files which depend on old configs.
check functions.checking for ruby_native_thread_p() in ruby.h... yes
.checking for rb_errinfo() in ruby.h... yes
.checking for rb_safe_level() in ruby.h... yes
.checking for rb_hash_lookup() in ruby.h... yes
.checking for rb_proc_new() in ruby.h... yes
.checking for rb_obj_untrust() in ruby.h... yes
.checking for rb_obj_taint() in ruby.h... yes
.checking for rb_set_safe_level_force() in ruby.h... yes
.checking for rb_sourcefile() in ruby.h... yes

check struct members.checking for struct RArray.ptr in ruby.h... no
.checking for struct RArray.len in ruby.h... no

Use ActiveTcl libraries (if available).
Search tclConfig.sh (in ../../tcl/lib) and tkConfig.sh (in ../../tk/lib)..
WARNING: found "../../tcl/lib/tclConfig.sh", but cannot find valid Tcl/Tk libraries on the same directory. So, ignore it.

WARNING: found "../../tcl/lib/tclconfig.sh", but cannot find valid Tcl/Tk libraries on the same directory. So, ignore it.

Fail to find [tclConfig.sh, tkConfig.sh]
Search Tcl library.checking for Tcl_FindExecutable() in -ltcl85s... yes

Search Tk library.checking for Tk_Init() in -ltk85s... no

Warning:: cannot find Tk library. tcltklib will not be compiled (tcltklib is disabled on your Ruby == Ruby/Tk will not work). Please check configure options.
Can't find proper Tcl/Tk libraries. So, can't make tcltklib.so which is required by Ruby/Tk.
*** ../extconf.rb failed ***

Checking libtk85s.a for symbols (using nm) the Tk_Init symbol is defined.

I've tried this way and also using --with-old-extconf, both with similar results.

If extconf can't be make compatible with the default build process, what tweak is required for a successful compilation?

Thank you.
=end

Related issues:

Related to Backport192 - Backport #4802: Please backport r31742	**Rejected**

Associated revisions

Revision 542190ce - 05/26/2011 11:36 PM - nagai (Hidetoshi Nagai)

- ext/tk/lib/tk.rb: hang-up at exit before calling Tk.mainloop.
- ext/tk/lib/tk/extconf.rb: cannot make on MinGW [Ruby 1.9 - Bug #4141].

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@31742 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 31742 - 05/26/2011 11:36 PM - nagai (Hidetoshi Nagai)

- ext/tk/lib/tk.rb: hang-up at exit before calling Tk.mainloop.
- ext/tk/lib/tk/extconf.rb: cannot make on MinGW [Ruby 1.9 - Bug #4141].

Revision 31742 - 05/26/2011 11:36 PM - nagai (Hidetoshi Nagai)

- ext/tk/lib/tk.rb: hang-up at exit before calling Tk.mainloop.
- ext/tk/lib/tk/extconf.rb: cannot make on MinGW [Ruby 1.9 - Bug #4141].

Revision 31742 - 05/26/2011 11:36 PM - nagai (Hidetoshi Nagai)

- ext/tk/lib/tk.rb: hang-up at exit before calling Tk.mainloop.

- ext/tk/lib/tk/extconf.rb: cannot make on MinGW [Ruby 1.9 - Bug #4141].

Revision 31742 - 05/26/2011 11:36 PM - nagai (Hidetoshi Nagai)

- ext/tk/lib/tk.rb: hang-up at exit before calling Tk.mainloop.
- ext/tk/lib/tk/extconf.rb: cannot make on MinGW [Ruby 1.9 - Bug #4141].

Revision 31742 - 05/26/2011 11:36 PM - nagai (Hidetoshi Nagai)

- ext/tk/lib/tk.rb: hang-up at exit before calling Tk.mainloop.
- ext/tk/lib/tk/extconf.rb: cannot make on MinGW [Ruby 1.9 - Bug #4141].

Revision 31742 - 05/26/2011 11:36 PM - nagai (Hidetoshi Nagai)

- ext/tk/lib/tk.rb: hang-up at exit before calling Tk.mainloop.
- ext/tk/lib/tk/extconf.rb: cannot make on MinGW [Ruby 1.9 - Bug #4141].

History

#1 - 12/27/2010 10:05 AM - nagai (Hidetoshi Nagai)

=begin
Hi,

From: Luis Lavena redmine@ruby-lang.org
Subject: [ruby-core:33656] [Ruby 1.9-Bug#4141][Open] Tk extension is not accepting any type of parameter combination
Date: Fri, 10 Dec 2010 05:02:24 +0900
Message-ID:

> Bug #4141: Tk extension is not accepting any type of parameter combination
> http://redmine.ruby-lang.org/issues/show/4141

I'm very sorry. My reply is too late for a new RubyInstaller.
Could you try the following patch (and new options)?

Index: README.tcltklib
===
--- README.tcltklib (revision 30168)
+++ README.tcltklib (working copy)
@@ -29,10 +29,32 @@
Old "extconf.rb" doesn't support this option.

--with-tcltkversion=

- --with-tcltkversion=, force version of Tcl/Tk libaray
- (e.g. libtcl8.4g.so ==> --with-tcltkversion=8.4g)
- (e.g. libtcl8.4g.so & libtk8.4g.so ==> --with-tcltkversion=8.4g

- [redacted]
- --without-tcl-config / --without-tk-config
- --enable-tcl-h-ver-check/--disable-tcl-h-ver-check

- --enable-tk-h-ver-check/--disable-tk-h-ver-check
- [redacted]
- [redacted]
- [redacted]

  +
- --with-tcl-build-dir=
- --with-tk-build-dir= If you want to compile with pre-installed Tcl/Tk
- [redacted]
- [redacted]
- [redacted]
- [redacted]
- [redacted]
- [redacted]

  +
- --with-tclConfig-file=
- --with-tkConfig-file= file path of tclConfig.sh/tkConfig.sh.
- [redacted]
- [redacted]
- [redacted]

  +
  --with-tclConfig-dir=

  --with-tkConfig-dir= the directory contains 'tclConfig.sh' and
  'tkConfig.sh'.

Index: extconf.rb

--- extconf.rb (revision 30168)
+++ extconf.rb (working copy)
@@ -10,7 +10,9 @@

%w[8.7 8.6 8.5 8.4 8.3 8.2 8.1 8.0]

%w[8.7 8.6 8.5 8.4 8.0] # to shorten search steps

+TkLib_Config['major_nums'] = '87'

+
##
use old extconf.rb ?
##
@@ -111,23 +113,25 @@
/darwin/ =~ RUBY_PLATFORM
end

+def maybe_64bit?

- /64|universal/ =~ RUBY_PLATFORM
  +end
  +
  def check_tcltk_version(version)
  return [nil, nil] unless version.kind_of? String

- version = version.strip
- tclver, tkver = version.split(',')
- tclver = tclver.strip
- return [tclver, tkver.strip] if tkver
- tclver = version.dup
- **tkver = version.dup**

  dot = major = minor_dot = minor = plvl_dot = plvl = ext = nil
- if version =~ /^(.?)(\d)(.?)(\d*)(.*)$/
- if tclver =~ /^(.?)(\d)(.?)(\d*)(.*)$/
  major = $1; minor_dot = $2; minor = $3; plvl_dot = $4; plvl = $5; ext = $6
  dot = ! minor_dot.empty?
  if plvl_dot.empty? && ! plvl.empty?
  minor << plvl
  end
- elsif version =~ /^(.?)(\d?)(.*)$/
- elsif tclver =~ /^(.?)(\d?)(.*)$/
  major = $1; minor_dot = $2; minor = $3; ext = $4
  dot = ! minor_dot.empty?
  else # unknown -> believe user
  @@ -140,9 +144,12 @@
  tkver = "4" + ((dot)? ".": "") + ((minor.empty)? "": "2") + ext
  elsif major == "4" # Tk4.2 (not support Tkversion < 4.2)
  # Tcl7.6
- tkver = tclver
  tclver = "7" + ((dot)? ".": "") + ((minor.empty)? "": "6") + ext
  end
- tkver = tclver unless tkver
  +
  [tclver, tkver]
  end

@@ -187,11 +194,12 @@
if CROSS_COMPILING
elsif is_win32?
if TkLib_Config["ActiveTcl"]

- path_head.concat ["c:/ActiveTcl", "c:/Program Files/ActiveTcl"]
- path_head.concat ["c:/ActiveTcl", "c:/Program Files/ActiveTcl",
- "c:/Program Files (x86)/ActiveTcl"]   end   path_head.concat [
- "c:/Tcl", "c:/Program Files/Tcl",
- "/Tcl", "/Program Files/Tcl"
- "c:/Tcl", "c:/Program Files/Tcl", "c:/Program Files (x86)/Tcl",
- "/Tcl", "/Program Files/Tcl", "/Program Files (x86)/Tcl"   ]   path_head.each{|dir| path_dirs << "#{dir}"}

@@ -203,6 +211,7 @@
].each{|dir|
next unless File.directory?(dir)

- path_dirs << "#{dir}/lib64" if maybe_64bit?   path_dirs << "#{dir}/lib"   path_dirs << "#{dir}" unless Dir.glob("#{dir}/lib*.*",
  File::FNM_CASEFOLD).empty?

@@ -371,8 +380,8 @@
end

def get_libpath(lib_flag, lib_spec)

- # get libpath fro {TCL,Tk}_LIB_FLAG and {TCL,Tk}_LIB_SPEC
- libpath = lib_spec.gsub(/(#{lib_flag}|-L)/, "").strip
- # get libpath from {TCL,Tk}_LIB_FLAG and {TCL,Tk}_LIB_SPEC
- lib_spec.gsub(/(#{lib_flag}|-L)/, "").strip end

def get_tclConfig_dirs
@@ -387,14 +396,14 @@
end
dirs.concat [
"c:/ActiveTcl*/lib", "c:/Tcl*/lib",

- "c:/Program Files/ActiveTcl*/lib", "c:/Program Files/Tcl*/lib",
- "c:/Program Files*/ActiveTcl*/lib", "c:/Program Files*/Tcl*/lib", "/ActiveTcl*/lib", "/Tcl*/lib",
- "/Program Files/ActiveTcl*/lib", "/Program Files/Tcl*/lib"
- "/Program Files*/ActiveTcl*/lib", "/Program Files*/Tcl*/lib"] else dirs = [
- "c:/Tcl*/lib", "c:/Program Files/Tcl*/lib",
- "/Tcl*/lib", "/Program Files/Tcl*/lib"
- "c:/Tcl*/lib", "c:/Program Files*/Tcl*/lib",

- [redacted]
  ]

  end
  dirs.collect{|d| Dir.glob(d, File::FNM_CASEFOLD)}.flatten!
  @@ -443,22 +452,28 @@
  end
  end

- config_dir.concat [
- [redacted]
-]
- config_dir << RbConfig::CONFIG['libdir']
- ((maybe_64bit?)? ['lib64', 'lib']: ['lib']).each{|dir|
- [redacted]
- }
  +
  config_dir.concat [
  '/opt', '/pkg', '/share',
  '/usr/local/opt', '/usr/local/pkg', '/usr/local/share', '/usr/local',
  '/usr/opt', '/usr/pkg', '/usr/share', '/usr/contrib', '/usr'
  ].map{|dir|

- [redacted]

- [redacted]
- [redacted]
  ```
  Dir.glob(dir + '/{tcltk,tcl,tk}/lib', File::FNM_CASEFOLD)
  Dir.glob(dir + '/{tcltk,tcl,tk}', File::FNM_CASEFOLD)
  ```

  }.flatten!
  @@ -489,8 +504,8 @@
  ]
  paths.reverse! unless TkLib_Config["ActiveTcl"]

- paths.each{|framework|
- [redacted]
- paths.each{|frmwk|
- [redacted]
  ```
  config_dir << [
    File.join(base, 'Tcl.framework'), File.join(base, 'Tk.framework')
  ]
  ```

  @@ -511,7 +526,7 @@
  config_dir
  end

-def libcheck_for_tclConfig(dir, tclconf, tkconf)
+def libcheck_for_tclConfig(tcldir, tkdir, tclconf, tkconf)
tcllib_ok = tklib_ok = false

```
if TkLib_Config["tcltk-stubs"]
```

@@ -524,29 +539,35 @@
tkfunc = "Tk_Init"
end

- incflags = $INCFLAGS
  libpath = $LIBPATH
  tcllibs = nil

  begin

- tcllib_ok ||= Dir.glob(File.join(dir, "*tcl#{stub}#{tclconf['TCL_MAJOR_VERSION']}{.,}#{tclconf['TCL_MINOR_VERSION']}.**"),
  File::FNM_CASEFOLD).find{|file|

- tcllib_ok ||= Dir.glob(File.join(tcldir, "*tcl#{stub}#{tclconf['TCL_MAJOR_VERSION']}{.,}#{tclconf['TCL_MINOR_VERSION']}."),
  File::FNM_CASEFOLD).find{|file|
  if file =~ /(tcl#{stub}#{tclconf['TCL_MAJOR_VERSION']}(.|)#{tclconf['TCL_MINOR_VERSION']}.*).[*]$/

- [redacted]
- [redacted]
- [redacted]
- [redacted]
  ```
    tcllibs = append_library($libs, $1)
  ```
- [redacted]
- [redacted]
  ```
    try_func(tclfunc, tcllibs)
  end
  ```

  }

- tklib_ok ||= Dir.glob(File.join(dir, "*tk#{stub}#{tkconf['TK_MAJOR_VERSION']}{.,}#{tkconf['TK_MINOR_VERSION']}.**"),
  File::FNM_CASEFOLD).find{|file|

- tklib_ok ||= Dir.glob(File.join(tkdir, "*tk#{stub}#{tkconf['TK_MAJOR_VERSION']}{.,}#{tkconf['TK_MINOR_VERSION']}."),
  File::FNM_CASEFOLD).find{|file|
  if file =~ /(tk#{stub}#{tkconf['TK_MAJOR_VERSION']}(.|)#{tkconf['TK_MINOR_VERSION']}.*).[*]$/

- [redacted]

- [redacted]
- [redacted]
- [redacted]
- [redacted]

  +
- [redacted]
- [redacted]
- [redacted]
- [redacted]
- [redacted]
- [redacted]

  ```
    try_func(tkfunc, tklibs)
  end
  ```

  }
  ensure

- $INCFLAGS = incflags
  $LIBPATH = libpath
  end

@@ -630,7 +651,8 @@
tcllib_ok = tklib_ok = true
else
tcllib_ok, tklib_ok = libcheck_for_tclConfig(File.dirname(tclpath),

- tclconf, tkconf)
- File.dirname(tkpath),

- [redacted]

  =begin
  tcllib_ok = tklib_ok = false
  if TkLib_Config["tcltk-stubs"]
  @@ -877,8 +899,8 @@

  if !CROSS_COMPILING and is_win32?
  default_paths.concat [

- [redacted]
- [redacted]
- [redacted]
- [redacted]

  ].find_all{|dir| File.directory?(dir)}
  end

@@ -928,15 +950,19 @@
print(".")
[path, find_library(tcllib, func, path)]
else

- sufx_list = ['', 't', 'g', 's', 'x'] st = search_vers_on_path(versions, path, lib, 'tcl').find{|ver|
- (print(".");find_library("#{lib}#{ver}", func, path)) or
- (print(".");find_library("#{lib}#{ver.delete('.')}", func, path)) or
- (print(".");find_library("#{lib}#{ver}g", func, path)) or
- (print(".");find_library("#{lib}#{ver.delete('.')}g", func, path)) or
- (print(".");find_library("tcl#{ver}", func, path)) or
- (print(".");find_library("tcl#{ver.delete('.')}", func, path)) or

- (print(".");find_library("tcl#{ver}g", func, path)) or
- (print(".");find_library("tcl#{ver.delete('.')}g", func, path))
- dir_enum = Dir.foreach(path)
- no_dot_ver = ver.delete('.')
- libnames = ["#{lib}#{ver}", "#{lib}#{no_dot_ver}"]
- libnames << "tcl#{ver}" << "tcl#{no_dot_ver}" if lib != "tcl"
- libnames.find{|libname|
- sufx_list.find{|sufx|
- print(".")
- dir_enum.find{|fname| fname =~ /#{libname + sufx}/} &&
- find_library(libname + sufx, func, path)
- }

- [redacted]

```
} || (!version && (print(".");find_library(lib, func, path)))
[path, st]
```

end
@@ -989,8 +1015,8 @@

if !CROSS_COMPILING and is_win32?
default_paths.concat [

- [redacted]
- [redacted]
- [redacted]
- [redacted]

].find_all{|dir| File.directory?(dir)}
end

@@ -1039,15 +1065,19 @@
print(".")
[path, find_library(tklib, func, path)]
else

- sufx_list = ['', 't', 'g', 's', 'x']   st = search_vers_on_path(versions, path, lib, 'tk').find{|ver|
- (print(".");find_library("#{lib}#{ver}", func, path)) or
- (print(".");find_library("#{lib}#{ver.delete('.')}", func, path)) or
- (print(".");find_library("#{lib}#{ver}g", func, path)) or
- (print(".");find_library("#{lib}#{ver.delete('.')}g", func, path)) or
- (print(".");find_library("tk#{ver}", func, path)) or
- (print(".");find_library("tk#{ver.delete('.')}", func, path)) or
- (print(".");find_library("tk#{ver}g", func, path)) or
- (print(".");find_library("tk#{ver.delete('.')}g", func, path))
- dir_enum = Dir.foreach(path)
- no_dot_ver = ver.delete('.')
- libnames = ["#{lib}#{ver}", "#{lib}#{no_dot_ver}"]
- libnames << "tk#{ver}" << "tk#{no_dot_ver}" if lib != "tk"
- libnames.find{|libname|
- sufx_list.find{|sufx|
- print(".")
- dir_enum.find{|fname| fname =~ /#{libname + sufx}/} &&
- find_library(libname + sufx, func, path)
- }

- [redacted]

```
} || (!version && (print(".");find_library(lib, func, path)))
[path, st]
```

end
@@ -1085,8 +1115,10 @@

if !CROSS_COMPILING && is_win32?
base_dir.concat [

- [redacted]
- [redacted]

- [redacted]
- [redacted]
- [redacted]
- [redacted]

  ].find_all{|dir| File.directory?(dir)}
  end

@@ -1097,41 +1129,81 @@
if TclConfig_Info['TCL_INCLUDE_SPEC'] &&
have_tcl_h = try_cpp('#include ', TclConfig_Info['TCL_INCLUDE_SPEC'])
$INCFLAGS << " " << TclConfig_Info['TCL_INCLUDE_SPEC']

- elsif have_tcl_h = have_header('tcl.h')
- # find else
- if tclver && ! tclver.empty?
- versions = [tclver]
- if enable_config("tcl-h-ver-check", true) &&
- tclver && tclver =~ /\D*(\d).?(\d)/
- major = $1; minor = $2   else
- versions = TkLib_Config['search_versions']
- major = minor = nil   end
- paths = base_dir.dup
- versions.each{|ver|
- paths.concat(base_dir.map{|dir|
- [dir + '/tcl' + ver, dir + '/tcl' + ver.delete('.')]
- }.flatten)
- }
- have_tcl_h = find_header('tcl.h', *paths)
- if major && minor
- # version check on tcl.h
- have_tcl_h = try_cpp("#include \n#if TCL_MAJOR_VERSION != #{major} || TCL_MINOR_VERSION != #{minor}\n#error VERSION does not match\n#fi")
- else
- have_tcl_h = have_header('tcl.h')
- end
- unless have_tcl_h
- if tclver && ! tclver.empty?
- versions = [tclver]
- else
- versions = TkLib_Config['search_versions']
- end
- paths = base_dir.dup
- versions.each{|ver|
- paths.concat(base_dir.map{|dir|
- [dir + '/tcl' + ver, dir + '/tcl' + ver.delete('.')]
- }.flatten)
- }
- if enable_config("tcl-h-ver-check", true)
- # version check on tcl.h
- have_tcl_h = paths.find{|path|
- try_cpp("#include \n#if TCL_MAJOR_VERSION != #{major} || TCL_MINOR_VERSION != #{minor}\n#error VERSION does not match\n#fi", "-I#{path.quote}")
- }
- $INCFLAGS << " " << "-I#{have_tcl_h.quote}" if have_tcl_h
- else
- have_tcl_h = find_header('tcl.h', *paths)
- end

- end
  end

  if TkConfig_Info['TK_INCLUDE_SPEC'] &&
  have_tk_h = try_cpp('#include ', TkConfig_Info['TK_INCLUDE_SPEC'])
  $INCFLAGS << " " << TkConfig_Info['TK_INCLUDE_SPEC']

- elsif have_tk_h = have_header('tk.h')

- # find
  else

- if tkver && ! tkver.empty?
- [redacted]
- if enable_config("tk-h-ver-check", true) &&
- [redacted]
- [redacted]

  else
- [redacted]
- [redacted]

  end
- paths = base_dir.dup
- versions.each{|ver|
- [redacted]
- [redacted]
- [redacted]
- }
- have_tk_h = find_header('tk.h', *paths)
- if major && minor
- [redacted]
- [redacted]
- else
- [redacted]
- end
- unless have_tk_h
- [redacted]
- [redacted]
- [redacted]
- [redacted]
- [redacted]
- [redacted]
- [redacted]
- [redacted]
- [redacted]
- [redacted]
- [redacted]
- [redacted]
- [redacted]

- [redacted]
- [redacted]
- [redacted]
- [redacted]
- [redacted]
- [redacted]
- end
  end

  have_tcl_h && have_tk_h
  @@ -1200,11 +1272,13 @@
  end

def search_X_libraries

- use_tkConfig = false if TkConfig_Info['config_file_path'] # use definitions on tkConfig.sh
- if TkConfig_Info['TK_XINCLUDES'] && TkConfig_Info['TK_XLIBSW'] &&
- !TkConfig_Info['TK_XINCLUDES'].strip.empty? &&
- !TkConfig_Info['TK_XLIBSW'].strip.empty?
- if (TkConfig_Info['TK_XINCLUDES'] &&
- !TkConfig_Info['TK_XINCLUDES'].strip.empty?) ||
- (TkConfig_Info['TK_XLIBSW'] && !TkConfig_Info['TK_XLIBSW'].strip.empty?)

- [redacted]
  `#use_X = true && with_config("X11", ! is_win32?)`
  `use_X = with_config("X11", true)`

  else
  @@ -1216,21 +1290,28 @@
  use_X = with_config("X11", !(is_win32? || TkLib_Config["tcltk-framework"]))
  end

- if TkConfig_Info['TK_XINCLUDES'] &&
- [redacted]
- $INCFLAGS << " " << TkConfig_Info['TK_XINCLUDES'].strip
- end
  +
  if use_X
- puts("Use X11 libraries.")
- puts("Use X11 libraries (or use TK_XINCLUDES/TK_XLIBSW information on tkConfig.sh).")
  x11_idir, x11_ldir = dir_config("X11")
  x11_ldir2 = with_config("X11-lib")
  unless find_X11(x11_ldir2, x11_ldir)
- [redacted]
- [redacted]
- [redacted]
- [redacted]
- [redacted]
- [redacted]
- [redacted]
- [redacted]

- [redacted]
- [redacted]

  end
  end

- if TkConfig_Info['TK_XINCLUDES'] &&
- [redacted]
- $INCFLAGS << " " << TkConfig_Info['TK_XINCLUDES'].strip
- **end**

  use_X
  end

@@ -1272,7 +1353,7 @@
else
puts("Warning: '#{TclConfig_Info['config_file_path']}' may not be a tclConfig file.")
end

- tclConfig = false
- #tclConfig = false end end end @@ -1280,7 +1361,7 @@ if tcl_enable_thread == nil && !TclConfig_Info['config_file_path'] # tcl-thread is unknown and tclConfig is unavailable begin
- try_run_available = try_run("int main() { exit(0); }")
- try_run("int main() { exit(0); }") rescue Exception # cannot try_run. Is CROSS-COMPILE environment? puts(%Q'\ @@ -1439,15 +1520,61 @@

#--
+TclConfig_Info = {}
+TkConfig_Info = {}
+
+# use Tcl/Tk build dir? (has highest priority)
+TkLib_Config["tcl-build-dir"] = with_config("tcl-build-dir")
+TkLib_Config["tk-build-dir"] = with_config("tk-build-dir")
+if TkLib_Config["tcl-build-dir"]

- puts("use Tcl build (pre-install) dir \"#{TkLib_Config["tcl-build-dir"]}\"")
- TkLib_Config["tcl-build-dir"] = File.expand_path(TkLib_Config["tcl-build-dir"])
- base = File.dirname(TkLib_Config["tcl-build-dir"])
- $INCFLAGS << " -I#{File.join(base, "generic").quote} -I#{TkLib_Config["tcl-build-dir"].quote}"
- $LIBPATH |= [TkLib_Config["tcl-build-dir"]] +end +if TkLib_Config["tk-build-dir"]
- puts("use Tk build (pre-install) dir \"#{TkLib_Config["tk-build-dir"]}\"")
- TkLib_Config["tk-build-dir"] = File.expand_path(TkLib_Config["tk-build-dir"])
- base = File.dirname(TkLib_Config["tk-build-dir"])
- $INCFLAGS << " -I#{File.join(base, "generic").quote} -I#{TkLib_Config["tk-build-dir"].quote}"
- $LIBPATH |= [TkLib_Config["tk-build-dir"]] +end + # check requirement of Tcl/tk version tcltk_version = with_config("tcltkversion") -tclver, tkver =
- TkLib_Config["tcltkversion"] = check_tcltk_version(tcltk_version) -puts("Specified Tcl/Tk version is #{[tclver, tkver].inspect}") if tclver&&tkver +TkLib_Config["tcltkversion"] = check_tcltk_version(tcltk_version)

+if TkLib_Config["tcl-build-dir"]

- if (cfgfile = with_config("tclConfig-file", Dir.glob(File.join(TkLib_Config["tcl-build-dir"], "tclConfig*.sh"), File::FNM_CASEFOLD)[0]))
- TclConfig_Info['config_file_path'] = cfgfile
- TkLib_Config["tclConfig_info"] = cfginfo = parse_tclConfig(cfgfile)
- if tclver = TkLib_Config["tcltkversion"][0]
- TkLib_Config["tcltkversion"][0].sub!(/\d(.?)\d/, "#{cfginfo['TCL_MAJOR_VERSION']}\1#{cfginfo['TCL_MINOR_VERSION']}")
- else
- TkLib_Config["tcltkversion"][0] = "#{cfginfo['TCL_MAJOR_VERSION']}.#{cfginfo['TCL_MINOR_VERSION']}"
- end
- end +end +if TkLib_Config["tk-build-dir"]
- if (cfgfile = with_config("tkConfig-file", Dir.glob(File.join(TkLib_Config["tk-build-dir"], "tkConfig*.sh"), File::FNM_CASEFOLD)[0]))
- TkConfig_Info['config_file_path'] = cfgfile
- TkLib_Config["tkConfig_info"] = cfginfo = parse_tclConfig(cfgfile)
- if TkLib_Config["tcltkversion"][1]
- TkLib_Config["tcltkversion"][1].sub!(/\d(.?)\d/, "#{cfginfo['TK_MAJOR_VERSION']}\1#{cfginfo['TK_MINOR_VERSION']}")
- else
- TkLib_Config["tcltkversion"][1] = "#{cfginfo['TK_MAJOR_VERSION']}.#{cfginfo['TK_MINOR_VERSION']}"
- end
- end +end + +tclver, tkver = TkLib_Config["tcltkversion"] +puts("Specified Tcl/Tk version is #{[tclver, tkver].inspect}") if tclver||tkver + # use ActiveTcl ? #if activeTcl = with_config("ActiveTcl") -if activeTcl = with_config("ActiveTcl", true) +#if activeTcl = with_config("ActiveTcl", true) +if activeTcl = with_config("ActiveTcl", !(TkLib_Config["tcl-build-dir"] && TkLib_Config["tk-build-dir"])) puts("Use ActiveTcl libraries (if available).")

unless activeTcl.kind_of? String # set default ActiveTcl path @@ -1490,8 +1617,18 @@ end

```
# directory configuration of Tcl/Tk libraries
-tk_idir,  tk_ldir  = dir_config("tk")
-tcl_idir, tcl_ldir = dir_config("tcl")
+if TkLib_Config["tcl-build-dir"]
```

- tcl_idir = File.join(File.dirname(TkLib_Config["tcl-build-dir"]),"generic")
- tcl_ldir = TkLib_Config["tcl-build-dir"] +else
- tcl_idir, tcl_ldir = dir_config("tcl") +end +if TkLib_Config["tk-build-dir"]
- tk_idir = File.join(File.dirname(TkLib_Config["tk-build-dir"]),"generic")
- tk_ldir = TkLib_Config["tk-build-dir"] +else
- tk_idir, tk_ldir = dir_config("tk") +end

```
tcl_idir = tk_idir unless tcl_idir
tcl_ldir = tk_ldir unless tcl_ldir
@@ -1500,14 +1637,21 @@

# get tclConfig.sh/tkConfig.sh
TkLib_Config["tcltk-NG-path"] = []
-tclcfg, tkcfg = get_tclConfig(with_config("tclConfig-file", true),
```

- with_config("tkConfig-file", true),
- with_config("tclConfig-dir", tcl_ldir || true),
- with_config("tkConfig-dir", tk_ldir || true)) -TclConfig_Info = TkLib_Config["tclConfig_info"] || {} -TkConfig_Info = TkLib_Config["tkConfig_info"] || {} -TclConfig_Info['config_file_path'] = tclcfg -TkConfig_Info['config_file_path'] = tkcfg +tclcfg, tkcfg =
- get_tclConfig(
- TclConfig_Info['config_file_path'] || with_config("tclConfig-file", true),
- TkConfig_Info['config_file_path'] || with_config("tkConfig-file", true),
- (TclConfig_Info['config_file_path'])?
- File.dirname(TclConfig_Info['config_file_path']) :
- with_config("tclConfig-dir", tcl_ldir || true),
- (TkConfig_Info['config_file_path'])?
- File.dirname(TkConfig_Info['config_file_path']) :
- with_config("tkConfig-dir", tk_ldir || true)
-) +TclConfig_Info.merge!(TkLib_Config["tclConfig_info"]) +TkConfig_Info.merge!(TkLib_Config["tkConfig_info"]) +TclConfig_Info['config_file_path'] ||= tclcfg +TkConfig_Info['config_file_path'] ||= tkcfg

```
TclConfig_Info['TCL_INCLUDE_SPEC'] = "-I#{tcl_idir.quote}" if tcl_idir
TkConfig_Info['TK_INCLUDE_SPEC']   = "-I#{tk_idir.quote}"  if tk_idir
@@ -1515,9 +1659,26 @@
tk_cfg_dir  = File.dirname(TkConfig_Info['config_file_path'])  rescue nil
tcl_cfg_dir = File.dirname(TclConfig_Info['config_file_path']) rescue nil

-tk_ldir_list  = [tk_ldir,  tk_cfg_dir]
-tcl_ldir_list = [tcl_ldir, tcl_cfg_dir]
+tk_ldir_list  = [tk_ldir,  tk_cfg_dir].uniq
+tcl_ldir_list = [tcl_ldir, tcl_cfg_dir].uniq

+if TkConfig_Info['config_file_path']
```

- if TkLib_Config["tk-build-dir"]
- spec_dir = get_libpath(TkConfig_Info['TK_LIB_FLAG'], TkConfig_Info['TK_BUILD_LIB_SPEC'])
- else
- spec_dir = get_libpath(TkConfig_Info['TK_LIB_FLAG'], TkConfig_Info['TK_LIB_SPEC'])
- end
- tk_ldir_list << spec_dir if File.directory?(spec_dir) +end +if TclConfig_Info['config_file_path']
- if TkLib_Config["tcl-build-dir"]
- spec_dir = get_libpath(TclConfig_Info['TCL_LIB_FLAG'], TclConfig_Info['TCL_BUILD_LIB_SPEC'])
- else
- spec_dir = get_libpath(TclConfig_Info['TCL_LIB_FLAG'], TclConfig_Info['TCL_LIB_SPEC'])
- end
- tcl_ldir_list << spec_dir if File.directory?(spec_dir) +end + # check tk_shlib_search_path check_shlib_search_path(with_config('tk-shlib-search-path'))

```
@@ -1529,22 +1690,49 @@
# MacOS X Frameworks?
if TkLib_Config["tcltk-framework"]
puts("Use MacOS X Frameworks.")
```

- $LDFLAGS << " -L#{TkLib_Config["tcl-build-dir"].quote}" if TkLib_Config["tcl-build-dir"] if tcl_cfg_dir $INCFLAGS << ' ' << TclConfig_Info['TCL_INCLUDE_SPEC'] $LDFLAGS << ' ' << TclConfig_Info['TCL_LIBS']   if stubs
- $LDFLAGS << ' ' << TclConfig_Info['TCL_STUB_LIB_SPEC']
- if TkLib_Config["tcl-build-dir"] &&
- TclConfig_Info['TCL_BUILD_STUB_LIB_SPEC'] &&

- !TclConfig_Info['TCL_BUILD_STUB_LIB_SPEC'].strip.empty?
- $LDFLAGS << ' ' << TclConfig_Info['TCL_BUILD_STUB_LIB_SPEC']
- else
- $LDFLAGS << ' ' << TclConfig_Info['TCL_STUB_LIB_SPEC']
- end else
- $LDFLAGS << ' ' << TclConfig_Info['TCL_LIB_SPEC']
- if TkLib_Config["tcl-build-dir"] &&
- TclConfig_Info['TCL_BUILD_LIB_SPEC'] &&
- !TclConfig_Info['TCL_BUILD_LIB_SPEC'].strip.empty?
- $LDFLAGS << ' ' << TclConfig_Info['TCL_BUILD_LIB_SPEC']
- else
- $LDFLAGS << ' ' << TclConfig_Info['TCL_LIB_SPEC']
- end end end +
- $LDFLAGS << " -L#{TkLib_Config["tk-build-dir"].quote}" if TkLib_Config["tk-build-dir"] if tk_cfg_dir $INCFLAGS << ' ' << TkConfig_Info['TK_INCLUDE_SPEC'] $LDFLAGS << ' ' << TkConfig_Info['TK_LIBS'] if stubs
- $LDFLAGS << ' ' << TkConfig_Info['TK_STUB_LIB_SPEC']
- if TkLib_Config["tk-build-dir"] &&
- TclConfig_Info['TK_BUILD_STUB_LIB_SPEC'] &&
- !TclConfig_Info['TK_BUILD_STUB_LIB_SPEC'].strip.empty?
- $LDFLAGS << ' ' << TkConfig_Info['TK_BUILD_STUB_LIB_SPEC']
- else
- $LDFLAGS << ' ' << TkConfig_Info['TK_STUB_LIB_SPEC']
- end else
- $LDFLAGS << ' ' << TkConfig_Info['TK_LIB_SPEC']
- if TkLib_Config["tk-build-dir"] &&
- TclConfig_Info['TK_BUILD_LIB_SPEC'] &&
- !TclConfig_Info['TK_BUILD_LIB_SPEC'].strip.empty?
- $LDFLAGS << ' ' << TkConfig_Info['TK_BUILD_LIB_SPEC']
- else
- $LDFLAGS << ' ' << TkConfig_Info['TK_LIB_SPEC']
- end end end setup_for_macosx_framework(tclver, tkver) if tcl_cfg_dir && tk_cfg_dir

--
Hidetoshi NAGAI (nagai@ai.kyutech.ac.jp)
Department of Artificial Intelligence, Kyushu Institute of Technology
=end

#2 - 12/27/2010 10:05 AM - nagai (Hidetoshi Nagai)

=begin
Hi,

From: Luis Lavena redmine@ruby-lang.org
Subject: [ruby-core:33656] [Ruby 1.9-Bug#4141][Open] Tk extension is not accepting any type of parameter combination
Date: Fri, 10 Dec 2010 05:02:24 +0900
Message-ID:

> Bug #4141: Tk extension is not accepting any type of parameter combination
> http://redmine.ruby-lang.org/issues/show/4141

I'm very sorry. My reply is too late for a new RubyInstaller.
Could you try the following patch (and new options)?

Index: README.tcltklib
===
--- README.tcltklib (revision 30168)
+++ README.tcltklib (working copy)
@@ -29,10 +29,32 @@
Old "extconf.rb" doesn't support this option.

--with-tcltkversion=

- --with-tcltkversion=, force version of Tcl/Tk libaray
- (e.g. libtcl8.4g.so ==> --with-tcltkversion=8.4g)
- (e.g. libtcl8.4g.so & libtk8.4g.so ==> --with-tcltkversion=8.4g

- [redacted]

- --without-tcl-config / --without-tk-config

- --enable-tcl-h-ver-check/--disable-tcl-h-ver-check

- --enable-tk-h-ver-check/--disable-tk-h-ver-check

- [redacted]
- [redacted]
- [redacted]

  +

- --with-tcl-build-dir=
- --with-tk-build-dir= If you want to compile with pre-installed Tcl/Tk
- [redacted]
- [redacted]
- [redacted]
- [redacted]
- [redacted]
- [redacted]

  +

- --with-tclConfig-file=
- --with-tkConfig-file= file path of tclConfig.sh/tkConfig.sh.
- [redacted]
- [redacted]
- [redacted]

  +
  --with-tclConfig-dir=

  --with-tkConfig-dir= the directory contains 'tclConfig.sh' and
  'tkConfig.sh'.

Index: extconf.rb

--- extconf.rb (revision 30168)
+++ extconf.rb (working copy)
@@ -10,7 +10,9 @@

%w[8.7 8.6 8.5 8.4 8.3 8.2 8.1 8.0]

%w[8.7 8.6 8.5 8.4 8.0] # to shorten search steps

+TkLib_Config['major_nums'] = '87'

+
##
use old extconf.rb ?
##
@@ -111,23 +113,25 @@
/darwin/ =~ RUBY_PLATFORM
end

+def maybe_64bit?

- /64|universal/ =~ RUBY_PLATFORM
  +end
  +
  def check_tcltk_version(version)
  return [nil, nil] unless version.kind_of? String

- version = version.strip

- tclver, tkver = version.split(',')
- tclver = tclver.strip
- return [tclver, tkver.strip] if tkver
- tclver = version.dup
- **tkver = version.dup**

  dot = major = minor_dot = minor = plvl_dot = plvl = ext = nil
- if version =~ /')(.?)(\d)(.?)(\d*)(.*)$/
- if tclver =~ /')(.?)(\d)(.?)(\d*)(.*)$/
  major = $1; minor_dot = $2; minor = $3; plvl_dot = $4; plvl = $5; ext = $6
  dot = ! minor_dot.empty?
  if plvl_dot.empty? && ! plvl.empty?
  minor << plvl
  end
- elsif version =~ /')(.?)(\d?)(.*)$/
- elsif tclver =~ /')(.?)(\d?)(.*)$/
  major = $1; minor_dot = $2; minor = $3; ext = $4
  dot = ! minor_dot.empty?
  else # unknown -> believe user
  @@ -140,9 +144,12 @@
  tkver = "4" + ((dot)? ".": "") + ((minor.empty)? "": "2") + ext
  elsif major == "4" # Tk4.2 (not support Tkversion < 4.2)
  # Tcl7.6
- tkver = tclver
  tclver = "7" + ((dot)? ".": "") + ((minor.empty)? "": "6") + ext
  end
- tkver = tclver unless tkver
  +
  [tclver, tkver]
  end

@@ -187,11 +194,12 @@
if CROSS_COMPILING
elsif is_win32?
if TkLib_Config["ActiveTcl"]

- path_head.concat ["c:/ActiveTcl", "c:/Program Files/ActiveTcl"]
- path_head.concat ["c:/ActiveTcl", "c:/Program Files/ActiveTcl",
- "c:/Program Files (x86)/ActiveTcl"]   end   path_head.concat [
- "c:/Tcl", "c:/Program Files/Tcl",
- "/Tcl", "/Program Files/Tcl"
- "c:/Tcl", "c:/Program Files/Tcl", "c:/Program Files (x86)/Tcl",
- "/Tcl", "/Program Files/Tcl", "/Program Files (x86)/Tcl"   ]   path_head.each{|dir| path_dirs << "#{dir}"}

@@ -203,6 +211,7 @@
].each{|dir|
next unless File.directory?(dir)

- path_dirs << "#{dir}/lib64" if maybe_64bit?   path_dirs << "#{dir}/lib"   path_dirs << "#{dir}" unless Dir.glob("#{dir}/lib*.*",
  File::FNM_CASEFOLD).empty?

@@ -371,8 +380,8 @@
end

def get_libpath(lib_flag, lib_spec)

- # get libpath fro {TCL,Tk}_LIB_FLAG and {TCL,Tk}_LIB_SPEC
- libpath = lib_spec.gsub(/(#{lib_flag}|-L)/, "").strip
- # get libpath from {TCL,Tk}_LIB_FLAG and {TCL,Tk}_LIB_SPEC
- lib_spec.gsub(/(#{lib_flag}|-L)/, "").strip end

def get_tclConfig_dirs
@@ -387,14 +396,14 @@

end
dirs.concat [
"c:/ActiveTcl*/lib", "c:/Tcl*/lib",

- "c:/Program Files/ActiveTcl*/lib", "c:/Program Files/Tcl*/lib",
- "c:/Program Files*/ActiveTcl*/lib", "c:/Program Files*/Tcl*/lib", "/ActiveTcl*/lib", "/Tcl*/lib",
- "/Program Files/ActiveTcl*/lib", "/Program Files/Tcl*/lib"
- "/Program Files*/ActiveTcl*/lib", "/Program Files*/Tcl*/lib"] else dirs = [
- "c:/Tcl*/lib", "c:/Program Files/Tcl*/lib",
- "/Tcl*/lib", "/Program Files/Tcl*/lib"
- "c:/Tcl*/lib", "c:/Program Files*/Tcl*/lib",

- [redacted]
  ]

  end
  dirs.collect{|d| Dir.glob(d, File::FNM_CASEFOLD)}.flatten!
  @@ -443,22 +452,28 @@
  end
  end

- config_dir.concat [
- [redacted]
- [redacted]
- [redacted]
- [redacted]
- [redacted]
- [redacted]
-]
- config_dir << RbConfig::CONFIG['libdir']
- ((maybe_64bit?)? ['lib64', 'lib']: ['lib']).each{|dir|
- [redacted]
- [redacted]
- [redacted]
- [redacted]
- [redacted]
- [redacted]
- [redacted]
- }
  +
  config_dir.concat [
  '/opt', '/pkg', '/share',
  '/usr/local/opt', '/usr/local/pkg', '/usr/local/share', '/usr/local',
  '/usr/opt', '/usr/pkg', '/usr/share', '/usr/contrib', '/usr'
  ].map{|dir|
- [redacted]
- [redacted]
- [redacted]
- [redacted]
- [redacted]

- [redacted]
  ```
  Dir.glob(dir + '/{tcltk,tcl,tk}/lib', File::FNM_CASEFOLD)
  Dir.glob(dir + '/{tcltk,tcl,tk}', File::FNM_CASEFOLD)
  ```

  }.flatten!
  @@ -489,8 +504,8 @@
  ]
  paths.reverse! unless TkLib_Config["ActiveTcl"]

- paths.each{|framework|
- [redacted]
- paths.each{|frmwk|
- [redacted]
  ```
  config_dir << [
    File.join(base, 'Tcl.framework'), File.join(base, 'Tk.framework')
  ]
  ```

  @@ -511,7 +526,7 @@
  config_dir
  end

-def libcheck_for_tclConfig(dir, tclconf, tkconf)
+def libcheck_for_tclConfig(tcldir, tkdir, tclconf, tkconf)
tcllib_ok = tklib_ok = false

```
if TkLib_Config["tcltk-stubs"]
```

@@ -524,29 +539,35 @@
tkfunc = "Tk_Init"
end

- incflags = $INCFLAGS
  libpath = $LIBPATH
  tcllibs = nil

  begin

- tcllib_ok ||= Dir.glob(File.join(dir, "*tcl#{stub}#{tclconf['TCL_MAJOR_VERSION']}{.,}#{tclconf['TCL_MINOR_VERSION']}.**"),
  File::FNM_CASEFOLD).find{|file|

- tcllib_ok ||= Dir.glob(File.join(tcldir, "*tcl#{stub}#{tclconf['TCL_MAJOR_VERSION']}{.,}#{tclconf['TCL_MINOR_VERSION']}.*"),
  File::FNM_CASEFOLD).find{|file|
  if file =~ /(tcl#{stub}#{tclconf['TCL_MAJOR_VERSION']}(.|)#{tclconf['TCL_MINOR_VERSION']}.*)*.[*]$/*
- [redacted]
- [redacted]
- [redacted]
- [redacted]
  ```
    tcllibs = append_library($libs, $1)
  ```
- [redacted]
- [redacted]
  ```
    try_func(tclfunc, tcllibs)
  end
  ```

  }

- tklib_ok ||= Dir.glob(File.join(dir, "*tk#{stub}#{tkconf['TK_MAJOR_VERSION']}{.,}#{tkconf['TK_MINOR_VERSION']}.**"),
  File::FNM_CASEFOLD).find{|file|

- tklib_ok ||= Dir.glob(File.join(tkdir, "*tk#{stub}#{tkconf['TK_MAJOR_VERSION']}{.,}#{tkconf['TK_MINOR_VERSION']}.*"),
  File::FNM_CASEFOLD).find{|file|
  if file =~ /(tk#{stub}#{tkconf['TK_MAJOR_VERSION']}(.|)#{tkconf['TK_MINOR_VERSION']}.*)*.[*]$/*
- [redacted]
- [redacted]

- [redacted]
- [redacted]
- [redacted]

  +
- [redacted]
- [redacted]
- [redacted]
- [redacted]
- [redacted]
- [redacted]
  ```
    try_func(tkfunc, tklibs)
  end
  ```

  }
  ensure

- $INCFLAGS = incflags
  $LIBPATH = libpath
  end

@@ -630,7 +651,8 @@
tcllib_ok = tklib_ok = true
else
tcllib_ok, tklib_ok = libcheck_for_tclConfig(File.dirname(tclpath),

- tclconf, tkconf)
- File.dirname(tkpath),

- [redacted]

  =begin
  tcllib_ok = tklib_ok = false
  if TkLib_Config["tcltk-stubs"]
  @@ -877,8 +899,8 @@

  if !CROSS_COMPILING and is_win32?
  default_paths.concat [

- [redacted]
- [redacted]
- [redacted]
- [redacted]

  ].find_all{|dir| File.directory?(dir)}
  end

@@ -928,15 +950,19 @@
print(".")
[path, find_library(tcllib, func, path)]
else

- sufx_list = ['', 't', 'g', 's', 'x'] st = search_vers_on_path(versions, path, lib, 'tcl').find{|ver|
- (print(".");find_library("#{lib}#{ver}", func, path)) or
- (print(".");find_library("#{lib}#{ver.delete('.')}", func, path)) or
- (print(".");find_library("#{lib}#{ver}g", func, path)) or
- (print(".");find_library("#{lib}#{ver.delete('.')}g", func, path)) or
- (print(".");find_library("tcl#{ver}", func, path)) or
- (print(".");find_library("tcl#{ver.delete('.')}", func, path)) or
- (print(".");find_library("tcl#{ver}g", func, path)) or
- (print(".");find_library("tcl#{ver.delete('.')}g", func, path))

- dir_enum = Dir.foreach(path)
- no_dot_ver = ver.delete('.')
- libnames = ["#{lib}#{ver}", "#{lib}#{no_dot_ver}"]
- libnames << "tcl#{ver}" << "tcl#{no_dot_ver}" if lib != "tcl"
- libnames.find{|libname|
- sufx_list.find{|sufx|
- print(".")
- dir_enum.find{|fname| fname =~ /#{libname + sufx}/} &&
- find_library(libname + sufx, func, path)
- }

- [redacted]

```
} || (!version && (print(".");find_library(lib, func, path)))
[path, st]

end
@@ -989,8 +1015,8 @@

if !CROSS_COMPILING and is_win32?
default_paths.concat [
```

- [redacted]
- [redacted]
- [redacted]
- [redacted]

```
].find_all{|dir| File.directory?(dir)}
end


@@ -1039,15 +1065,19 @@
print(".")
[path, find_library(tklib, func, path)]
else
```

- sufx_list = ['', 't', 'g', 's', 'x']   st = search_vers_on_path(versions, path, lib, 'tk').find{|ver|
- (print(".");find_library("#{lib}#{ver}", func, path)) or
- (print(".");find_library("#{lib}#{ver.delete('.')}", func, path)) or
- (print(".");find_library("#{lib}#{ver}g", func, path)) or
- (print(".");find_library("#{lib}#{ver.delete('.')}g", func, path)) or
- (print(".");find_library("tk#{ver}", func, path)) or
- (print(".");find_library("tk#{ver.delete('.')}", func, path)) or
- (print(".");find_library("tk#{ver}g", func, path)) or
- (print(".");find_library("tk#{ver.delete('.')}g", func, path))
- dir_enum = Dir.foreach(path)
- no_dot_ver = ver.delete('.')
- libnames = ["#{lib}#{ver}", "#{lib}#{no_dot_ver}"]
- libnames << "tk#{ver}" << "tk#{no_dot_ver}" if lib != "tk"
- libnames.find{|libname|
- sufx_list.find{|sufx|
- print(".")
- dir_enum.find{|fname| fname =~ /#{libname + sufx}/} &&
- find_library(libname + sufx, func, path)
- }

- [redacted]

```
} || (!version && (print(".");find_library(lib, func, path)))
[path, st]

end
@@ -1085,8 +1115,10 @@

if !CROSS_COMPILING && is_win32?
base_dir.concat [
```

- [redacted]
- [redacted]
- [redacted]

- [redacted]
- [redacted]
- [redacted]

  ].find_all{|dir| File.directory?(dir)}
  end

@@ -1097,41 +1129,81 @@
if TclConfig_Info['TCL_INCLUDE_SPEC'] &&
have_tcl_h = try_cpp('#include ', TclConfig_Info['TCL_INCLUDE_SPEC'])
$INCFLAGS << " " << TclConfig_Info['TCL_INCLUDE_SPEC']

- elsif have_tcl_h = have_header('tcl.h')
- # find else
- if tclver && ! tclver.empty?
- versions = [tclver]
- if enable_config("tcl-h-ver-check", true) &&
- tclver && tclver =~ /\D*(\d).?(\d)/
- major = $1; minor = $2   else
- versions = TkLib_Config['search_versions']
- major = minor = nil   end
- paths = base_dir.dup
- versions.each{|ver|
- paths.concat(base_dir.map{|dir|
- [dir + '/tcl' + ver, dir + '/tcl' + ver.delete('.')]
- }.flatten)
- }
- have_tcl_h = find_header('tcl.h', *paths)
- if major && minor
- # version check on tcl.h
- have_tcl_h = try_cpp("#include \n#if TCL_MAJOR_VERSION != #{major} || TCL_MINOR_VERSION != #{minor}\n#error VERSION does not match\n#fi")
- else
- have_tcl_h = have_header('tcl.h')
- end
- unless have_tcl_h
- if tclver && ! tclver.empty?
- versions = [tclver]
- else
- versions = TkLib_Config['search_versions']
- end
- paths = base_dir.dup
- versions.each{|ver|
- paths.concat(base_dir.map{|dir|
- [dir + '/tcl' + ver, dir + '/tcl' + ver.delete('.')]
- }.flatten)
- }
- if enable_config("tcl-h-ver-check", true)
- # version check on tcl.h
- have_tcl_h = paths.find{|path|
- try_cpp("#include \n#if TCL_MAJOR_VERSION != #{major} || TCL_MINOR_VERSION != #{minor}\n#error VERSION does not match\n#fi", "-I#{path.quote}")
- }
- $INCFLAGS << " " << "-I#{have_tcl_h.quote}" if have_tcl_h
- else
- have_tcl_h = find_header('tcl.h', *paths)
- end

- end
  end

  if TkConfig_Info['TK_INCLUDE_SPEC'] &&
  have_tk_h = try_cpp('#include ', TkConfig_Info['TK_INCLUDE_SPEC'])
  $INCFLAGS << " " << TkConfig_Info['TK_INCLUDE_SPEC']

- elsif have_tk_h = have_header('tk.h')

- # find
  else

- if tkver && ! tkver.empty?

- [redacted]
- if enable_config("tk-h-ver-check", true) &&
- [redacted]
- [redacted]

  else
- [redacted]
- [redacted]

  end
- paths = base_dir.dup
- versions.each{|ver|
- [redacted]
- [redacted]
- [redacted]
- }
- have_tk_h = find_header('tk.h', *paths)
- if major && minor
- [redacted]
- [redacted]
- else
- [redacted]
- end
- unless have_tk_h
- [redacted]
- [redacted]
- [redacted]
- [redacted]
- [redacted]
- [redacted]
- [redacted]
- [redacted]
- [redacted]
- [redacted]
- [redacted]
- [redacted]
- [redacted]
- [redacted]
- [redacted]

- [redacted]
- [redacted]
- [redacted]
- [redacted]
- [redacted]
- end
  end

  have_tcl_h && have_tk_h
  @@ -1200,11 +1272,13 @@
  end

def search_X_libraries

- use_tkConfig = false if TkConfig_Info['config_file_path'] # use definitions on tkConfig.sh
- if TkConfig_Info['TK_XINCLUDES'] && TkConfig_Info['TK_XLIBSW'] &&
- !TkConfig_Info['TK_XINCLUDES'].strip.empty? &&
- !TkConfig_Info['TK_XLIBSW'].strip.empty?
- if (TkConfig_Info['TK_XINCLUDES'] &&
- !TkConfig_Info['TK_XINCLUDES'].strip.empty?) ||
- (TkConfig_Info['TK_XLIBSW'] && !TkConfig_Info['TK_XLIBSW'].strip.empty?)

- [redacted]
  `#use_X = true && with_config("X11", ! is_win32?)`
  `use_X = with_config("X11", true)`

  else
  @@ -1216,21 +1290,28 @@
  use_X = with_config("X11", !(is_win32? || TkLib_Config["tcltk-framework"]))
  end

- if TkConfig_Info['TK_XINCLUDES'] &&
- [redacted]
- $INCFLAGS << " " << TkConfig_Info['TK_XINCLUDES'].strip
- end
  +
  if use_X
- puts("Use X11 libraries.")
- puts("Use X11 libraries (or use TK_XINCLUDES/TK_XLIBSW information on tkConfig.sh).")
  x11_idir, x11_ldir = dir_config("X11")
  x11_ldir2 = with_config("X11-lib")
  unless find_X11(x11_ldir2, x11_ldir)
- [redacted]
- [redacted]
- [redacted]
- [redacted]
- [redacted]
- [redacted]
- [redacted]
- [redacted]
- [redacted]
- [redacted]

- [redacted]

  end
  end

- if TkConfig_Info['TK_XINCLUDES'] &&

- [redacted]

- $INCFLAGS << " " << TkConfig_Info['TK_XINCLUDES'].strip

- **end**

  use_X
  end

@@ -1272,7 +1353,7 @@
else
puts("Warning: '#{TclConfig_Info['config_file_path']}' may not be a tclConfig file.")
end

- tclConfig = false
- #tclConfig = false end end end @@ -1280,7 +1361,7 @@ if tcl_enable_thread == nil && !TclConfig_Info['config_file_path'] # tcl-thread is unknown and tclConfig is unavailable begin
- try_run_available = try_run("int main() { exit(0); }")
- try_run("int main() { exit(0); }") rescue Exception # cannot try_run. Is CROSS-COMPILE environment? puts(%Q\ @@ -1439,15 +1520,61 @@

#---
+TclConfig_Info = {}
+TkConfig_Info = {}
+
+# use Tcl/Tk build dir? (has highest priority)
+TkLib_Config["tcl-build-dir"] = with_config("tcl-build-dir")
+TkLib_Config["tk-build-dir"] = with_config("tk-build-dir")
+if TkLib_Config["tcl-build-dir"]

- puts("use Tcl build (pre-install) dir \"#{TkLib_Config["tcl-build-dir"]}\"")
- TkLib_Config["tcl-build-dir"] = File.expand_path(TkLib_Config["tcl-build-dir"])
- base = File.dirname(TkLib_Config["tcl-build-dir"])
- $INCFLAGS << " -I#{File.join(base, "generic").quote} -I#{TkLib_Config["tcl-build-dir"].quote}"
- $LIBPATH |= [TkLib_Config["tcl-build-dir"]] +end +if TkLib_Config["tk-build-dir"]
- puts("use Tk build (pre-install) dir \"#{TkLib_Config["tk-build-dir"]}\"")
- TkLib_Config["tk-build-dir"] = File.expand_path(TkLib_Config["tk-build-dir"])
- base = File.dirname(TkLib_Config["tk-build-dir"])
- $INCFLAGS << " -I#{File.join(base, "generic").quote} -I#{TkLib_Config["tk-build-dir"].quote}"
- $LIBPATH |= [TkLib_Config["tk-build-dir"]] +end + # check requirement of Tcl/tk version tcltk_version = with_config("tcltkversion") -tclver, tkver =
- TkLib_Config["tcltkversion"] = check_tcltk_version(tcltk_version) -puts("Specified Tcl/Tk version is #{[tclver, tkver].inspect}") if tclver&&tkver +TkLib_Config["tcltkversion"] = check_tcltk_version(tcltk_version)

+if TkLib_Config["tcl-build-dir"]

- if (cfgfile = with_config("tclConfig-file", Dir.glob(File.join(TkLib_Config["tcl-build-dir"], "tclConfig*.sh"), File::FNM_CASEFOLD)[0]))
- TclConfig_Info['config_file_path'] = cfgfile
- TkLib_Config["tclConfig_info"] = cfginfo = parse_tclConfig(cfgfile)
- if tclver = TkLib_Config["tcltkversion"][0]
- TkLib_Config["tcltkversion"][0].sub!(/\d(.?)\d/, "#{cfginfo['TCL_MAJOR_VERSION']}\1#{cfginfo['TCL_MINOR_VERSION']}")
- else
- TkLib_Config["tcltkversion"][0] = "#{cfginfo['TCL_MAJOR_VERSION']}.#{cfginfo['TCL_MINOR_VERSION']}"
- end
- end +end +if TkLib_Config["tk-build-dir"]
- if (cfgfile = with_config("tkConfig-file", Dir.glob(File.join(TkLib_Config["tk-build-dir"], "tkConfig*.sh"), File::FNM_CASEFOLD)[0]))
- TkConfig_Info['config_file_path'] = cfgfile
- TkLib_Config["tkConfig_info"] = cfginfo = parse_tclConfig(cfgfile)
- if TkLib_Config["tcltkversion"][1]
- TkLib_Config["tcltkversion"][1].sub!(/\d(.?)\d/, "#{cfginfo['TK_MAJOR_VERSION']}\1#{cfginfo['TK_MINOR_VERSION']}")
- else
- TkLib_Config["tcltkversion"][1] = "#{cfginfo['TK_MAJOR_VERSION']}.#{cfginfo['TK_MINOR_VERSION']}"
- end
- end +end + +tclver, tkver = TkLib_Config["tcltkversion"] +puts("Specified Tcl/Tk version is #{[tclver, tkver].inspect}") if tclver||tkver + # use ActiveTcl ? #if activeTcl = with_config("ActiveTcl") -if activeTcl = with_config("ActiveTcl", true) +#if activeTcl = with_config("ActiveTcl", true) +if activeTcl = with_config("ActiveTcl", !(TkLib_Config["tcl-build-dir"] && TkLib_Config["tk-build-dir"])) puts("Use ActiveTcl libraries (if available).") unless activeTcl.kind_of? String # set default ActiveTcl path @@ -1490,8 +1617,18 @@ end

```
# directory configuration of Tcl/Tk libraries
-tk_idir,  tk_ldir  = dir_config("tk")
-tcl_idir, tcl_ldir = dir_config("tcl")
+if TkLib_Config["tcl-build-dir"]
```

- tcl_idir = File.join(File.dirname(TkLib_Config["tcl-build-dir"]),"generic")
- tcl_ldir = TkLib_Config["tcl-build-dir"] +else
- tcl_idir, tcl_ldir = dir_config("tcl") +end +if TkLib_Config["tk-build-dir"]
- tk_idir = File.join(File.dirname(TkLib_Config["tk-build-dir"]),"generic")
- tk_ldir = TkLib_Config["tk-build-dir"] +else
- tk_idir, tk_ldir = dir_config("tk") +end

```
tcl_idir = tk_idir unless tcl_idir
tcl_ldir = tk_ldir unless tcl_ldir
@@ -1500,14 +1637,21 @@

# get tclConfig.sh/tkConfig.sh
TkLib_Config["tcltk-NG-path"] = []
-tclcfg, tkcfg = get_tclConfig(with_config("tclConfig-file", true),
```

- with_config("tkConfig-file", true),
- with_config("tclConfig-dir", tcl_ldir || true),
- with_config("tkConfig-dir", tk_ldir || true)) -TclConfig_Info = TkLib_Config["tclConfig_info"] || {} -TkConfig_Info = TkLib_Config["tkConfig_info"] || {} -TclConfig_Info['config_file_path'] = tclcfg -TkConfig_Info['config_file_path'] = tkcfg +tclcfg, tkcfg =
- get_tclConfig(
- TclConfig_Info['config_file_path'] || with_config("tclConfig-file", true),
- TkConfig_Info['config_file_path'] || with_config("tkConfig-file", true),
- (TclConfig_Info['config_file_path'])?
- File.dirname(TclConfig_Info['config_file_path']) :
- with_config("tclConfig-dir", tcl_ldir || true),
- (TkConfig_Info['config_file_path'])?
- File.dirname(TkConfig_Info['config_file_path']) :
- with_config("tkConfig-dir", tk_ldir || true)
-) +TclConfig_Info.merge!(TkLib_Config["tclConfig_info"]) +TkConfig_Info.merge!(TkLib_Config["tkConfig_info"]) +TclConfig_Info['config_file_path'] ||= tclcfg +TkConfig_Info['config_file_path'] ||= tkcfg

```
TclConfig_Info['TCL_INCLUDE_SPEC'] = "-I#{tcl_idir.quote}" if tcl_idir
TkConfig_Info['TK_INCLUDE_SPEC']   = "-I#{tk_idir.quote}"  if tk_idir
@@ -1515,9 +1659,26 @@
tk_cfg_dir  = File.dirname(TkConfig_Info['config_file_path'])  rescue nil
tcl_cfg_dir = File.dirname(TclConfig_Info['config_file_path']) rescue nil

-tk_ldir_list  = [tk_ldir,  tk_cfg_dir]
-tcl_ldir_list = [tcl_ldir, tcl_cfg_dir]
+tk_ldir_list  = [tk_ldir,  tk_cfg_dir].uniq
+tcl_ldir_list = [tcl_ldir, tcl_cfg_dir].uniq

+if TkConfig_Info['config_file_path']
```

- if TkLib_Config["tk-build-dir"]
- spec_dir = get_libpath(TkConfig_Info['TK_LIB_FLAG'], TkConfig_Info['TK_BUILD_LIB_SPEC'])
- else
- spec_dir = get_libpath(TkConfig_Info['TK_LIB_FLAG'], TkConfig_Info['TK_LIB_SPEC'])
- end
- tk_ldir_list << spec_dir if File.directory?(spec_dir) +end +if TclConfig_Info['config_file_path']
- if TkLib_Config["tcl-build-dir"]
- spec_dir = get_libpath(TclConfig_Info['TCL_LIB_FLAG'], TclConfig_Info['TCL_BUILD_LIB_SPEC'])
- else
- spec_dir = get_libpath(TclConfig_Info['TCL_LIB_FLAG'], TclConfig_Info['TCL_LIB_SPEC'])
- end
- tcl_ldir_list << spec_dir if File.directory?(spec_dir) +end + # check tk_shlib_search_path check_shlib_search_path(with_config('tk-shlib-search-path'))

```
@@ -1529,22 +1690,49 @@
# MacOS X Frameworks?
if TkLib_Config["tcltk-framework"]
puts("Use MacOS X Frameworks.")
```

- $LDFLAGS << " -L#{TkLib_Config["tcl-build-dir"].quote}" if TkLib_Config["tcl-build-dir"] if tcl_cfg_dir $INCFLAGS << ' ' << TclConfig_Info['TCL_INCLUDE_SPEC'] $LDFLAGS << ' ' << TclConfig_Info['TCL_LIBS']   if stubs
- $LDFLAGS << ' ' << TclConfig_Info['TCL_STUB_LIB_SPEC']
- if TkLib_Config["tcl-build-dir"] &&
- TclConfig_Info['TCL_BUILD_STUB_LIB_SPEC'] &&
- !TclConfig_Info['TCL_BUILD_STUB_LIB_SPEC'].strip.empty?
- $LDFLAGS << ' ' << TclConfig_Info['TCL_BUILD_STUB_LIB_SPEC']

- else
- $LDFLAGS << ' ' << TclConfig_Info['TCL_STUB_LIB_SPEC']
- end else
- $LDFLAGS << ' ' << TclConfig_Info['TCL_LIB_SPEC']
- if TkLib_Config["tcl-build-dir"] &&
- TclConfig_Info['TCL_BUILD_LIB_SPEC'] &&
- !TclConfig_Info['TCL_BUILD_LIB_SPEC'].strip.empty?
- $LDFLAGS << ' ' << TclConfig_Info['TCL_BUILD_LIB_SPEC']
- else
- $LDFLAGS << ' ' << TclConfig_Info['TCL_LIB_SPEC']
- end end end +
- $LDFLAGS << " -L#{TkLib_Config["tk-build-dir"].quote}" if TkLib_Config["tk-build-dir"] if tk_cfg_dir $INCFLAGS << ' ' << TkConfig_Info['TK_INCLUDE_SPEC'] $LDFLAGS << ' ' << TkConfig_Info['TK_LIBS'] if stubs
- $LDFLAGS << ' ' << TkConfig_Info['TK_STUB_LIB_SPEC']
- if TkLib_Config["tk-build-dir"] &&
- TclConfig_Info['TK_BUILD_STUB_LIB_SPEC'] &&
- !TclConfig_Info['TK_BUILD_STUB_LIB_SPEC'].strip.empty?
- $LDFLAGS << ' ' << TkConfig_Info['TK_BUILD_STUB_LIB_SPEC']
- else
- $LDFLAGS << ' ' << TkConfig_Info['TK_STUB_LIB_SPEC']
- end else
- $LDFLAGS << ' ' << TkConfig_Info['TK_LIB_SPEC']
- if TkLib_Config["tk-build-dir"] &&
- TclConfig_Info['TK_BUILD_LIB_SPEC'] &&
- !TclConfig_Info['TK_BUILD_LIB_SPEC'].strip.empty?
- $LDFLAGS << ' ' << TkConfig_Info['TK_BUILD_LIB_SPEC']
- else
- $LDFLAGS << ' ' << TkConfig_Info['TK_LIB_SPEC']
- end end end setup_for_macosx_framework(tclver, tkver) if tcl_cfg_dir && tk_cfg_dir

--
Hidetoshi NAGAI (nagai@ai.kyutech.ac.jp)
Department of Artificial Intelligence, Kyushu Institute of Technology

=end

#3 - 12/27/2010 09:04 PM - luislavena (Luis Lavena)

=begin
On Sun, Dec 26, 2010 at 10:05 PM, Hidetoshi NAGAI
nagai@ai.kyutech.ac.jp wrote:

> Hi,

Hello,

> > Bug #4141: Tk extension is not accepting any type of parameter combination
> > http://redmine.ruby-lang.org/issues/show/4141

> I'm very sorry. My reply is too late for a new RubyInstaller.

No need to apologize. I appreciate you took the time to answer.

> Could you try the following patch (and new options)?

I've tried the patch you provided, and this is what I got in the console:

```
compiling syslog
compiling tk
check functions.....
check struct members..
Use ActiveTcl libraries (if available).
Search tclConfig.sh and tkConfig.sh.....
Fail to find [tclConfig.sh, tkConfig.sh]
c:/Users/Luis/Projects/oss/ruby/ext/tk/extconf.rb:1651:in merge!':
can't convert nil into Hash (TypeError)
from c:/Users/Luis/Projects/oss/ruby/ext/tk/extconf.rb:1651:in
'
from ../../../../ruby/ext/extmk.rb:157:in load'
from ../../../../ruby/ext/extmk.rb:157:inextmake'
from ../../../../ruby/ext/extmk.rb:444:in block in <main>'
```

from ../../../../ruby/ext/extmk.rb:440:ineach'
from ../../../../ruby/ext/extmk.rb:440:in `'
make: *** [mkmain.sh] Error 1

===

Some additional information:

1) tcl and tk were build statically (configure --enable-static --disable-shared)

2) sandbox/tcl contains:

bin: tclsh85s.exe
lib: libtcl85s.a, libtclstub85s.a, tclConfig.sh (along some tcl8.5 directories)

3) sandbox/tk contains:

bin: wish85s.exe
lib: libtk85s.a, libtkstub85s.a, tkConfig.sh

Please note the following:

tcl and tk are compiled and installed in different folders, as tk
requires tcl, first I compile and 'activate it' for the build process
work.

All this is automated with RubyInstaller recipes under the tcl-tk branch:

https://github.com/oneclick/rubyinstaller/tree/tcl-tk

This can be easily tested and hacked, using a Ruby 1.8.7 interpreter
as baseruby:

git clone git://github.com/oneclick/rubyinstaller.git
cd rubyinstaller
git checkout tcl-tk
rake ruby19

(That will build using latest 1.9.2-p136 codebase)
If you want to build against a SVN checkout or git workcopy, you can
use LOCAL and the PATH:

rake ruby19 LOCAL=C:\Path\To\Ruby

Thank you for your time looking into this issue.

Regards,
--
Luis Lavena
AREA 17
-
Perfection in design is achieved not when there is nothing more to add,
but rather when there is nothing more to take away.
Antoine de Saint-Exupéry

=end

#4 - 12/28/2010 10:34 AM - nagai (Hidetoshi Nagai)

=begin
From: Luis Lavena luislavena@gmail.com
Subject: [ruby-core:33922] Re: [Ruby 1.9-Bug#4141][Open] Tk extension is not accepting any type of parameter combination
Date: Mon, 27 Dec 2010 21:04:29 +0900
Message-ID: AANLkTikngfu6Ty0o9BqCb3ab3pdn5ETR+CVF8YKZHwee@mail.gmail.com

> I've tried the patch you provided, and this is what I got in the console:
> (snip)
> c:/Users/Luis/Projects/oss/ruby/ext/tk/extconf.rb:1651:in `merge!':
> can't convert nil into Hash (TypeError)

Oh, I'm sorry. Please replace line 1651 - 1652 from

TclConfig_Info.merge!(TkLib_Config["tclConfig_info"])

TkConfig_Info.merge!(TkLib_Config["tkConfig_info"])

---

to

TclConfig_Info.merge!(TkLib_Config["tclConfig_info"]) if TkLib_Config["tclConfig_info"]
TkConfig_Info.merge!(TkLib_Config["tkConfig_info"]) if TkLib_Config["tkConfig_info"]

> tcl and tk are compiled and installed in different folders, as tk

I'm not enough to assume such case.
Could you tell me configure options which you gave for tcltklib?
And, if you can, please show me tclConfig.sh and tkConfig.sh.
--
Hidetoshi NAGAI (nagai@ai.kyutech.ac.jp)
Department of Artificial Intelligence, Kyushu Institute of Technology

=end

#5 - 12/28/2010 11:50 PM - luislavena (Luis Lavena)

=begin
On Mon, Dec 27, 2010 at 10:34 PM, Hidetoshi NAGAI
nagai@ai.kyutech.ac.jp wrote:

> ## Oh, I'm sorry. Please replace line 1651 - 1652 from
>
> TclConfig_Info.merge!(TkLib_Config["tclConfig_info"])
>
> ## TkConfig_Info.merge!(TkLib_Config["tkConfig_info"])
>
> ## to
>
> TclConfig_Info.merge!(TkLib_Config["tclConfig_info"]) if TkLib_Config["tclConfig_info"]
>
> ## TkConfig_Info.merge!(TkLib_Config["tkConfig_info"]) if TkLib_Config["tkConfig_info"]

Replaced, output:

compiling tk
check functions.....
check struct members..
Use ActiveTcl libraries (if available).
Search tclConfig.sh and tkConfig.sh.....
Fail to find [tclConfig.sh, tkConfig.sh]
Search Tcl library
Warning:: cannot find Tcl library. tcltklib will not be compiled
(tcltklib is disabled on your Ruby
== Ruby/Tk will not work). Please check configure options.
Can't find proper Tcl/Tk libraries. So, can't make tcltklib.so which
is required by Ruby/Tk.
compiling tk/tkutil

> > tcl and tk are compiled and installed in different folders, as tk
>
> I'm not enough to assume such case.
> Could you tell me configure options which you gave for tcltklib?

I gave no configure options to tcltklib extension, I'm compiling
tcltklib as part of Ruby own configure/make process.

> And, if you can, please show me tclConfig.sh and tkConfig.sh.

Find attached both files.

tcl was configure with:

--disable-shared --enable-static

tk was configure with

--disable-shared --enable-static --with-tcl=/path/to/tcl/build/directory

my ENV:

```
echo %PATH%
C:\Users\Luis\Projects\oss\oci\rubyinstaller\sandbox\tk\bin;C:\Users\Luis\Projects\oss\oci\rubyinstaller\sandbox\tcl\bin;C:\Users\Luis\Projects\oss
\oci\rubyinstaller\sandbox\pdcurses\bin;C:\Users\Luis\Projects\oss\oci\rubyinstaller\sandbox\openssl\bin;C:\Users\Luis\Projects\oss\oci\rubyinstal
ler\sandbox\zlib\bin;C:\Users\Luis\Projects\oss\oci\rubyinstaller\sandbox\iconv\bin;C:\Users\Luis\Projects\oss\oci\rubyinstaller\sandbox\gdbm\bin;
\usr\bin;\usr\mingw\bin;C:\Users\Luis.gem\ruby\x86-mingw32\1.8\bin;C:\Users\Luis\Tools\Ruby\ruby-1.8.7-p330-i386-mingw32\bin;C:\Windows\s
ystem32;C:\Windows;C:\Windows\System32\Wbem;C:\Windows\System32\WindowsPowerShell\v1.0\;C:\Program
Files
(x86)\GNU\GnuPG\pub;C:\Users\Luis\Tools\bin;C:\Users\Luis\Tools\Git\cmd;C:\Users\Luis\Tools\Svn\bin;C:\Users\Luis\Tools\Hg;C:\Users\Luis\T
ools\Unix\bin;C:\Users\Luis\Tools\Vim;C:\Program
Files (x86)\Java\jdk1.6.0_18\bin

echo %CPATH%
```

C:\Users\Luis\Projects\oss\oci\rubyinstaller\sandbox\tk\include;C:\Users\Luis\Projects\oss\oci\rubyinstaller\sandbox\tcl\include;C:\Users\Luis\Projects\
oss\oci\rubyinstaller\sandbox\libyaml\include;C:\Users\Luis\Projects\oss\oci\rubyinstaller\sandbox\pdcurses\include;C:\Users\Luis\Projects\oss\oci\rub
yinstaller\sandbox\openssl\include;C:\Users\Luis\Projects\oss\oci\rubyinstaller\sandbox\zlib\include;C:\Users\Luis\Projects\oss\oci\rubyinstaller\sandb
ox\iconv\include;C:\Users\Luis\Projects\oss\oci\rubyinstaller\sandbox\gdbm\include;C:\Users\Luis\Projects\oss\oci\rubyinstaller\sandbox\libffi\include;

```
echo %LIBRARY_PATH%
```

C:\Users\Luis\Projects\oss\oci\rubyinstaller\sandbox\tk\lib;C:\Users\Luis\Projects\oss\oci\rubyinstaller\sandbox\tcl\lib;C:\Users\Luis\Projects\oss\oci\ru
byinstaller\sandbox\libyaml\lib;C:\Users\Luis\Projects\oss\oci\rubyinstaller\sandbox\pdcurses\lib;C:\Users\Luis\Projects\oss\oci\rubyinstaller\sandbox\o
penssl\lib;C:\Users\Luis\Projects\oss\oci\rubyinstaller\sandbox\zlib\lib;C:\Users\Luis\Projects\oss\oci\rubyinstaller\sandbox\iconv\lib;C:\Users\Luis\Proj
ects\oss\oci\rubyinstaller\sandbox\gdbm\lib;C:\Users\Luis\Projects\oss\oci\rubyinstaller\sandbox\libffi\lib;

These correspond to PATH, INCLUDE and LIB definitions of Visual
Studio, but for GCC

These parameters worked perfectly for extensions like fiddle and psych
that depend on libffi and libyaml respectively.

Thank you again for your time looking into this issue.
--
Luis Lavena
AREA 17
-
Perfection in design is achieved not when there is nothing more to add,
but rather when there is nothing more to take away.
Antoine de Saint-Exupéry

Attachment: tclConfig.sh
Attachment: tkConfig.sh
=end

#6 - 03/26/2011 10:25 PM - shyouhei (Shyouhei Urabe)

- Status changed from Open to Assigned

#7 - 05/27/2011 08:36 AM - nagai (Hidetoshi Nagai)

- Status changed from Assigned to Closed

- % Done changed from 0 to 100

This issue was solved with changeset r31742.
Luis, thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.
May Ruby be with you.

- ext/tk/lib/tk.rb: hang-up at exit before calling Tk.mainloop.
- ext/tk/lib/tk/extconf.rb: cannot make on MinGW [Ruby 1.9 - Bug #4141].

#8 - 05/31/2011 01:26 AM - luislavena (Luis Lavena)

Hidetoshi Nagai, thank you, It is now possible to compile Tcl/Tk with MinGW without issues.

I've open a backport request so this get merged back into 1.9.2:

http://redmine.ruby-lang.org/issues/4802

On a sidenote I'm going to check if is possible build 1.8.7 out of the repository too, as we aim to provide the same binaries for it.

Thank you!