

Porting a large scale enterprise application from Tcl/Tk 8.4 to 8.5 - A Case Study

Prashant Thakre, Tushar Gupta
{prashant_thakre,
tushar_gupta}@mentor.com

1.0 Abstract

This paper describes the challenges involved in porting a large scale enterprise application from *Tcl/Tk* 8.4 to *Tcl/Tk* 8.5, and lists down all issues encountered during the migration. The migration to *Tcl/Tk* 8.5 required adjustments in application code, necessitated due to changes in interface, or as a result of unexpected functional behaviour. On some occasions it was required to patch the *Tcl/Tk* 8.5 distribution with fixes which are now expected to be included into standard *Tcl/Tk* release.

The case-study describes in detail various external and internal dependencies requiring resolution and covers performance related benchmarks/issues. Also, changes in various packages (core and external) resulting in regressions are discussed. The objective of the case-study is to provide other developers an insight into typical issues faced, and suggest solutions wherever available.

2.0 Glossary

Description of terms used in the paper:

[incr Tcl] - is an OO system for *Tcl*, and *[incr Tcl]* provides a similar object model, including multiple inheritance, and public and private classes and variables.

[incr Tk] - Megawidget framework based upon *[incr Tcl]*.

Iwidgets - is an object-oriented mega-widget set which extends *Tcl/Tk* and is based on *[incr Tcl]* and *[incr Tk]*.

[mti Widgets] - Collection of mega-widget, additions and enhancements to *Iwidgets*. This megawidget requires TIP #125 (*wm toplevel*)

3.0 Motivation

The application Veloce GUI, is from the Mentor Emulation Division (MED) at Mentor Graphics Corp. The Veloce GUI is used for emulation compiles, emulation runtime control, and project management and most importantly for design debug. Typically customers use emulation for some of the largest and the most complex System-on-a-chip (SOC) designs, and the Veloce GUI is tuned to handle large design databases as well as huge

waveform data. Each such set of generated debug data having information about design hierarchy; waveform and design connectivity is called a dataset. The application uses object oriented programming provided by *[incr Tcl]* package, to support multiple simultaneous dataset debug, potentially belonging to different design databases.

The code base for Veloce GUI primarily consists of *Tcl/Tk* and *C/C++*. Veloce GUI in general uses various *Tcl/Tk* packages and required custom build of *Tk* (8.4) necessitated by usage of *[mti Widgets]* (TIP 125: Converting between *frame* and *oplevel* windows). Over time, we have observed various *Tcl/Tk* related issues in 8.4 have been fixed in 8.5. Also, since 8.5 release of *Tcl/Tk* has incorporated TIP #125 (using new subcommands *wm manage* and *wm forget*), has various new features and is actively maintained, migrating from 8.4 was a necessity.

As with any migration to a major release (8.5 in this case), it's bound to throw challenges during build, implementation and testing stage. The paper describes the changes required to build process along with changes to code base. Changes to code are result of core *Tcl/Tk* changes or due to changes in the packages like *[incr Tk]*. Also, the paper covers areas where change in behaviour of

Tcl/Tk commands and rendering is observed as compared to 8.4 release. Performance benchmark data is also reported in a separate section.

The paper concludes by sharing the findings and by providing our recommendations on how to catch issues during early stages of migration.

4.0 Changes required

This section describes in detail various external and internal dependencies requiring resolution. Also, change in various packages (core and external) resulting in regressions are discussed.

- *tk/library/button.tcl*:

Package:

This is part of standard distribution.

Issue:

Application uses *check buton* and *radio button*, and global variable linked to the *buttons* are specified using *itcl::scope*. This caused *Tcl* exception at runtime.

Solution:

Patch the *tcl* distribution with available fix for bug # 87409. This fix uses *oplevel set* instead of *set ::* to work with *itcl*.

- *[list]*-Quoting of the '#' character:

Package:

This is part of standard distribution.

Issue:

Application dumps the tree contents during automated testing to check for new regressions if any. Starting with *Tcl/Tk* 8.5, name of tree nodes starting with '#' are quoted by default.

Solution:

Application could either modify the routine dumping the contents of the tree if possible or update the gold files.

- *[info level]*

Package:

This is part of standard distribution.

Issue:

Change in information returned by *[info level]* command. Application in question disables *cd* command for a user. However, various *tk* routines like *tk_chooseDirectory*, *tk_getOpenFile*, and *tk_getSaveFile* internally call *cd*. Until *Tcl/Tk* 8.4 application would call

[info level] with level argument as -1 in order to verify that the caller is *tk::dialog::file*. However, this no longer returns the required information in *Tcl/Tk* 8.5.

Solution:

Level argument passed to *[info level]* should be -2 instead of -1.

- *[pwd]*

Package:

This is part of standard distribution.

Issue:

In *Tcl/Tk* 8.4 if a present working directory gets deleted a call to *pwd* throws exception. However, starting with *Tcl/Tk* 8.5 *pwd* returns an empty string and no longer throws exception.

Solution:

Add check for empty string on return value of *pwd*.

- *[tk_getOpenFile]*

Package:

This is part of standard distribution.

Issue:

In *Tcl/Tk* 8.4 multiple file and directory

selection using ctrl
throws an exception.

Solution:

This has been fixed
in *Tcl/Tk* 8.5. No change
is required in application
code.

- *[incr Tk]: 3.4*

Package:

This package is
included with *[incr Tcl]*
distribution.

Issue:

[incr Tk] supplies
three base classes that
reside in the *itk*
namespace:

1. *itk::Archetype*
2. *itk::Widget*
3. *itk::Toplevel*

Application uses
[incr Tk] 3.2 with *Tcl/Tk*
8.4 and “-menu” option is
defined in base class
itk::Archetype. However,
with *[incr Tk]* 3.4 this
option is now defined only
for toplevel windows i.e.
itk::Toplevel.

Solution:

Widgets derived
from *itk::Archetype* will
no longer have “-menu”
option defined and will
require necessary
changes i.e. use
itk::Toplevel instead.

- *[TIP 125]: wm manage and
wm forget*

Package:

This is part of
standard distribution.

Issue:

Application allows
docking and undocking of
child windows. This was
implemented in *Tcl/Tk* 8.4
using patch for *[wm
toplevel]*.

However, this TIP
was incorporated into
standard distribution with
new *wm* subcommands
manage and *forget*.

Solution:

A procedure
wm_toplevel was defined
to handle all calls to *[wm
toplevel]*. Following is the
definition of the
procedure.

```
proc wm_toplevel {win
{bool {}}} {
    if {[llength [info level
0]] == 2} {
        return [expr {[wininfo
manager $win] eq "wm"}]
    } else {
        if {$bool} {
            wm manage $win
        } else {
            wm forget $win
        }
    }
}
```

- *BWidget: 1.9.2*

Package:

This package is part
of *Tcllib* and is
available at
<http://tcllib.sf.net/>.

Issue:

Application was using *BWidget* 1.6 and many interfaces level changes have occurred since 1.6. The command *Widget::create* does the renaming of the widgets to `$path:cmd` and creates the proc to redirect the widget commands.

Solution:

Change usage of *BWlabel::create* to *Label::create*. Also, *BWidget* 1.9.1 release was a development snapshot from the Tile enabled version. This release has some compatibility issues and users are encouraged to upgrade to 1.9.2.

- *Incorrect gridding/packing of elements*

Package:

This is part of standard distribution.

Issue:

This relates to incorrect display in *Tcl/Tk* 8.5 as compared to 8.4. For example following would display the middle element correctly in 8.4 but not in 8.5 where `button_frame` contains add and remove buttons.

```

pack $left
$button_frame $right -
expand yes -fill both -
side left =padx -pady 5

```

Solution:

Required changes to application code on case by case basis. Above mentioned code had to be replaced with

```

pack $left -expand
yes -fill both -side left
-padx 5 -pady 5
pack $button_frame -
dise left -padx 5 -pady 5
pack $right -expand
yes -fill both -side left
-padx 5 -pady 5

```

- *Drag and drop*

Package:

This is part of tkdnd distribution.

Issue:

Application uses tkdnd for supporting drag and drop. Incorrect text was dropped in some cases.

Solution:

A close inspection revealed that unnecessary quoting was causing an issue i.e. "%T" instead of %T in the bind script.

- *Tk crash while trying to dock/undock a window repeatedly.*

Package:

This is part of standard distribution.

Issue:

Repeated docking and undocking of child windows would result in a Tk crash. On closer inspection it seemed that this was due to change in the way “-menu” option was configured. Most of these child windows were derived from *itk::Archetype*. Also, due to changes to “menu” in [incr Tk] 3.4, “menu” was configured after first undock operation. This resulted in abnormal termination during docking process.

Solution:

Configure “menu” option to empty during all docking operations and configure it back to correct menu during undocking.

- *Change in font size.*

Package:

This is part of standard distribution.

Issue:

Veloce GUI uses custom widget and it was observed that default font size has changed across release i.e. in *Tcl/Tk* 8.5. This was cause of many

regressions in our test suites.

Solution:

Preferences were modified in the application to select a larger font to fix regression issues found during automated testing.

5.0 Performance

Moving to newer release of *Tcl/Tk* 8.5 has caused some unexpected regressions in terms of performance. In a particular computation intensive task (happens on C/C++ side) it was observed that the performance penalty can be as high as 30%. Application is busy with the computation and does not take user inputs i.e. no change in display of the application.

Solution:

Currently we are trying to profile the application in order to narrow down the exact cause of the problem. However, the same application without any modification when linked to *Tcl/Tk* 8.4 produces much better performance results. We have tried to build the application by linking it to single threaded version of *Tcl/Tk*

libraries. However, this hasn't yielded desired results.

6.0 Recommendations

1. A regression setup helps to track initial set of issues.
2. Change log of all external and internal packages required by an application can help narrow down code level changes.
3. Performance benchmarking is required for critical portion of the code.

7.0 Bibliography

1. Tcl wiki, <http://wiki.tcl.tk>
2. *[incr Widgets]* An Object Oriented Mega-Widget Set, Mark L. Ulferts, http://incrtcl.sourceforge.net/incr_widgets/paper/paper.html