

Migration of tDOM to Tcl 9

Rolf Ade

rolf@pointsman.de

Nothing special with tDOM

- Almost all binary extensions need code changes
- tDOM works on text (XML and JSON are text, not binary data)
- tDOM uses external libraries (expat, gumbo)
- tDOM has “own” C code (DOM builder, XPath engine etc.), not only library interface
- tDOM has a (small) scripted part

Core changes made migration easier

- TIP 660
Without that, migration would require an even much more careful code review.
- TIP 661
With that the compiler helps you to spot the API functions calls which changed with warnings as you would expect
- I assume that your binary extension should build and work for Tcl 9 and for Tcl 8

There is help out there

- The core repository wiki:
<https://core.tcl-lang.org/tcl/wiki?name=Migrating+C+extensions+to+Tcl+9>
- The wiki:
<https://wiki.tcl-lang.org/page/Porting+extensions+to+Tcl+9>
<https://wiki.tcl-lang.org/page/Tcl+9+functions+using+Tcl%5FSize>
- You should read at least the core repository wiki article

Things you probably need to do

- You have to ensure that a `Tcl_Size` type (in Tcl 9 technically a signed 64-bit integer) is defined as `int` when building with Tcl 8.6
 - Modern TEA claims to ensure this for you
 - Or have that in a home grown `#define` you probably may need:

```
#if TCL_MAJOR_VERSION < 9
#  define Tcl_Size int
#  define TCL_SIZE_MODIFIER ""
#  define Tcl_GetSizeIntFromObj Tcl_GetIntFromObj
...
#endif (perhaps with #ifndef for safety)
```
- (The proposed `#ifndef TCL_SIZE_MAX ...` seems to be less clear to me)

Things you probably need to do II

- Consider system-wide installation of Tcl 8.6 and Tcl 9 builds of the same extension alongside
- TEA provides
 - Tcl 8.6 build results in libtdom0.9.4.so
 - Tcl 9.0 build results in libtcl9tdom0.9.4.so
- Check your pkgIndex.tcl template/creation

Things you probably need to do III

- (Try to) build your untouched extension code with Tcl 9
- This will probably produce a lot of warnings. With the tdom 0.9.3 release code: 65 warnings
- Now visit any place and do the necessary variable type changes, typically from `int` to `Tcl_Size`
- Which sometimes means, function definitions and `.h` files need to be changed
- As a consequence, the callers of the changed functions have to be changed
- The compiler claims what you've been missing
- For the type warnings in `(f|s)printf()` calls look if `TCL_SIZE_MODIFIER` helps

The not so automated tasks

- The third-party library functions used by your extension may have limits
- tDOM uses several times this expat function:

```
XML_Parse(XML_Parser parser, const char *s, int len, int isFinal)
```

- This function digests any Tcl 8.6 string in one call
- For Tcl 9 the code loops over the now potentially much longer string and parse in chunk
- Sometimes you'll need guards: gumbo parse in one go and up to (“only”) 4 GByte

The not so automated tasks II

- Things you left out because they made not much sense with Tcl 8.6
- For tDOM: The JSON parser had to be enhanced
- The JSON syntax allows to write a non-BMP character as `\u` escaped UTF-16 surrogate pairs:

JSON: `"\uD834\uDD1E"` = Tcl 9: `\U1D11E`

- `tclDOM_utfToUniChar()`

Pitfalls

- Look at this:

```
int a;  
Tcl_Size b;  
a = strlen (str);  
a = b;
```
- Look out for this if your C code parses Tcl_Obj strings
- Neither gcc nor clang warns you about such assignment even with -Wall (the TEA warning options default)
- The VC compiler does it much better (at least with the flags provided by Ashoks extension build framework for native VC builds)

Pitfalls II

- Even a good and large test suite will not spot such problems for you
- Tcl_Obj with string representation longer than 2 GByte were not possible with Tcl 8.6 and that problems show up only with strings > 2GBytes
- With gcc and clang add (temporally) -Wconversion to -Wall
- That will be noisy. An additional -Wno-sign-conversion will silence it a bit
- But spots the same as the VC compiler (in between of the probable noise)

Tcl 9 still is a moving target

- Two weeks ago TIP 699 affected tDOM
- (Though tDOM at this place (checking for BOM) was one of the rare examples that the old `-encoding binary` was exactly right, now `-translation` must be explicitly set back for text format)
- Help is also on its way: Ashoks migration tool for Tcl 9 helps with the script parts of the extension:
<https://github.com/apnadkarni/tcl9-migrate/releases>

Finis.

Questions?