

# TransSkel for TML Modula-2 and MPW

The TransSkel.doc file on this disk is the original TransSkel documentation for the LightSpeed™ C version written by Paul DuBois. This TML Modula-2 version was written by:

Dennis Cohen  
215 N. Kenwood St., #102  
Glendale, CA 91206

I would like to thank Paul DuBois for an excellently written and well-designed set of tools which were quite easy to port to TML Modula-2. I would also like to thank Tom Leonard, Carl Nelson, and especially Bob Campbell for providing me with a fine compiler and assistance when I ran afoul of my own carelessness or the rare bug in the pre-release version of the compiler with which I was working.

I apologize for not providing a complete conversion of the documentation for the Modula-2 version; however, Paul DuBois wrote a superb manual describing the package and although it is directed to the C programmer, it should be easily understandable to a Modula-2 programmer who has a listing of TransSkel.def available (and many who don't). All of the sources are in 10-pt Courier font with a tab-setting of 2, as these are my personal preferences.

## Differences between this version and the original C version

The most obvious difference is the language. I will not go into much detail on the language differences. The major change outside of language syntax is that the Modula-2 version uses procedure variables and types to distinguish between the many action procs where the C version just used variables of type ProcPtr. The more rigid type-checking of Modula-2 also forced a number of type conversions (via VAL) in comparisons and assignments which were made quite blithely in C, either implicitly or via a type coercion. I also treat the modifiers word as a BITSET rather than an integer as it is a more natural concept for the modifiers, allowing simple set membership tests.

## To build the examples

Obviously, the first thing you will need to do is to compile TransSkel.def and TransSkel.mod.

TransSkel.SBM should be kept in a place where it will be found when compiling any of the other modules ({MInterfaces} is a good place for it).

Compile the various examples and link as described in the following example link directive for the Skel example:

```
link -o Skel.Code Skel.MOD.o TransSkel.MOD.o  $\partial$ 
      -sg "Main=Skel,TransSkel,%Mod2Init,Strings,MacLib"  $\partial$ 
      "{MLibraries}"MRuntime.o "{MLibraries}"MInterface.o  $\partial$ 
      "{Libraries}"Runtime.o "{Libraries}"Interface.o
```

The .r files are Rez input files and will need to have the first lines edited to reflect where things can be found on your system. You should invoke Rez as follows (again the example is for Skel):

```
Rez -o Skel -t APPL -c SKEL Skel.R
```

### **Farewell and enjoy**

PS: Just as Paul is interested in any additions or corrections to TransSkel, I am interested in anything that might enhance this rendering and would appreciate being informed of such at the address above.