TransSkel 3.18 — Release Notes

This document is an adjunct to the TransSkel Programmer's Manual. It lists changes and additions specific to release 3.18.

The main enhancements for release 3.18 were:

- The code compiles cleanly using the universal headers.
- The code compiles under THINK C 6, as before, but now also under THINK C 7 and under Metrowerks C 4.5 (68K or PPC).
- Power Macintosh support was added, allowing native PowerPC code to be generated if your compiler supports it. (PowerPC code generation has been tested under Metrowerks only. If you're using Symantec's PPC CDK, I'd be interested in hearing about the results.)

Release 3.18 can be thought of as "do the same thing as 3.17, but more generally." That is, 3.18 doesn't do much new, but it can be used in a wider range of environments. The 3.18 interface should be compatible with the release 3.17 interface. That is, all other factors being equal, any 68K application that you built with the 3.17 header file and library should build without change using the 3.18 header file and library.

A few bugs were fixed:

- The dialog filter provided by `SkelDlogFilter()` and `SkelDlogFilterYD()` would leave the cursor as an i-beam if told to track the cursor in dialog edittext items and the user dismissed a dialog via keypress while the cursor was in a text item. The cursor is now restored to the arrow when the dialog is dismissed.

- In release 3.11 the TransSkel library has been made compatible with THINK Pascal, but that compatibility was broken by 3.17, which changed references to QuickDraw globals to be written in terms of the global `qd` struct. For example, references to `thePort` and `gray` became `qd.thePort` and `qd.gray`. Since THINK Pascal doesn't know about the `qd` struct, that broke THINK Pascal programs. In 3.18, TransSkel still compiles by default to use the `qd` struct, but it can optionally be compiled to use non-`qd` references. The *TransSkel/TP.π* project compiles THINK Pascal-compatible library. A prebuilt library *TransSkel/TP* is included in the Interface folder, which Pascal programs should include rather than the *TransSkel* library.

The TransSkel library and all of the demonstration applications have been tested in the 68K environment (THINK C, Metrowerks) and the PowerPC environment (Metrowerks). Because the source may now be compiled by different compilers and produce code for multiple architectures, I'm using the following conventions:

- THINK C project names have a "*.π*" suffix. In the top level TransSkel folder, *TransSkel.π* compiles code using QuickDraw global references that use the `qd` struct. *TransSkel/TP.π* compiles code that doesn't use the `qd` struct. The library compiled from the latter project is compatible with THINK Pascal, which doesn't understand the `qd` struct. (See *TransSkel.h* for explanation and check the Prefix in the *TransSkel/TP.π* project.)

- Metrowerks project names have "*/MW-68K.μ*" or "*/MW-PPC.μ*" suffixes. Projects with the 68K suffix compile 680x0 code and projects with the PPC suffix compile PowerPC code.

Prototypes for several routines were changed to use universal procedure pointer (UPP) types. These are identical to the old `ProcPtr` types when compiling 68K code, but for PowerPC code generation you'll need to pass routine descriptors via UPP's instead of function pointers. The files and routines affected by this change are:

SkelAlert.c:
The type of the filter argument to `SkelAlert()` is now `ModalFilterUPP` rather than `ModalFilterProcPtr`.

SkelDlogBtnOutliner.c:
`SkelSetDlogButtonOutliner()` now installs a drawing proc using a routine descriptor when generating PowerPC code.

SkelDlogFilter.c:
The types of the `SkelDlogFilter()` filter argument and return value are now `ModalFilterUPP` rather than `ModalFilterProcPtr`. The types of the `SkelDlogFilterYD()` filter argument and return value are now `ModalFilterYDUPP` rather than `ModalFilterYDProcPtr`.

SkelGetDlogProc.c:
The `SkelGetDlogProc()` return value type has been changed from `SkelDlogItemProcPtr` to `UserItemUPP`.

SkelSetDlogProc.c:
The caller of `SkelSetDlogProc()` should pass a procedure using a routine descriptor when generating PowerPC code. The procedure parameter type has been changed from `SkelDlogItemProcPtr` to `UserItemUPP`.

Note that in *TransSkel.c*, the filter argument to `SkelDialog()` is still a `ModalFilterProcPtr`, which means that on PPC machines the filter must either be written in native mode, or you must supply a wrapper function to pass to `SkelDialog()` that calls the non-native filter properly.

The general implications of the changes made to support compiling with the universal headers and generating Power Macintosh code are discussed in TransSkel Programmer's Note 13. Remarks specific to the grow zone procedure used with `SkelInit()` may be found in TransSkel Programmer's Note 5.