TransSkel 3.13 — Release Notes

This document is an adjunct to the TransSkel Programmer's Manual. It lists changes and additions specific to release 3.13. If you have only programmed with pre-3.0 releases of TransSkel, be sure to read the earlier 3.XX release notes as well.

- **The distinction between core, auxiliary, and convenience routines has become somewhat hazy, due, e.g., to the fact that on occasion it becomes convenient to use a convenience routine from within a core routine. Since the distinction is now less useful, I'm dropping it. The initial manifestation is that the "Core", "Auxiliary", and "Convenience" folders have been combined into a single "Source" folder.**

- **New function `SkelGetWindPropData()` to directly return the data value for a property (in contrast to returning the handle, from which you extract the data value).**

- **The event router now notices Command-period (or whatever the internationalized equivalent is) and doesn't send the event through the menu handlers. Instead the handler for the front window can detect it and respond to it as a Cancel event, or ignore it if it doesn't apply. For modeless or movable modal dialog windows this is done in the dialog filter; for non-dialog windows it's done in the key handler.**

- **Window-specific idle-time procedures are now run in `SkelRouteEvent()` rather than in `SkelEventLoop().` Functionally, this is the same as before except for an idle-time procedure that should run whether or not the associated window is frontmost. Before, the idle-time procedure would freeze when a modal dialog or alert came up. Now the procedure continues to run when you present a dialog or alert, at least if you use `SkelDlogFilter()` to make sure non-dialog events get sent into the main event router. The application idle-time routine no longer freezes, either.**

- **Window-specific idle-time routines now run when the application is suspended if they are set to run even when the window isn't frontmost.**

- **New function `SkelSetDlogCursor()` for setting the cursor to an I-beam or an arrow depending on whether or not it's in an edit text item.**

- **The dialog filters provided by `SkelDlogFilter()` and `SkelDlogFilterYD()` now pass command-clicks in the drag region of underlying windows through TransSkel's event router. This allows windows beneath a modal dialog to be dragged around in their current plane as per Apple's Human Interface Guidelines.**

- **Support for modeless dialogs has been much revised and support for movable modal dialogs has been added, as described in the next several paragraphs. TransSkel Programmer's Note (12) discusses the details of the changes made.**

- Incompatible change. **`SkelDialog()` now takes an argument allowing you to specify an event filter procedure. The filter allows dialog handlers to do things that before were difficult. For example, you can use a filter procedure to track mouse clicks in user items, map key clicks to button clicks (Return key becomes OK button, etc.), set hiliting state of controls and text selections on activates and deactivates, or track the cursor if there are edit text items.**

- Incompatible change. **The `doEvent` function argument to `SkelDialog()` is now called `doSelect,` and its type is called `SkelWindSelectProcPtr` rather than `SkelWindEventProcPtr`. These are better names because the purpose of the function is**

really for notification that a dialog item was selected, not to pass information about the event. The event is no longer passed to the selection function, and a pointer to the dialog is. If for some reason you really want the event from within the selection function, you can call `SkelGetCurrentEvent().`

- Movable modal dialog support has been added. A movable modal dialog treated much like a modeless dialog in that you create the dialog and register it with `SkelDialog().` The event router automatically handles events such as mouse clicks in the content region, key clicks, window dragging, updates and activates, just as for modeless dialogs. In addition, the router also handles beeping when you click outside the dialog window (unless you click in the menu bar or Command-click in another window's drag region)

- Since more than one kind of dialog can now be registered with `SkelDialog(),` the property added by that routine is now `skelWPropModeless` or `skelWPropMovableModal` depending on the type of dialog window.

- Due to the tighter integration of dialog support into the event router, there is no separate `DoDialog()` function any more. Dialog support is always compiled in and the `#ifdef supportDialogs` stuff is obsolete.

- Incompatible change. **The dialog event mask no longer applies, either, so the `SkelSetDlogMask()` and `SkelGetDlogMask()` functions have been dropped.**

- New routines `SkelIsDlog()` and `SkelIsMMDlog()` for determining whether or not a window is a dialog or movable modal dialog, respectively. These functions can be useful in menu handlers when the action associated with an item varies depending on the kind of window that's frontmost.

- New function `SkelDlogMapKeyToButton()` that determines whether or not a key event should be mapped onto a click in a dialog's default or cancel button. It's useful in modeless and movable modal dialog event filters, and the standard filter returned by `SkelDlogFilter()` for processing modal dialogs has been rewritten in terms of it as well.

- Revised the DialogSkel demonstration application to add a filter showing how to handle updates, activates, key-to-button mapping, cursor tracking, and enabling/disabling the default button.

- Revised the Button demonstration application to add a modeless dialog showing how a modeless dialog event filter works for updates, activates, and key-to-button mapping. Added a movable modal dialog (available for System 7 only). Shows menu adjustment when movable modal is shown/hidden. (Menu adjustment doesn't work in the Pascal version because I've, er, forgotten how to access a function in file A from file B when file B must be compiled before file A.)

- Removed redundant section from reference manual on using TransSkel from Pascal.