TransSkel 2.0 — Release Notes and Compatibility Issues

Applications created with versions of TransSkel prior to 2.0 are not compatible with version 2.0, which differs from earlier versions in the following respects.

- Better handling of out of memory conditions. All TransSkel calls which allocate space on the heap now return a value to indicate success or failure. Desk accessories now make sure they have at least the minimal amount of memory before opening.
- There is an explicit model of port-setting behavior, discussed in more detail under the description of the `SkelWindow()` function. Many extraneous `SetPort()` calls have been deleted.
- `SkelInit()` has become much smarter now. It will call `MoreMasters()` for you a specified number of times, and optionally install a user-supplied `GrowZone()` function.
- `SkelMenu()` now has a parameter which allows drawing of the menu bar to be delayed. This eliminates the annoying redrawing of the bar after each menu is added.
- The mechanism that determines the window to which events should be routed has been much improved. If you had many windows, for each event TransSkel would do an exhaustive search of its internal window list. Windows at the end of the list required a search of the entire list. There is now a cache, based on the presumption that the last window to receive an event is the window most likely to get the next event. This window is checked first before searching the list. The overhead for this is more than compensated by the savings in list traversal.

Converting Pre-Version 2.0 TransSkel Programs to Version 2.0

Since the TransSkel calls are all fairly easy to find, especially given THINK's fast searching capabilities, conversion is quite simple. The following illustrates how to quickly convert TransSkel calls for applications based on versions of TransSkel older than version 2.0.

- `SkelInit()`
  This will be called once near the beginning of the application.

  Change from: `SkelInit ();`
  to: `SkelInit (6, nil);`

- `SkelWindow()`

Calls to `SkelWindow()` may appear anywhere in an application. Strictly speaking, you do not need to change these, if you wish to ignore the return value. However, if you wish to explicitly discard the return value or assign it for testing, do the following:

Change from: `SkelWindow (args...);`
to: `(void) SkelWindow (args...);`
or to: `result = SkelWindow (args...);`

- `SkelMenu()`
  Calls to `SkelMenu()` may appear anywhere in an application. You must change these to supply an extra parameter. You can ignore the return value or assign it for testing, as follows:

  Change from: `SkelMenu (args...);`
  to: `(void) SkelMenu (args..., true);`

or to: `result = SkelMenu (args..., true);`

Note that this emulates pre-version 2.0 behavior, i.e., the menu bar is redrawn for every menu installed. It is better to pass false for the last argument for all but the last `SkelMenu()` call.

- If you find that you're drawing things in the wrong windows, you're probably relying on TransSkel to set the port for you. You may need to explicitly set the port.