

TransSkel 3.19 — Release Notes

This document is an adjunct to the TransSkel Programmer's Manual. It lists changes and additions specific to release 3.19.

This release is an update to allow compiling under release 2 of the universal headers, such as are supplied with Metrowerks 5.0 and 5.5.

- **The distribution now compiles under Metrowerks 5.0 and 5.5. I have been able to compile the library source with the following:**

- THINK C 7, using either the old Apple headers or the universal headers release 1
- Metrowerks 3.0 through 5.5 (68K or PPC) using whatever universal headers come with the compiler

Some support for older constructs has been removed, so the distribution no longer compiles under THINK C 6 without some tweaking (see below about function call renaming).

- **The distribution now compiles under release 2 of the universal headers. The macro that indicates whether or not the universal headers are being used (`skelUnivHeaders`) now may have three values:**

0	Univers
1	Univers
2	Univers

Thus `#if skelUnivHeaders` or `#if !skelUnivHeaders` still distinguish universal from non-universal header compilation, but you can also test whether the value of `skelUnivHeaders` is 1 or 2 for cases when you must make decisions based on which release of the headers is being used. TransSkel Programmer's Note 13 was updated to reflect changes in the method by which the value of `skelUnivHeaders` is determined. The Note also has a new section that describes allowable values for macros like `OLDROUTINENAMES` and `STRICT_{CONTROLS,WINDOWS,LISTS,MENUS}` when you precompile the headers..

- **Changed instances of old routine names like `GetIText()` to new routine names like `GetDialogItemText()`. The following routines were renamed:**

<u>Old Name</u>	<u>New Name</u>
<code>DelMenuItem()</code>	<code>DeleteMenuItem()</code>
<code>GetItem()</code>	<code>GetMenuItemText()</code>
<code>AddResMenu()</code>	<code>AppendResMenu()</code>
<code>GetMHandle()</code>	<code>GetMenuHandle()</code>
<code>RsrvMem()</code>	<code>ReserveMem()</code>
<code>KeyTrans()</code>	<code>KeyTranslate()</code>
<code>GetCRefCon()</code>	<code>GetControlReference()</code>
<code>SetCRefCon()</code>	<code>SetControlReference()</code>
<code>GetCtlValue()</code>	<code>GetControlValue()</code>
<code>SetCtlValue()</code>	<code>SetControlValue()</code>
<code>GetCtlMax()</code>	<code>GetControlMaximum()</code>
<code>SetCtlMax()</code>	<code>SetControlMaximum()</code>
<code>GetIText()</code>	<code>GetDialogItemText()</code>
<code>SetIText()</code>	<code>SetDialogItemText()</code>
<code>SelIText()</code>	<code>SelectDialogItemText()</code>

FindDItem()	FindDialogItem()
GetDItem()	GetDialogItem()
SetDItem()	SetDialogItem()
UpdtDialog()	UpdateDialog()
DlogCut()	DialogCut()
DlogCopy()	DialogCopy()
DlogPaste()	DialogPaste()
DlogDelete()	DialogDelete()

New names for the following routines were also used, except that the old name must still be used with non-universal headers (the new names apparently did not appear until the universal headers were introduced):

Old Name

SizeResource()	GetResourceSizeOnDisk()
GetEnvirons()	GetScriptManagerVariable()

These name changes do not affect the THINK Pascal interface—calls to the routines named above compile to A-traps anyway, so no link errors occur. The name changes do affect compilation under THINK C 6 — the header files that come with that compile don't understand all of the new names, so they need to be renamed back. Sorry.

- **Some minor cleanup of source files was done to remove unused variables and get rid of compiler warnings.**
- **The distribution contains a new file *TestSymbol.c* in the *Misc* folder. It's probably not generally useful but I find it helpful for determining which architecture- and library-dependent symbols are defined in *MacHeaders* and what values they have. To use it, open a project, then open *TestSymbol.c* and with the open document window frontmost, select Preprocess from the appropriate menu.**