

# New Technical Notes

Macintosh



---

Developer Support

## TE 26 – Inside Macintosh Text and Life Before 7.1 Text

Written by: Norbert Lindenberg

June 1993

*Inside Macintosh: Text* documents the text handling functionality in System 7.1. This Technical Note gives you some additional information that lets you design software using *Inside Macintosh: Text* and make it work on system software before System 7.1, going back to System 6.0.7.

---

### Introduction

*Inside Macintosh: Text* is the best documentation on Macintosh text handling (especially its international aspects) that you've ever had, but it focuses on the functionality available in System 7.1. Many of the features documented in the book were newly introduced in System 7.0 or even in System 7.1. If your software has to be able to run on System 7.0 and System 6.0.7, you will have to use some older, now obsolete routines, that are no longer documented in *Inside Macintosh: Text*, or fill in some gaps by writing your own code.

This technical note gives you an overview of the trouble areas. The idea is that you design or internationalize your software using the information in *Inside Macintosh: Text* (which, after all, is the most coherent documentation available), and then check this Technical Note to find out where you have to add code in order to be able to run on older system software.

If your software already makes use of the old Script Manager and International Utilities calls, you may want to check the appendix "Renamed and Relocated Text Routines" in *Inside Macintosh: Text* to find out which new routines you can take advantage of in System 7.1.

### References

This technical note may refer you to the following documentation:

*IM: Inside Macintosh*; if a name follows, we are talking about a volume of the second edition, if a Roman number follows, about a volume of the first edition; published by Addison-Wesley.

*TN*: Macintosh Technical Notes; available on the Developer CD and from many other sources.

*WWD*: *Macintosh Worldwide Development: Guide to System Software* (beta draft); available from APDA.

## New Features in System 7.1

### Managers

The Text Services Manager and the Dictionary Manager are new in System 7.1. Before using them, make sure to check for their existence using Gestalt.

### Routines

The following table shows the routines of other text managers that are new in System 7.1, which earlier routines can be used instead of them, and where to find documentation for the precursors:

Routine Name	Precursor	Documentation for Precursor
CharacterByteType	CharByte	IM V-306; WWD 142
CharacterType	CharType	IM V-306–307; WWD 142–143
FillParseTable	ParseTable	WWD 178–179
FindWordBreaks	FindWord	IM V-309; WWD 145–146
GetScriptQDPatchAddress	—	—
GetScriptUtilityAddress	—	—
LowercaseText	LowerText	IM VI-14-62
SetScriptQDPatchAddress	—	—
SetScriptUtilityAddress	—	—
StripDiacritics	7.0: StripText 6.x: — (see Note)	IM VI-14-63
TransliterateText	Transliterate	IM V-311–312
UppercaseStripDiacritics	7.0: StripUpperText 6.x: — (see Note)	IM VI-14-63
UppercaseText	7.0: UpperText 6.x: — (see Note)	IM VI-14-63

**Note:** The routines StripText, StripUpperText, and UpperText are available on some machines running System 6.x—namely the Macintosh IIfx, IIfx, IIfx, LC, and Portable. If you don't want to check for individual machines, however, you should assume that they are not available before System 7.0.

**Note:** IM VI mentions the NFindWord routine as a replacement for the older FindWord routine. We do not recommend using NFindWord—FindWord works and is a functional superset of NFindWord.

### Script Manager Constants

The following constants for Script Manager routines are new in System 7.1 and cannot be used in earlier systems:

- Selector for GetScriptManagerVariable and SetScriptManagerVariable: smKeyDisableState.
- For bits in the long word accessed by GetScriptManagerVariable and SetScriptManagerVariable with the smGenFlags selector: smfUseAssocFontInfo.
- For bits in the word accessed by GetScriptVariable and SetScriptVariable with the smScriptFlags selector: smsfUnivExt, smsfSynchUnstyledTE.

- Codes for KeyScript: smKeySwapKybd, smKeySetDirLeftRight, smKeySetDirRightLeft, smKeyRoman.
- Return values of CharacterType: smCharContextualLR, smCharNonContextualLR, smCharBopomofo.

## International Resources

“itlc”: The field itlcSysFlags was added to this resource type in System 7.1.

“itl2”: a new format of the script run table was defined for System 7.1. Both the old and the new format are documented in IM Text B-40 to B-43, and both continue to be supported.

“trsl”: This resource type is new in System 7.1 and cannot be used in earlier systems.

## New Features in System 7.0

### Routines

The following table shows the routines that are new in System 7.0, which earlier routines can be used instead of them, and where to find documentation for the precursors:

Routine Name	Precursor	Documentation for Precursor
CharToPixel	Char2Pixel	IM V-308–309; WWD 140–142
ClearIntlResourceCache	—	
CompareString	IUCompString	IM I-506
CompareText	IUMagString	IM I-506
DrawJustified	DrawJust	IM V-310–311; WWD 144
FlushFonts	— (see Note)	
GetIntlResourceTable	—	
GetOutlinePreferred	— (see Note)	
GetPreserveGlyph	— (see Note)	
IdenticalString	IUEqualString	IM I-506–507
IdenticalText	IUMagIDString	IM I-507
IsOutline	— (see Note)	
LanguageOrder	—	
MeasureJustified	MeasureJust	IM V-311; WWD 178
OutlineMetrics	— (see Note)	
PixelToChar	Pixel2Char	IM V-307–308; TN Pixel2Char;
	WWD 179–181	
PortionLine	PortionText	WWD 182–183
ReplaceText	—	
ScriptOrder	—	

SetOutlinePreferred — (see Note)  
SetPreserveGlyph — (see Note)  
StringOrder —  
TextOrder —  
TruncString —  
TruncText —

**Note:** The Font Manager routines supporting TrueType may be available on machines running System 6.x, for example, if StyleWriter software is installed. If Gestalt with the `gestaltFontMgrAttr` tells you that outline fonts are available (bit `gestaltOutlineFonts` is set), the Font Manager routines `FlushFonts`, `GetOutlinePreferred`, `GetPreserveGlyph`, `IsOutline`, `OutlineMetrics`, `SetOutlinePreferred`, and `SetPreserveGlyph` are available.

## TextEdit Constants

The following constants for TextEdit routines are new in System 7.0 and cannot be used in earlier systems:

- Feature codes for `TEFeatureFlag`: `teFInlineInput`, `teFUseTextServices`. Actually, these feature codes are supported in versions of System 6.0.7 that have nonroman scripts installed.
- Hook code for `TECustomHook`: `intTextWidthHook`.

## Script Manager Constants

The following constants for Script Manager routines are new in System 7.0 and cannot be used in earlier systems:

- Selectors for `GetScriptManagerVariable` and `SetScriptManagerVariable`: `smDoubleByte`, `smKCHRCache`, `smRegionCode`.
- Implicit script codes for `GetScriptVariable` and `SetScriptVariable`: `smSystemScript`, `smCurrentScript`.
- Selectors for `GetScriptVariable` and `SetScriptVariable`: `smScriptMonoFondSize`, `smScriptPrefFondSize`, `smScriptSmallFondSize`, `smScriptSysFondSize`, `smScriptAppFondSize`, `smScriptHelpFondSize`, `smScriptValidStyles`, `smScriptAliasStyle`.
- Codes for `KeyScript`: `smKeyNextKybd`, `smKeyDisableKybds`, `smKeyEnableKybds`, `smKeyToggleInline`, `smKeyToggleDirection`, `smKeyNextInputMethod`, `smKeySwapInputMethod`, `smKeyDisableKybdSwitch`.
- Field masks for `CharType`: `smcOrientationMask`.
- Return values of `CharType`: `smCharBidirect`, `smCharHangul`, `smCharJamo`, `smPunctRepeat`, `smPunctGraphic`, `smJamoJaeum`, `smJamoBogJaeum`, `smJamoMoeum`, `smJamoBogMoeum`, `smKanaSmall`, `smKanaHardOK`, `smKanaSoftOK`, `smCharHorizontal`, `smCharVertical`.
- Source mask and target values for `Transliterate`: `smMaskHangul2`, `smMaskJamo2`,

smMaskBopomofo2, smTransHangul, smTransJamo2, smTransBopomofo.

## **International Resources**

“itlc”: The field itlcRegionCode was added to this resource type in System 7.0. In general, you don’t need to access “itlc” resources, because all relevant information is available through the GetScriptManagerVariable and SetScriptManagerVariable routines.

“itlb”: Several fields were added to this resource type in System 7.0. In general, you don’t need to access “itlb” resources, because all relevant information is available through the GetScriptVariable and SetScriptVariable routines. If you need to access resources directly, and run on system software before System 7.0, you should use the data type ItlbRecord instead of ItlbExtRecord to access the content of “itlb” resources.

**Note:** You may find extended “itlb” resources in some versions of System 6.0.7, but they are not guaranteed to exist or be supported in all versions.

“itl1”: The optional extension that can contain additional month and day names as well as abbreviated month and day names is new in System 7.0.

“itl2”: The length information for code and tables is new in System 7.0. The break tables use a new format, NBreakTable. If you need to define break tables for earlier systems, or need to access older break tables, use the BreakTable data type. See WWD 209–216 for more information about the old break table format. The optional script run table was added in System 7.0.

“itl4”: The length information for code and tables is new in System 7.0. For systems before 7.0, you should use the data type Itl4Rec instead of NItl4Rec to access the content of “itl4” resources. In the untoken table, the following tokens were added in System 7.0: tokenEllipsis, tokenCenterDot. The white space table is new in System 7.0.

“itl5”: This resource type is new in System 7.0 and cannot be used in earlier systems.

“itlm”: This resource type is new in System 7.0 and cannot be used in earlier systems.

## Font Resources

“sfnt”: This resource type is new with TrueType and supported only if TrueType is available. TrueType became generally available with System 7.0, but may also be available on machines running System 6.x, for example, if StyleWriter software is installed. If Gestalt with the gestaltFontMgrAttr tells you that outline fonts are available (bit gestaltOutlineFonts is set), “sfnt” resources are supported.

## Acknowledgments

Thanks to Jeannette Cheng, Peter Edberg, and Kenny Tung for reviewing this document and providing me with some historical information.

## Further Reference:

---

- *Inside Macintosh: Text*
- *Inside Macintosh*, Volume I, The International Utilities Package

- *Inside Macintosh*, Volume V, The Script Manager
- *Inside Macintosh*, Volume VI, The Font Manager
- *Inside Macintosh*, Volume VI, Worldwide Software Overview
- *Inside Macintosh*, Volume VI, TextEdit
- Macintosh Technical Note: “TE 12 - Script Manager’s Pixel2Char”
- *Macintosh Worldwide Development: Guide to System Software* (beta draft)