The 'KCHR' resource specifies the mapping of virtual key codes to character codes (for example, ASCII). Each installed script system has one or more 'KCHR' resources; there may be one or more for each language or region to suit the preference of the user. The resource ID for each 'KCHR' resource is in the script's resource number range. The default 'KCHR' resource for a script is specified by the script's 'itlb' resource. For more on virtual key codes, see the Toolbox Event Manager chapter in Volume V. If you do not have access to the MPW file SysTypes.r, which contains the Rez type definition for the 'KCHR' resource, consult Macintosh Developer Technical Support.

Note: Prior to system software version 7.0, an application could force a script system to load a 'KCHR' resource that only existed inside its resource fork. Since 'KCHR' resources are used across the system (that is, are not application-specific), this was an extremely undesirable situation. If an application failed to restore the appropriate script system variables properly when it terminated or was switched into the background, the Script Manager attempted to find a 'KCHR' resource that was no longer available.

With version 7.0, the Script Manager has been revised to load only 'KCHR' resources that are installed in the System file. However, if your application needs to modify the keyboard layout temporarily without forcing users to install a new keyboard, you should load a 'KCHR' resource from your resource fork and pass a pointer to that 'KCHR' resource directly to the _KeyTrans trap.

Figure 14-28 shows the parts of the 'KCHR' resource and how they are used. First, the modifier state information—8 bits, each indicating the state of one modifier key—is treated as a byte that is mapped through a 256-byte table-selection table to a table code. The table code specifies which of several 128-byte mapping tables is used to map the virtual key code to a character-code byte. If the mapping table has a nonzero entry for a particular virtual key code, that entry is the desired character code. If the entry is 0, the dead-key tables are searched for a match with the virtual key code. If there is a match, dead-key state information is set that affects how the next virtual key code is processed, but no event is posted.

With system software version 7.0, the following changes in the U.S. 'KCHR' resource (0) have been introduced for greater consistency:

• Since the Shift key is ignored if the Command key is pressed, the 'KCHR' resource has been changed so the Caps Lock key is also ignored if the Command key is pressed.

• Table 14-13 lists the changes in the handling of Option-Shift-key and Option–Caps Lock–key combinations. These changes are based on the following principles:

  • If either the Option-key or the Option-Shift-key combination produces a letter, then the Option–Caps Lock–key combination should produce the same character as the Option-key, not the Option-Shift-key, combination.

  • If the Option-key combination is a dead key for a particular accent, then the Option-Shift-key combination produces the accent directly.

- System software version 7.0 also includes some changes in default dead-key completers, so that the completer is a "real" accent character instead of a low-ASCII approximation. The default completer is used when a dead key is entered, but the following key is either a space or a key for a character that cannot take the accent corresponding to the dead key. Table 14-14 summarizes these changes.

Table 14-13 Changes in handling 'KCHR' (0) key combinations

Key combination	Old character	New character	Reason
Option–Caps Lock–E	‰	´	Acute accent for dead-key Option-E
Option–Caps Lock–G	Ì	©	Be like Option-G, not Option-Shift-G
Option–Caps Lock–I	È	^	Circumflex accent for dead-key Option-I
Option–Caps Lock–K	($F0)	°	Be like Option-K, not Option-Shift-K
Option–Caps Lock–M	˜	Â	
Option–Caps Lock–N	^	˜	Tilde accent for dead-key Option-N
Option–Caps Lock–R	Â	®	Be like Option-R, not Option-Shift-R
Option–Caps Lock–T	Ê	†	Be like Option-T, not Option-Shift-T
Option–Caps Lock–U	Ë	¨	Umlaut for dead-key Option-U
Option–Caps Lock–V	◊	√	Be like Option-V, not Option-Shift-V
Option–Caps Lock–W	„ ($E3)	Σ	Be like Option-W, not Option-Shift-W
Option–Caps Lock–X	Ù	≈	Be like Option-X, not Option-Shift-X
Option–Caps Lock–Z	Û	Ω	Be like Option-Z, not Option-Shift-Z
Option-Shift-E	‰ ($E4)	´	Acute accent for dead-key Option-E
Option-Shift-G	Ì	″ ($FD)	Make ″ (double-acute accent) available for keyboard
Option-Shift-I	È	^ ($F6)	Circumflex accent for dead-key Option-I
Option-Shift-M	˜ ($F7)	Â	
Option-Shift-N	^ ($F6)	˜ ($F7)	Tilde accent for dead-key Option-N
Option-Shift-R	Â	‰	Moved ‰ from Option-Shift-E
Option-Shift-T	Ê	ˇ ($FF)	Make ˇ (hacek) available from keyboard
Option-Shift-U	Ë	¨	Umlaut for dead-key Option-U
Option-Shift-X	Ù	˛ ($FE)	Make ˛ (ogonek) available for keyboard
Option-Shift-Z	Û	¸ ($FC)	Make ¸ (cedilla) available for keyboard
Option-Shift-`	Ÿ	`	Grave accent for dead key Option-`

Table 14-14 Changes in 'KCHR' dead-key completers

Dead key	Old default completer	New default completer
Option-N	~ ($7E)	˜ ($F7)
Option-U	^ ($5E)	^ ($F6)

Using the Standard Roman Character Set

Be aware that the "traditional" Macintosh character set (that is, the original set described in Volume I as the Macintosh character set) stops at code $D8 and contains a limited set of European accented forms. The standard Roman character set now includes the remaining character codes ($D9–$FF); it supplies uppercase versions of all of the lowercase accented forms in the traditional set, new symbols, and other forms. These characters are available in most LaserWriter® and TrueType™ fonts, but not in the Apple bitmap versions of Chicago, Geneva, or Monaco. See the Font Manager chapter in this volume for an illustration of the standard Roman character set. See *Macintosh Worldwide Development: Guide to System Software* for further information on the standard Roman character set (there referred to as the extended Roman character set).

Since system software version 6.0.4, there has been full support for the standard Roman character set. This version has supplied more completeness and consistency in the handling of accented forms in the fonts that contain these forms.

Also with system software version 6.0.4, the U.S. keyboard resource 'KCHR' (0) has been modified to make it possible to enter the accented forms with dead keys. Users can enter all the accented forms in the original Macintosh character set with dead keys.

  Note: With system software version 7.0, the keyboard entry of the following four
  characters is now possible using the U.S. 'KCHR' resources: cedilla (¸),
  double-acute accent (˝), ogonek (˛), and hacek (ˇ). These characters are
  represented by codes $FC–FF.

The tables used by the Script Manager's CharType, Transliterate, and FindWord routines (in 'itl2'), by the IntlTokenize function (in 'itl4'), and by the International Utilities Package (in Pack6) have been modified in the U.S. system for proper treatment of the character-set extensions as well as the traditional characters ß, ø , and Ø.