Creating Simple Script Systems

You can create simple script systems-that is, script systems that contain small character sets and are noncontextual and left to right (for example, Greek or Cyrillic). You are only required to supply the appropriate fonts and the following international, keyboard, and font resources: 'itlb' (with the smsfAutoInit bit on, as described below), 'itl0', 'itl1', 'itl2', 'itl4', 'KCHR', and the keyboard color icon family ('kcs#', 'kcs4', and 'kcs8'). You must also supply 'FOND' and 'NFNT' or 'sfnt' resources.

Such simple script systems can use the Roman Script System routines; however, to operate as a script system, they need their own local variables.

> **Note:** The capability to create simple script systems is available beginning with system software version 6.0.7.

If the flag word in a script's 'itlb' resource in the System file has the smsfAutoInit bit set, the **Script Manager** initializes the local variables for that script and fills in the font and style information from fields in 'itlb'.

To provide built-in support for all simple script systems, the Roman **FindScriptRun** function has been modified to use an optional table in the 'itl2' resource that specifies the location of Roman characters in a non-Roman font. See **The 'itl2' Resource** for details.

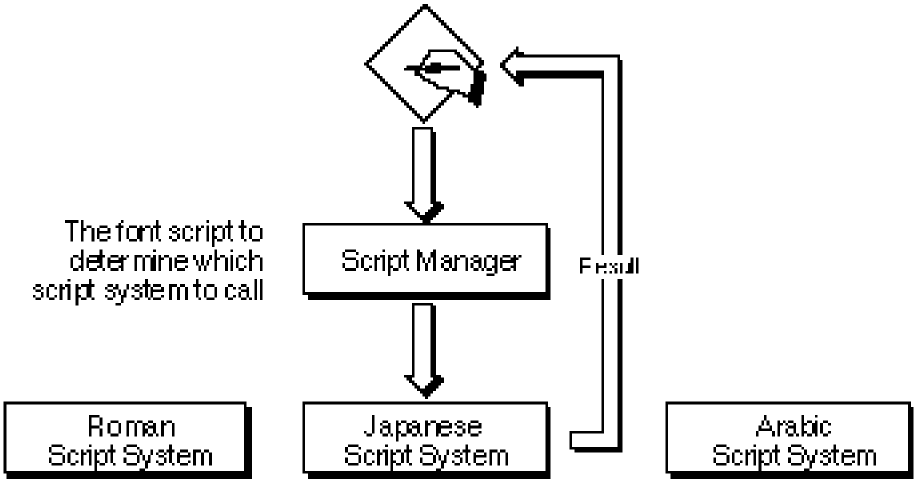
Calling the Script Manager

The **Script Manager** implements several routines itself, but for many other routines it acts as a dispatcher to the appropriate script system. For example, each script system provides a **CharType** function. When a program calls **CharType**, the **Script Manager** uses the current **font script** (that is, the script that corresponds to the font of the current grafPort) to dispatch the call to the correct script system. Your application does not need to know whether a particular routine is implemented by the **Script Manager** or by a script system.

Assembly-language note: All **Script Manager** routines except the **LowerText**, **UpperText**, **StripText**, and **StripUpperText** procedures are called via the _ScriptUtil trap.

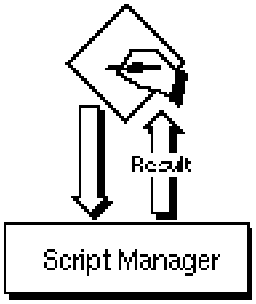
> **Warning:** You should always have the grafPort and **A5 world** set appropriately before you call any **Script Manager** or **International Utilities Package** routine. **A5** must point to the **QuickDraw** global variables, and thePort->txFont must be set correctly.

The Figure below shows how the **Script Manager** calls a script system when an application calls a **Script Manager** routine that is implemented by each script system (for example, the **Pixel2Char** function). When your application calls **Pixel2Char**, the **Script Manager** uses the font script to determine which script system to call. In the example in the Figure below, the assumption is that a Japanese font is the font of the current grafPort, so the **Script Manager** dispatches to the Japanese Script System. The **Pixel2Char** function in the Japanese Script System is used, and it returns the result directly to the application that called **Pixel2Char**.



Calling the **Script Manager** routines implemented by a script system

When an application calls a **Script Manager** routine that is implemented directly by the **Script Manager** itself (for example, the **FontScript** function), the flow of control is as shown in the Figure below.



Calling the **Script Manager** routines