New Technical Notes

Macintosh

Developer Support

Accessing the Script Manager Print Action Routine

Text **M.TE.PrintAction**

Revised by: March 1988
Written by: Mark Davis November 1987

This technical note describes how Print Drivers can access the Script Manager Print Action routine to print unconventional text, such as Japanese or Arabic.

General Notes

Scripts such as Japanese or Arabic modify the normal QuickDraw text handling in order to represent text properly. On the screen, this is done by trapping `StdText` and `StdTxtMeasure`, and transforming the text before printing. For example, for Hebrew or Arabic the text might be reversed, since text normally goes from right to left in those scripts.

Print drivers require slightly different handling, for two reasons:

1. A print driver might not call the standard QuickDraw procedures. For example, the LaserWriter writes directly in PostScript instead.

2. A print driver might need to format the text, for accurate line-layout. In this case, the text needs to be transformed before the driver performs line-layout. If the driver is spooling the text, and will replay the text a second time, the text cannot be transformed a second time, since that would ruin the appearance.

For example, the ImageWriter driver calls QuickDraw procedures twice, once to spool and once to unwind the spooling. The text must be transformed when spooling, so that line layout can be done, but when unwinding, the transformation must be turned off completely.

Note that some drivers, such as the LaserWriter, use QuickDraw re-entrantly: the application program calls a QuickDraw routine, which is directed to the driver's `grafProcs`, which in turn call QuickDraw internally to put up status messages on the screen. The Print Action procedure handles the text properly so that the text transformations are enabled during the re-entrant calls, so that the status messages will be properly formatted.

When To Call the Print Action Routine

The Script Manager Print Action routine allows the print driver to be independent of the particular scripts being used. The printing driver should call this routine whenever it changes the `grafProcs` in the printing `grafPort`. The Print Action routine will then substitute `grafProcs` of its own in the `grafProcs` record, saving the original routine addresses.

The Print Action routine will actually call a Print Action routine for each script system that is currently installed. Each of the script Print Action routines will do the appropriate tasks for its system.

Calling the Print Action Routine

To call the Print Action routine, the driver should use the following code:

```
intlGlobals   equ    $ba0   ; international globals
printActionOff      equ    $16    ; offset to PrintAction proc ptr

; get procedure pointer to call

       tst.w  Rom85                     ; on a Macintosh + or better?
       blt.s  @PrintActionDone          ; no, skip
       move.l intlGlobals,d2            ; get international globals
       ble.s  @PrintActionDone          ; not there, skip
       move.l d2,a0                     ; in address register
       move.l printActionOff(a0),d2     ; get print action address
       beq.s  @PrintActionDone          ; not there, skip
       move.l d2,a0                     ; in address register

; set up arguments to call

       move.l <myPort>,d0               ; pass the port
       move.w <myVerb>,d1               ; pass the verb
       jsr    (a0)                      ; call the procedure
@PrintActionDone
```

Print Action Routine Verbs

There are currently three verbs to pass to the Print Action routine.

```
       paUnwindText        equ    -1
       paSpoolText         equ    1
       paNoQuickDraw       equ    3
```

Use the `paUnwind` verb to ensure that the text is not transformed before your `StdText` procedure receives the text. This verb is used when playing back stored text that has already been transformed.

The other two verbs (`paNoQuickDraw` and `paSpoolText`) are used to ensure that the text is transformed before your `StdText` procedure receives the text. The `paSpoolText` verb is used when your driver will use QuickDraw to image the text in the printing `grafPort`. The `paNoQuickDraw` verb is used when the text is not drawn into the printing port by going through QuickDraw (e.g. the LaserWriter). In that case some languages (e.g. Japanese) which use an extended font structure may need to recast the text calls as `CopyBits` calls.

As mentioned above, some applications may call QuickDraw from within the driver, as when a status window is updated. During any `StdTxtMeasure` calls in the driver during the application's call to `StdText`, the port is checked against the printer port. If they match, then the text is not transformed. Otherwise, the text is transformed.

The solutions adopted by the Print Action routine assume that the print driver does not measure or draw text except within calls to `StdTxtMeasure` or `StdText`. If your driver does text buffering (as for line layout), make sure that any measurements are performed within these two calls. For example, you might buffer both the text and its screen width as measured by QuickDraw.

Further Reference:

- The Script Manager