

TECHNOTE: Inside Macintosh: GX Series Addenda

By Ingrid Kelly
inge@apple.com
Apple Developer Technical Support (DTS)

This Technote discusses addenda to the *Inside Macintosh: QuickDraw GX Series*.

This Note is intended for Macintosh QuickDraw GX developers who are using any version of QuickDraw GX up to and including 1.1.3.

Topics:

Addendum to QuickDraw GX Environment and Utilities

- GX Gestalt Versions February 1996

Addendum to QuickDraw GX Objects

- GXGetShapeClipType February 1996

Addenda to QuickDraw GX Printing Extensions and Drivers

- PostScript Render Options February 1996
- GXSetupPageImageData message February 1996
- GXPostScriptEjectPendingPage message February 1996
- GXHandleAltDestination message February 1996

- The Alternate Destination ('dsta') resource February 1996
- The Custom Halftone Matrix ('dmat') resource February 1996
- The Manual Feed Preferences ('mfpr') resource February 1996
- The Desktop Printer Output ('outp') resource February 1996
- Change Default DTP ('pfpr') Apple Event February 1996

Addenda to QuickDraw GX Typography

- GXGetLayoutJustificationGap February 1996
- GXGetLayoutJustificationFactors February 1996
- GXGetFontDefaultFeatures February 1996

Addendum to QuickDraw GX Environment and Utilities

GX Gestalt Versions

To determine the current version of QuickDraw GX in general, you call the Gestalt functions with the various gestalt selectors as outlined on p. 1-4 of *Inside Macintosh: QuickDraw GX Environment and Utilities*. The function returns a value indicating the version of QuickDraw GX printing/graphics/etc. currently installed.

This is how the QuickDraw GX gestalt 'qdgx' works:

major revision	minor revision	revision stage
0000	00	00

Table 1 shows you version gestalt values for QuickDraw GX.

Table 1 Gestalt values for QuickDraw GX

'qdgx' Value	GX Version
0x00010000	GX 1.0
0x00010001	GX 1.0.1
0x00010002	GX 1.0.2
0x00010100	GX 1.1
0x00010101	GX 1.1.1
0x00010102	GX 1.1.2
0x00010103	GX 1.1.3

Addendum to QuickDraw GX Objects

GXGetShapeClipType

You can use the `GXGetShapeClipType` function to retrieve the clip shape type that a shape is being clipped to.

```
gxShapeType GXGetShapeClipType(gxShape source)
```

source A reference to the shape you want to examine the clip property of.

function result A reference to the clip shape type.

DESCRIPTION

This function is used to determine just the clip shape and eliminates three expensive steps that were necessary in QuickDraw GX 1.0 to get this information.

The clip shape, which you specify using the clip parameter, may be a geometric shape, a bitmap shape, or a glyph shape. In the current implementation, it may not be a picture, text, or layout shape.

- If you specify a geometric shape, it must be in primitive form — that is, all the stylistic information about the shape must be incorporated into the

shape's geometry — because this function copies only the geometry-related information from the shape you specify. It does not copy the information contained in the shape's style. You can convert a shape to its primitive form using the `GXPrimitiveShape` function, which is described in *Inside Macintosh: QuickDraw GX Graphics*. You can also specify an empty or full shape for a clip.

- If you specify a bitmap shape, it must have a pixel size of 1 and its color profile reference must be `nil`. In the bitmap, pixel values of 0 obscure drawing; pixel values of 1 do not restrict visibility.
- If you specify a glyph shape, this function uses information from the glyph shape's style object as well as its style list to determine the size, form, and position of the glyph outlines; those outlines are then used to clip drawing. The style list cannot have `nil` entries. A style object referenced by the glyph shape cannot be complex — that is, it cannot have a cap, join, dash, pattern, text face, font variation, tag list, or any of the properties used only by layout shapes.

ERRORS

`out_of_memory`
`shape_is_nil`

Addenda to QuickDraw GX Printing Extensions and Drivers

PostScript Render Options

`gxNoIllegalEPSOperator` and `gxEPSTargetOption` flag have been added to PostScript Render Options.

```
enum
{
    gxNeedsHexOption           = 0x00000001,
    gxNeedsCommentsOption     = 0x00000002,
    gxBoundingBoxesOption     = 0x00000004,
    gxPortablePostScriptOption = 0x00000008,
```

TECHNOTE: Inside Macintosh: GX Series Addenda

```
gxUseLevel2ColorOption      = 0x00000080,  
gxNoEPSIllegalOperators    = 0x00000100,  
gxEPSTargetOption          = gxNoEPSIllegalOperators +  
                             gxNeedsCommentsOption +  
                             gxBoundingBoxesOption  
};  
  
typedef long gxPostScriptRenderOptions;
```

CONSTANT DESCRIPTIONS

`gxNoEPSIllegalOperators` The driver should only emit operators that are allowed by Encapsulated PostScript specifications (Version 3.0).

`gxEPSTargetOption` The driver should issue PostScript intended for EPS use.

For descriptions of other constants, please see *Inside Macintosh: QuickDraw GX Printing Extensions and Drivers*, p. 4-27.

GXSetupPageImageData message

QuickDraw GX sends the `GXSetupPageImageData` message to indicate the final format, page shape, and imaging data for any given page. The message is sent once for every page, just prior to `gxStartSendPage`. You may wish to override this message in order to determine the final characteristics of the data which will be sent to the printer. Your override of the `GXSetupPageImageData` message must match the following declaration:

```
OSErr MySetupPageImageData(gxFormat theFormat, gxShape thePage, void  
                           *imageData);
```

`theFormat` The format object for the page.

`thePage` A reference to the page shape.

`imageData` A pointer to imaging-specific data for initializing the printing device.

function result An error code. The value `noErr` indicates that the operation was successful.

SPECIAL CONSIDERATIONS

You never send the `GXSetupPageImageData` message yourself. This message is *not* a device communications time message. Overrides should always forward this message.

RESULT CODES

`gxSegmentLoadFailedErr`

A required code segment could not be found, or there was not enough memory to load it.

`gxPrUserAbortErr`

The user has cancelled printing.

GXPostScriptEjectPendingPage message

QuickDraw GX sends the `GXPostScriptEjectPendingPage` message to check if there is a previously imaged page waiting to be ejected. If so, it will send the `gxPostScriptEjectPage` message for that page. Your override of the `GXPostScriptEjectPendingPage` message must match the following declaration:

```
OSErr MyPostScriptEjectPendingPage(Boolean *pageWasEjected)
```

`*pageWasEjected`

A pointer to a Boolean value. On return, the value is true if a page was ejected and false otherwise.

function result An error code. The value `noErr` indicates that the operation was successful.

DESCRIPTION

You can override this message to implement your own page handling. If your driver is going to display a manual feed alert, it is usually desirable to eject the previous page before asking the user to insert the next sheet of paper.

SPECIAL CONSIDERATIONS

Overrides should always forward this message.

RESULT CODES

`gxSegmentLoadFailedErr`

A required code segment could not be found, or there was not enough memory to load it.

`gxPrUserAbortErr`

The user has cancelled printing.

GXHandleAltDestination message

QuickDraw GX sends the `GXHandleAltDestination` message when a destination other than "Print" is selected in the Print dialog. Your override of the `GXHandleAltDestination` message must match the following declaration:

```
OSErr MyGXHandleAltDestination (Boolean *userCancels)
```

`*userCancels` A pointer to a Boolean value. On return, the value is true if the user cancels.

function result An error code. The value `noErr` indicates that the operation was successful.

DESCRIPTION

This message is sent when the destination pop-up in the Print dialog is something other than Print. For instance, currently, if a user selects File in the Print dialog, the Print button changes to Save. If you have a FAX driver and you want the Print dialog to be more user-friendly, you simply override this message to add your destination (e.g., TheFax) and to change the Save button to FAX for example.

SPECIAL CONSIDERATIONS

The override should check the `gxFileFormatTag` in the job collection. If the `fileFormatName` is an alternate destination, the driver or extension can either display its own dialog to get more information from the user or forward the message for the default 'Standard File' dialog.

In order to change the destination, you must have an alternate destination ('dsta') resource declared. For more information on the 'dsta' resource, continue reading this document.

RESULT CODES

gxSegmentLoadFailedErr

A required code segment could not be found, or there was not enough memory to load it.

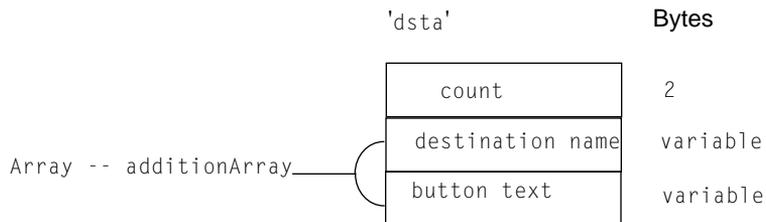
gxPrUserAbortErr

The user has cancelled printing.

The Alternate Destination ('dsta') Resource

The alternate destination ('dsta') resource, of type **gxDestinationAdditionType**, allows a driver or application to specify a destination to be added to the destination pop-up menu in the Print dialog. Figure 1 shows the structure of the alternate destination resource.

Figure 1 The alternate destination resource



destination name

The Pascal string name to show in the print dialog pop-up menu.

button text

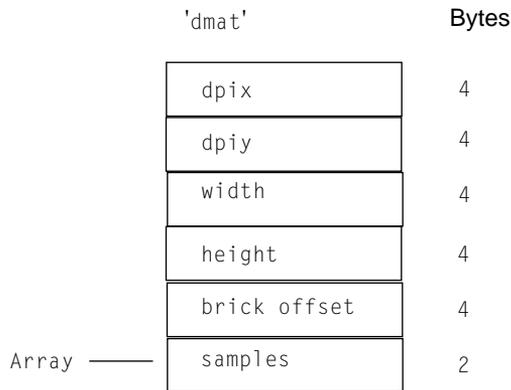
The Pascal string name to show in the "OK" button when the user selects the destination name from above in the pop-up menu.

The Custom Halftone Matrix ('dmat') Resource

The custom halftone matrix ('dmat') resource, of type **gxCustomMatrixType**, allows a driver or application to specify a custom halftone matrix to be used

rather than one of the several built-in halftone dot types provided by QuickDraw GX. Figure 2 shows the structure of a custom halftone matrix.

Figure 2 The custom halftone matrix resource



dpix The dots per inch in the x direction. This is a fixed point value.

dpiy The dots per inch in the y direction. This is a fixed point value.

width The width of the matrix.

height The height of the matrix.

brick offset The tile shift.

Each entry in the samples array is for a matrix that outlines a specific tile shift.

In order to make use of this resource, you must set the dither type of your 'rdip' resource to `gxCustomDot`.

For more information on the 'rdip' resource, please see *Inside Macintosh: QuickDraw GX Printing Extensions and Drivers*, p. 6-66.

The Manual Feed Preferences ('mfpr') Resource

The manual feed preferences ('mfpr') resource, of type `gxManualFeedAlertPrefsType`, allows the disabling of manual feed alerts on a desktop printer. This resource is optional.

Figure 3 shows the structure of the manual feed preferences resource.

Figure 3The manual feed preferences resource



`alertFlags` A collection of flag values that you can combine together to specify when manual feed alerts will be shown. The first word is for a driver's private use.

Table 2Flags used in the manual feed preferences resource

Constant	Value	Explanation
<code>gxShowAlerts</code>	0x01	Tells GX to show alerts for this desktop printer
<code>gxAlertOnPaperChange</code>	0x02	Tells GX to show alerts only if the paper type changes

The default settings if no resource is found is `gxDefaultMFeedAlertSettings` (`=gxShowAlerts | gxAlertOnPaperChange`).

SPECIAL CONSIDERATIONS

The 'mfpr' and 'outp' resources must be stored in the desktop printer by sending the `GXWriteDTPData` message. The desktop printer is the only place QuickDraw GX looks for these resources. A driver might store these resources when the desktop printer is created by overriding `GXDefaultDesktopPrinter`, forwarding the message, and then using `Send_GXWriteDTPData` to write the data.

For more information on `GXDefaultDesktopPrinter` and `GXWriteDTPData`, see *Inside Macintosh: QuickDraw GX Printing Extensions and Drivers*.

See the section "The Desktop Printer Output Characteristics ('outp') Resource" in this document for more information on 'outp'.

The Desktop Printer Output Characteristics ('outp') Resource

The Desktop Printer Output Characteristics ('outp') Resource, of type `gxDriverOutputType`, allows printer drivers to indicate the characteristics of a desktop printer's output device. Currently, this resource only indicates whether or not the desktop printer represents an output device that can be configurable to different paper types.

Figure 4 shows the structure of the desktop printer output characteristics resource.

Figure 4The desktop printer output characteristics resource

'outp'	Bytes
driverFlags	4
output settings	4

`driverFlags` Available for driver-defined options.

`outputSettings`

A collection of flag values that you can combine together to specify whether a printer is a device with a paper feed. Currently predefined with `gxCanConfigureTrays` or 0.

Table 3 Flags used in the manual feed preferences resource

Constant	Value	Explanation
<code>gxCanConfigureTrays</code>	0x01	Desktop printer represents a device with a paper feed

If no 'outp' resource is found in the desktop printers, QuickDraw GX assumes that the desktop printer is tray-configurable and will display the manual feed and paper mismatch alerts.

SPECIAL CONSIDERATIONS

The 'mfpr' and 'outp' resources must be stored in the desktop printer by sending the `GXWriteDTPData` message. The desktop printer is the only place

QuickDraw GX looks for these resources. A driver might store these resources when the desktop printer is created by overriding `GXDefaultDesktopPrinter`, forwarding the message, and then using `Send_GXWriteDTPData` to write the data.

For more information on `GXDefaultDesktopPrinter` and `GXWriteDTPData`, see *Inside Macintosh: QuickDraw GX Printing Extensions and Drivers*.

See the section "The Manual Feed Preferences ('mfpr') Resource" in this document for more information on 'mfpr'.

Change Default DTP 'pfpr' Apple Event

This Apple Event is used to change the current default desktop printer. In order to use it, you need to send an Apple Event of type 'pfpr' (not very mnemonic) to the printing extension (creator of 'pxtn') with the data of the event being the name of the desktop printer that you want to make the default. A good sample of this can be found on the Developer CD series: Tool Chest Edition under Tool Chest: QuickDraw GX: SetDefaultDTP.

▲ **WARNING**

'pfpr' is currently not a registered part of the Apple Event Suite. ▲

Addenda to QuickDraw GX Typography

GXGetLayoutJustificationGap

You can use the `GXGetLayoutJustificationGap` function to retrieve information about the amount of space QuickDraw GX tries to fill when justifying a line.

Fixed `GXGetLayoutJustificationGap(gxShape layout)`

`layout` A reference to the layout shape whose information you need.

function result The signed difference between the specified width for the layout and the measured (unjustified) width.

ERRORS

shape_is_nil

GXGetLayoutJustificationFactors

You can use the `GXGetLayoutJustificationFactors` function to retrieve information about how much space is added for different kinds of characters (such as kashida, whitespace, and non-whitespace) when a line is justified.

```
void GXGetLayoutJustificationFactors(gxShape layout, Fixed
                                   constrainedFactors[], Fixed unconstrainedFactors[])
```

`layout` A reference to the layout shape whose information you need.

`constrainedFactors`
An array of Fixed values. On return, the array contains factors that represent the portions of available padding by priority (`gxWhiteSpacePriority`,..) that the layout ended up using in order to justify the line.

`unconstrainedFactors`
An array of Fixed values. On return, the array contains factors that represent the characters that were designated as being able to absorb unlimited gap.

function result The signed difference between the specified width for the layout and the measured (unjustified) width.

ERRORS

shape_is_nil

GXGetFontDefaultFeatures

You can use the `GXGetFontDefaultFeatures` function to retrieve information about which layout features in a font happen by default.

```
long GXGetFontDefaultFeatures(gxFont fontID, gxRunFeature features[])
```

`fontID` A reference to the font whose information you need.

`features` On return, an array of the features for this font.

function result A list of the feature type/selector pairs that correspond to those features the font designates as default behaviors.

ERRORS

out_of_memory
internal_font_error
illegal_font_parameter

Acknowledgments

Thanks to Dave Bice, Tom Dowdy, Dave Hersey, Guillermo Ortiz, and Harita Patel.