

This chapter describes the contents and related data structures for each of the resources you need to define for printing extensions and printer drivers. This chapter consists primarily of reference information. For specific information on using each resource in the context of developing an extension or driver, see the “Printing Extensions” and “Printer Drivers” chapters in this book.

You need to read this chapter if you are developing a printing extension or a printer driver for use with QuickDraw GX. This chapter constitutes the complete reference guide to the resources you can use to develop these programs.

Before reading this chapter, you need to have a basic understanding of Macintosh resources and the Macintosh Resource Manager, which are described in *Inside Macintosh: More Macintosh Toolbox*.

This chapter begins with a brief overview of the resources that QuickDraw GX uses for printing and then describes

- the numbers that you can use to identify resources
- the attributes that you specify in every resource
- information about file type and creator type for a resource
- each resource that you use for your extensions and drivers, including a description of each field in each of the resources

About the Printing Resources

Each printing extension and printer driver that you develop consists of a number of resources that you must create and package together. Some of the resources are standard Macintosh resources for the icons and panels that are part of the user interface to your extension or driver. Other resources are used specifically for QuickDraw GX printing.

Some of the resources described here are used for both printing extensions and printer drivers, some are used only for printing extensions, and some are used only for printer drivers. You define some resources for a printing extension or printer driver that you develop for any imaging system, and you define others only if your extension or driver is designed for a specific imaging system. The available imaging systems are raster, PostScript, and vector.

These resources contain information such as

- code to override message
- user interface data, such as dialog boxes and icons
- data that defines how to manage color handling on a specific device
- data used to establish appropriate communications with a device
- strings displayed to a user during printing
- paper types and formats used for printing

Resource ID Numbering

You assign a unique identifier, or resource ID, to each resource that you use in a Macintosh program. The printing resources that you define for printing extensions need to be in a certain range: from 0x9600 (-27136) to 0x97FF (-26625).

The printing resources that you define for printer drivers need to be in a certain range: from 0x9400 (-27648) to 0x95FF (-27137). For most of the printing-specific resources in your drivers, you define the ID as a value added to a constant named `gxPrintingDriverBaseID` (-28672), as in this example:

```
resource gxOverrideType (gxPrintingDriverBaseID+1, sysHeap,
                        purgeable)
{
    {
        gxRasterPackageBitmap, segmentID, firstOffset+40,
        gxRasterLineFeed,      segmentID, firstOffset+44
    };
};
```

QuickDraw GX defines constants for the resource ID of many of the resources that you define for your extensions and drivers. These constants are described in the reference description for each resource.

Resource Attributes

All of the resources that you define for your printing extensions and printer drivers need to be loaded into the system heap and need to be purgeable. System resources are stored in the system heap as opposed to the application heap, where application resources are stored. Purgeable resources can be purged by the Memory Manager when space is required, as described in *Inside Macintosh: Memory*.

You need to specify these attributes in the first line of every resource that you define for your extensions and drivers, as in this example:

```
resource gxOverrideType (gxPrintingDriverBaseID+1, sysHeap,
                        purgeable)
{
};

resource statusType (kDriverStatus, sysHeap, purgeable)
{
};
```

▲ WARNING

You need to store your resources in the system heap to allow the memory that they use to be shared when multiple copies of your driver are active at the same time. You must, however, take care to not release a resource that is in use by another copy of your driver. ▲

Extension and Driver Resource Files

Each printing extension and printer driver is implemented in a file of resources. Code segments of a printing extension must have the file type `'pext'`, and code segments of a printer driver must have the file type `'pdvr'`. If the file does not have the correct resource type, QuickDraw GX cannot recognize it as an extension or driver.

The creator type of each printing extension must be unique. The only way to guarantee uniqueness is to register the creator type with Macintosh Developer Technical Support. This is highly recommended because QuickDraw GX relies on the uniqueness of the creator type.

Printing Resources Reference

This section describes the data structures and printing resources that are specific to printing extension and printer driver files.

The “Constants and Data Types” section shows enumerations and structures for some of the resources.

The remaining sections describe the resources used in printing extensions and printer drivers—resources used in both, those used only in printing extensions, and those used only in printer drivers. Most of the resources used for printing extensions and printer drivers have resource templates defined for them. These templates are shown in the illustration that accompanies the section on each resource in this chapter.

Almost all of the fields in the resources are simple, unstructured values. Numerous fields use constant values with specific meanings; the constants that can be used in these fields are shown in the tables that accompany the field descriptions in this section.

The resources that you define for printing extensions and printer drivers are presented in three sections:

- the resources that you use for printing extensions and printer drivers
- the resources that you use only for printing extensions
- the resources that you use only for printer drivers

Some resources required for extensions and drivers are described in other *Inside Macintosh* books. For information on the standard version (`'vers'`), bundle (`'BNDL'`), file reference (`'FREF'`), and various icon resources, all of which are required resources, see *Inside Macintosh: Macintosh Toolbox Essentials*. For information on required resources that define the Chooser interface for your driver, see *Inside Macintosh: Devices*.

Printing Resources

If you want to provide a user interface to your printing extension or printer driver, see information on the necessary resources to do this in *Inside Macintosh: QuickDraw GX Printing*.

Constants and Data Types

QuickDraw GX defines structures and enumerations for some of the printing resources. These data types are defined in this section.

The Buffering and Input/Output Preferences Structure

The buffering and input/output preferences structure, of data type `gxIOPrefsRec`, is used to specify the contents of the buffering and input/output preferences ('iobm') resource.

```
struct gxIOPrefsRec {
    unsigned long  communicationsOptions;
    unsigned long  numBuffers;
    unsigned long  bufferSize;
    unsigned long  numReqBlocks;
    unsigned long  openCloseTimeout;
    unsigned long  readWriteTimeout
};

typedef struct gxIOPrefsRec gxIOPrefsRec, *gxIOPrefsPtr,
**gxIOPrefsHdl;
```

Field descriptions

<code>communicationsOptions</code>	The options for how a driver handles input and output communications. The value of this field is 0 for standard I/O mechanisms and <code>gxUseCustomIO</code> (1) for nonstandard mechanisms, including SCSI.
<code>numBuffers</code>	The number of buffers created for the driver.
<code>bufferSize</code>	The number of bytes in each buffer.
<code>numReqBlocks</code>	The maximum number of input or output requests that can be pending at any time.
<code>openCloseTimeout</code>	The number of clock ticks that constitute a timeout when trying to open or close the device.
<code>readWriteTimeout</code>	The number of clock ticks that constitute a timeout when trying to read from or write to the device.

The Customization Structure

The customization structure, of data type `gxCustomizationRec`, defines the format of the customization ('cust') resource.

```
struct gxCustomizationRec {
    short horizontalResolution;
    short verticalResolution;
    short upDriverType;
    Point patternStretch;
    short translatorSettings
};

typedef struct gxCustomizationRec gxCustomizationRec,
*gxCustomizationPtr, **gxCustomizationHdl;
```

Field descriptions

`horizontalResolution`

The horizontal resolution to use for the device in dots per inch.

`verticalResolution`

The vertical resolution to use for the device in dots per inch.

`upDriverType`

The Macintosh Printing Manager interface driver with which your device is compatible. The values for this field are shown in Table 6-20 on page 6-48.

`patternStretch`

The scaling factor for printing bitmap patterns.

`translatorSettings`

Settings for translating Macintosh Printing Manager driver calls into messages for your driver. The values for this field are shown in Table 6-21 on page 6-48.

The Resolution Structure

The resolution structure, of data type `gxResolutionRec`, defines the format of the resolution ('resl') resource.

```
struct gxResolutionRec {
    short    rangeType;
    short    xMinimumResolution;
    short    xMaximumResolution;
    short    yMinimumResolution;
    short    yMaximumResolution;
    short    resolutionCount;
    Point    resolutions[1];
};

typedef struct gxResolutionRec gxResolutionRec, *gxResolutionPtr,
**gxResolutionHdl;
```

Printing Resources

Field descriptions

<code>rangeType</code>	The type of resolution range being defined. This value is currently always 1.
<code>xMinimumResolution</code>	The minimum horizontal resolution supported by the driver in dots per inch (dpi).
<code>xMaximumResolution</code>	The maximum horizontal resolution supported by the driver in dpi.
<code>yMinimumResolution</code>	The minimum vertical resolution supported by the driver in dpi.
<code>yMaximumResolution</code>	The maximum vertical resolution supported by the driver in dpi.
<code>resolutionCount</code>	The number of entries in the <code>resolutions</code> array.
<code>resolutions</code>	An array of points, each of which defines a printing resolution, in dots per inch, supported by the driver. The x value of each point defines the horizontal resolution, and the y value of each point defines the vertical resolution.

Raster Preferences Structure

The raster preferences structure, of type `gxRasterPrefsRec`, defines the format of the raster preferences ('rdip') resource.

```

struct  gxRasterPrefsRec{
    gxRasterRenderOptions
        renderOptions;
    Fixed    hImageRes;
    Fixed    vImageRes;
    short    minBandSize;
    short    maxBandSize;
    Fixed    ramPercentage;
    long     ramSlop;
    short    depth;
    short    numPlanes;
    gxPlaneSetupRec
        planeSetup[1];
};

typedef struct gxRasterPrefsRec gxRasterPrefsRec,
*gxRasterPrefsPtr, **gxRasterPrefsHdl;

```

Printing Resources

Field descriptions

<code>renderOptions</code>	Rendering options for raster imaging. The constants that you can combine into a single value for this field are the values of the raster render options enumeration, described in the next section.
<code>hImageRes</code>	The horizontal resolution for imaging.
<code>vImageRes</code>	The vertical resolution for imaging.
<code>minBandSize</code>	The minimum band size to use, in pixels.
<code>maxBandSize</code>	The maximum band size to use, in pixels. A value of 0 in this field indicates that the maximum band size is the full page.
<code>ramPercentage</code>	The maximum percentage of available RAM to use.
<code>ramSlop</code>	The minimum amount of RAM to leave available.
<code>depth</code>	The depth, in pixels, for each plane.
<code>numPlanes</code>	The number of planes.
<code>planeSetup</code>	An array of structures containing the setup data for each plane.

Raster Render Options

The raster render options enumeration defines constants that you can use in the `renderOptions` field of the raster preferences structure, which is described in the previous section.

```
enum {
    gxDefaultRaster      = 0x00000000,
    gxDontResolveTransferModes
                        = 0x00000001,
    gxRenderInReverse    = 0x00000002,
    gxOnePlaneAtATime    = 0x00000004,
    gxSendAllBands       = 0x00000008
};

typedef long gxRasterRenderOptions;
```

Constant descriptions

<code>gxDefaultRaster</code>	The driver uses the default raster options.
<code>gxDontResolveTransferModes</code>	If this is included, the driver does not need to resolve transfer modes. The default is to resolve transfer modes.
<code>gxRenderInReverse</code>	The driver needs to traverse the raster image in reverse order.
<code>gxOnePlaneAtATime</code>	The driver needs to render each plane separately.
<code>gxSendAllBands</code>	The driver needs to send every band of data, even if it is all empty.

Raster Package Structure

The raster package structure, of data type `gxRasterPackageRec`, defines the format of the raster package ('rpck') resource.

```
struct gxRasterPackageRec {
    Ptr      bufferSize;
    short     colorPasses;
    short     headHeight;
    short     numberPasses;
    short     passOffset;
    gxRasterPackgeOptions
            packageOptions;
};

typedef struct gxRasterPackageRec gxRasterPackageRec,
*gxRasterPackagePtr, **gxRasterPackageHdl;
```

Field descriptions

<code>bufferSize</code>	The buffer size for packaging. This value must be greater than or equal to the maximum head-pass size.
<code>colorPasses</code>	The number of color passes. This value is typically 1 for monochrome printers and 4 for CMYK printers.
<code>headHeight</code>	The height of the print head, in pixels.
<code>numberPasses</code>	The number of passes it takes to print one print head of data.
<code>passOffset</code>	The offset between passes, in pixels.
<code>packageOptions</code>	The packaging options. The constants that you can use in this field are the values of the raster package options enumeration, described in the next section. You can combine several of the constants into one value for this field.

Raster Package Options

The raster package options enumeration defines constants that you can use in the `packageOptions` field of the raster package structure, which is described in the previous section.

```
enum {
    gxSendAllColors    = 0x00000001,
    gxInterlaceColor   = 0x00000002,
    gxOverlayColor     = 0x00000004,
    gxUseColor         = (gxInterlaceColor|gxOverlayColor);
};

typedef long gxRasterPackageOptions;
```


Printing Resources

Constant descriptions`gxSendAllColors`

The driver needs to send all bands of data, even when the entire band is empty.

`gxInterlaceColor`

The driver needs to interlace colors because ribbon contamination is a concern on this color printer. If you use this option, you might need to use negative line feed values.

`gxOverlayColor`

The driver does not need to interlace colors because ribbon contamination is not a concern on this printer.

`gxUseColor`

This is a color printer.

Raster Package Controls Structure

The raster package controls structure, of data type `gxRasterPackageControlsRec`, defines the format of the raster package controls ('ropt') resource.

```
struct gxRasterPackageControlsRec {
    short    startPageStringID;
    short    formFeedStringID;
    short    forwardMax;
    gxStandardNumberRec
            forwardLineFeed;
    short    reverseMax;
    gxStandardNumberRec
            reverseLineFeed;
};

typedef struct gxRasterPackageControlsRec
gxRasterPackageControlsRec, *gxRasterPackageControlsPtr,
**gxRasterPackageControlsHdl;
```

Field descriptions`startPageStringID`

The ID of the wide string ('wstr') to send to the device at the start of each page. Wide strings are strings that require a 16-bit, short integer value to define their length. The length value is stored in the first 2 bytes of the string.

`formFeedStringID`

The ID of the wide string ('wstr') to send to the device to generate a form feed.

`forwardMax`

The maximum amount of a forward line feed.

`forwardLineFeed`

The standard number structure that defines how to express the forward line-feed value.

Printing Resources

<code>reverseMax</code>	The maximum amount of a reverse line feed.
<code>reverseLineFeed</code>	The standard number structure that defines how to express the reverse line-feed value.

Standard Number Structure

The standard numbering structure, of data type `StandardNumberRec`, defines how to output numbers. The raster package controls structure contains two of these structures.

```
struct gxStandardNumberRec {
    short    numberType;
    short    minWidth;
    char     padChar;
    char     alignment;
    Str31    startString;
    Str31    endString;
};
```

```
typedef struct gxStandardNumberRec gxStandardNumberRec,
*gxStandardNumberPtr;
```

Field descriptions

<code>numberType</code>	The type of numeric output to be used. This is one of the types shown in Table 6-33 on page 6-76.
<code>minWidth</code>	If you are using the <code>RasterNumToASCII</code> number type, this is the minimum number of characters in the output number string. If you are using the <code>RasterNumDirect</code> number type, this is the minimum number of bytes in the output number string.
<code>padChar</code>	The character used to pad (in front) numbers that are shorter than the minimum width value.
<code>alignment</code>	A single byte used for alignment of data in this structure.
<code>startString</code>	The string sent in front of the number string.
<code>endString</code>	The string sent after the number string.

Resources Used for Printing Extensions and Printer Drivers

Six resources are used for both printing extensions and printer drivers and are described in this section.

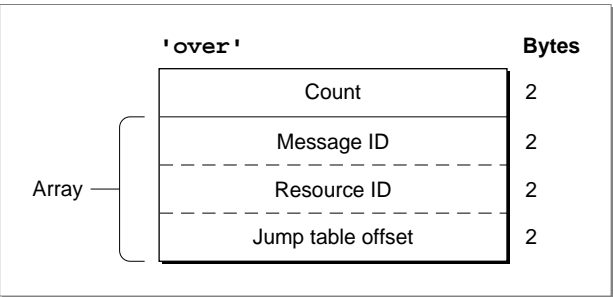
- The `override` ('over') resource defines which of the printing messages your printing extension or printer driver is overriding. All printing extensions and printer drivers must have this resource.

- The version ('vers') resource defines with which version of QuickDraw GX your printing extension or printer driver is compatible. All printing extensions and printer drivers must have at least one version resource that is used for this purpose.
- The status ('stat') resource defines status messages for display to the user in the desktop printer window during the printing of a job.
- The printing alert ('plrt') resource defines alert messages that are displayed in printing alert boxes when a user's attention is required during the printing of a job.
- The tray count ('tray') resource specifies how many paper trays your printing extension or printer driver supports.
- A tray name ('tryn') resource specifies the name of each paper tray.

The Override ('over') Resource

You must provide at least one override resource, of type `gxOverrideType`, for any printing extension or printer driver that you develop. This resource provides QuickDraw GX with a list of the messages that you are overriding in your extension or driver, along with the ID of the resource in which to find the code for your override's implementation. Figure 6-1 shows the structure of an override resource.

Figure 6-1 The override resource



The override resource consists of a variable number of records, each of which specifies a message, a resource type, a resource ID, and an offset value. Each entry tells QuickDraw GX the name of a printing message that you are overriding and where to find the code for your override.

- Count. The number of message entries in the resource.
- Message ID. The ID of the message that you are overriding. QuickDraw GX defines an integer constant that you use here for each message. These constants are shown in Table 6-2 on page 6-15.
- Resource ID. The ID of the resource in which the code for your override can be found.

Printing Resources

- **Jump table offset.** The offset into the jump table . This is the number of bytes from the beginning of the jump table to the jump instruction for your override code. The first 4 bytes in the jump table must be reserved for use by QuickDraw GX, so your first offset must be 4. You can read about the jump table in the chapter “Printing Extensions” in this book.

You need to provide separate override resources for different kinds of printing messages. There are three kinds of messages:

- **Universal messages,** used for all imaging systems.
- **Imaging system messages,** used for specific imaging systems. Some messages are raster imaging messages, some messages are PostScript imaging messages, and some are vector imaging messages.
- **Macintosh Printing Manager messages,** used to provide compatibility with the Macintosh Printing Manager interface for printing. Printing extensions cannot override Macintosh Printing Manager messages; if an extension does specify that it is overriding a Macintosh Printing Manager call, that message override is ignored.

The ID of each override resource indicates which kind of messages are specified in the resource. Different values are used in override resources for printing extensions than are used for printer drivers, as shown in Table 6-1.

Table 6-1 Override resource IDs

Message type	Extension resource ID	Driver resource ID
Universal	gxExtensionUniversalOverrideID	gxDriverUniversalOverrideID
Imaging system	0	gxDriverImagingOverrideID
Macintosh Printing Manager	Not allowed	gxDriverCompatibilityOverrideID

Imaging system messages are actually separated into categories: messages for the raster imaging system, PostScript imaging system, and vector imaging system. The identifier used for each message type is shown in Table 6-2.

Printing Resources

Table 6-2 Printing message constants

Message type	Constant	Value
Universal	gxInitialize	0
	gxShutDown	1
	gxJobIdle	2
	gxJobStatus	3
	gxPrintingEvent	4
	gxJobFormatDialog	5
	gxFormatDialog	6
	gxJobPrintDialog	7
	gxFilterPanelEvent	8
	gxHandlePanelEvent	9
	gxParsePageRange	10
	gxDefaultJob	11
	gxDefaultFormat	12
	gxDefaultPaperType	13
	gxDefaultPrinter	14
	gxCreateSpoolFile	15
	gxSpoolPage	16
	gxSpoolData	17
	gxSpoolResource	18
	gxCompleteSpoolFile	19
	gxCountPages	20
	gxDespoolPage	21
	gxDespoolData	22
	gxDespoolResource	23
	gxCloseSpoolFile	24
	gxStartJob	25
	gxFinishJob	26
	gxStartPage	27
	gxFinishPage	28
	gxPrintPage	29
	gxSetupImageData	30
	gxImageJob	31
	gxImageDocument	32
	gxImagePage	33
	gxRenderPage	34
	gxCreateImageFile	35
	gxOpenConnection	36
	gxCloseConnection	37
	gxStartSendPage	38
	gxFinishSendPage	39
	gxWriteData	40
	gxBufferData	41
	gxDumpBuffer	42
	gxFreeBuffer	43

continued

Printing Resources

Table 6-2 Printing message constants (continued)

Message type	Constant	Value
	gxCheckStatus	44
	gxGetDeviceStatus	45
	gxFetchTaggedData	46
	gxGetDTPMenuList	47
	gxDTPMenuSelect	48
	gxHandleAlertFilter	49
	gxJobFormatModeQuery	50
	gxWriteStatusToDTPWindow	51
	gxInitializeStatusAlert	52
	gxHandleAlertStatus	53
	gxHandleAlertEvent	54
	gxCleanupStartJob	55
	gxCleanupStartPage	56
	gxCleanupOpenConnection	57
	gxCleanupStartSendPage	58
	gxDefaultDesktopPrinter	59
	gxCaptureOutputDevice	60
	gxOpenConnectionRetry	61
	gxExamineSpoolFile	62
	gxFinishSendPlane	63
	gxDoesPaperFit	64
	gxChooserMessage	65
	gxFindPrinterProfile	66
	gxFindFormatProfile	67
	gxSetPrinterProfile	68
	gxSetFormatProfile	69
Macintosh Printing Manager	gxPrOpenDoc	0
	gxPrCloseDoc	1
	gxPrOpenPage	2
	gxPrClosePage	3
	gxPrintDefault	4
	gxPrStlDialog	5
	gxPrJobDialog	6
	gxPrStlInit	7
	gxPrJobInit	8
	gxPrDlgMain	9
	gxPrValidate	10
	gxPrJobMerge	11
	gxPrGeneral	12
	gxConvertPrintRecordTo	13
	gxConvertPrintRecordFrom	14
	gxPrintRecordToJob	15
Raster imaging system	gxRasterDataIn	0
	gxRasterLineFeed	1
	gxRasterPackageBitmap	2

Table 6-2 Printing message constants (continued)

Message type	Constant	Value
PostScript imaging system	gxPostscriptQueryPrinter	0
	gxPostscriptInitializePrinter	1
	gxPostscriptResetPrinter	2
	gxPostscriptExitServer	3
	gxPostscriptGetStatusText	4
	gxPostscriptGetPrinterText	5
	gxPostscriptScanStatusText	6
	gxPostscriptScanPrinterText	7
	gxPostscriptGetDocumentProcSetList	8
	gxPostscriptDownloadProcSetList	9
	gxPostscriptGetPrinterGlyphsInformation	10
	gxPostscriptStreamFont	11
	gxPostscriptDoDocumentHeader	12
	gxPostscriptDoDocumentSetUp	13
	gxPostscriptDoDocumentTrailer	14
	gxPostscriptDoPageSetUp	15
	gxPostscriptSelectPaperType	16
	gxPostscriptDoPageTrailer	17
	gxPostscriptEjectPage	18
	gxPostscriptProcessShape	19
Vector imaging system	gxVectorPackageShape	0
	gxVectorLoadPens	1
	gxVectorVectorizeShape	2

Listing 6-1 shows a typical override resource definition. You can also find examples of override resource definitions in Listing 2-24 on page 2-38 in the chapter “Printing Extensions” and in Listing 3-20 on page 3-56 in the chapter “Printer Drivers.”

Listing 6-1 An example of an override resource

```
#define mySegment 0
#define firstOffset 4

resource gxOverrideType (gxExtensionUniversalOverrideID, sysHeap,
                        purgeable)
{
    {
        /* message to override      segmentID      jump table offset */

        gxInitialize,              mySegment,      firstOffset+0,
        gxShutDown,                mySegment,      firstOffset+4,
        gxJobPrintDialog,          mySegment,      firstOffset+8,
        gxHandlePanelEvent,        mySegment,      firstOffset+12,
```

Printing Resources

```

        gxDespoolPage,          mySegment,      firstOffset+16
    };
};

```

This resource specifies the universal messages that an extension is overriding. In this case, five messages are being overridden. Each of these messages is found in a code segment with a resource ID of `mySegment`. The jump ('JMP') statement for the `gxInitialize` message is found at the first offset (4 bytes) from the start of the jump table, and the jump statements for the subsequent messages are each located 4 bytes apart.

IMPORTANT

Each jump table entry is 4 bytes long, and QuickDraw GX reserves the first 4 bytes in the table for its own use. This means that your first jump table entry must be located at an offset of 4 bytes from the beginning of the table. For this reason, the `firstOffset` constant in Listing 6-1 is defined with a value of 4. ▲

The Version ('vers') Resource

You need to include a version resource that defines with which version of QuickDraw GX your printing extension or printer driver is compatible. In the resource files for your printing extension, you can also include additional standard version resources, as described in *Inside Macintosh: Macintosh Toolbox Essentials*.

For printing extensions, you need to include a version resource with an ID of `gxPrintingExtensionBaseID` that defines with which version of QuickDraw GX your extension is compatible. For printer drivers, you need to include a version resource with an ID of `gxPrintingDriverBaseID` that defines with which version of QuickDraw GX your driver is compatible.

The first byte defines the QuickDraw GX version. For the current version of QuickDraw GX, the first byte must have a value of either 1 or 0. Listing 6-2 shows the version resources that define QuickDraw GX compatibility for the background picture printing extension and for the ImageWriter II printer driver.

Listing 6-2 QuickDraw GX version resources

```

resource 'vers' (gxPrintingExtensionBaseID, sysHeap, purgeable) {
    0x0,
    0x0,
    release,
    0x0,
    verUS,
    " ",
    " "
};

```


Printing Resources

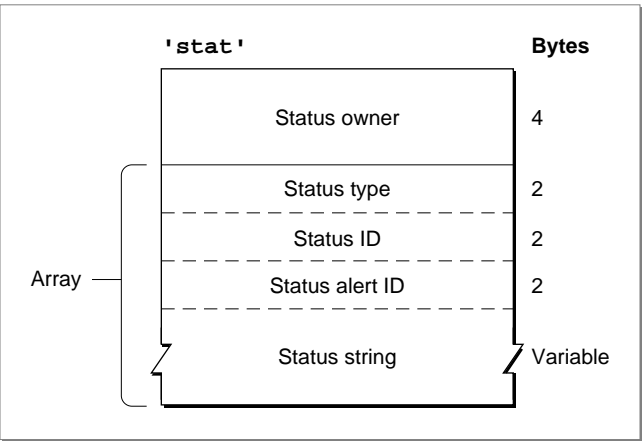
```
resource 'vers' (gxPrintingDriverBaseID, sysHeap, purgeable)
{
    0x01, 0x00, release, 0x00,
    verUS,
    "1.00",
    "1.00, Copyright \251 Apple Computer, Inc. 1989-1993"
};
```

The Status ('stat') Resource

You need to include a status resource, of type `statusType`, to define the status messages that are displayed during the printing process. The driver forces status messages to be displayed by calling either the `GXReportStatus` function (for simple messages) or the `GXAlertTheUser` function (for messages that require user attention).

Figure 6-2 shows the structure of the status resource.

Figure 6-2 The status resource



The status resource contains a count of the status entries and an array of status definitions.

- **Status owner.** The signature of the printing extension or printer driver to which this status resource belongs.

Printing Resources

Each status definition contains four values:

- **Status type.** The kind of status message that this is. The status type constants are shown in Table 6-3.
- **Status ID.** The ID of this status message within the status resource. You typically assign sequential numbers to the status messages within each status resource, as shown in the example at the end of this section.
- **Status alert ID.** The ID of the printing alert associated with this status message. Use the ID 0 to indicate that this status message does not require a printing alert.
- **Status string.** The status message string to display to the user.

Most of the status types produce side effects. For example, if you send a status message with status type `gxSpoolingPageStatus`, the page count is incremented in the spooling status that is displayed on the user's screen. Table 6-3 shows the status type constants and the side effects associated with each.

Table 6-3 Status types

Constant	Value	Explanation of side effects
<code>gxNonFatalError</code>	1	Affects the icon that is displayed during spooling
<code>gxFatalError</code>	2	Displays a printing alert during spooling
<code>gxPrinterReady</code>	3	Signals that alert mode is done
<code>gxUserAttention</code>	4	Signals initiation of a modal alert
<code>gxUserAlert</code>	5	Signals initiation of a printing alert
<code>gxPageTransmission</code>	6	Signals that a page has been sent to the printer and increments the printed page count
<code>gxOpenConnectionStatus</code>	7	Signals that animation of the printer icon is to begin
<code>gxInformationalStatus</code>	8	Displays an informational status message with no side effects
<code>gxSpoolingPageStatus</code>	9	Signals that a page has been spooled and increments the spooled page count
<code>gxEndStatus</code>	10	Signals the end of spooling
<code>gxPercentageStatus</code>	11	Signals the percentage of the current print job that is currently complete

Listing 6-3 shows an example of a status resource.

Listing 6-3 An example of a status resource

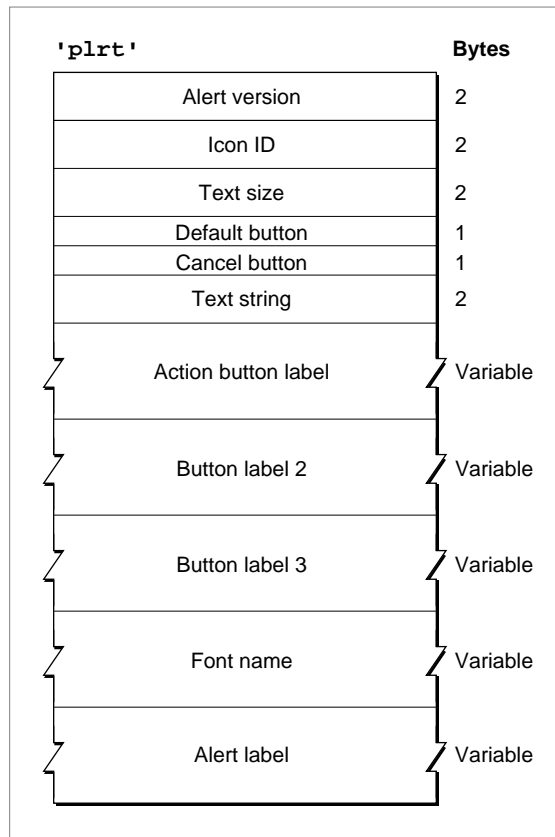
```
#define kDrvrCreatorType 'IWII'

resource statusType (kDriverStatus, sysHeap, purgeable)
{
    kDrvrCreatorType,
    {
        gxInformationalStatus, 1, 0, "Sending data to printer";
        gxUserAlert, 1, kDriverStatus,
            "Please check that the printer is on-line";
    };
};
```

Listing 6-3 defines a status resource for the ImageWriter II printer driver. This resource defines two status text strings. The first text string is an informational message that does not require a printing alert box and is displayed in the desktop printer window. The second text string requires user attention and is displayed in a printing alert box.

The Printing Alert ('plrt') Resource

You need to include a printing alert resource to define the messages that require user attention during the printing process. Figure 6-3 shows the structure of the printing alert resource.

Figure 6-3 The printing alert resource

The printing alert resource contains a number of fields that you can use to define alert messages for display to the user in printing alert boxes.

- **Alert version.** The type of printing alert that this resource represents. There are currently 2 versions, as shown in Table 6-4.
- **Icon ID.** The type of icon to display in the upper-left corner of the alert. The icon types are shown in Table 6-5.
- **Text size.** The size of the text to display in the alert. You can use the value `defaultSystemSize` to indicate that the default text size for the system is to be used.
- **Default button.** The string to display on the default button. You use one of the values shown in Table 6-7.
- **Cancel button.** The string to display on the cancel button. You use one of the values shown in Table 6-7.
- **Text string.** The string to display in the alert window.

Printing Resources

- Action button label. The string that is displayed on the action button. This is a required value.
- Button label 2. The string that is displayed on button 2. This is an optional value.
- Button label 3. The string that is displayed on button 3. This is an optional value.
- Font name. The name of the font used to display the text in this alert.
- Alert title. The title displayed for this alert.

Table 6-4 shows the constants that you can use to specify which kind of printing alert you are defining.

Table 6-4 Printing alert versions

Constant	Value	Explanation
printingAlert	1	Produces a modal alert dialog box
printingStatus	2	Produces a printing alert box that can only be used with QuickDraw GX printing

Table 6-5 shows the constants that you can use to specify the icon ID for a printing alert resource.

Table 6-5 Icon IDs for a printing alert resource

Constant	Value	Explanation
noIcon	-1	No icon is displayed
stopIcon	0	The stop icon is displayed
noteIcon	1	The note icon is displayed
cautionIcon	2	The caution icon is displayed

Table 6-7 shows the constants that you can use to specify the default button strings in a printing alert resource.

Table 6-6 Default button string values for a printing alert resource

Constant	Value	Explanation
noDefaultTitle	0	There is no default button
defaultAction	1	Use the action button label string
defaultTitle2	2	Use the button label 2 string
defaultTitle3	3	Use the button label 3 string

Printing Resources

Table 6-7 shows the constants that you can use to specify the cancel button strings in a printing alert resource.

Table 6-7 Cancel button string values for a printing alert resource

Constant	Value	Explanation
noCancelTitle	0	There is no cancel button
cancelAction	1	Use the action button label string
cancelTitle2	2	Use the button label 2 string
cancelTitle3	3	Use the button label 3 string

Listing 6-4 shows an example of a printing alert resource.

Listing 6-4 An example of a printing alert resource

```
resource 'plrt' (kDriverStatus, sysHeap, purgeable)
{
    printingStatus,
    cautionIcon,
    defaultSystemSize,
    defaultAction,
    noCancelTitle,
    "The document \"!1\" cannot be printed, because the printer \"
    \"!0\" is offline. To continue printing, please make \"
    \"sure the printer is \"
    \"properly connected and turned on. If you wish \"
    \"to cancel printing, please click Cancel Printing.\",
    "Cancel Printing",
    "\"",
    "\"",
    "\" ",
    "Printer offline"
};
```

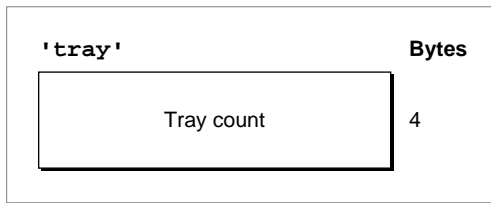
This resource defines the contents of the printing alert box that corresponds to the second entry in the status resource that is shown in Listing 6-3 on page 6-21. The printing alert box contains a caution icon and a Cancel button. The text is displayed using the default system font and text size.

The Tray Count ('tray') Resource

You need to include a tray count resource, of type `gxTrayCountDataType`, if you support named paper trays in your printing extension or printer driver. You use the tray count resource to specify how many paper trays there are on the output device, and you provide a tray name resource (as described in the next section) for each of the trays.

Figure 6-4 shows the structure of the tray count resource.

Figure 6-4 The tray count resource



The tray count resource contains a single value.

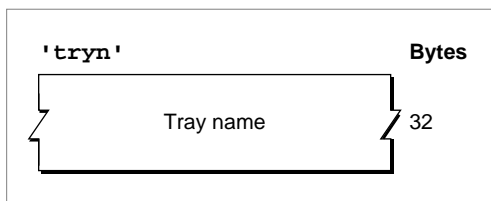
- **Tray count.** The number of paper trays that your printing extension or printer driver supports on the output device. This is a long integer value.

The Tray Name ('tryn') Resource

You need to include a tray name resource, of type `gxTrayNameDataType`, for each paper tray that your printing extension or printer driver is supporting. You must specify the number of the paper tray as the ID of the resource.

Figure 6-5 shows the structure of the tray name resource.

Figure 6-5 The tray name resource



The tray name resource contains a single value.

- **Tray name.** The name of the paper tray. This is a Pascal string with a length byte and 31 characters.

Resources Used Only for Printing Extensions

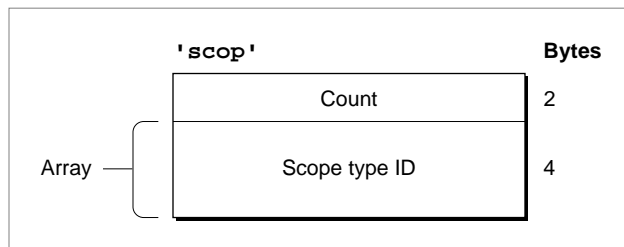
Some resources are used for printing extensions, but not for drivers. These resources are described in this section. They are required for all printing extensions.

- The extension scope ('scop') resource specifies with which imaging systems and drivers your extension is compatible.
- The extension load ('load') resource tells QuickDraw GX where to load your extension in the print message handler chain.
- The extension optimization ('eopt') resource provides QuickDraw GX with information about when your extension needs to be in memory, which allows QuickDraw GX to optimize printing performance.

The Extension Scope ('scop') Resource

You must define at least one extension scope resource, of type `gxExtensionScopeType`, for each printing extension that you develop. This resource indicates the types of drivers and imaging systems with which your extension is compatible. Figure 6-6 shows the structure of an extension scope resource.

Figure 6-6 The extension scope resource



The extension scope resource consists of a variable number of scope values.

- **Count.** The number of scope types defined in this resource.

- Scope type ID. A long integer value that specifies an imaging system or driver. These values are specified as four-character resource-type names in your resource definition. Constants that you can use for the resource types are shown in Table 6-8.

Table 6-8 Imaging system identifiers for the extension scope resource

Constant	Value	Explanation
gxAnyPrinterType	'univ'	All output devices
gxRasterPrinterType	'rast'	Raster output devices
gxPostscriptPrinterType	'post'	PostScript output devices
gxVectorPrinterType	'vect'	Vector output devices

The ID of each extension scope resource defines the type of information that is contained in the resource values. Constants for the scope resource IDs are shown in Table 6-9.

Table 6-9 Scope resource identifiers

Constant	Value	Explanation
gxDriverScopeID	gxPrintingExtensionBaseID	The imaging system types supported by the extension
gxPrinterScopeID	gxPrintingExtensionBaseID+1	Specific driver types supported by the extension
gxPrinterExceptionScopeID	gxPrintingExtensionBaseID+2	Specific driver types not supported by the extension

You can include several different extension scope resources in your extension file to pinpoint the scope of your extension. The values in all scope resources are unioned together to produce the exact scope of your extension. A typical pair of scope resources for an extension is shown in Listing 6-5. You can find other examples of scope resources in Listing 2-21 on page 2-35 and Listing 2-22 on page 2-36 in the chapter “Printing Extensions.”

Listing 6-5 An example of a pair of extension resources used together

```

resource gxExtensionScopeType (gxPrinterScopeID, sysHeap,
                                purgeable)
{
    {
        gxPostscriptPrinterType,
        gxRasterPrinterType
    };
};

resource gxExtensionScopeType (gxPrinterExceptionScopeID, sysHeap,
                                purgeable)
{
    {
        'ODD1 '
    };
};

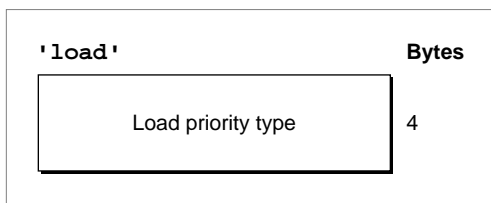
```

In this example, the first extension scope resource defines the extension as compatible with all PostScript and raster printers. The second resource specifies that the extension is not compatible with the printer driver name 'ODD1'. The net result is that the extension is compatible with all PostScript and raster printers that do not use the 'ODD1' driver.

The Extension Load ('load') Resource

The extension load resource, of type `gxExtensionLoadType`, tells QuickDraw GX where to load an extension into the printing message chain. You must include a load resource in your extension file. The value that you define in your extension load resource specifies the default loading order for your extension when it is first added to the system. The user can modify the loading order of extensions by using the Extension Setup dialog box for each desktop printer.

Figure 6-7 shows the structure of an extension load resource.

Figure 6-7 The extension load resource

The extension load resource consists of a single value.

- Load priority type. The default location in the message chain where your extension is to be loaded. Constants for the three load priority types are shown in Table 6-10.

Table 6-10 Extension load priority constants

Constant	Value	Explanation
<code>gxExtensionLoadFirst</code>	<code>0x00000100</code>	Load this extension as the first handler in the print message handling chain so that it has the first opportunity to respond to messages that it overrides
<code>gxExtensionLoadAnywhere</code>	<code>0x7FFFFFFF</code>	Load this extension anywhere in the print message handling chain
<code>gxExtensionLoadLast</code>	<code>0xFFFFFFFF00</code>	Load this extension last in the print message handling chain so that it receives its message after any other handlers have performed their operations

The extension load resource ID needs to be the constant `gxExtensionLoadID`. Listing 6-6 shows an example of an extension load resource.

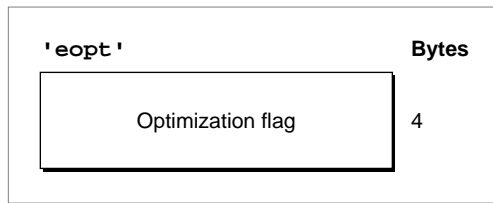
Listing 6-6 An example of an extension load resource

```
resource gxExtensionLoadType (gxExtensionLoadID, sysHeap,
                                purgeable)
{
    gxExtensionLoadLast
};
```

This resource tells QuickDraw GX to load the extension last in the message chain, which allows the extension to modify the contents of a page after any other message handlers have performed their modifications. However, the user can change this by using the Extension Setup dialog box for each desktop printer.

The Extension Optimization ('eopt') Resource

The extension optimization resource, of type `gxExtensionOptimizationType`, provides QuickDraw GX with additional information about the activities of an extension. The information in this resource allows QuickDraw GX to optimize its use. You must include an extension optimization resource in each printing extension. Figure 6-8 shows the structure of an extension optimization resource.

Figure 6-8 The extension optimization resource

- **Optimization flag.** The flags that define when your extension needs to be loaded into memory. Each extension optimization resource contains one or more optimization flags. The constants for these flags are shown in Table 6-11. The flags actually come in pairs: one that specifies that an extension needs to be activated during a certain phase of printing, and a complementary one that specifies that the extension does not need to be active during that phase. QuickDraw GX defines each flag as a Boolean value.

Table 6-11 Extension optimization resource flags

Constant	Value	Explanation
<code>gxExecuteDuringImaging</code>	<code>true</code>	Use if your extension overrides any messages that are sent during the imaging phase of printing.
<code>gxDontExecuteDuringImaging</code>	<code>false</code>	Use if your extension only overrides messages that are sent for working with dialog boxes or during the spooling phase of printing.
<code>gxNeedDeviceStatus</code>	<code>true</code>	Use if your extension performs any form of two-way communications with the device. This applies mostly to extensions driving add-on devices.
<code>gxDontNeedDeviceStatus</code>	<code>false</code>	Use if your extension does not need to perform two-way communications.
<code>gxChangePageAtGXDespoolPage</code>	<code>true</code>	Use if your extension changes the contents of a page during processing of an override for the <code>GXDespoolPage</code> message.
<code>gxDontChangePageAtGXDespoolPage</code>	<code>false</code>	Use if your extension does not change the contents of a page during processing of the <code>GXDespoolPage</code> message.
<code>gxChangePageAtGXImagePage</code>	<code>true</code>	Use if your extension changes the contents of a page during processing of an override for the <code>GXImagePage</code> message.
<code>gxDontChangePageAtGXImagePage</code>	<code>false</code>	Use if your extension does not change the contents of a page during processing of the <code>GXImagePage</code> message.

Table 6-11 Extension optimization resource flags (continued)

Constant	Value	Explanation
<code>gxChangePageAtGXRenderPage</code>	<code>true</code>	Use if your extension changes the contents of a page during processing of an override for the <code>GXRenderPage</code> message.
<code>gxDontChangePageAtGXRenderPage</code>	<code>false</code>	Use if your extension does not change the contents of a page during processing of the <code>GXRenderPage</code> message.
<code>gxServerPresenceRequired</code>	<code>true</code>	Use if your extension must be available on the print server's system.
<code>gxNotServerPresenceRequired</code>	<code>false</code>	Use if your extension does not have to be available on the print server's system.
<code>gxClientPresenceRequired</code>	<code>true</code>	Use if your extension must be available on the user's system.
<code>gxNotClientPresenceRequired</code>	<code>false</code>	Use if your extension does not have to be available on the user's system.

IMPORTANT

You need to be careful about setting these flags appropriately. Unnecessarily setting these flags can cause performance degradation during printing to certain devices. ▲

The extension optimization resource ID needs to be the constant `gxExtensionOptimizationID`. Listing 6-7 shows an example of an extension optimization resource. You can also find an example of an extension optimization resource definition in Listing 2-23 on page 2-37 in the chapter “Printing Extensions.”

Listing 6-7 An example of an extension optimization resource

```
resource gxExtensionOptimizationType (gxExtensionOptimizationID,
                                     sysHeap, purgeable)
{
    gxDontExecuteDuringImaging,
    gxDontNeedDeviceStatus,
    gxChangePageAtGXDespoolPage,
    gxDontChangePageAtGXImagePage,
    gxDontChangePageAtGXRenderPage,
    gxNotServerPresenceRequired,
    gxClientPresenceRequired
};
```

This example specifies that the extension does not have to be available during the imaging phase of printing, does not need to perform two-way communications, and only changes the page during the despooling process of the imaging phase of printing.

Printing Resources

This extension does not need to be present on the server, but must be present on the client machine.

Resources Used Only in Printer Drivers

Some resources are used for printer drivers, but not for extensions. These resources, which are summarized in Table 6-12, are described in this section.

Table 6-12 Resources used only in printer drivers

Resource	Function	Status
Imaging system (<code>'isys'</code>)	Identifies the imaging system that the driver uses.	Required
Look (<code>'look'</code>)	Specifies the kinds of communications that a desktop printer uses.	Required
Communications (<code>'comm'</code>)	Specifies the parameters of device communications that are used by the driver.	Required
Customization (<code>'cust'</code>)	Defines translation parameters for compatibility with the Macintosh Printing Manager.	Optional (used for compatibility)
Resolution (<code>'resl'</code>)	Defines the horizontal and vertical resolution values supported by a driver for compatibility with the Macintosh Printing Manager.	Optional (used for compatibility)
Print record (<code>'PREC'</code>)	Defines a default print record for driver compatibility with the Macintosh Printing Manager.	Optional (used for compatibility)
Dialog control (<code>'dctl'</code>)	Automates the definition of control items in dialog boxes that are used for compatibility with the Macintosh Printing Manager.	Optional (used for compatibility)
Scaling table (<code>'stab'</code>)	Defines the discrete page size reduction values that a user can select when printing with the Macintosh Printing Manager.	Optional (used for compatibility)
Buffering (<code>'iobm'</code>)	Defines the timeout and buffering parameters for a driver.	Optional
Capture (<code>'cpts'</code>)	Defines the strings that allow a device to be removed and replaced on a network during printer sharing.	Optional

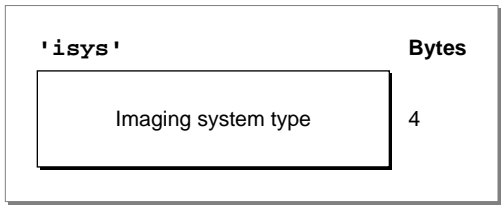
Table 6-12 Resources used only in printer drivers (continued)

Resource	Function	Status
Color set ('crst')	Defines color information for a raster printer driver.	Optional
Raster preferences ('rdip')	Defines rendering options for a raster driver.	Required for raster drivers
Raster package ('rpck')	Defines how bitmap data is packed into rasters for raster drivers.	Optional
Raster package controls ('ropt')	Defines how some forms of line feeding are performed on a raster device.	Optional
PostScript scanning ('scan')	Specifies a set of replacement strings that a printer driver uses when interpreting the information returned by a PostScript printer.	Optional
PostScript procset ('prec')	Defines a PostScript procedure set that can be downloaded to an output device.	Optional
PostScript font type ('pfnt')	Defines parameters for a specific font type used on a PostScript printer.	Optional
PostScript preferences ('pdip')	Defines rendering options for a PostScript printer driver.	Required for PostScript drivers

The Imaging System ('isys') Resource

The imaging system resource, of type `gxImagingSystemSelectorType`, tells QuickDraw GX which imaging system a driver uses. You must include an imaging system resource in your driver file. Figure 6-9 shows the structure of an imaging system resource, which consists of one value.

Figure 6-9 The imaging system resource



- **Imaging system type.** The imaging system with which your driver operates. This value is specified in your resource files as a four-character imaging system name, for which you can use the imaging system constants that are shown in Table 6-8 on page 6-27. Note that you cannot specify the `gxAnyPrinterType` value in this resource.

Printing Resources

The imaging system resource ID needs to be the constant `gxImagingSystemSelectorID`. Listing 6-8 shows an example of an imaging system resource.

Listing 6-8 An example of an imaging system resource

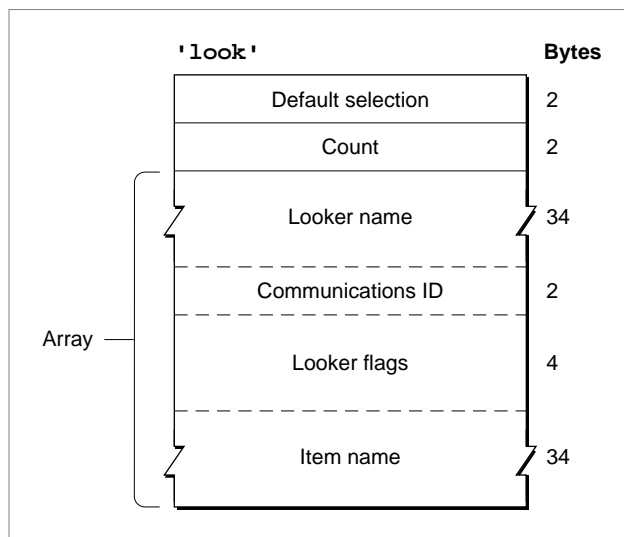
```
resource gxImagingSystemSelectorType (gxImagingSystemSelectorID,
                                     sysHeap, purgeable)
{
    gxRasterPrinterType
};
```

This resource tells QuickDraw GX that the driver uses the raster imaging system.

The Look ('look') Resource

The look ('look') resource specifies the kinds of communications that a driver uses. It contains a list of entries, each of which can be used when the user creates a desktop printer with the Chooser. The values in the look resource are also used for printer sharing; you must include a communications resource for each entry in the look resource to make printer sharing work properly. Figure 6-10 shows the structure of a look resource.

Figure 6-10 The look resource



Printing Resources

The look resource consists of a default indicator, a count, and one or more “looker” entries.

- **Default selection.** A value that specifies which looker entry is the default.
- **Count.** The number of looker entries in this resource.

Each looker entry tells QuickDraw GX about one kind of communications that the driver can perform and includes the following fields:

- **Looker name.** The name to display in the list of communications choices that are presented to the user.
- **Communications ID.** The resource ID of the communications (‘comm’) resource that corresponds to this looker entry.
- **Looker flags.** A collection of flags that can be combined to define what kind of connection the entry is naming. The flag constants shown are in Table 6-13.
- **Item name.** The name that is displayed for the item in the Chooser. For serial connections, this is the default port name. For AppleTalk and PrinterShare connections, this is the name-binding protocol (NBP) type.

Table 6-13 shows the constants that you can use to specify the kind of connection that a driver uses.

Table 6-13 Flag constants for the look resource

Constant	Value	Explanation
isAppleTalk	1	This entry specifies an AppleTalk Printer Access Protocol (PAP) connection that requires a name-binding protocol (NBP) lookup.
iconCells	2	This entry specifies that the Chooser displays icons in its list rather than names.
isPrinterShare	4	This entry specifies an Apple PrinterShare connection to a server.

The look resource ID needs to be the constant -4096. Listing 6-9 shows an example of a look resource.

Listing 6-9 An example of a look resource

```
resource 'look' (-4096, sysHeap, purgeable)
{
    2,                /* use the 2nd entry in list as default */
    {
        "AppleTalk",   -4096,   isAppleTalk,       "ImageWriter";
        "Serial",      -4095,   iconCells,         "Modem Port";
        "Servers",     -4094,   isAppleTalk+isPrinterShare,
                                "ImageWriterIIIS";
    };
};
```

This is a look resource for an ImageWriter II printer, which supports PAP (the Printer Access Protocol used in AppleTalk), serial, and PrinterShare communications interfaces. The communications resource that defines the AppleTalk interface has resource ID -4096, the communications resource for the serial interface has resource ID -4095, and the communications resource for the PrinterShare interface has resource ID -4094. The default connection type is serial, the Chooser uses icons for choosing the serial connection port, and the default icon that would be selected by the Chooser is the "Modem Port" icon.

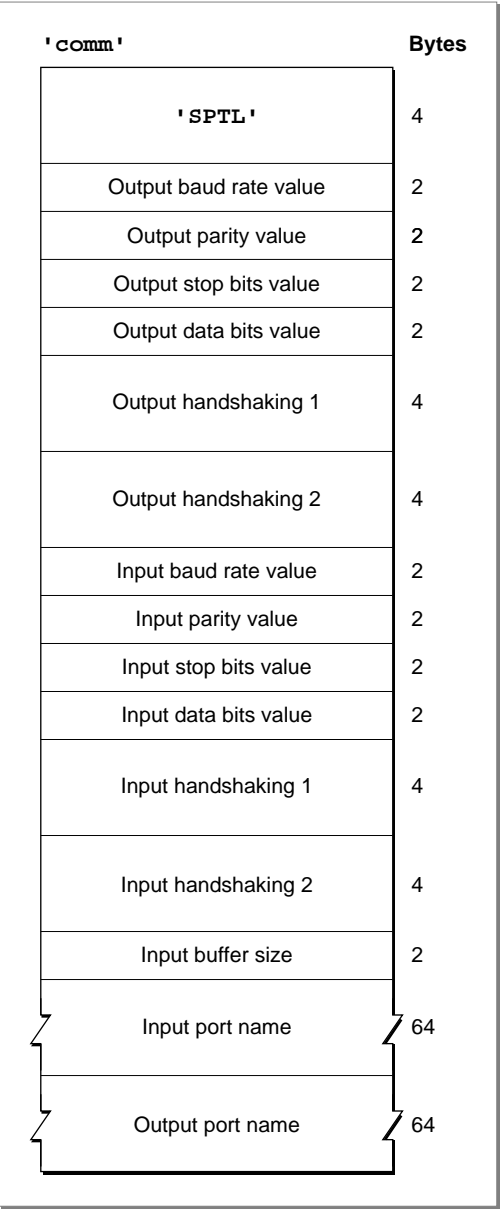
The Communications ('comm') Resource

Each communications ('comm') resource, of type `gxDeviceCommunicationsType`, defines parameters for a specific communications protocol that is supported by a printer driver. You must include one communications resource for each entry in your look resource. You can define printer communications resources in four different formats: serial communications resources, PAP communications resources, PrinterShare communications resources, and SCSI communications resources.

Serial Communications Resource

You need to define a serial communications resource to support serial communications in your driver. Figure 6-11 shows the structure of this form of a communications resource.

Figure 6-11 The serial communications resource



The serial communications resource consists of a communications type identifier and a number of input and output communications values.

- **Type identifier.** 'SPTL' (or the constant `Serial`). Used to identify serial communications resources.
- **Output baud-rate value.** The baud rate used for output, constants for which are shown in Table 6-14.

Printing Resources

- **Output parity value.** The parity type used for output, constants for which are shown in Table 6-15.
- **Output stop-bits value.** The number of stop bits used for output, constants for which are shown in Table 6-16.
- **Output data-bits value.** The number of data bits used for output, constants for which are shown in Table 6-17.
- **Output handshaking.** The handshaking protocol used for output, as defined on page 6-40.
- **Input baud-rate value.** The input baud rate, constants for which are shown in Table 6-14.
- **Input parity value.** The parity type used for input, constants for which are shown in Table 6-15.
- **Input stop-bits value.** The number of stop bits used for input, constants for which are shown in Table 6-16.
- **Input data-bits value.** The number of data bits used for input, constants for which are shown in Table 6-17.
- **Input handshaking.** The handshaking protocol used for input, as defined on page 6-40.
- **Input buffer size.** The number of bytes in the serial input buffer.
- **Input port name.** The name of the input port, filled in by the Chooser.
- **Output port name.** The name of the output port, filled in by the Chooser.

Table 6-14 shows the values that you can use to specify the baud rate in serial communications resources.

Table 6-14 Constants for the baud rate

Constant	Explanation
baud300	300 baud
baud600	600 baud
baud1200	1200 baud
baud1800	1800 baud
baud2400	2400 baud
baud3600	3600 baud
baud4800	4800 baud
baud7200	7200 baud
baud9600	9600 baud
baud19200	19200 baud
baud38400	38400 baud
baud57600	57600 baud

Printing Resources

Table 6-15 shows the constants that you can use to specify parity values in serial communications resources.

Table 6-15 Parity constants

Constant	Explanation
<code>noParity</code>	Parity not used
<code>oddParity</code>	Use odd parity
<code>evenParity</code>	Use even parity

Table 6-16 shows the constants that you can use to specify stop-bits values in serial communications resources.

Table 6-16 Constants for stop bits

Constant	Explanation
<code>oneStop</code>	1
<code>oneFiveStop</code>	1.5
<code>twoStop</code>	2

Table 6-17 shows the constants that you can use to specify data-bits values in serial communications resources.

Table 6-17 Constants for data bits

Constant	Explanation
<code>data5</code>	5 data bits
<code>data6</code>	6 data bits
<code>data7</code>	7 data bits
<code>data8</code>	8 data bits

The values that are used to define input and output handshaking in the serial communications resource are shown in Figure 6-12.

Figure 6-12 Input and output handshaking values

	Bytes
XOn/XOff flow output control flag	1
CTS hardware handshake flag	1
XOn character	1
XOff character	1
Handshake errors	1
Status changes	1
XOn/XOff input flow control flag	1
Unused	1

The handshake values define parameters of how the driver and the device communicate.

- **XOn/XOff output flow-control flag.** A value that defines whether output flow control is enabled or disabled. If this value is nonzero, XOn/XOff output flow is enabled.
- **CTS hardware handshake flag.** A value that defines whether CTS hardware handshake is enabled or disabled. If this value is nonzero, CTS hardware handshake is enabled.
- **XOn character.** The XOn character used for XOn/XOff flow control.
- **XOff character.** The XOff character used for XOn/XOff flow control.
- **Handshake errors.** The types of errors that cause input requests to be aborted. The constants that you can use for the error types are shown in Table 6-18. These constants can be combined into a single value to specify all of the error types that cause input requests to be aborted.
- **Status changes.** A value that indicates whether changes in the CTS or break status cause the Serial Driver to post device-driver events. The constants that you can use for the status event types are shown in Table 6-18. These constants can be combined into a single value to specify all of the status changes that cause device driver events to be posted.
Use of the status change option is discouraged because interrupts will be disabled for a long time while the driver event is posted.
- **XOn/XOff input flow-control flag.** A value that defines whether input flow control is enabled or disabled. If this value is nonzero, XOn/XOff input flow is enabled.
- **Unused.** A single byte reserved for use in the future.

Printing Resources

Table 6-18 shows the constants that you can use to specify handshake error-handling in serial communications resources.

Table 6-18 Handshake error types

Constant	Value	Explanation
<code>parityErr</code>	16	Abort the input request if a parity error occurs
<code>hwOverrunErr</code>	32	Abort the input request if a hardware overrun error occurs
<code>framingErr</code>	64	Abort the input request if a framing error occurs

Table 6-19 shows the constants that you can use to specify which status changes cause driver events to be posted.

Table 6-19 Status changes that cause driver events to be posted

Constant	Value	Explanation
<code>ctsEvent</code>	32	Causes driver events to be posted if the CTS status changes
<code>breakEvent</code>	128	Causes driver events to be posted if the break status changes

Listing 6-10 shows an example of a serial communications resource.

Listing 6-10 An example of a serial communications resource

```
resource gxDeviceCommunicationsType (-4095, sysHeap, purgeable)
{
    Serial
    {
        baud9600,      /* output baud rate */
        noParity,      /* output parity */
        oneStop,       /* output stop bits */
        data8,         /* output data size */
        0x00010000,    /* output handshaking */
        0x00000000,
        baud9600,      /* input baud rate */
        noParity,      /* input parity */
        oneStop,       /* input stop bits */
        data8,         /* input data size */
        0,             /* input handshaking */
        0,
```

Printing Resources

```

1024,          /* input buffer size */
".Ain",        /* input driver name; Chooser fills in */
".Aout"        /* output driver name; Chooser fills in */
};
};

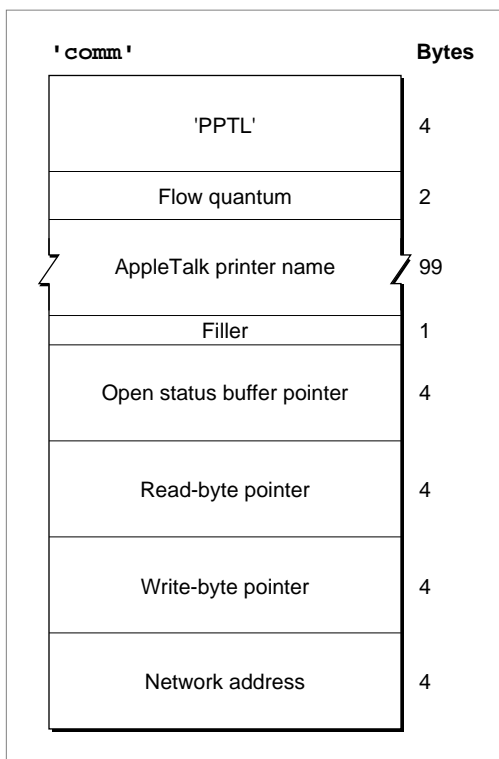
```

This example defines a serial port that supports input and output using the same parameters: 9600 baud, no parity, 1 stop bit, and 8 data bits. The input buffer size is 1024 bytes long.

AppleTalk PAP Communications Resource

You need to define a PAP communications resource to support the AppleTalk Printer Access Protocol in your driver. Figure 6-13 shows the structure of this form of a communications resource.

Figure 6-13 The PAP communications resource



Printing Resources

The PAP communications resource consists of a communications type identifier and a number of input and output communications values.

- **Type identifier.** 'PPTL' (or the constant `PAP`). Used to identify PAP communications resources.
- **Flow quantum.** A value that specifies the flow quantum, which is the number of packets that your device sends or receives in one transaction. The flow quantum value is hardware dependent: it is defined by the hardware manufacturer. The flow quantum value for LaserWriter printers is 8, the value for ImageWriters is 1. You can read more about this value in *Inside AppleTalk*.
- **AppleTalk printer name.** A string value that is filled in by the Chooser with the compacted, network name of the printer.
- **Filler.** An unused byte added for alignment purposes.
- **Open status buffer pointer.** An internally used value. This must be 0 in your resource definition.
- **Read-byte pointer.** An internally used value. This must be 0 in your resource definition.
- **Write-byte pointer.** An internally used value. This must be 0 in your resource definition.
- **Network address.** An internally used value. This must be 0 in your resource definition.

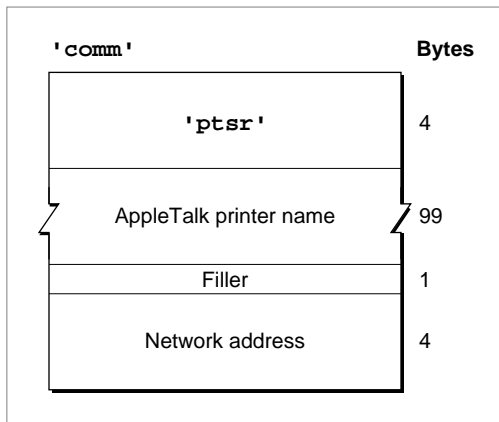
Listing 6-11 shows an example of a PAP communications resource that defines an AppleTalk connection for an ImageWriter II printer, which uses a flow quantum value of 1.

Listing 6-11 An example of an AppleTalk communications resource

```
resource gxDeviceCommunicationsType(-4096, sysHeap, purgeable)
{
    PAP
    {
        1,                /* flow quantum */
        "",               /* AppleTalk address; Chooser fills in */
        0, 0, 0, 0
    };
};
```

PrinterShare Communications Resource

You need to define a PrinterShare communications resource to support a server connection to your driver. Figure 6-14 shows the structure of this form of a communications resource.

Figure 6-14 The PrinterShare communications resource

The PrinterShare communications resource consists of a communications type identifier and a number of input and output communications values.

- Type identifier. 'ptsr' (or the constant `PrinterShare`). Used to identify PrinterShare communications resources.
- AppleTalk printer name. A string value filled in by the Chooser with the compacted, network name of the printer.
- Filler. An unused byte added for alignment purposes.
- Network address. The most recent network address of the printer. This value is filled in at run time and needs to be 0 in your resource definition.

Listing 6-12 shows an example of a PrinterShare communications resource that defines a print server's connection for an ImageWriter II printer.

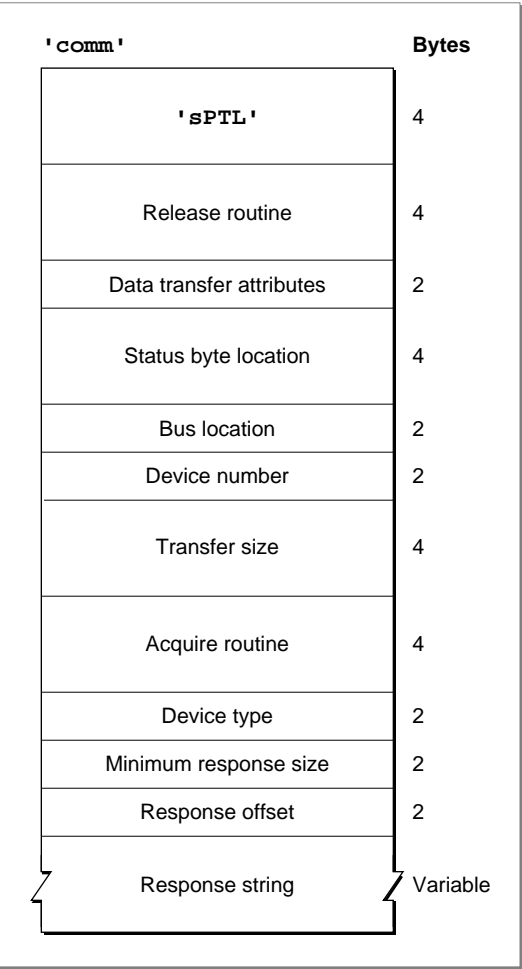
Listing 6-12 An example of a PrinterShare communications resource

```
resource gxDeviceCommunicationsType (-4094, sysHeap, purgeable)
{
    PrinterShare,
    {
        "",          /* AppleTalk address; Chooser fills in */
        0
    };
};
```

SCSI Communications Resource

You need to define a SCSI communications resource for SCSI output devices. The structure of this form of a communications resource is shown in Figure 6-15. Although QuickDraw GX provides the SCSI communications resource, it does not provide a default implementation of the device communications messages for SCSI devices. If you are implementing a printer driver for a SCSI device, you must override the device communications messages, which are described in the chapter “Printing Messages” in this book.

Figure 6-15 The SCSI communications resource



The SCSI communications resource consists of a communications type identifier and a number of SCSI control values.

Printing Resources

- **Type identifier.** 'sPTL' (or the constant SCSI) is used to identify SCSI communications resources.
- **Release routine.** An internally used value. This must be 0 in your resource definition.
- **Data transfer attributes.** The SCSI input/output attributes that apply to data transfers.
- **Status byte location.** An internally used value. This must be 0 in your resource definition.
- **Bus location.** The SCSI bus number for the device. The value 0 indicates the motherboard.
- **Device number.** The SCSI device number for the device.
- **Transfer size.** The number of bytes for each transfer (at the SCSI TIB level). If this value is 0, it is ignored.
- **Acquire routine.** An internally used value. This must be 0 in your resource definition.
- **Device type.** The type of SCSI device to look for.
- **Minimum response size.** The minimum amount of data, in bytes, in the response data.
- **Response offset.** The location in the response data to look for the response string. This is specified as a byte offset from the start of the response data.
- **Response string.** The string to look for in the response data.

Listing 6-13 shows an example of a SCSI communications resource that defines a SCSI communications connection for a LaserWriter IISC printer.

Listing 6-13 An example of a SCSI communications resource

```
resource gxDeviceCommunicationsType (-4095, sysHeap, purgeable)
{
    SCSI
    {
        0,          /* release device routine */
        0,          /* I/O attributes */
        0,          /* status byte */
        0,          /* SCSI bus number */
        0,          /* SCSI ID */
        0,          /* chunk level */
        0,          /* acquire routine (use default) */
        2,          /* device type to look for */
        27,         /* minimum length of additional response data */
        8,          /* bytes from start for search */
        "APPLE     PERSONAL LASER  ", /* search string */
    };
};
```

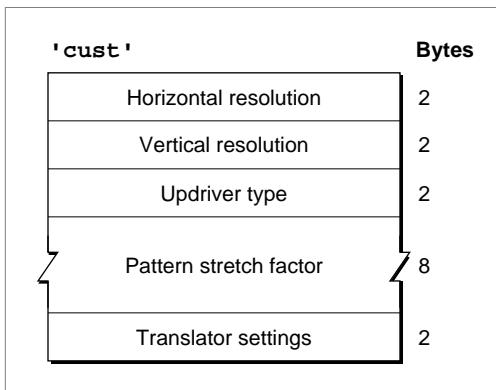
Note

QuickDraw GX provides the SCSI communications resource type and supports it in the Chooser. However, QuickDraw GX does not provide a default implementation of SCSI printer communications. If you are implementing a driver for a SCSI printer, you must override all of the communications messages. These messages are described in the chapter “Printing Messages” in this book. ♦

The Customization ('cust') Resource

The driver customization ('cust') resource, of type `gxCustType`, tells QuickDraw GX how to support the use of your printer driver by Macintosh Printing Manager applications. It defines values that help QuickDraw GX to translate Macintosh Printing Manager calls into messages for your printer driver. If you do not include a customization resource for your driver, a default version is used by QuickDraw GX. The default version works well for most drivers. Figure 6-16 shows the structure of a customization resource.

Figure 6-16 The customization resource



The customization resource defines the following:

- **Horizontal resolution.** The horizontal resolution to use for the device in dots per inch (dpi).
- **Vertical resolution.** The vertical resolution to use for the device in dots per inch.
- **Macintosh Printing Manager interface type.** The style of Macintosh Printing Manager interface driver with which your driver is compatible. Constants for this field are shown in Table 6-20.
- **Pattern stretch factor.** A point that indicates how to scale bitmap patterns when printing.

Printing Resources

- **Translator settings.** Settings for translating Macintosh Printing Manager driver calls into messages for your driver. Constants for the translator setting values, which you can combine together into a single value, are shown in Table 6-21.

Table 6-20 shows the constants that you can use to specify the updriver type in customization resources.

Table 6-20 Updriver values for the customization resource

Constant	Value	Explanation
defaultUpDriver	0	LaserWriter interface
laserWriter	0	LaserWriter interface
laserWriterSC	1	LaserWriter SC interface

Table 6-21 shows the constants that you can use to specify translator settings in customization resources.

Table 6-21 Translator setting values

Constant	Value	Explanation
gxDefaultOptionsTranslation	0x0000	These default settings produce the most accurate representation of the QuickDraw data.
gxOptimizedTranslation	0x0001	The translator fills in the center of outline text shapes that are drawn in <code>srcOr</code> mode with white, producing the same effect as black-and-white QuickDraw.
gxReplaceLineWidthTranslation	0x0002	The line width specified in the PostScript <code>SetLineWidth</code> command becomes the new line width. The default (without this option) is to scale the line width by the specified value.
gxSimpleScalingTranslation	0x0004	The translator scales data via a simple multiply, without compensating for increased resolution. The resulting scaled image is similar to what QuickDraw would have produced when it scaled the data.
gxSimpleGeometryTranslation	0x0008	The translator translates QuickDraw data without taking into account the QuickDraw hanging pen. The translator does not reproduce the QuickDraw 6-sided line and 7-sided triangle. This means faster drawing with QuickDraw GX with some loss of accuracy. This option turns on the <code>gxSimpleScalingTranslation</code> option.

Table 6-21 Translator setting values (continued)

Constant	Value	Explanation
<code>gxSimpleLinesTranslation</code>	<code>0x000C</code>	The translator maintains the width of lines when lines are at an angle, rather than producing a thicker diagonal line. With this option, line width is the same as pen width. This option turns on the <code>simpleScalingTranslation</code> and <code>simpleGeometryTranslation</code> options.
<code>gxLayoutTextTranslation</code>	<code>0x0010</code>	The translator allows layout to perform default substitutions, which produce text that is more attractive than, but different from, the original QuickDraw text.
<code>gxRasterTargetTranslation</code>	<code>0x0020</code>	The translator produces output for a raster device.
<code>gxPostScriptTargetTranslation</code>	<code>0x0040</code>	The translator produces output for a PostScript device.

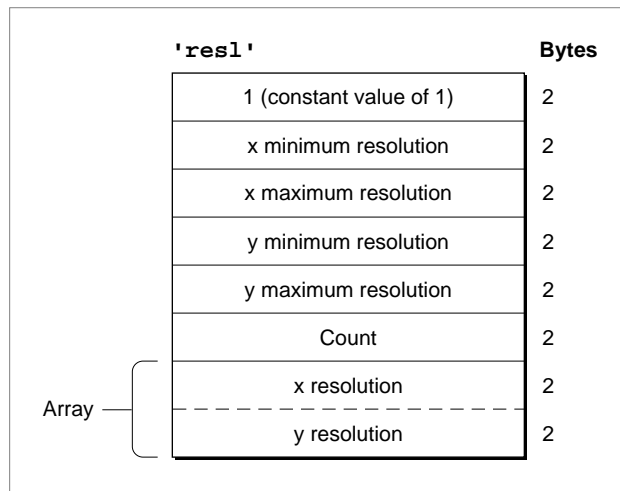
The customization resource ID needs to be the constant `gxCustID` (-8192). Listing 6-14 shows an example of a customization resource for the ImageWriter II printer driver, which uses the optimized translation settings and scales bitmaps by a factor of 2 both horizontally and vertically.

Listing 6-14 An example of a customization resource

```
resource gxCustType (gxCustID, sysHeap, purgeable)
{
    144, 144,          /* 144 dpi device */
    defaultUpDriver,   /* use default upDriver */
    {2,2},             /* pattern scaling (144dpi/72dpi)*/
    gxOptimizedTranslation/* use optimized translator settings */
};
```

The Resolution ('resl') Resource

The resolution ('resl') resource, of type `gxReslType`, is used to define the resolution values that a driver supports. This resource is optional: if you do not provide one, the resolution values in the customization resource are used. If you do not provide a resolution resource or a customization resource, the default customization values (the defaults for the updriver) are used. You can define a range of resolution values in a single resolution resource. Figure 6-17 shows the structure of resolution resource.

Figure 6-17 The resolution resource

The resolution resource defines the resolution range for a driver and contains a variable number of specific resolution entries.

- **Range type.** A constant value of 1.
- **x minimum resolution.** The minimum horizontal resolution supported by the driver in dots per inch (dpi).
- **x maximum resolution.** The maximum horizontal resolution supported by the driver in dpi.
- **y minimum resolution.** The minimum vertical resolution supported by the driver in dpi.
- **y maximum resolution.** The maximum vertical resolution supported by the driver in dpi.
- **Count.** The number of resolution pairs that follow in this resource.

Two values are defined for each specific resolution entry.

- **x resolution.** The horizontal resolution of this entry in dpi.
- **y resolution.** The vertical resolution of this entry in dpi.

The resolution resource ID needs to be -8192. Listing 6-15 shows an example of a resolution resource.

Listing 6-15 An example of a resolution resource

```
resource gxRes1Type (-8192, sysHeap, purgeable)
{
    rangeType,
    25, 1500,
```


Printing Resources

```

    25, 1500,
    {
    300, 300;
    600, 600;
    };
};

```

This resource is for a printing device that supports resolutions from 25 to 1500 dpi. The recommended resolutions values are 300 and 600 dpi.

The Print Record ('PREC') Resource

The print record ('PREC') resource defines a default print record for a driver, which is used to create TPrint records for compatibility with Macintosh Printing Manager applications. When the user of a Macintosh Printing Manager application prints a document, this resource is used to define the initial values to use for printing that document. This is an optional resource. If you don't supply one, QuickDraw GX uses the Apple default values.

The Apple default version of the 'PREC' resource is 120 bytes long and matches the structure of the universal print structure, which is described in the section "The Universal Print Structure" beginning on page 4-12 in the chapter "Printing Messages."

The structure of the print record resource is driver dependent. Many printer drivers use the Apple default values and then modify the universal print structure fields to meet their needs.

Listing 6-16 shows an example of a print record resource that is the print record for the ImageWriter II printer driver.

Listing 6-16 An example of a print record resource

```

resource 'PREC' (0, sysHeap, purgeable) {
    4,                /* increase from 3 to 4 for QuickDraw GX */
    0,
    72,
    80,
    {0, 0, 752, 640},
    {-36, -20, 756, 660},
    1,
    0,
    NoDftBts,
    NoResSet,
    NoScroll,
    NoZoom,
    NormPix,

```

Printing Resources

```

    Portrait,
    LowRes,
    1320,
    1020,
    0,
    Fanfold,
    0,
    72,
    80,
    {0, 0, 752, 640},
    80,
    32,
    640,
    3200,
    24,
    norm,
    1,
    1,
    1,
    scanTB,
    1,
    9999,
    1,
    Spool,
    1,
    0,
    0,
    0,
    0,
    0,
    0,
    0,
    1
};

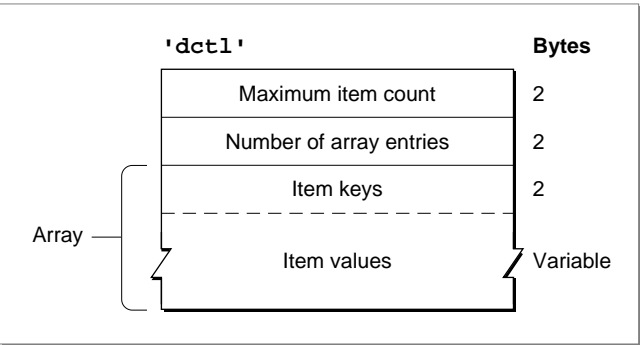
```

The Dialog Control ('dctl') Resource

You need to include a dialog control resource to define control items for the print dialog boxes used with Macintosh Printing Manager applications. This resource provides you with a means of mapping the behavior of items in an item list ('DITL') resource to fields in the universal print structure that QuickDraw GX uses for compatibility with Macintosh Printing Manager applications. The resource consists of a list of action types, each of which maps to one or more items in the item list resource and to a location in the universal print record.

The item list resource is described in *Inside Macintosh: Macintosh Toolbox Essentials*. The universal print structure is described in the section “The Universal Print Structure” beginning on page 4-12 in the chapter “Printing Messages.” Figure 6-18 shows the structure of the dialog control resource.

Figure 6-18 The dialog control resource



The dialog control resource contains a count of the control-item entries and an array of control-item definitions.

- **Maximum item count.** The maximum item count for the item list resource.
 - **Number of array entries.** The number of control-item definitions that follow.
- Each control item definition has a unique item-key value and some number of values.
- **Item key.** The type of control item. The values that you can use in this field are shown in Table 6-22.
 - **Item values.** The values that are needed to define the control item. The number and types of the values that you include in this field depend on the type of item that you are defining, as shown in the individual action sections beginning on page page 6-55.

The types of control items that you can define in a dialog control resource are shown in Table 6-22. Each item is described in more detail in the sections that follow the table.

Table 6-22 Control item types

Item type	Value	Explanation
Button	1	Implements the Cancel button
Cluster	2	Implements options that are presented to the user as a series of radio buttons
Copies	3	Implements the editable text item into which the user enters the number of copies to be printed

continued

Printing Resources

Table 6-22 Control item types (continued)

Item type	Value	Explanation
DialogBtn	4	Implements dialog boxes on top of the current dialog box
Frill	5	Used to draw the double-line bar, the version number, the heavy outline around the default button, and the printer name in the title of the dialog box
Moof	6	Draws Moof, the circus trick dog, which is displayed in the Page Setup dialog box to denote page orientation
OKButton	7	Implements the OK button
Orientation	8	Implements the orientation icon buttons
PageRange	9	Allows the user to specify the range of pages to be printed
PaperSizes	10	Implements selection of paper sizes by the user
Scale	11	Implements the print-scaling (Reduce/Enlarge) option
Toggle	12	Implements checkboxes that set and clear various bits in the <code>iFlags</code> field of the universal print structure

A dialog control resource can contain a number of actions, each of which is named by its type and followed by a number of values. Listing 6-17 shows two definitions of a dialog control resource from the ImageWriter II printer driver.

Listing 6-17 Two examples of a dialog control resource

```
resource 'dctl' (-8192, sysHeap, purgeable) {
    20,
    {
        Button { 2, cancel },
        Frill { 4, line },
        PaperSizes { 0, 0, { 6, 7, 8, 9, 10, 11 } },
        Orientation { 13, 14, 0, 0 },

        Toggle { 16, bPreciseBitmap },
        Toggle { 17, bUser0 },
        Toggle { 18, bBiggerPages },

        Frill { 19, version },
        Frill { 20, default },
    }
};
```

Printing Resources

```
resource 'dctl' (-8191, sysHeap, purgeable) {
    21,
    {
        Button { 2, cancel },
        Frill { 4, line },
        Cluster { quality, { 6, 7, 8 } },
        PageRange { 10, 11, 12, 14 },
        Copies { 16 },
        Cluster { feed, { 18, 19 } },

        Frill { 20, version },
        Frill { 21, default },
    }
};
```

The following sections describe the values that you need to specify in the dialog control resource for each item type.

Button Actions

You use the button (`Button`) action-item type to implement the Cancel button in the Print dialog box used with Macintosh Printing Manager applications. When the user selects this button, QuickDraw GX automatically restores the values in the universal print structure to what they were when the dialog box was created. This action uses two values:

- **Item ID.** The integer ID of the item in the item list to which this button pertains.
- **Button kind.** The kind of button that this represents. At this time, you can only use this item type to define Cancel buttons. You use the integer constant `cancel` to do this.

Listing 6-17 on page 6-54 contains two examples of button actions.

Cluster Actions

You use the cluster (`Cluster`) action-item type to implement clusters of radio buttons. When the user selects one of the radio buttons, QuickDraw GX automatically updates the corresponding value in the universal print structure. This action uses two values:

- **Cluster type.** The kind of item that the cluster is used to define. This is one of the integer constants shown in Table 6-23.
- **Cluster items.** The IDs of the items to which the cluster corresponds.

Printing Resources

Table 6-23 shows the constants that you can use to define what kind of item the radio button cluster is defining. Listing 6-17 contains an example of a cluster action.

Table 6-23 Cluster type constants for the dialog control resource

Cluster type	Value	Explanation
feed	0	Defines the feed mode used for the print job by filling in the <code>feed</code> field in the universal print structure
quality	1	Defines the print-quality mode used by filling in the <code>qualityMode</code> field in the universal print structure
coverPage	2	Defines the type of cover page to print for the print job by filling in the <code>coverPage</code> field in the universal print structure
firstPage	3	Defines which of the printer's paper trays is used as the first paper tray for this print job by filling in the <code>firstTray</code> field in the universal print structure
restPage	4	Defines the other paper tray for this print job by filling in the <code>remainingTray</code> field in the universal print structure
headMotion	5	Defines the type of print-head motion to use for the print job by filling in the <code>headMotion</code> field in the universal print structure
createFile	6	Defines the type of file to save for this print job by filling in the <code>saveFile</code> field in the universal print structure
user0	7	Reserved to allow the driver to define its own clusters and fill in the <code>userCluster1</code> field in the universal print structure
user1	8	Reserved to allow the driver to define its own clusters and fill in the <code>userCluster2</code> field in the universal print structure
user2	9	Reserved to allow the driver to define its own clusters and fill in the <code>userCluster3</code> field in the universal print structure

Copies Actions

You use the copies (`Copies`) action-item type to implement the editable text item into which the user enters the desired number of print copies. When the user types a value into this field, QuickDraw GX verifies that the number is between 1 and 9999 and then stores that value in the `actualCopies` field in the universal print structure. This action uses one value:

- **Item ID.** The ID of the item to which this action pertains.

Listing 6-17 on page 6-54 contains an example of the copies action type.

Dialog-Button Actions

You use the dialog-button (`DialogBtn`) action-item type to implement dialog boxes that appear on top of the current dialog box. For example, in the LaserWriter printer driver, the Options button in the style dialog box displays the dialog box with Moof in it on top of the style dialog box. When the user selects the item, QuickDraw GX brings up the specified dialog box, which can itself contain any of the action-item types in it. This action uses three values:

- Item ID. The ID of the item to which this action pertains.
- Dialog ID. The ID of the dialog resource to display when this item is selected by the user.
- Control ID. The ID of the dialog control resource for the dialog box that is displayed when this item is selected by the user.

Frill Actions

You use the frill (`Frill`) action-item type to implement informational and decorative objects in the dialog box. This action uses two values:

- Item ID. The ID of the item to which this action pertains.
- Frill type. The type of frill. This is one of the integer values shown in Table 6-24.

Table 6-24 Frill type constants for the dialog control resource

Frill type	Value	Explanation
line	0	Used to draw the double-line bar in the dialog box.
version	1	Used to draw the version number in the dialog box.
default	2	Used to draw the heavy outline under the default button in the dialog box.
printerName	3	Used to draw the printer name in the dialog box. This item must be a static text item.

Listing 6-17 on page 6-54 contains several examples of frill actions.

Moof Actions

You use the Moof (`Moof`) action-item type to display Moof, the circus trick dog, as is done when the orientation of the page is displayed in the Page Setup dialog box. This action uses one value:

- Item ID. The ID of the item in which Moof is drawn when the dialog box is initialized.

Confirm-Button Actions

You use the confirm-button (`OkButton`) action-item type to implement the OK button in a dialog box. This item is only needed if there is a Create File cluster that needs to change the name of the OK button for printing (to switch the name of the OK button depending on whether the user is printing or saving a file). This action uses three values:

- Item ID. The ID of the OK button item. This value must be 1.
- String ID 1. The ID of the string resource to display for the “Print” choice.
- String ID 2. The ID of the string resource to display for the “Save” choice.

Orientation Actions

You use the orientation (`Orientation`) action-item type to implement the orientation icon buttons. You can use up to four different buttons to establish the value of the `orientation` field in the universal print structure. QuickDraw GX assumes that the driver contains as many icon (`'ICON'`) resources as there are nonzero entries in this item definition. QuickDraw GX also assumes that the icon resource IDs start at 0 and increase by 1. This action uses four values:

- Portrait item ID. The ID of the portrait orientation item.
- Landscape item ID. The ID of the landscape orientation item.
- Alternate portrait item ID. The ID of the rotated portrait orientation item.
- Alternate landscape item ID. The ID of the rotated landscape orientation item.

Page-Range Actions

You use the page-range (`PageRange`) action-item type to implement the page-range selection. When the user enters page values in the “From” and “To” fields, QuickDraw GX updates the `firstpage` and `lastPage` values in the universal print structure. This action uses four values:

- All pages button ID. The ID of the “All Pages” button.
- Range button ID. The ID of the “Range” button.
- From field ID. The ID of the “From” or first-page editable text field.
- To field ID. The ID of the “To” or last-page editable text field.

Listing 6-17 on page 6-54 contains an example of a page-range action in a dialog control resource.

Paper-Size Actions

You use the paper-size (`PaperSizes`) action-item type to implement selection of paper sizes for a print job. This action has two values and an array of item IDs:

- Pop-up item ID. The item ID of the paper-sizes pop-up menu.
- Pop-up radio item ID. The ID of the paper-sizes pop-up menu.

- **Item IDs.** An array of IDs, one per radio button item that is referenced by this action. Listing 6-17 on page 6-54 contains an example of a paper-size action item in a dialog control resource.

Scale Actions

You use the scale (*Scale*) action-item type to implement the reduce and enlarge option in the Page Setup dialog box. This action uses two values:

- **Edit ID.** The ID of the editable text item that contains the scaling value.
- **Arrow ID.** The ID of the item that controls the up and down actions on the scaling value.
- **Scaling table ID.** The resource ID of the scaling table resource that defines the scaling values for this action item. The scaling table resource is described on page 6-61.

Toggle Actions

You use the toggle (*Toggle*) action-item type to implement the checkboxes that set and clear the flags in the *options* field of the universal print structure. This action uses two values:

- **Item ID.** The ID of the item to which this action pertains.
- **Toggle type.** The kind of toggle item to which this action pertains. You can combine any of the integer constants from Table 6-25 to form the value of this field. When you combine values and the user clicks the checkbox, all of the flags associated with the value are set.

Table 6-25 Toggle action-item flags for the dialog control resource

Toggle item type	Value	Explanation
bPreciseBitmap	0x0001	The driver needs to format pages as tall-adjusted for the Apple ImageWriter family of printers, and the driver needs to use precise bitmaps for the Apple LaserWriter family of printers. This is the same as the <i>gxPreciseBitmap</i> value described on page 4-15 in the chapter “Printing Messages.”
bBiggerPages	0x0002	The driver needs to not apply gaps if printing this job to one of the Apple ImageWriter family of printers, and the driver needs to use a large print area for the Apple LaserWriter family of printers. This is the same as the <i>gxBiggerPages</i> value described on page 4-15 in the chapter “Printing Messages.”

continued

Printing Resources

Table 6-25 Toggle action-item flags for the dialog control resource (continued)

Toggle item type	Value	Explanation
bGraphicSmoothing	0x0004	The driver needs to perform graphics smoothing on the Apple LaserWriter family of printers. This is the same as the <code>gxGraphicSmoothing</code> value described on page 4-15 in the chapter “Printing Messages.”
bTextSmoothing	0x0008	The driver needs to perform text smoothing for this print job. This is the same as the <code>gxTextSmoothing</code> value described on page 4-15 in the chapter “Printing Messages.”
bFontSubstitution	0x0010	The driver needs to perform font substitution for this print job. This is the same as the <code>gxFontSubstitution</code> value described on page 4-15 in the chapter “Printing Messages.”
bInvert	0x0020	The driver needs to invert the printed image (convert white to black and black to white) for this print job. This is the same as the <code>gxInvertPage</code> value described on page 4-15 in the chapter “Printing Messages.”
bFlipHoriz	0x0040	The driver needs to flip pages horizontally for this print job. This is the same as the <code>gxFlipPageHoriz</code> value described on page 4-15 in the chapter “Printing Messages.”
bFlipVert	0x0080	The driver needs to flip pages vertically for this print job. This is the same as the <code>gxFlipPageVert</code> value described on page 4-15 in the chapter “Printing Messages.”
bColorMode	0x0100	This print job uses color printing. This is the same as the <code>gxColorMode</code> value described on page 4-15 in the chapter “Printing Messages.”
bBidirectional	0x0200	This print job uses bidirectional printing. This is the same as the <code>gxBidirectional</code> value described on page 4-15 in the chapter “Printing Messages.”
bUser0	0x0400	Available for driver-defined options.
bUser1	0x0800	Available for driver-defined options.
bUser2	0x1000	Available for driver-defined options.
bReserved0	0x2000	Reserved for future options.
bReserved1	0x4000	Reserved for future options.
bReserved2	0x8000	Reserved for future options.

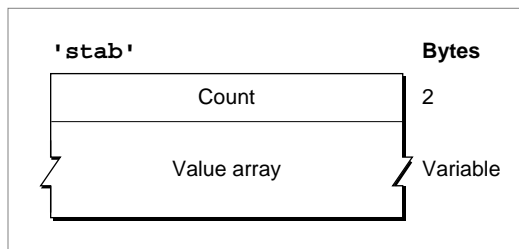
Listing 6-17 on page 6-54 contains several examples of toggle action items in a dialog control resource.

The Scaling Table ('stab') Resource

You need to include a scaling table resource when you include a scale action-item type in your dialog control ('dctl') resource, which is described in the previous section. The values in the scaling table resource define the discrete page-size reduction values that a user can select for a Macintosh Printing Manager application.

Figure 6-19 shows the structure of the scaling table resource.

Figure 6-19 The scaling table resource



The scaling table resource contains a count of the scaling value entries and an array of values.

- **Count.** The number of scaling values that follow.
- **Value array.** An array of scaling values. Each of these integers defines a discrete scaling value that the user can select when clicking on the up and down arrows for the scaling item in the dialog box.

Note

The user can directly enter any scaling value into the scaling item box. The values in the scaling table are only used when the user clicks on the up and down arrows. ♦

IMPORTANT

The values in the scaling table resource must be in ascending order. ▲

The Buffering and Input/Output Preferences ('iobm') Resource

The buffering and input/output preferences ('iobm') resource, of type `gxUniversalIOPrefType`, defines the timeout and buffering parameters for the driver. This resource is optional; if you do not supply one, QuickDraw GX uses standard I/O with two buffers, each 1 KB long, and defines each timeout value to be 10 seconds. Figure 6-20 shows the structure of the buffering and input/output preferences resource.

Figure 6-20 The buffering and input/output preferences resource

'iobm'	Bytes
Type of I/O	4
Number of buffers	4
Buffer size	4
Number of pending requests	4
Open/close timeout	4
Read/write timeout	4

The buffering and input/output preferences resource defines the buffering and timeout values for I/O.

- **Type of I/O.** The type of I/O to use. This is one of two values: `standardIO` or `customIO`. If you specify `customIO`, it means that your driver is handling input and output communications without using the built-in support.

Note

SCSI drivers must use the `customIO` value in this field. QuickDraw GX does not provide a default implementation of SCSI connections. If you do develop a SCSI driver, you must override the device communications messages and manage all of the communications yourself. ♦

- **Number of buffers.** The number of buffers to create for this driver.
- **Buffer size.** The size of each buffer.
- **Number of pending requests.** The maximum number of input or output requests that can be pending at any time.
- **Open/close timeout.** The number of clock ticks that constitute a timeout when trying to open or close the device.
- **Read/write timeout.** The number of clock ticks that constitute a timeout when trying to read from or write to the device.

Printing Resources

The buffering and input/output preferences resource ID needs to be the constant `gxUniversalIOPrefsID`. Listing 6-18 shows an example of a buffering and input/output preferences resource.

Listing 6-18 An example of a buffering and input/output preferences resource

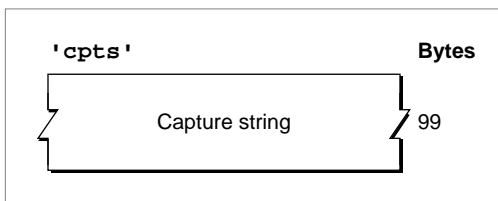
```
resource gxUniversalIOPrefsType (gxUniversalIOPrefsID, sysHeap,
                                purgeable)
{
    standardIO,
    4,          /* 4 buffers */
    2048,       /* each buffer 2 KB */
    10,         /* up to 10 I/O operations pending */
    1200,       /* open/close timeout of 1200 clock ticks */
    1200        /* read/write timeout of 1200 clock ticks */
};
```

This resource is for the ImageWriter II printer driver, which uses four buffers, each 2 KB long. Ten request blocks will be allocated for communications with this device, and requests to open, close, read, or write the device will time out in 1200 clock ticks (20 seconds).

The Capture ('cpts') Resource

The capture ('cpts') resource, of type `gxCaptureType`, allows devices to be removed and replaced on a network during printer sharing (through the `GXCaptureOutputDevice` message). This resource is not needed if your driver does not support the capture-and-release concept. The default version of the `GXCaptureOutputDevice` messages uses these resources and works only for PAP devices. Figure 6-21 shows the structure of a capture resource.

Figure 6-21 The capture resource



Printing Resources

- **Capture string.** The string that specifies capture information. There are four kinds of capture information, each of which you specify in a separate capture resource. The four capture-string resource type constants are shown in Table 6-26.

Table 6-26 Capture resource types

Constant	Explanation
<code>gxCaptureStringID</code>	Used to capture and release the device and does not contain a length byte
<code>gxReleaseStringID</code>	Used for a released device and does not contain a length byte
<code>gxUncapturedAppleTalkType</code>	The AppleTalk device name that is used for an uncaptured device; it includes a length byte
<code>gxCapturedAppleTalkType</code>	The AppleTalk device name that is used for a captured device; it includes a length byte

The system performs string substitution in any of the capture strings and defines constants that you can use for this purpose. Each of these constants is replaced with the appropriate value at run time. The constant strings are shown in Table 6-27.

Table 6-27 Substitution strings for the capture resource

Constant	Explanation
<code>PRINTERNAME</code>	Replaced with the name of the printer
<code>PRINTERTYPE</code>	Replaced with the name-binding protocol (NBP) type of the printer
<code>NAMELEN</code>	Replaced with a single byte that is the length of the printer name
<code>TYPELEN</code>	Replaced with a single byte that is the length of the printer NBP type

The four capture resources shown in Listing 6-19 are taken from the ImageWriter II printer driver.

Listing 6-19 Examples of capture resources for a printer driver

```

resource gxCaptureType (gxCapturedAppleTalkType, sysHeap,
                        purgeable)
{
    "\0D011ImageShared"
};

resource gxCaptureType (gxUncapturedAppleTalkType, sysHeap,
                        purgeable)
{
    "\0D011ImageWriter"
};

resource gxCaptureType (gxCaptureStringID, sysHeap, purgeable)
{
    "\0X1B\0X62NAMELENPRINTERNAMETYPEPELENPRINTERTYPE\0X01*"
};

resource gxCaptureType (gxReleaseStringID, sysHeap, purgeable)
{
    "\0X1B\0X62NAMELENPRINTERNAMETYPEPELENPRINTERTYPE\0X01*"
};

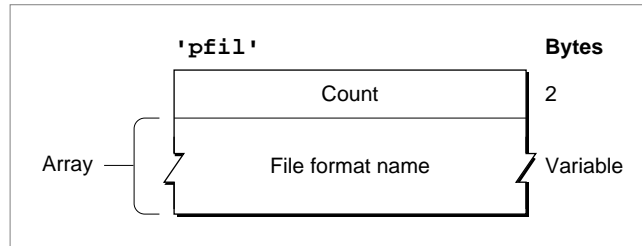
```

The Print-File Formats ('pfil') Resource

The print-file formats resource is used to specify driver-specific output file formats to display to the user for the print-to-file option. Each format name specified in this resource is displayed in the list of file formats shown in the standard file dialog box when the user confirms the Print dialog box after selecting the “Destination” option.

Figure 6-22 shows the structure of a print-file formats resource.

Figure 6-22 The print file formats resource



The print-file formats resource consists of a count and a number of file format names.

- **Count.** The number of file format names that follow in this resource.
- **File format name.** The Pascal string name to use when creating the output file.

Listing 6-20 shows an example of a print-file formats resource that defines the file format name for the LaserWriter IIfx printer driver.

Listing 6-20 An example of a print-file formats resource

```
resource 'pfil' ( gxDriverFileFormatID, sysHeap, purgeable) {
    {
        "PostScript™"
    };
};
```

The Raster Preferences ('rdip') Resource

The raster preferences ('rdip') resource, of type `gxRasterPrefsType`, controls the rendering options for a raster driver. This resource is required for all raster printer drivers. Figure 6-23 shows the structure of a raster preferences resource.

Figure 6-23 The raster preferences resource

'rdip'		Bytes
Options		4
Horizontal resolution		4
Vertical resolution		4
Minimum band size		2
Maximum band size		2
RAM percentage		4
RAM slop		4
Imaging depth		2
Count		2
Plane flags		4
Angle		4
Frequency		4
Dither type		4
Tint type		4
Dot color space		4
Dot color profile		4
Dot color value 1		2
Dot color value 2		2
Dot color value 3		2
Dot color value 4		2
Background color space		4
Background color profile		4
Background color value 1		2
Background color value 2		2
Background color value 3		2
Background color value 4		2
Tint space		4
Plane color space		4
Plane color set		4
Plane color profile		4

Printing Resources

The raster preferences resource contains a number of values that define how imaging is performed on the device, followed by an array of plane values, each of which defines parameters for a specific color on the printer. For example, if your driver accepts CMYK (cyan, magenta, yellow, and black color specification) data, QuickDraw GX sends it planes for cyan, magenta, yellow, and black. The values in the plane array correspond to the number of times that you need to send raster data to the printer for color printing.

You need to understand how QuickDraw GX represents and uses colors to understand many of the values in this resource. Refer to two chapters in *Inside Macintosh: QuickDraw GX Objects*: the “View Objects” chapter and the “Colors and Color-Related Objects” chapter.

- **Options.** A collection of option values that you can combine together to specify how rendering occurs on a raster device. Constants for the raster preference options are shown in Table 6-28 on page 6-70.
- **Horizontal resolution.** The horizontal resolution for imaging as a fixed point number.
- **Vertical resolution.** The vertical resolution for imaging as a fixed point number.
- **Minimum band size.** The minimum band size, in pixels.
- **Maximum band size.** The maximum band size, in pixels. A value of 0 indicates that the maximum band size is the whole page.
- **RAM percentage.** The amount of available temporary memory that can be used. The actual amount used is computed by the following calculation:

$$((\text{Free memory} - \text{RAMSlop}) * \text{RamPercentage});$$
 The result is then rounded into a multiple of the minimum band size, but is never made larger than the maximum band size.
- **RAM slop.** The amount to subtract from the total available temporary memory before computing the percentage, as shown above.
- **Imaging depth.** The number of pixels per plane for imaging.
- **Count.** The number of entries in the plane array for this resource. This value is filled in at run time and is not part of the resource definition.

Each entry in the plane array is for a specific color. For example, a driver that accepts CMYK data is sent planes for cyan, magenta, yellow, and black. This data may or may not be equivalent to a band on a ribbon, an ink color, or a wax color; however, each plane that you define in this array does represent one pass of data by the device. Each entry in the plane array contains the following data values.

- **Plane flags.** A collection of flag values that you can combine together to specify controls for rendering the plane. Constants for the plane flags are shown in Table 6-29 on page 6-70.
- **Angle.** The apparent direction to use for halftones, in degrees. This is a fixed point value. The chapter “View Objects” in *Inside Macintosh: QuickDraw GX Objects* describes using halftones.
- **Frequency.** The size of the dot, in cells per inch, to use for halftones. This is a fixed point value. The chapter “View Objects” in *Inside Macintosh: QuickDraw GX Objects* describes using halftones.

Printing Resources

- Dither type. The dithering style, or halftone method, to use for this plane. Constants for color-plane dithering types are shown in Table 6-30 on page 6-71. The results of using these values are shown in the discussion of halftones in the chapter “View Objects” in *Inside Macintosh: QuickDraw GX Objects*.
- Tint type. The type of tinting to apply to this plane. Constants for the plane tinting types are shown in Table 6-31 on page 6-71. The use of these values is explained in the chapter “View Objects” in *Inside Macintosh: QuickDraw GX Objects*.
- Dot color space. Which color space the values in this plane use. The constants for color-space values are shown in the chapter “Colors and Color-Related Objects” in *Inside Macintosh: QuickDraw GX Objects*.
- Dot color profile. The color profile to use for the dot color on this plane. The value `gxNoProfile` (0) indicates that a color profile is not to be used. Any other value specifies the ID of the color matching (‘`cmat`’) resource to use. Color profiles are explained in the chapter “Colors and Color-Related Objects” in *Inside Macintosh: QuickDraw GX Objects*.
If you specify `gxNoProfile`, no color matching is performed.
- Dot color value 1. The first value for defining the dot color for this plane.
- Dot color value 2. The second value for defining the dot color for this plane.
- Dot color value 3. The third value for defining the dot color for this plane.
- Dot color value 4. The fourth value for defining the dot color for this plane.
- Background color space. The color space to use for the background color on this plane. The color-space values are shown in chapter “Colors and Color-Related Objects” in *Inside Macintosh: QuickDraw GX Objects*.
- Background color profile. The color profile to use for the dot color on this plane. The value `gxNoProfile` (0) indicates that a color profile is not to be used. Any other value specifies the ID of the color matching (‘`cmat`’) resource to use. Color profiles are explained in the chapter “Colors and Color-Related Objects” in *Inside Macintosh: QuickDraw GX Objects*.
If you specify `gxNoProfile`, no color matching is performed.
- Background color value 1. The first value for defining the background color for this plane.
- Background color value 2. The second value for defining the background color for this plane.
- Background color value 3. The third value for defining the background color for this plane.
- Background color value 4. The fourth value for defining the background color for this plane.
- Tint space. The color space to use for halftoning. Use one of the values shown in the chapter “Colors and Color-Related Objects” in *Inside Macintosh: QuickDraw GX Objects*.
- Plane color space. The color space of the plane (the original data’s color space). Color spaces, color sets, and color profiles are described in the chapter “Colors and Color-Related Objects” in *Inside Macintosh: QuickDraw GX Objects*.

Printing Resources

- **Plane color set.** The resource ID of the color set to use for the plane. You can specify the value `gxNoSet` to indicate no color set.
- **Plane color profile.** The resource ID of the color profile to use for the plane. You can specify the value `gxNoProfile` to indicate no color profile.

Table 6-28 shows the constants that you can use to specify the option values for the raster preferences resource.

Table 6-28 Raster preference option values

Constant	Value	Explanation
<code>gxDefaultRaster</code>	0	Uses default options
<code>gxDontResolveTransferModes</code>	0x01	Tells the system to not resolve transfer modes because your 32-bit device can do it faster on its own
<code>gxRenderInReverse</code>	0x02	Traverses the image in reverse order
<code>gxOnePlaneAtATime</code>	0x04	Renders each plane separately
<code>gxSendAllBands</code>	0x08	Sends bands, even if empty (all white)

Table 6-29 shows the constants that you can use to specify the plane flags values for the raster preferences resource.

Table 6-29 Flags used for each plane in the raster preferences resource

Constant	Value	Explanation
<code>gxDefaultOffscreen</code>	0	Specifies that the default settings are used, which means that the view port's halftone is based on the information in the imaging system data.
<code>gxDontSetHalftone</code>	1	Specifies that the system must not set the view port's halftone. Use this option when you do not want halftones on a device.
<code>gxDotTypeIsDitherLevel</code>	2	Specifies that the system must set the view port's dither level to the value stored in the <code>gxDotType</code> parameter and ignore other halftone information.

Table 6-30 shows the constants that you can use to specify the plane dithering types for the raster preferences resource.

Table 6-30 Plane dithering types

Constant	Value
gxRoundDot	1
gxSpiralDot	2
gxSquareDot	3
gxLineDot	4
gxEllipticDot	5
gxTriangleDot	6
gxDispersedDot	7

Table 6-31 shows the constants that you can use to specify the plane tinting types for the raster preferences resource.

Table 6-31 Plane tinting types

Constant	Value	Explanation
gxLuminanceTint	1	Use the luminance of the color
gxAverageTint	2	Add all the components, then divide by the number of components
gxMixtureTint	3	Find the closest color on the axis between the foreground and background colors
gxComponent1Tint	4	Use the value of the first component of the color space
gxComponent2Tint	5	Use the value of the second component of the color space
gxComponent3Tint	6	Use the value of the third component of the color space
gxComponent4Tint	7	Use the value of the fourth component of the color space

Printing Resources

The ID of the raster preferences resource must be the constant `gxRasterPrefsID`. Listing 6-21 shows an example of a raster preferences resource for a 144-dpi four-plane color device, the ImageWriter II raster printer.

Listing 6-21 An example of a raster preferences resource

```
resource gxRasterPrefsType (gxRasterPrefsID, sysHeap, purgeable)
{
    gxDefaultRaster,          /* default options are fine */

    0x00900000,0x00900000,    /* 144X144 dpi device */
    16,                        /* min band size is 2 head heights */
    0,                          /* max band size (0 is full page) */
    0x00004000,                /* RAM percentage (25%) */
    100*1024,                  /* RAM slop (100K) */
    4,                          /* 4 bit per plane device */
    {
        /* dithering offscreen */
        3,                      /* gxDontSetHalftone+gxDotTypeIsDitherLevel */

        0x002D0000,            /* angle unused for dithering */
        0x003C0000,            /* freq unused for dithering */
        4,                      /* dithering with a level of 4 */
        gxLuminanceTint,        /* tint space unused for dithering */

                                /* dot color is unused for dithering */
        gxRGBSpace, gxNoProfile, 0, 0, 0, 0,
                                /* background is unused for dithering */
        gxRGBSpace, gxNoProfile, 0xFFFF, 0xFFFF, 0xFFFF, 0,

        gxRGBSpace,            /* halftone space unused for dithering */

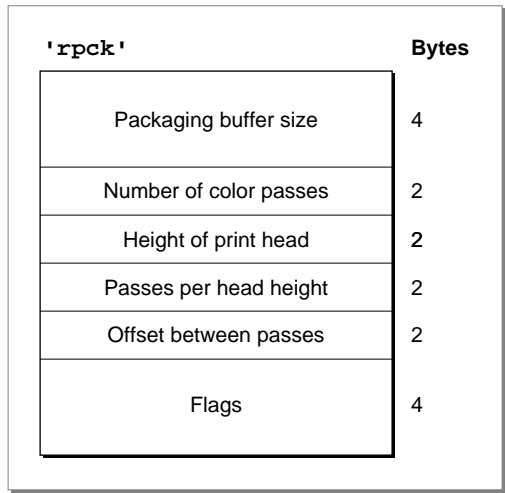
        gxIndexedSpace,         /* indexed color space */
        gxPrintingDriverBaseID, /* ID of the color set to use */
        gxPrintingDriverBaseID, /* ID of the color profile to use */
    };
};
```

The Raster Package ('rpck') Resource

The raster package ('rpck') resource, of type `gxRasterPackType`, controls how bitmap data is packed into rasters for raster drivers. This resource is optional.

Figure 6-24 shows the structure of a raster package resource.

Figure 6-24 The raster package resource



The raster package resource contains the following elements:

- **Packaging buffer size.** The number of bytes in the packing buffer. This value must be greater than or equal to the maximum number of bytes needed for one head pass on the printer.
- **Number of color passes.** The number of color passes required by the device to print an image. This value is typically 1 (for monochrome printers) or 4 (for CMYK printers).
- **Height of print head.** The height of the print head, in pixels.
- **Passes per head height.** The number of head passes required to achieve the height of the print head.
- **Offset between passes.** The offset, in pixels, between head passes.

Printing Resources

- **Flags.** The raster package options, which can be combined together into a single value. The constants for the raster packing options are shown in Table 6-32.

Table 6-32 Raster package options

Constant	Value	Explanation
<code>gxSendAllColors</code>	<code>0x00000001</code>	Send raster bands even if all clear
<code>gxInterlaceColor</code>	<code>0x00000002</code>	Interlace colors because ribbon contamination is of concern
<code>gxOverlayColor</code>	<code>0x00000004</code>	This is a color printer that does not have a ribbon contamination problem

The ID of the raster package resource must be the constant `gxRasterPackID`. Listing 6-22 shows an example of a raster package resource for an ImageWriter II printer.

Listing 6-22 An example of a raster package resource

```
resource gxRasterPackType (gxRasterPackID, sysHeap, purgeable)
{
    2500,          /* ImageWriter packing buffer size */
    4,             /* 4 color passes */
    16,           /* print head is 16 pixels high */
    2,            /* requires 2 passes to achieve 16 pixels */
    1,            /* 1 pixel difference between passes */
    gxInterlaceColor /* use interlace to avoid contamination */
};
```

This resource specifies that the ImageWriter II printer uses a packing buffer that is 2500 bytes long, which is long enough for the largest line packaged by the driver. Since the driver uses CMYK color, 4 passes are required to print an image. The print head is 16 pixels high, and it requires 2 printing passes to print 16 pixels of data.

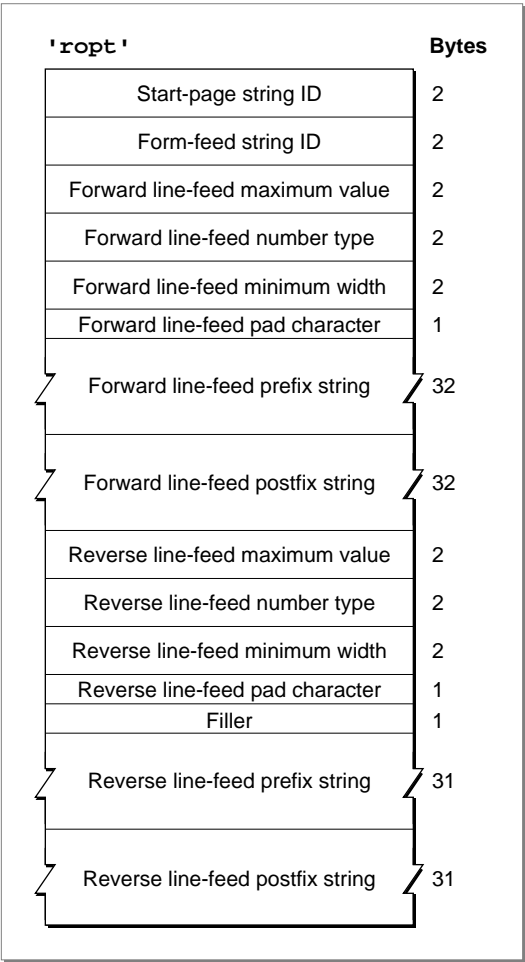
If you use the default implementation of the `GXRasterDataIn` message, you must define a raster package ('rpck') resource. The `GXRasterDataIn` message is described on page 4-97 in the chapter "Printing Messages" in this book.

The Raster Package Controls ('ropt') Resource

The raster package controls ('ropt') resource, of type `gxRasterPackOptionsType`, is used to define how some forms of line feeding are performed on a raster device. This resource is optional.

Figure 6-25 shows the structure of a raster package controls resource.

Figure 6-25 The raster package controls resource



The raster package controls resource contains the following elements:

- **Start-page string ID.** The ID of the wide string ('wstr') resource that is used as the start-page sequence. Wide strings are strings that require a 16-bit, short integer value to define their length. The length value is stored in the first 2 bytes of the string.
- **Form-feed string ID.** The ID of the wide string ('wstr') resource that is used as the form-feed sequence.
- **Forward line-feed maximum value.** The maximum amount of a forward line feed.
- **Forward line-feed number type.** The type of the number used to specify the forward line feed. Constants for the number types are shown in Table 6-33.
- **Forward line-feed minimum width.** The minimum width of the forward line feed.
- **Forward line-feed pad character.** The pad character used for the forward line feed.
- **Forward line-feed prefix string.** The prefix string used for the forward line feed.

Printing Resources

- Forward line-feed postfix string. The postfix string used for the forward line feed.
- Reverse line-feed maximum value. The maximum amount of a reverse line feed.
- Reverse line-feed number type. The type of the number used to specify the reverse line feed. The possible values are shown in Table 6-33.
- Reverse line-feed minimum width. The minimum width of the reverse line feed.
- Reverse line-feed pad character. The pad character used for the reverse line feed.
- Reverse line-feed prefix string. The prefix string used for the reverse line feed.
- Reverse line-feed postfix string. The postfix string used for the reverse line feed.

Table 6-33 shows the constants that you can use to specify the number types used in defining line feeds in raster package controls resources.

Table 6-33 Number types for specifying line feeds

Constant	Value	Explanation
<code>gxRasterNumNone</code>	0	The number is not sent at all. Only the prefix and postfix strings are sent.
<code>gxRasterNumDirect</code>	1	The number is first padded to minimum width bytes by prepending 0's to it and is then sent as a sequence of hex bytes.
<code>gxRasterNumToASCII</code>	2	The number is converted to ASCII and then front-padded with the specified pad character to make it minimum width bytes long. The ASCII characters are then sent to the printer.

The ID of a raster package controls resource must be the constant `gxRasterPackOptionsID`. Listing 6-23 shows an example of a raster package controls resource for the ImageWriter II printer driver.

Listing 6-23 An example of a raster package controls resource

```
resource gxRasterPackOptionsType (gxRasterPackOptionsID, sysHeap,
                                purgeable)
{
    gxPrintingDriverBaseID,      /* ID of start page wstr res */
    gxPrintingDriverBaseID+10,   /* ID of form-feed wstr res */
    /* forward line-feed characteristics */
    98,                          /* max line-feed amount is 98 */
    gxRasterNumToASCII,          /* express line-feed as ASCII */
    2,                          /* minimum width is 2 */
    "0",                        /* pad with zeros */
    "\0X1BT",                   /* <esc>T is set line-feed size */
}
```

Printing Resources

```

"\0x1Br\0x0A" /* <esc>r<lf> is direction reverse, line feed */

/* reverse line-feed characteristics */
98, /* max line-feed amount is 98 */
gxRasterNumToASCII, /* express line-feed as ASCII */
2, /* minimum width is 2 */
"0", /* pad with zeros */
"\0x1BT", /* <esc>T is set line-feed size */
"\0x1Br\0x0A" /* <esc>r<lf> is direction reverse, line feed */
};

resource 'wstr' (gxPrintingDriverBaseID, sysHeap, purgeable)
{
/*
Start page string: unidirectional, 144 dpi ==
ESC 5A 0700 ESC>ESCp
*/
"\0x1BO" "\0x1B>" "\0x1Bp"
};

resource 'wstr' (gxPrintingDriverBaseID+10, sysHeap, purgeable)
{
/* End-page string: a control-L for IW's form feed */
"\0x0C",
};

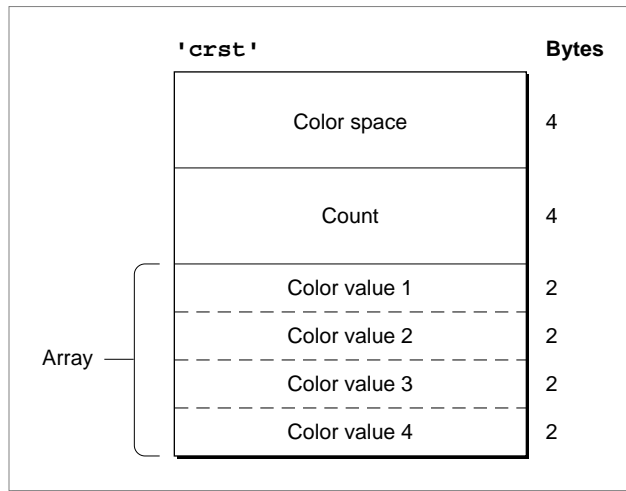
```

If you use the default implementation of the `GXRasterLineFeed` or `GXRasterPackageBitmap` messages, you must define a raster package controls ('ropt') resource. The `GXRasterLineFeed` and `GXRasterPackageBitmap` messages are described in the chapter “Printing Messages” in this book.

The Color Set ('crst') Resource

The color set ('crst') resource is used with raster imaging drivers to specify a set of replacement strings that are used when interpreting the information returned by a printer. This resource is optional.

Figure 6-26 shows the structure of a color set resource.

Figure 6-26 The color set resource

The color set resource consists of a number of color values that are defined for a certain color space.

- **Color space.** The color space that the colors are defined for. Use one of the values described in the chapter “Colors and Color-Related Objects” in *Inside Macintosh: QuickDraw GX Objects*.

- **Count.** The number of color values defined in the array that follows.

Each color entry in the array consists of up to four integer values. What each value specifies depends on the type of color space, as described in the chapter “Colors and Color-Related Objects” in *Inside Macintosh: QuickDraw GX Objects*.

- **Color value 1.** The first color value.
- **Color value 2.** The second color value.
- **Color value 3.** The third color value.
- **Color value 4.** The fourth color value.

Listing 6-24 shows an example of a color set resource for the ImageWriter II printer driver. This printer uses the RGB color space and defines only 8 colors.

Listing 6-24 An example of a color set resource

```
resource gxColorSetResType (gxPrintingDriverBaseID, sysHeap,
                           purgeable)
{
    gxRGBSpace,
    {
        /* R      G      B      unused */
```

Printing Resources

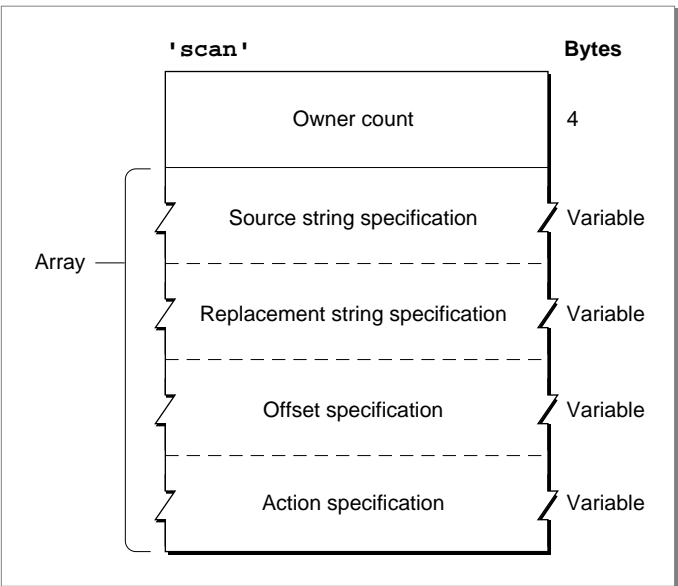
```
0xFFFF,0xFFFF,0xFFFF,0x0000; /* white */
0xFFFF,0xFFFF,0x0000,0x0000; /* yellow */
0xFFFF,0x0000,0xFFFF,0x0000; /* magenta */
0xFFFF,0x0000,0x0000,0x0000; /* red */
0x0000,0xFFFF,0xFFFF,0x0000; /* cyan */
0x0000,0xFFFF,0x0000,0x0000; /* green */
0x0000,0x0000,0xFFFF,0x0000; /* blue */
0xFFFF,0xFFFF,0xFFFF,0x0000; /* white */
0x0000,0x0000,0x0000,0x0000; /* black */
};
};
```

The PostScript Scanning ('scan') Resource

The PostScript scanning ('scan') resource, of type `gxPostscriptScanningType`, is used with PostScript drivers to specify a set of replacement strings that are used when interpreting the status information returned by a PostScript printer. This resource is optional.

Figure 6-27 shows the structure of a PostScript scanning resource.

Figure 6-27 The PostScript scanning resource



The PostScript scanning resource consists of an owner count field and an array of string specifications.

- Owner count.

Printing Resources

Each string specification contains four strings. These strings together form the specification of how to replace strings in the status information that is returned by the printer.

- Source string specification. Specifies the string to be searched for, as described in Table 6-34.
- Replacement string specification. Specifies the string with which to replace the source string.
- Offset specification. Specifies the offset at which to make the replacement, as described in Table 6-35.
- Action specification. Specifies the action to take after making the replacement, as described in Table 6-36.

You define the source string and replacement strings as `StringScan` specifications, which have five variations, as shown in Table 6-34.

Table 6-34 Scan string specifications for the PostScript scanning resource

String type	Value	Explanation	Parameters
<code>SimpleScan</code>	0	The parameter defines the string to uses	A wide string
<code>UserNameScan</code>	1	Uses the name of the user as the string	None
<code>DocumentNameScan</code>	2	Uses the name of the document as the string	None
<code>PrinterNameScan</code>	3	Uses the name of the printer as the string	None
<code>NilPtrScan</code>	4	Uses the empty string	None

You define the offset with an `OffsetScan` specification, which has six variations, as shown in Table 6-35. None of these requires the use of parameters.

Table 6-35 Scan offset specifications for the PostScript scanning resource

Offset type	Value	Explanation
<code>SimpleOffset</code>	0	No offset used—start at the beginning of the buffer
<code>SameAsPreviousOffset</code>	1	Used to insert text immediately before a source string
<code>ReturnedOffset</code>	2	Used to insert text immediately after a source string

Table 6-35 Scan offset specifications for the PostScript scanning resource (continued)

Offset type	Value	Explanation
SimpleRepeat	16	Same as SimpleOffset and the replacement repeats for all occurrences of the source string
SampleAsPreviousRepeat	17	Same as SameAsPreviousOffset and repeats for all occurrences of the source string
ReturnedRepeat	18	Same as ReturnedOffset and repeats for all occurrences of the source string

You define the action with an ActionScan specification, which has two variations, as shown in Table 6-36.

Table 6-36 Scan-action specifications for the PostScript scanning resource

Action type	Value	Explanation	Parameters
NoAction	0	No action taken	None
SimpleAction	1	If an error occurs, a printing alert box is displayed	Two integers The first defines the type of error to generate (normal, nonFatalError, or fatalError) The other specifies the resource ID of the printing alert to use

Listing 6-25 shows the PostScript scanning resources from the Apple LaserWriter IIg printer driver.

Listing 6-25 An example of PostScript scanning resources for a printer driver

```
resource gxPostscriptScanningType (gxPostscriptScanningID,
                                sysHeap, purgeable)
{
    0,
    {
        SimpleScan {"busy"}, SimpleScan { "printer busy" },
        SimpleOffset {}, NoAction {};
        SimpleScan {"waiting"}, SimpleScan {"preparing data"},
        SimpleOffset {}, NoAction {};
        SimpleScan {"job:"}, SimpleScan {"User"},
```

Printing Resources

```

        SimpleOffset {}, NoAction {};
    }
};

resource gxPostscriptScanningType (gxPostscriptScanningID + 1,
                                   sysHeap, purgeable)
{
    0,
    {
        SimpleScan {"%["}, SimpleScan {""},
        SimpleOffset {}, NoAction {};
        SimpleScan {"}%"}, SimpleScan {""},
        SimpleOffset {}, NoAction {};
        SimpleScan {"PrinterError"}, SimpleScan {"Printer"},
        SimpleOffset {}, NoAction {};
    }
};

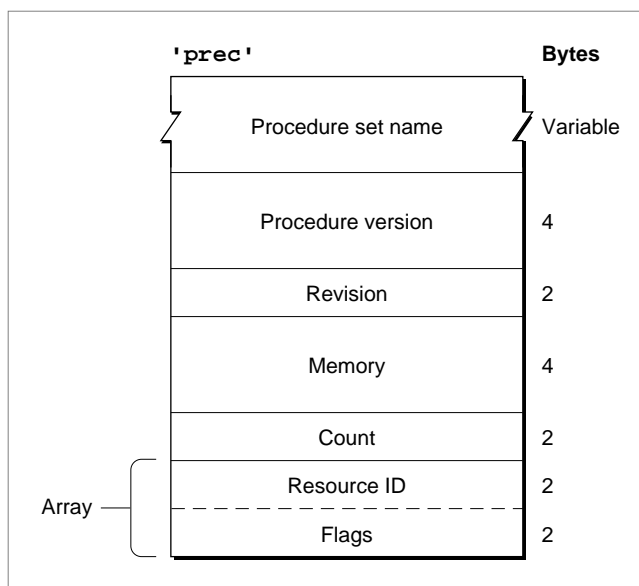
```

The PostScript Procedure Set Control ('prec') Resource

The PostScript procedure set control ('prec') resource, of type `gxPostscriptProcSetControlType`, is used with PostScript drivers to define a procedure set that can be downloaded to a PostScript printer. This resource is optional.

Figure 6-28 shows the structure of a procedure set control resource.

Figure 6-28 The PostScript procedure set control resource



Printing Resources

The procedure set control resource contains naming, version, and memory information about the procedure set, followed by the IDs of the resources that comprise the procedure set.

- Procedure set name. The Pascal string name of the procedure set.
- Procedure set version. The procedure set type. This is a fixed version number.
- Revision. The revision number (an integer) of this procedure set.
- Memory usage. The amount of printer memory required for the procedure set.
- Count. The number of resources that comprise the procedure set.

Each procedure in the set is defined in a resource. Each entry in the array consists of the ID of the procedure resource and a flag that specifies what to do with the procedure data.

- Resource ID. The ID of the resource containing a PostScript procedure.
- Flags. A value that describes what to do with the data in the resource. The constants that you can use in this field are shown in Table 6-37.

Table 6-37 PostScript procedure data flags

Constant	Value	Explanation
donothing	0	The imaging system should send the entire contents of this resource to the output device without modification.
dumpwidestring	1	The imaging system needs to treat the first two bytes (the first short) of the resource as the number of bytes to send to the printer. That many bytes, starting at the third byte, are then sent to the output device without modification.
dumpstringlist	2	The imaging system needs to treat the resource like a string list ('STR#') resource. The imaging system dumps each string after appending a line feed and carriage return to the end of each.
converttohex	0x0100	This is a modifier that specifies that the imaging system needs to convert the data to hexadecimal format before sending it to the output device. You can specify this modifier with the donothing or dumpwidestring flags; however, you cannot use it in conjunction with the dumpstringlist flag.

The string list resource is described in *Inside Macintosh: Macintosh Toolbox Essentials*.

The PostScript Printer Font Type ('pfont') Resource

Each PostScript printer font type ('pfont') resource defines parameters for a specific kind of font that you can use with a PostScript device:

- Adobe™ character set
- Apple character set
- Equivalent character set
- Encoded font

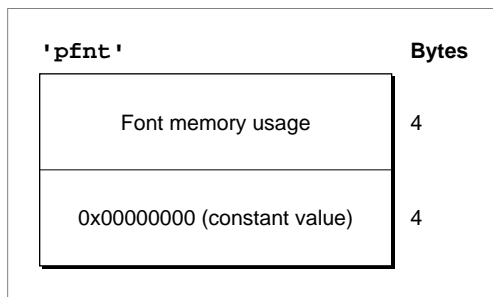
The resources for these kinds of fonts are described in the following sections.

The PostScript Printer Font Type ('pfont') Resource for the Adobe Character Set

The PostScript printer font type ('pfont') resource for the Adobe character set is used with PostScript drivers to specify information about fonts that use the Adobe character set. This resource is optional.

Figure 6-29 shows the structure of a PostScript printer font type resource for the Adobe character set.

Figure 6-29 The PostScript printer font type resource for the Adobe character set



The PostScript printer font type resource for the Adobe character set specifies the memory used by the font and contains a constant value.

- Font memory usage. The amount of memory used for the font.
- The constant value for this resource is 0x00000000.

Listing 6-26 shows several of the PostScript printer font type resources for the Adobe character set from the Apple LaserWriter printer driver.

Listing 6-26 Examples of PostScript printer font type resources for the Adobe character set

```
resource gxPostscriptPrinterFontType (gxPrintingDriverBaseID + 4,
                                     "Helvetica", sysHeap, purgeable)
{
```

Printing Resources

```
        ROMFont,
        AdobeCharacterSet {}};
};

resource gxPostscriptPrinterFontType (gxPrintingDriverBaseID + 5,
                                     "Helvetica-Bold", sysHeap, purgeable)
{
    ROMFont,
    AdobeCharacterSet {}};
};

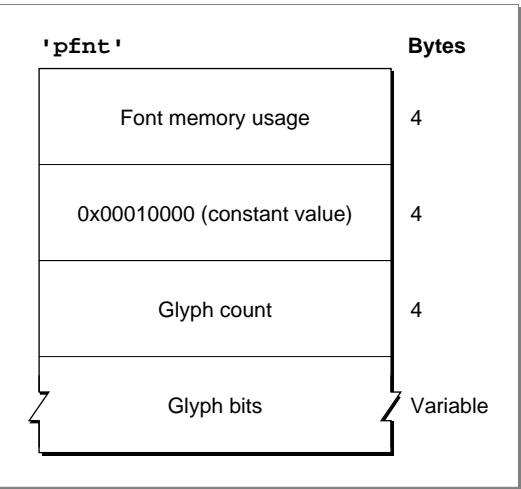
resource gxPostscriptPrinterFontType (gxPrintingDriverBaseID + 6,
                                     "Helvetica-BoldOblique", sysHeap, purgeable)
{
    ROMFont,
    AdobeCharacterSet {}};
};
```

The PostScript Printer Font Type ('pfont') Resource for the Apple Character Set

The PostScript printer font type ('pfont') resource for the Apple character set is used with PostScript drivers to specify information about fonts that use the Apple character set. This resource is optional.

Figure 6-30 shows the structure of a PostScript printer font type resource for the Apple character set.

Figure 6-30 The PostScript printer font type resource for the Apple character set



Printing Resources

The PostScript printer font type resource for the Apple character set specifies the memory used by the font, a constant value, and glyph data.

- Font memory usage. The amount of memory used for the font.
- The constant value for this resource is 0x00010000.
- Glyph count. The number of glyphs defined in this resource.
- Glyph bits. The data for the glyphs of this font.

Listing 6-27 shows a PostScript printer font type resource for the Apple character set from the Apple LaserWriter printer driver.

Listing 6-27 An example of a PostScript printer font type resource for the Apple character set

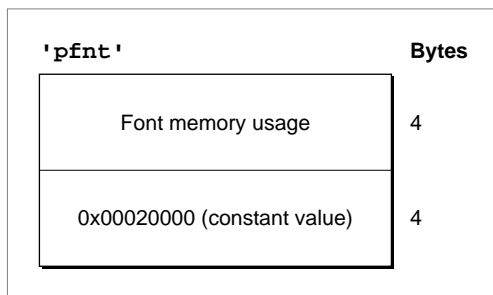
```
resource gxPostscriptPrinterFontType (gxPrintingDriverBaseID + 7,
                                     "Symbol", sysHeap, purgeable)
{
    ROMFont,
    AppleCharacterSet {
        191,
        $"9FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF"
    };
};
```

The PostScript Printer Font Type ('pfnt') Resource for the Equivalent Character Set

The PostScript printer font type ('pfnt') resource for the equivalent character set is used with PostScript drivers to specify information about a font that uses the equivalent character set. This resource is optional.

Figure 6-31 shows the structure of a PostScript printer font type resource for an equivalent character set.

Figure 6-31 The PostScript printer font type resource for an equivalent character set



The PostScript printer font type resource for an equivalent character set specifies the memory used by the font and contains a constant value.

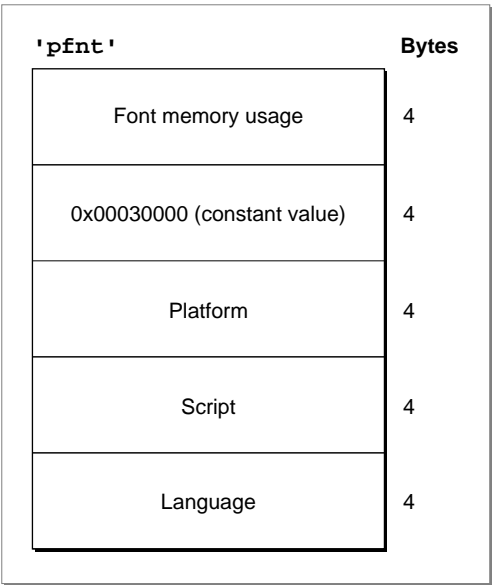
- Font memory usage. The amount of memory used for the font.
- The constant value for this resource is 0x00020000.

The PostScript Printer Font Type ('pfnt') Resource for an Encoded Font

The PostScript printer font type ('`pfnt`') resource for an encoded font is used with PostScript drivers to specify information about encoded fonts. This resource is optional.

Figure 6-32 shows the structure of a PostScript printer font type resource for an encoded font.

Figure 6-32 The PostScript printer font type resource for an encoded font



The PostScript printer font type resource for an encoded font specifies the memory used by the font, a constant value, and information about the platform, script, and language of the font. The platform, script, and language values used in this resource are described in the chapter “Fonts” in *Inside Macintosh: QuickDraw GX Typography*.

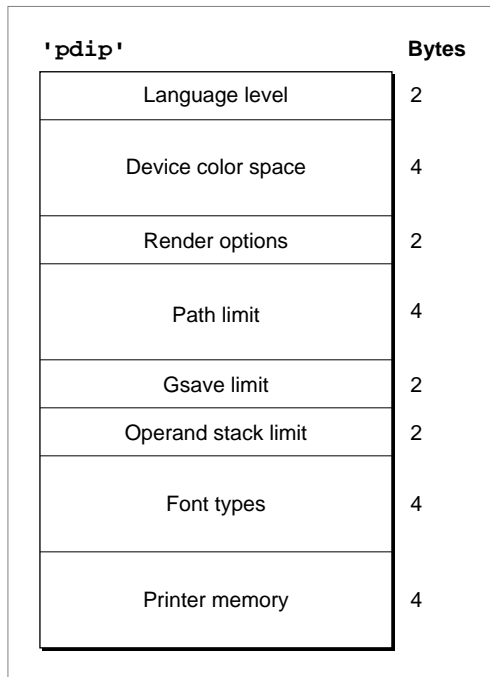
- Font memory usage. The amount of memory used for the font.
- The constant value for this resource is 0x00030000.
- Platform. The platform ID for this font.
- Script. The ID of the script system for this font.
- Language. The language ID for this font.

The PostScript Preferences ('pdip') Resource

The PostScript preferences ('pdip') resource, of type `gxPostscriptPrefsType`, is used with PostScript drivers to specify information about a specific PostScript device. This resource is optional.

Figure 6-33 shows the structure of a PostScript preferences resource.

Figure 6-33 PostScript preferences resource



The PostScript preferences resource describes a number of attributes of the printer:

- **Language level.** The PostScript language level supported by the printer.
- **Device color space.** The color space supported by the printer. The color-space value constants are shown in the chapter “Colors and Color-Related Objects” in *Inside Macintosh: QuickDraw GX Objects*. The values that are valid for this field are `gxGraySpace`, `gxCMYKSpace`, and `gxRGBSpace`.
- **Render options.** Rendering options for the PostScript printer. This value is the combined value of the constants you that you include from the choices shown in Table 6-38.
- **Path limit.** The largest number of points that can be active in the device during the imaging process.
- **Gsave limit.** The greatest number of gsave's that can be performed on the device.

Printing Resources

- **Operand stack limit.** The maximum number of objects that can be placed in the stack at any one time.
- **Font types.** The stream types that the imaging system needs to use when downloading a font. The possible values are shown in Table 6-39.
- **Printer memory.** The amount of memory available in the printer.

Table 6-38 shows the constants that you can use to define the render options field of the PostScript preferences resource.

Table 6-38 PostScript render options

Constant	Value	Explanation
<code>gxNeedsAsciiOption</code>	1	The driver needs to convert all binary data to 7-bit ASCII values.
<code>gxNeedsCommentsOption</code>	2	The driver needs to issue PostScript comments.
<code>gxBoundingBoxOption</code>	4	The driver needs to calculate values for the <code>%%BoundingBox:</code> and <code>%%PageBoundingBox:</code> variables. This option implies the <code>gxNeedsCommentsOption</code> constant.
<code>gxPortablePostScriptOption</code>	8	The driver needs to generate PostScript that is not device specific.
<code>gxUseLevel2ColorOption</code>	128	The driver is to use Level 2 device-independent color when printing to a Level 2 output device.

Table 6-39 shows the values that you can specify in the font types field of the PostScript preferences resource.

Table 6-39 Font stream types

Constant	Value	Explanation
<code>trueTypeStreamType</code>	0x0001	TrueType font format
<code>type1StreamType</code>	0x0002	PostScript Type 1 font format
<code>type3StreamType</code>	0x0004	PostScript Type 3 font format
<code>type42StreamType</code>	0x0008	PostScript Type 4.2 font format
<code>type42GXStreamType</code>	0x0010	GX PostScript Type 4.2 font format
<code>portableStreamType</code>	0x0020	Portable font format
<code>flattenedStreamType</code>	0x0040	Flattened font format

Summary of Printing Resources

C Summary

Constants and Data Types

```

    /* basic client types */
#define gxPrintingManagerType 'pmgr'
#define gxPrinterDriverType   'pdvr'
#define gxPrintingExtensionType 'pext'
#define gxAnyPrinterType      'univ'
#define gxPortableDocPrinterType 'gxpdp'

#define gxRasterPrinterType    'rast'
#define gxPostscriptPrinterType 'post'
#define gxVectorPrinterType    'vect'

#define gxPrintingTagID (-28672) /* gxTag ID for printing collections */

/*
    The following constants are for resources used by both extensions and
    drivers.
*/

    /* base IDs for extension & driver resources */
#define gxPrintingDriverBaseID (-28672)
#define gxPrintingExtensionBaseID (-27136)

#define gxOverrideType 'over' /* override resource type */

#define gxDriverUniversalOverrideID (gxPrintingDriverBaseID)
#define gxDriverImagingOverrideID (gxPrintingDriverBaseID + 1)
#define gxDriverCompatibilityOverrideID (gxPrintingDriverBaseID + 2)

#define gxExtensionUniversalOverrideID gxPrintingExtensionBaseID
#define gxExtensionImagingOverrideSelectorID gxPrintingExtensionBaseID

```


Printing Resources

```

/*
    The following constants are used to identify printing messages for use
    in both extensions and drivers.
*/

    /* identifiers for universal message overrides... */

#define gxInitialize            0
#define gxShutDown            1

#define gxJobIdle              2
#define gxJobStatus            3
#define gxPrintingEvent        4

#define gxJobFormatDialog      5
#define gxFormatDialog         6
#define gxJobPrintDialog       7
#define gxFilterPanelEvent     8
#define gxHandlePanelEvent     9
#define gxParsePageRange      10

#define gxDefaultJob           11
#define gxDefaultFormat        12
#define gxDefaultPaperType     13
#define gxDefaultPrinter       14

#define gxCreateSpoolFile      15
#define gxSpoolPage            16
#define gxSpoolData            17
#define gxSpoolResource        18
#define gxCompleteSpoolFile    19

#define gxCountPages           20
#define gxDespoolPage          21
#define gxDespoolData          22
#define gxDespoolResource      23
#define gxCloseSpoolFile       24

#define gxStartJob             25
#define gxFinishJob            26
#define gxStartPage            27
#define gxFinishPage           28

```

Printing Resources

#define gxPrintPage	29
#define gxSetupImageData	30
#define gxImageJob	31
#define gxImageDocument	32
#define gxImagePage	33
#define gxRenderPage	34
#define gxCreateImageFile	35
#define gxOpenConnection	36
#define gxCloseConnection	37
#define gxStartSendPage	38
#define gxFinishSendPage	39
#define gxWriteData	40
#define gxBufferData	41
#define gxDumpBuffer	42
#define gxFreeBuffer	43
#define gxCheckStatus	44
#define getDeviceStatus	45
#define gxFetchTaggedData	46
#define gxGetDTPMenuList	47
#define gxDTPMenuSelect	48
#define gxDTPHandleAlertFilter	49
#define gxJobFormatModeQuery	50
#define gxWriteStatusToDTPWindow	51
#define gxInitializeStatusAlert	52
#define gxHandleAlertStatus	53
#define gxHandleAlertEvent	54
#define gxCleanupStartJob	55
#define gxCleanupStartPage	56
#define gxCleanupOpenConnection	57
#define gxCleanupStartSendPage	58
#define gxDefaultDesktopPrinter	59
#define gxCaptureOutputDevice	60

Printing Resources

```

#define gxOpenConnectionRetry    61
#define gxExamineSpoolFile      62

#define gxFinishSendPlane       63
#define gxDoesPaperFit          64
#define gxChooserMessage        65

#define gxFindPrinterProfile     66
#define gxFindFormatProfile     67
#define gxSetPrinterProfile     68
#define gxSetFormatProfile      69

    /* identifiers for Macintosh Printing Manager message overrides */
#define gxPrOpenDoc              0
#define gxPrCloseDoc             1
#define gxPrOpenPage            2
#define gxPrClosePage           3
#define gxPrintDefault           4
#define gxPrStlDialog           5
#define gxPrJobDialog            6
#define gxPrStlInit              7
#define gxPrJobInit              8
#define gxPrDlgMain              9
#define gxPrValidate            10
#define gxPrJobMerge             11
#define gxPrGeneral              12
#define gxConvertPrintRecordTo   13
#define gxConvertPrintRecordFrom 14
#define gxPrintRecordToJob       15

    /* identifiers for raster imaging message overrides */
#define gxRasterDataIn           0
#define gxRasterLineFeed         1
#define gxRasterPackageBitmap    2

    /* identifiers for PostScript imaging message overrides */
#define gxPostscriptQueryPrinter 0
#define gxPostscriptInitializePrinter 1
#define gxPostscriptResetPrinter 2
#define gxPostscriptExitServer   3
#define gxPostscriptGetStatusText 4
#define gxPostscriptGetPrinterText 5
#define gxPostscriptScanStatusText 6
#define gxPostscriptScanPrinterText 7

```

Printing Resources

```

#define gxPostscriptGetDocumentProcSetList      8
#define gxPostscriptDownloadProcSetList        9
#define gxPostscriptGetPrinterGlyphsInformation 10
#define gxPostscriptStreamFont                 11
#define gxPostscriptDoDocumentHeader           12
#define gxPostscriptDoDocumentSetUp            13
#define gxPostscriptDoDocumentTrailer          14
#define gxPostscriptDoPageSetUp                15
#define gxPostscriptSelectPaperType            16
#define gxPostscriptDoPageTrailer              17
#define gxPostscriptEjectPage                  18
#define gxPostscriptProcessShape               19

/* identifiers for vector imaging message overrides */
#define gxVectorPackageData                    0
#define gxVectorLoadPens                      1
#define gxVectorVectorizeShape                 2

/* identifiers for status message types */
enum {
    gxNonFatalError = 1,      /* effects icon on spooling dialog */
    gxFatalError = 2,         /* sends up printing alert on spooling dialog */
    gxPrinterReady = 3,       /* signals QuickDraw GX to leave alert mode */
    gxUserAttention = 4,       /* signals initiation of a modal alert */
    gxUserAlert = 5,          /* signals initiation of a moveable modal
                               alert */
    gxPageTransmission = 6,   /* signals page sent to printer, increments
                               page count in strings to user */
    gxOpenConnectionStatus = 7, /* signals QuickDraw GX to begin animation
                               on printer icon */
    gxInformationalStatus = 8, /* default status type, no side effects */
    gxSpoolingPageStatus = 9,  /* signals page spooled, increments page
                               count in spooling dialog */
    gxEndStatus = 10,          /* signals end of spooling */
    gxPercentageStatus = 11    /* signals the QuickDraw GX as to the amount
                               of the job that is currently complete */
};

/*
The following resource types and IDs are used by extensions.
*/

```

Printing Resources

```

#define gxExtensionScopeType  'scop'
#define gxDriverScopeID      gxPrintingExtensionBaseID
#define gxPrinterScopeID     gxPrintingExtensionBaseID+1
#define gxPrinterExceptionScopeID  gxPrintingExtensionBaseID+2

#define gxExtensionLoadType 'load'
#define gxExtensionLoadID    gxPrintingExtensionBaseID

#define gxExtensionLoadFirst  0x00000100
#define gxExtensionLoadAnywhere 0x7FFFFFFF
#define gxExtensionLoadLast   0xFFFFF00

#define gxExtensionOptimizationType 'eopt'
#define gxExtensionOptimizationID  gxPrintingExtensionBaseID

    /* extension optimization values*/

#define gxExecuteDuringImaging      TRUE
#define gxNeedDeviceStatus          TRUE
#define gxChangePageAtGXDespoolPage TRUE
#define gxChangePageAtGXImagePage  TRUE
#define gxChangePageAtGXRenderPage TRUE
#define serverPresenceRequired      FALSE
#define clientPresenceRequired      FALSE
#define dontexecuteDuringImaging    FALSE
#define dontneedDeviceStatus        FALSE
#define dontchangePageAtDespoolPage FALSE
#define dontchangePageAtImagePage  FALSE
#define dontchangePageAtRenderPage FALSE
#define notServerPresenceRequired   FALSE
#define notClientPresenceRequired   FALSE

/*
    The following resource types and IDs are used by writers of printer
    drivers.
*/

    /* imaging Resources */
#define gxImagingSystemSelectorType 'isys'
#define gxImagingSystemSelectorID(gxPrintingDriverBaseID)

    /* raster rendering preferences resources */
#define gxRasterPrefsType  'rdip'
#define gxRasterPrefsID    gxPrintingDriverBaseID

```

Printing Resources

```

/* resource type for specifying a color set*/
#define gxColorSetResType      'crst'

/* resource type and ID for raster generic driver packaging preferences */
#define gxRasterPackType      'rpck'
#define gxRasterPackID        gxPrintingDriverBaseID

/* resource type and ID for raster generic driver packaging options */
#define gxRasterNumNone      0 /* number isn't output at all */
#define gxRasterNumDirect    1 /* lowest minWidth bytes as data */
#define gxRasterNumToASCII    2 /* minWidth ASCII characters */

#define gxRasterPackOptionsType 'ropt'
#define gxRasterPackOptionsID   gxPrintingDriverBaseID

/* resource type for the PostScript procedure set control resource */
#define gxPostscriptProcSetControlType 'prec'

/* resource type for the PostScript printer gxFont resource */
#define gxPostscriptPrinterFontType 'pfnt'

/* resource type and id for the PostScript imaging preferences */
#define gxPostscriptPrefsType    'pdip'
#define gxPostscriptPrefsID      gxPrintingDriverBaseID

/* resource type and id for the PostScript default scanning code */
#define gxPostscriptScanningType 'scan'
#define gxPostscriptScanningID0

/* resource for type for color matching */
#define gxColorMatchingDataType  'prof'
#define gxColorMatchingDataID    gxPrintingDriverBaseID

/* resource type and id for the default bin and paper specifications */
#define gxTrayCountDataType      'tray'
#define gxTrayCountDataID        gxPrintingDriverBaseID

/*
The following resource types and IDs are used to define input and output
parameters for printer drivers.
*/

/* resource type and ID for default IO and buffering resources */
#define gxUniversalIOPrefsType  'iobm'
#define gxUniversalIOPrefsID    gxPrintingDriverBaseID

```

Printing Resources

```

/* resource defines for default implementation of */
/*  GXCaptureOutputDevice, which only handles PAP devices */
#define gxCaptureType          'cpts'
#define gxCaptureStringID      (gxPrintingDriverBaseID)
#define gxReleaseStringID      (gxPrintingDriverBaseID + 1)
#define gxUncapturedAppleTalkType(gxPrintingDriverBaseID + 2)
#define gxCapturedAppleTalkType (gxPrintingDriverBaseID + 3)

/* resource type and ID for driver papertypes in individual files */
#define gxSignatureType        'sig '
#define gxPapertypeSignatureID  0

/* file type for driver papertypes placed in individual files */
#define gxDrvvrPaperType       'drpt'

/*
   The following resource types and IDs are used to support Macintosh
   Printing Manager compatibility.
*/

#define gxCustType             'cust'
#define gxCustID               -8192

#define gxReslType             'resl'
#define gxReslID               -8192
#define gxDiscreteResolution 0

#define gxStlDialogResID       -8192
#define gxJobDialogResID       -8191

```

The Buffering and I/O Preferences Structure

```

struct gxIoPrefsRec {
    unsigned long  communicationsOptions; /* options for input & output */
    unsigned long  numBuffers;             /* number of buffers to allocate */
    unsigned long  bufferSize;            /* number of bytes per buffer */
    unsigned long  numReqBlocks;          /* number of I/O request blocks */
    unsigned long  openCloseTimeout;      /* timeout for open and close */
    unsigned long  readWriteTimeout       /* timeout for read and write */
};

typedef struct gxIoPrefsRec gxIoPrefsRec, *gxIoPrefsPtr, **gxIoPrefsHdl;

```

The Customization Structure

```

struct gxCustomizationRec {
    short horizontalResolution; /* horizontal resolution */
    short verticalResolution;   /* vertical resolution */
    short upDriverType;         /* type of Macintosh Printing Manager
                                interface to use */
    Point patternStretch;       /* pattern stretching factor */
    short translatorSettings    /* settings for translator */
};

typedef struct gxCustomizationRec gxCustomizationRec, *gxCustomizationPtr,
**gxCustomizationHdl;

```

The Resolution Structure

```

struct gxResolutionRec {
    short    rangeType;          /* always 1 */
    short    xMinimumResolution;
    short    xMaximumResolution;
    short    yMinimumResolution;
    short    yMaximumResolution;
    short    resolutionCount;
    Point    resolutions[1];     /* array of points */
};

typedef struct gxResolutionRec gxResolutionRec, *gxResolutionPtr,
**gxResolutionHdl;

```

Raster Preferences Structure

```

struct gxRasterPrefsRec {
    gxRasterRenderOptions
        renderOptions;          /* raster imaging options */
    Fixed    hImageRes;          /* horiz imaging resolution */
    Fixed    vImageRes;          /* vert imaging resolution */
    short    minBandSize;        /* minimum band size to use */
    short    maxBandSize;        /* maximum band size to use */
    Fixed    ramPercentage;      /* maximum percentage of RAM to use */
    long     ramSlop;             /* minimum RAM to leave free */
    short    depth;              /* depth, in pixels, per plane */
    short    numPlanes;          /* number of planes to render */
};

```


Printing Resources

```

    gxPlaneSetupRec
        planeSetup[1];    /* one for each plane */
};

typedef struct gxRasterPrefsRec gxRasterPrefsRec, *gxRasterPrefsPtr,
**gxRasterPrefsHdl;

```

Raster Render Options

```

typedef long gxRasterRenderOptions;

enum {
    gxDefaultRaster      = 0x00000000,  /* default options */
    gxDontResolveTransferModes
                        = 0x00000001,  /* 0 means resolve, 1 means don't */
    gxRenderInReverse    = 0x00000002,  /* traverse in reverse */
    gxOnePlaneAtATime    = 0x00000004,  /* render each separately */
    gxSendAllBands       = 0x00000008  /* send all bands, even if empty */
};

```

Raster Package Structure

```

struct gxRasterPackageRec {
    Ptr      bufferSize;  /* buffer size of packaging */
    short    colorPasses; /* number of color passes */
    short    headHeight;  /* height of print head in pixels */
    short    numberPasses; /* number of passes per head height */
    short    passOffset;  /* offset between passes, in pixels */
    gxRasterPackageOptions
        packageOptions; /* packaging options */
};

typedef struct gxRasterPackageRec gxRasterPackageRec, *gxRasterPackagePtr,
**gxRasterPackageHdl;

```

Raster Package Options

```

enum {          /* bit fields in gxRasterPackageOptions */
    gxSendAllColors    = 0x00000001,  /* send all bands, even if empty */
    gxInterlaceColor    = 0x00000002,  /* ribbon contamination */
    gxOverlayColor      = 0x00000004,  /* no ribbon problem */
    gxUseColor          = (gxInterlaceColor|gxOverlayColor);
};

typedef long gxRasterPackageOptions;

```

Raster Package Controls Structure

```

struct gxRasterPackageControlsRec {
    short    startPageStringID; /* ID of string to send at start of page */
    short    formFeedStringID; /* ID of string to send for form feed */
    short    forwardMax;        /* maximum amount of forward line feed */
    gxStandardNumberRec
        forwardLineFeed; /* number struct to express line feed */
    short    reverseMax;        /* maximum amount of reverse line feed */
    gxStandardNumberRec
        reverseLineFeed; /* number struct to express reverse line
                           feed */
};

typedef struct gxRasterPackageControlsRec gxRasterPackageControlsRec,
*gxRasterPackageControlsPtr, **gxRasterPackageControlsHdl;

```

Standard Number Structure

```

struct gxStandardNumberRec {
    short    numberType; /* type of numeric output desired */
    short    minWidth;    /* minimum output width of number */
    char     padChar;     /* pad character */
    char     alignment;
    Str31    startString; /* the prefix string */
    Str31    endString;   /* the postfix string */
};

typedef struct gxStandardNumberRec gxStandardNumberRec, *gxStandardNumberPtr;

```