Technical Note TN2049

'pdat' specification

CONTENTS

This Technote describes the `'pdat'` resource that printer vendors must include in their drivers to support CarbonLib printing under Mac OS 8 and 9.

[Apr 10 2002]

Introduction

Mac OS X introduced a new printing system that is significantly different from the Classic Printing Manager available on Mac OS 8 and 9. The Carbon Printing Manager provides a bridge between these two architectures, allowing Carbon applications to print on both platforms.

On Mac OS 8 and 9, the Carbon Printing Manager is built into CarbonLib. Carbon applications running on Mac OS 8 and 9 continue to print using Classic printer drivers, and most Carbon Printing Manager functions call through to their Classic counterparts. On Mac OS X, the Printing Framework belongs to the Carbon umbrella framework. When running on Mac OS X, Carbon applications automatically use the new printing system.

The Carbon Printing Manager is designed to make the transition from Classic to Carbon as straightforward as possible. While the function names have changed, there is a high level of correspondence between Carbon and Classic printing functions, and the basic print loop model has been preserved.

Because it does not fundamentally alter the structure of an application's print loop, adopting the Carbon Printing Manager in an application is relatively simple. The most significant change is that the data structures corresponding to the Classic print record are opaque in Carbon - meaning their contents are hidden from applications. There are new functions for accessing fields within these structures. Most other Carbon Printing Manager functions are direct replacements for Classic Printing Manager functions.

In order to support those new functions, the Carbon Printing Manager needs to know where your Classic printer driver stores those settings in its Classic Print Record. That's where the `'pdat'` resource comes in.

There are 3 categories of Carbon Printing Manager functions:

1. Functions that map directly to Classic Printing Manager functions. For example, `PMSessionPageSetupDialog` calls `PrStlDialog`.
2. Functions that are reserved for Mac OS X only. For example, `PJCSetPSInjectionData`. Note that some functions in this category are in CarbonLib and return `kPMNotImplemented` (like `PMSessionUseSheets`) while others aren't in CarbonLib at all (for example, `PMSessionCreatePageFormatList`) and will cause link errors.
3. Functions that have no Classic Printing Manager equivalent and cause the Carbon Printing Manager to directly

access the Classic driver's print record. For example, PMSetOrientation and PMSetScale.

The purpose of 'pdat' resources is to support this third category of functions. Some examples include:

```
PMSessionValidatePageFormat
PMSessionValidatePrintSettings
PMSessionConvertOldPrintRecord
PMSessionMakeOldPrintRecord
PMGetScale
PMSetScale
PMGetResolution
PMSetResolution
PMGetOrientation
PMSetOrientation
PMGetCopies
PMSetCopies
PMGetColorMode
PMSetColorMode
```

Important notes:

The session functions have corresponding non-session counterparts that fall into the same categories. However, the non-session APIs have been deprecated in favor of the session APIs.

Regarding category (1) functions, the Carbon Printing Mgr depends on Classic drivers adhering to the Classic Printing Mgr specification. This means drivers **MUST** use TPrint fields as defined in Chapter 9 of *Inside Macintosh* : Imaging with QuickDraw.

In addition, unlike the Classic Printing Manager, where both Page Setup and Print settings are stored in one data structure, the Carbon Printing Manager maintains two objects, one for Page Setup settings and another for Print settings. When it is necessary to update the Classic print record, the Carbon Printing Manager calls PrJobMerge() to copy settings from its Page Setup and Print Settings objects. For this reason, the Classic print driver must implement PrJobMerge properly. When PrJobMerge isn't working properly, you'll notice Page Setup and/or Print settings that aren't preserved after a call to PrJobMerge.

For example:

```
PrStlDialog(prec1);  // turn stuff on in page setup
PrJobDialog(prec2);  // turn stuff on in print dialog
PrJobMerge(prec2,prec1);

PrStlDialog(prec1);  // are settings from the
                     // first page setup preserved?

PrJobDialog(prec1);  // does this have all the settings
                     // from the first job dialog?
```

The external view of what PrJobMerge is supposed to do is described in the QuickDraw Printing Manager Reference and as indicated in Technote 1161: Extending the Print Record for LaserWriter 8, your PrJobMerge implementation must properly handle the fields in extended print records.

When PrJobMerge is called, the destination print record contains the Page Setup settings. If the destination record validates properly, then you should only copy over your job-specific print dialog settings. If the destination record fails validation, you should copy over all of the settings from the source record.

Back to top

What is a 'pdat' resource

A 'pdat' resource is a data structure that defines the location and encoding of settings in the Classic Print Record (TPrint) that are used by Carbon Printing Mgr functions.

The CarbonLib system extension contains a database of 'pdat' resources to support "legacy" Classic printer drivers. The organization of this database is described in the following sections.

> **Important:**
> Developers are expected to add 'pdat' resources to all printer drivers released after CarbonLib version 1.4.5 as Apple will no longer be adding new 'pdat' resources to the CarbonLib extension.

Back to top

Format for the 'pdat' resource

A 'pdat' resource contains information about the data format and location of Print Record settings such as orientation, scaling, color modes, etc.

The 'pdat' resource format is as follows:

1. Each setting is identified by a 4-byte tag name (listed below).
2. The location of a setting is not restricted to a contiguous range of bits.
3. For settings that can be represented by enumerated types (for example, page orientation) the 'pdat' resource specifies the value of the "ON" setting. This is described below in the section on 'Type1' entries.
4. For settings that are stored as scalar values (for example, drawing resolution), the 'pdat' resource specifies the data format. See the description of 'Type2' entries below.

Back to top

Tags

There are 11 tags to identify settings that the Carbon Printing Manager needs to access in your driver's Print Record. They are:

```
#define pdatPortrait 'port' // required
#define pdatLandscape 'land'
#define pdatReversePortrait 'rpor'
#define pdatReverseLandscape 'rlan'
#define pdatBlackAndWhite 'blkw' // required
#define pdatGrayscale 'gray'
#define pdatColormode 'colr'
#define pdatScale 'scal' // required
#define pdatHRes 'hres' // required
#define pdatVRes 'vres' // required
#define pdatCopies 'copy' // required
```

We recommend providing data for all 11 tags. Of these, 6 are mandatory.

Back to top

Type1 data format

The Carbon Printing Manager recognizes two types of Print Record settings. 'Type1' settings are for enumerated types like page orientation (Portrait, Landscape, and sometimes Reverse Portrait and Reverse Landscape). 'Type2' settings are for parameters like scaling and resolution.

Each 'Type1' setting consists of a sequence of 1 or more entries, defined as follows:

```
struct PdatRecord1Data {
 UInt16 format;  // 0-bits 1-1byte 2-2bytes
 UInt16 offset;  // offset into the print record
 UInt16 mask;  // mask to apply to the offset area
 UInt16 value;  // value to activate this mode
};
typedef struct PdatRecord1Data PdatRecord1Data;

struct PdatRecord1 {  // used to switch modes on and off
 UInt32 tag;  // the tag
 UInt16 count;  // 1-n how many of the following entries
    // there are to switch on the tag's mode
 PdatRecord1Data entry[1]; // must have at least one
};
typedef struct PdatRecord1 PdatRecord1;
```

This structure gives us full control over 'Type1' settings. For example, your driver may use 2 bits (non-contiguous) in the Print Record to store the page orientation setting. In this case, the 'pdat' record would contain a count of 2. One entry would specify where the Portrait/Landscape bit was located and the meaning of the bit value (e.g., 1=Portrait, 0=Landscape). The other entry would specify the Reverse setting (e.g., 0=Normal, 1=Reverse).

Back to top

Type2 data format

'Type2' entries are used for settings which take a scalar value, such as resolution and scaling. A 'Type2' entry looks like this:

```
struct PdatRecord2 {  // used to set values
 UInt32  tag;  // the tag
 UInt16  format; // 0-bits 1-byte 2-bytes 4-bytes (how the data is stored
 UInt16  offset; // offset into the print record
 UInt32  mask;  // mask to apply to the offset area
 UInt16  adjustment; // decimal pt adjustment
};
typedef struct  PdatRecord2 PdatRecord2;

This structure enables us to handle various data formats.  The formula for
getting a setting's value out of the Print Record is:
    ((value at offset location) & ~mask)/adjustment
and to change a setting's value:
    ((value to set)*adjustment) & mask
```

Back to top

Locating the'pdat' resource

The Carbon Printing Manager applies the following rules to find and load a'pdat' record for your printer driver:

1. Look up the current printer driver by getting 'STR ' -8192 from the System file.
2. Locate the printer driver file and - if present - load its 'pdat' resource. Future versions of your drivers should include their own 'pdat' resource. Note that the resource ID for the 'pdat' resource doesn't matter since the Carbon Printing Manager loads your 'pdat' by type and not by resource ID. However, you should make sure you have only one 'pdat' resource in your driver.
3. If the driver does not contain a 'pdat' resource, get the driver's creator type and NumVersion field from its 'vers' 1 resource. This information is used to a create a key into the Carbon Printing Manager's database. Note that this means that each printer driver must have a unique creator code. At one point Apple was accepting 'pdat' resources for existing drivers for inclusion in the Carbon Printing Manager's database but that is no longer the case. For existing drivers, you should revise your driver and include a 'pdat' resource.
4. Concatenate the creator type and the version data and look for a matching string in the CarbonLib database. If that doesn't work, ignore the 'vers' string and look for a match using only the driver's creator type.
5. If a match is found, load the corresponding 'pdat' resource and use it to locate the needed Print Record parameters.
6. If no match is found, the Carbon Printing Manager will follow the procedure described in the next section of this specification (see below).

Note:
Multiple keys in the 'pdst' resource in CarbonLib can point to a single 'pdat' entry enabling a group of drivers to share the same 'pdat' data.

Back to top

Support for drivers without'pdat' resources

In the case where a 'pdat' resource is unavailable for the current driver, the Carbon Printing Manager will fall back to 2 default 'pdat' resources. If the driver's type (prStl.wDev) is 3, we use a 'pdat' based on the LaserWriter 8 driver. For other drivers, we use a 'pdat' based on Apple's StyleWriter driver. There will be compatibility problems, especially with raster printer drivers, which is why it is important for you to provide us with 'pdat' information for your drivers.

Back to top

Other issues

We've had issues with some printer drivers where, for example, our grayscale and color mode tags don't map to their driver's options. Map your driver's capabilities to our settings as best you can, making sure that when a user selects monochrome (grayscale or B&W) they don't get color, and vice versa.

Other drivers keep scalar quantities (copies, for example) in more than one place. Such print records can't be represented by a 'pdat'. Scalar quantities should be stored in only one place.

Some developers have asked how to access settings in the print record that aren't represented in the 'pdat' data structure. The sole purpose of the 'pdat' resource is to support application APIs that access generic features like copies, scaling, etc. As such, 'pdat' resources don't support access to any extra settings.

Some printer drivers are used to handling features like copies, for example, that are now handled by Carbon applications themselves. The PrintLoop sample in /Developer/Examples/Printing/App provides the definitive example of how Carbon applications should handle printing. You should make sure your driver compliments that model.

Back to top

Development Tools

Apple has provided a 'pdat' tool to help you create your 'pdat' resources for inclusion with your driver. You can download it by clicking on the link in the downloads section below.

pdat Creator Window

The top left corner of the window contains the driver name and below that are the two possible tags for the driver. The middle list box contains the available pdat tags which you can create. Required tags are always selected for saving and are dimmed. You should check off the other features which your driver supports. You can only test a checked tag and only checked tags will be saved. The pdat Creator Window also indicates if it found a pdat associated with the driver.

Print Record Window

This window is used when testing a tag. It will highlight the differences between a default print record and one that has been set with the 'pdat' tag data. The program will bring up the appropriate printing dialog for confirmation. Option-clicking over an entry will bring up a popup menu which allows you to change the display from decimal to hex and back. Option-clicking in an empty space of the window will toggle between decimal and hex for all entries

Saving a pdat file

Clicking this button will bring up the save options available for creating a 'pdat' rsrc file. Use the default settings.

A sample .r file for the LaserWriter is included along with the necessary type description for the rez and derez tool in MPW.

Back to top

Summary

It is very important that you provide 'pdat' resources for your printer drivers. Otherwise users that run Carbon applications on Mac OS 9 will have problems printing to your printer. All new drivers and revisions of existing drivers should contain a 'pdat' resource since Apple won't be adding any more 'pdat' resources to CarbonLib.

Here are the important things you need to do to make sure your Classic printer driver properly supports the Carbon Printing Manager:

1. Include one and only one 'pdat' resource in your driver.
2. The 'pdat' resource ID doesn't matter.
3. Make sure that PrJobMerge works correctly.
4. Make sure your driver uses the TPrint fields as defined in Inside Mac: Imaging with QuickDraw, Chapter 9.
5. Make sure your driver has a unique creator code.

Back to top

Appendix A

'pdat' resources in CarbonLib 1.4.5

The following is a list of the 'pdat' resources included with CarbonLib 1.4.5 and their associated printer drivers/manufacturers.

'pdat' Resource IDs	Associated Printer Driver
-6001 to -6000	ALPS
-5000	PrintToPDF
-4012 to -4000	Epson
-3001 to -3000	HP DeskJets
-2020 to -2000	Canon BubbleJets
-1004 to -1002	Color StyleWriter
-1001	StyleWriter
-1000	LaserWriter

Back to top

Appendix B

The LaserWriter 8 'pdat'

```
resource 'pdat' (12888, "LaserWriter") {
 { /* array PDatRecords: 10 elements */
  /* [1] */
  Copies {
```

```
 2,
 46,
 0xFFFFFFFF,
 1
},
/* [2] */
HRes {
 2,
 6,
 0xFFFFFFFF,
 1
},
/* [3] */
VRes {
 2,
 4,
 0xFFFFFFFF,
 1
},
/* [4] */
Scale {
 2,
 50,
 0xFFFFFFFF,
 1
},
/* [5] */
BlackAndWhiteMode {
 { /* array PdatDataType1: 1 elements */
  /* [1] */
  0,
  83,
  0x8,
  0
 }
},
/* [6] */
Portrait {
 { /* array PdatDataType1: 1 elements */
  /* [1] */
  0,
  25,
  0x2,
  2
 }
},
/* [7] */
ColorMode {
 { /* array PdatDataType1: 1 elements */
  /* [1] */
  0,
  83,
  0x8,
  8
 }
},
/* [8] */
Landscape {
 { /* array PdatDataType1: 1 elements */
  /* [1] */
  0,
  25,
  0x2,
  0
 }
},
/* [9] */
ReversePortrait {
 { /* array PdatDataType1: 1 elements */
  /* [1] */
  0,
  25,
  0x2,
  2
 }
},
/* [10] */
ReverseLandscape {
   { /* array PdatDataType1: 1 elements */
    /* [1] */
    0,
    25,
    0x2,
    0
   }
  }
 }
};
```

Back to top

Appendix C

The Apple StyleWriter 'pdat'

```
resource 'pdat' (1000, "StyleWriter") {
 { /* array PDatRecords: 9  elements */
  /* [1] */
  Copies {
   2,
   46,
   0xFFFFFFFF,
   1
  },
  /* [2] */
  HRes {
   2,
   6,
   0xFFFFFFFF,
   1
  },
  /* [3] */
  VRes {
   2,
   4,
   0xFFFFFFFF,
   1
  },
  /* [4] */
  Scale {
   2,
   50,
   0xFFFFFFFF,
   1
  },
  /* [5] */
  BlackAndWhiteMode {
   { /* 2 elements */
    /* [1] */
    1,
    48,
    0xFFFF,
    0x0,
    /* [2] */
    1,
    102,
    0xFF,
    0x1
   }
  },
  /* [6] */
  Portrait {
   { /* 1 elements */
    /* [1] */
    0,
    25,
    0x2,
    0x2
   }
  },
  /* [7] */
  GrayscaleMode {
   { /* 2 elements */
```

```
    /* [1] */
    1,
    48,
    0xFF,
    0x1,
    /* [2] */
    1,
    102,
    0xFF,
    0x8
   }
  },
  /* [8] */
  ColorMode {
   { /* 2 elements */
    /* [1] */
    1,
    48,
    0xFF,
    0x1,
    /* [2] */
    1,
    102,
    0xFF,
    0x20
   }
  },
  /* [9] */
  Landscape {
   { /* 1 elements */
    /* [1] */
    0,
    25,
    0x2,
    0x0
   }
  }
 }
};
```

Back to top

References

Inside Macintosh: Imaging With QuickDraw

Carbon Printing Manager

Back to top

Downloadables

□	Acrobat version of this Note (72K).	Download
	Pdat tool v.6 sample (176K).	Download

Back to top

Technical Notes by Date | Number | Technology | Title
Developer Documentation | Technical Q&As | Development Kits | Sample Code