

Technical Note TN1133

The New PrGeneral Version Opcode

CONTENTS

[The New PrGeneral Opcode](#)

[What Printer Driver Developers Need to Know](#)

[What Application Developers Need to Know](#)

[Go Forth and Print!](#)

[wDev Registration](#)

[References](#)

[Change History](#)

[Downloadables](#)

The Settings Library (`SettingsLib`), introduced in LaserWriter 8.5.1, allows applications to access and change the information created by LaserWriter 8 when a desktop printer is created. `SettingsLib` prevents contention over the printer database between different parts of a printer driver, different printer drivers, and other printer database clients such as the Desktop Print Monitor. The printer database is stored in the LaserWriter 8 preferences file. The Settings Library is especially useful for applications that previously relied on the format of the `PAPA` resource in the LaserWriter 8 driver. With the introduction of LaserWriter 8.5.1, the size of the `PAPA` resource changed, and applications that depended upon its size broke. Since the resource size may change again in the future, Apple has introduced this library to prevent such problems. This Technote overviews the Settings Library APIs.

Updated: [Jun 01 1998]

For the past 12 years, printer driver developers have used the print record's `wDev` field in the `TPPrSt1` data structure to store their device information. Specifically, they've consistently used the high-order byte of this field -- commonly known as the `wDev ID` -- to store the device code. Unfortunately, due to the design limitations of the print record and the `wDev` field, the Printing Manager allowed for a maximum of only 256 IDs. Surprisingly enough (ha ha), with the expansion of the printer market, we're running out of numbers! As a result, with LaserWriter driver version 8.4, we've added a `PrGeneral` opcode and new structures to the Printing Manager so developers can work around this limitation.

This Technote is directed at both printer driver and application developers. You need to add code to support this new information, but the changes are minimal. Below is sample code that shows you how to retrieve the new version information when your application is looking for the features of a particular printer driver properly.

[Back to top](#)

The New PrGeneral Opcode: kPrVersionOp

LaserWriter driver version 8.4 introduced several new `PrGeneral` opcodes, including a new opcode for driver version information: `kPrVersionOp`. Using `kPrVersionOp` is now the recommended way for applications to find out the printer driver currently in use. Remember that for printer drivers that haven't been updated to use this new `PrGeneral` opcode, your application needs to keep its `wDev`-checking code as well.

For more information on the extended print record opcodes introduced with LaserWriter version 8.4, read the Print Hints column [The All-New LaserWriter Driver Version 8.4](#) in *develop* Issue 27. There you'll find a table that lists all the other `PrGeneral` opcodes supported in Apple printer drivers. See the article [Meet PrGeneral, the Trap That Makes the Most of the Printing Manager](#) in *develop* Issue 3 for more information on `PrGeneral`.

[Back to top](#)

What Printer Driver Developers Need to Know

From now on, Apple's Developer Support Center will assign all printer drivers a `wDev` ID of 255. If you have an old `wDev` ID number (which is less than 255), you should still use the number you were assigned so that you don't break any applications that vary their behavior based on your driver's `wDev`. In addition, you should implement the new `kPrVersionOp` opcode as described here so that new applications can take advantage of it with your driver.

Listing 1 shows the new opcode and structures that you'll need to implement in your PDEF 7 resource. You need to place your creator type in the `driver.creator` field of the `TPrVersion` structure. You should also place your version number in `driver.release` so that application developers know what version of the driver is available.

The release and interfaces fields have the same format as the first long in a 'vers' resource -- that is, the following four 1-byte values: a major version number, a minor version number, a revision stage (development = 0x20, alpha = 0x40, beta = 0x60, and release = 0x80), and a prerelease version number.

Listing 1. The new version opcode and structures

```
enum {
    ...
    kPrVersionOp = 22          // LaserWriter version 8.4 and later only
};
typedef struct {
    OSType architecture;      // Architecture (for example, 'pwpc', 'm68k')
    OSType creator;           // Driver's creator type (for example, 'AAPL')
    unsigned long release;    // Release number (for example, 0x8480)
    unsigned long interfaces; // Interfaces version (for example, 0x8480)
} TVersion;

typedef struct {
    short iOpCode;            // Must be kPrVersionOp
    short iError;             // For returning errors
    long lReserved;           // Reserved, set to 0
    TVersion driver;          // Driver information
    TVersion reserved;        // Reserved, set to 0
} TPrVersion;
```

That's all you'll need to do to support the new `PrGeneral` opcode in your driver--hard to believe, but true!

[Back to top](#)

What Application Developers Need to Know

Although Apple doesn't recommend that application developers check the `wDev` field of the print record, many do. They assume that the `wDev` value means they can count on certain features being present in the printer. For instance, many developers assume that a `wDev` of 3 means the printer driver supports PostScript™. In fact, it indicates only that the current printer uses the Apple LaserWriter driver. This leaves out the possibility that a user has a non-Apple PostScript printer driver with an ID of, say, 243.

Even though Apple doesn't recommend that application developers rely on the `wDev` field (or the new `PrGeneral` opcode, for that matter), if you're going to change application behavior based on the number or creator returned, at least do so properly so that your applications don't break with future releases of printer drivers. We recommend that you use the `PrGeneral kPrVersionOp` opcode, if available, to get the structure and then check the `driver.creator` and `driver.release` fields to determine the current printer driver and version (see Listing 1). You should also continue to support the old `wDev` method for printer drivers that have not yet updated to the new `PrGeneral` opcode. For your reference, the `PrGeneral` Version sample that accompanies this column shows how to retrieve the version and `wDev` information properly.

[Back to top](#)

Go Forth and Print!

As you can see, the changes that printer driver and application developers need to make to support the new `PrGeneral` opcode are minimal. We've already adopted this new method with the LaserWriter version 8.4.x drivers. We hope you will, too!

[Back to top](#)

wDev Registration

Although from all printer drivers will now be assigned a `wDev` ID of 255, we still request that you register your printer driver with Apple's Developer Support Center to avoid creator type conflicts. To do so, register at our [wDev registration page](#). You need to provide the following information:

- Company name and address
- Contact name, phone number, and e-mail address
- Make and model of device supported (for example, Apple LaserWriter 16/600)
- Type of connection (for example, serial)
- Creator type (for example, 'AAPL')

You'll receive confirmation of your registration within five to ten working days. If you have questions about `wDev` registration, write to devprograms@apple.com.

[Back to top](#)

References

[Print Hints: The All-New LaserWriter Driver Version 8.4](#) by Dave Polaschek, *develop* Issue 27.

[Meet PrGeneral, the Trap that Makes the Most of the Printing Manager](#) by Pete "Luke" Alexander, *develop* Issue 3.

Change History

01-June-1997

Originally published in *MacTech* magazine.

[Back to top](#)

Downloadables



Acrobat version of this Note (60K).

[Download](#)

[Back to top](#)

Technical Notes by [API](#) | [Date](#) | [Number](#) | [Technology](#) | [Title](#)
[Developer Documentation](#) | [Technical Q&As](#) | [Development Kits](#) | [Sample Code](#)