

yesyesyesyesTRUEnono&AboutC&loseC&opy&PrintnoPrinter Dialog  
Testprinteryes20/03/95

## Table of Contents

**Version 4**  
**3/22/95**

On the "Fly" Printer Setup in Access 2.0

"On the Fly" release history: revisions dated 1/8/95, 1/28/95, and 3/19/95 as  
modifications to original Functions

Help file produced by **HELLLP!** v2.3a , a product of Guy Software, on 3/20/95 for Unregistered User.

The above table of contents will be automatically completed and will also provide an excellent cross-reference for context strings and topic titles. You may leave it as your main table of contents for your help file, or you may create your own and cause it to be displayed instead by using the I button on the toolbar. This page will not be displayed as a topic. It is given a context string of `__` and a HelpContextID property of 32517, but these are not presented for jump selection.

HINT: If you do not wish some of your topics to appear in the table of contents as displayed to your users (you may want them ONLY as PopUps), move the lines with their titles and contexts to below this point. If you do this remember to move the whole line, not part. As an alternative, you may wish to set up your own table of contents, see Help under The Structure of a Help File.

Do not delete any codes in the area above the Table of Contents title, they are used internally by HELLLP!

## On the Fly Printer Setup in Access 2.0

A major hurdle for developing Microsoft Access applications which are high on usability, is the inability for the user to direct or redirect a selected report to a selected printer device "on the fly" when a Print command button is pushed on a form.

Access 2.0 stores (for good reason) the printer setup within the report document itself and for most users the only way to change the selected printer device is to open a report in either design mode or in print preview and change the printer setup there before printing. Opening a report in preview prior to changing the printer setup requires two compiles for the report. Once to preview and once to print.

This becomes cumbersome, especially when a print device might be changed often, (in a multi-user environment), you don't want users to have to visibly be confronted with the design mode of the report, (they can't if you're using the run time version of Access,) or if you have designed your application for mobile users who may often switch between sending a report to a printer or to a fax device. In addition, if you have a series of reports (such as Invoices, Invoice details, and Mailing Labels) which might be compiled and sent to print in a batch, this would mean opening multiple reports to change print settings.

With these problems at hand the Printer Functions Module in the Printer.MDB database was our solution. This module includes two functions a) SetPrint and b) szPrinterDlg. SetPrint is called from a command button's "on click" event as demonstrated in the Printer form. SetPrint then calls szPrinterDlg to get the printer device selection of the user, and returns to device information to SetPrint to allow for changing the information in the reports setup.

We were interested in a broad run time routine not only changed the printer device but also allowed for selecting all the options normally allowed in printer device selection such as number of copies, paper tray etc. To accomplish these actions the functions change an Access 2 report's PrtDevNames property and PrtDevMode property. The PrtDevNames and PrtDevMode properties are scantily documented by Microsoft in the retail documentation or Language Reference of the ADT (These properties follow the format of the DevNames and devmode structures as best outlined in the Windows SDK documentation). As additional functionality, we wanted to be able to send the same print device specifications to multiple reports which would run in a batch to the same output device.

The SetPrint Function relies on opening the report in the background, changing the printer device and then saving the report before printing, all unseen to the user. This is the only way to accomplish changing the reports properties in Access. (You could set all the reports in your application to the default printer and then change the which device is the default printer in Windows, however, since this impacted applications other than Access we ruled this option out.) SetPrint also allows you to specify whether a report is a second or greater report in a single series of reports to be sent to the same selected device without calling the Print Dialog again. The Set Print function returns a 0 (false) if the user presses Cancel in the Print Dialog, allowing you to cancel actions which might be based on a successful print action in your Access Basic code, such as recording the sending of invoices through an update query.

(Note you may need to rethink about how you set up security and permissions for reports for your users if you haven't provided access to design mode for most objects in the database when used under retail Access.)

These functions rely on Windows API calls available in Windows 3.1 through the Commdlg.dll and other Kernel and User functions. If you're not familiar with the Windows API be cautious about changing much, otherwise you might encounter the ubiquitous "General Protection Fault".

We are providing no warranty for these functions. If they're useful to you great. We hope to save you some development time. If you use them in an application we would appreciate your support of our ongoing enhancements and publishing of these functions by sending us a nominal fee of \$10.00 to:

ATTAC Consulting Group  
2869 Baylis  
Ann Arbor, MI 48108

## "On the Fly" Release and Enhancement History

### Release 4 -- March 22, 1995 Changes

The functions were changed in one major and one minor way in this release. The first revision resolved a bug experienced by some users which caused an error in setting the new printer device parameters for a few specific printers which require inclusion of additional (supplemental) driver data in addition to the standard devmode structure parameters to initialize a printer. This would include many postscript printers which do not use solely the standard postscript driver, but also have additional file dependencies. The second enhancement eliminated much of the screen flicker which was noticeable when the Print Dialog box was opened and closed.

For those tracking the coding changes in the functions, we did the following to resolve these issues:

*To resolve the printer setup issue for printers requiring supplemental driver data:*

First (for reasons described below,) we continue to copy the values of the returned hdevmode member of the printdlg structure to a standard devmode structure *without* the necessary supplemental data which would normally be included in an optional dmdriverdata member. Access Basic does not support inclusion of the dmdriverdata member of the data type set up for the devmode structure because dmdriverdata member would be a variable length string. (Which in C is simply null terminated.) Variable length strings can not be incorporated into data structures which are acted on using the LSet command which we use to transfer the devmode structure data to the devmode string data type which is then copied to the prtdevmode property of the report. This standard portion of the devmode data is placed in a variable which we use to provide the individual components of the devmode for spooled or batched reports.

To obtain the devmode data including the optional dmdriverdata we also now copy the devmode data directly to a string variable. This string contains the entire devmode structure including the default supplemental driver data required by certain printers, and any changes to that data entered into the print dialog by the user. This string is then used as the input to the prtdevmode property of the first (or only) report, since it contains complete devmode data in the same format as that required by an Access report.

For any second or greater reports spooled into a batch, we continue use the data loaded into the standard devmode structure variable. This is because we need to adjust all the individual components of the devmode data of any reports following the first report *other* than the orientation member of the devmode structure, in order to preserve report design. To add the supplemental printer driver data necessary, we break down the devmode string copied previously. Since we know the standard portion of the devmode structure is exactly 68 bytes long, we strip the first 68 bytes from the devmode string to get just the additional supplemental data, and then load this into a variable. This supplemental data is then concatenated to the standard devmode data specified for the second or greater reports resulting in a complete devmode value which we then set into the prtdevmode property of the report(s).

*Screen Flicker* was resolved by adding the use of the Lock Window API call rather than just relying on use of Echo off commands. The additional advantage of this functionality is that if the report was saved with the properties box displayed in design mode, the properties box is no longer visible to the user when the report is opened to set up printing.

### Release 3 -- January 28, 1995 Changes

Resolved a bug which resulted in a General Protection Fault when the user selected a new printer device in the Print set up dialog and subsequently pressed cancel in the Print Dialog.

### Release 2 -- January 8, 1995

Changed SetPrint function to use the Print Action rather than the Open Report Method because Open Report will only print one copy of the report even though a change in the prtdevmode property set the report up for multiple copies.

By switching to the Print Action we were able to add new functionality including:

1. User capability to specify the number of copies for the reports(s)
2. User capability to select whether to collate the copies
3. User ability to select page numbers for the print job
4. Slightly enhanced speed for the process

Eliminated use of changing the dmcopies member of prtdevmode structure to determine number of copies to be printed (as is suggested in the example of use of the prtdevmode property by Microsoft in the retail version of their help file,) rather using the Print Action number of copies parameter. this allows the report to be saved with a default copy number of one which is clear coding for administration of future releases of any reports.

