

# Excerpt from *Power Reports* (Pinnacle Publishing)

By Helen Feddema

## ASCII Text Reports.

I wrote an article on creating ASCII Text reports for the Charter issue of Smart Access. Access has gone through two upgrades (1.1 and 2.0) since that time, but it still lacks the capacity to create useful ASCII Text reports; the continuing problem is evidenced by the feedback I still receive from that article, a year and a half later. You probably want to be able to output any report to ASCII Text, and have it emerge in a useful form—without special fonts, emphasis, and graphics, but basically laid out correctly—but that is not what happens. If you want a properly formatted ASCII Text report, you must create it from scratch with ASCII output in mind (or substantially modify it from the standard format). If you need to output a report both to the printer and to an ASCII Text file, you should make two versions of the report. I use the prefix “rasc” for ASCII Text reports (an add-on to the standard Leszynski-Reddick naming convention). See *Power Forms* in this series for a full discussion of this naming convention, and its many advantages for developers and power users—for purposes of this report, it suffices to mention that the standard report prefix is “rpt” and the subreport prefix is “rsub”.

As you probably know, ASCII Text reports are limited to the ASCII character set, which includes the letters of the alphabet, the numerals, the standard punctuation marks, and just a few symbols. You cannot use any special symbols, lines, graphics, colors, or even proportional fonts in an ASCII Text report. You might think that a simple business address report such as the one shown in Figure 1. 1 would fare well when printed to an ASCII Text file; after all, it has no graphics and no subreports, and only minimal text formatting (bold, italics and underlining).

**Figure 1. 1. A simple business address report (with regular printer selected) in print preview.**

However, that is not the case; Figure 1. 2 shows the ASCII Text file generated from this report in the Windows Notepad, illustrating typical problems with such files. If you try to print a normally formatted Access report to an ASCII Text file, by printing it to the Generic/Text Only printer (see below for how to do this), and then look at the outputted file in a text editor, you will see one or more of the following problems: badly misaligned text, labels either missing or duplicated several times (at staggered intervals), and numerous strange characters, represented by the black boxes or other graphic symbols, depending on the font—these are low-number ASCII characters such as ASCII #10 and ASCII #12 (the line feed and page break symbols). In addition, all the blank rows used to make extra space between sections of the report are missing.

**Figure 1. 2. An ASCII Text file generated from a simple business address report by printing to the Generic/Text Only printer.**

Since Access 2.0 introduced a new way of printing to an ASCII Text report, the Output To MS-DOS Text item on the File menu, I hoped that this would clear up the problems I had experienced with such reports, and eliminate (or at least reduce) the extensive pre-processing and cleanup required to generate clean ASCII Text reports. This method does eliminate some of the problems displayed in the report generated directly to the Generic/Text Only printer, such as the staggered and duplicated labels, and the extraneous low-number ASCII characters.

However, it has some problems of its own. It discards the underlining of the company names, which is converted into underline characters by the Generic/Text Only printer method. And it introduces numerous blank lines into the report—one for every text box, whether or not it contains any text. And, perhaps most significantly, the Output To MS-DOS format does not process any subreports. If you try to use this selection with a report that includes subreports or subforms, you will get a warning message that it will not process them—and indeed it does not; when I tried this option on a report with subreports, all I got was the headings and page footers.

Since neither of the methods for generating an ASCII Text report from Access 2.0 solves the problems I experienced in earlier versions, I am still generating ASCII Text reports directly to the Generic/Text Only

printer (since that method does support subreports, which I use extensively), and using the pre-processing and clean-up techniques I developed during the Access 1.0 beta, with some slight modifications. These techniques are listed below.

To print to the Generic/Text Only printer, you need to have that printer available for selection in the printer list in the Windows control panel. It may have already been installed to support some other program, but if it is not there, you can add it as follows:

1. Open the Control Panel and click on the Printers icon.
2. Click on the Add button.
3. Highlight *Generic/Text Only* from the list of printers and click on the Install button.
4. Now *Generic/Text Only on LPT1* (or your default port): is one of the installed printers; you need to change it to direct its output to a file instead of to a printer port, so highlight *Generic/Text Only on LPT1*: in the Installed Printers list and then click on the Connect button.
5. In the Connect dialog box, select *FILE:* from the list of ports, and click on the OK button.
6. Now *Generic/Text Only on FILE:* is on your list of installed printers. (Don't make it the default printer; you will select it as the printer for the Access reports you want to print to ASCII files).
7. Click on the Close button to close the Printers dialog box, and then close the Control Panel.

The next step is to prepare the Access report for printing to an ASCII file. First make a copy of the report and rename it with the "rasc" prefix (to identify it as an ASCII Text report). Then select the Generic/Text Only printer driver for the report (this information is saved with the report), as follows:

1. Open the report in design view.
2. Select Print Setup from the File menu. The Print Setup dialog box opens, as shown in Figure 1. 3. Select the Generic/Text Only printer from the Specific Printer selection list, and check the Data Only box.

**Figure 1. 3. Selecting the Generic/Text Only printer in the Print Setup dialog box.**

3. Click the More button to drop down an extra section at the bottom of the Print Setup dialog box, and check the *Same as Detail* box (if it is not already checked). Figure 1. 4 shows this screen, with all selections made.

**Figure 1. 4. The Print Setup dialog box, with the More section dropped down.**

4. Click on the Options button, and in the Generic/Text Only dialog box, check the *No Page Break* box. *Note:* This step is optional; for some reports, you may get better results by unchecking this box, allowing page break characters to appear in the report, and then stripping them out in the final cleanup.
5. Click on OK to close the Generic/Text Only dialog box, and on OK again to close the Print Setup dialog box.

Finally, there are a number of design considerations when laying out controls on a report to be printed to an ASCII text file.

1. Set Layout for Print to Yes in the report properties sheet (it is under Layout Properties). *Note:* to open the report properties sheet, open the properties sheet and choose *Select Report* from the Edit menu, or click on the white square in the upper left corner of the report.
2. Select all controls on the report (Edit, Select All) and choose Roman 10 cpi (a fixed-width font) from the font list on the tool bar (*Note:* If you don't see the Roman 10 cpi font, you don't have the Generic/Text Only printer selected; go back and select it in the Print Setup window, as described in the preceding section). There is a limited selection of printer fonts, and this is the one that works best for ASCII text reports.

3. If there are any labels on the report, remove them and replace them with text boxes, with the former label text as the control source ("Label Text"). Labels print *very* strangely (if they print at all) to an ASCII text file.
4. If you want blank rows on the report, in order to separate sections, use text boxes with a placeholder character such as "@" on those lines. It does not matter how close or far apart (vertically) controls are on your report; in printing to an ASCII text file, extra vertical space is stripped out—that is why you need a placeholder to get blank rows in your report.
5. If you want to produce a continuous report, remove page headers and footers; otherwise (i.e., if you want to produce a report broken up into separate pages), you can leave them in the report.
6. Remove all graphics from the report, and (if desired) replace graphical lines and boxes with controls using ASCII characters such as dashes, underlines, and plus signs. *Note:* if you want to use the pipe symbol (|) to create a vertical line, you can't enter it directly, as it is a reserved character; but you can insert it into a control as Chr(124). Figure 1. 5 shows a report in design view, with text box controls containing expressions with text and functions (or just text) as section and subsection headers, and other text box controls with dashes, plus signs, and the Chr(124) expression for making lines and boxes. Figure 1. 6 shows the same report in print preview. (*Note:* this report is from the full Crafts database; it is not in the sample database. Many similar ASCII Text reports produced from my CRAFTS.MDB are available for on-line reading or downloading from the CompuServe CRAFTS forum.)

**Figure 1. 5. An ASCII Text report in design view, showing use of ASCII characters to create lines and boxes.**

**Figure 1. 6. An ASCII Text report with ASCII character lines and boxes, in print preview.**

The next stage is to print the report to a file; click on the Print button from the print preview window, as shown in Figure 1. 7, or select Print from the File menu in either design view or print preview. A standard Windows Print dialog box appears; check that it lists Generic/Text Only as the printer, and if it has another printer listed, back out and select the Generic/Text Only printer under File, Print Setup. Next, click on OK button. *Note:* You do not need to check the Print to File box on this screen—the report will be printed to a file whether or not the box is checked.

Next a standard Windows Print dialog box appears, followed by a Print to File dialog box; enter the output file name (with path, unless you want the output file to be created in your Access directory) and click on the OK button. *Note:* Access does not automatically add the .TXT extension; if you want an extension, you must add it yourself.

**Figure 1. 7. The Print button on the print preview toolbar.**

The Print dialog box shows progress by pages (which may not be very meaningful, depending on how you have set up the report); when it closes, the ASCII text file has been generated.

When viewed in a text editor such as the Windows Notepad or the Norton Desktop Editor, the generated text file will contain at least some of the following extraneous characters: ASCII #10 (white circle in a black square), ASCII #12 (female symbol), and @ or other placeholder characters, if you used them to make blank rows. *Note:* The ASCII characters are not visible in all fonts; in the Norton Desktop Editor they are visible in the OEM Fixed Font. Cleanup of these extraneous symbols is dealt with in the following paragraphs.

If you are just going to import your ASCII Text report into a word processor, you don't need to do any further clean-up; the low-number ASCII characters will not cause any problems. However, if you plan to upload the file to CompuServe or another on-line service where users may read the file on line, you need to remove these characters, as they will be interpreted as actions (rather than text) when the file is read, with bizarre results.

You can search for and delete the three types of extraneous characters manually, using search and replace and replacing them with nothing (see Figure 1. 8 for a view of a new text file before cleanup). However, if you have a great number of ASCII reports to clean up, this is not an appealing prospect. To

make the clean-up process speedier, I originally made a Norton Desktop Batch Runner batch file to automate the process of searching for and removing the redundant characters from newly generated files. The ASCII #10 and #12 characters, in particular, must be removed from files intended for upload to CIS, as they will cause strange errors on-line. Norton Desktop has been upgraded since I made the original clean-up file, and I have also streamlined the clean-up file; it is provided on the Companion Disc as ASCII.SM (with a small companion file, ASCIINOS.TXT). ASCII.SM is a Norton Desktop 3.0 ScriptMaker file, which can be compiled into an .EXE file after modifying the paths as needed for your system.

**Figure 1. 8. A newly generated ASCII Text report, before cleanup, in the Norton Desktop Editor.**

I ran into a problem at first when I tried to use a SendKeys statement to send ASCII # 10 and #12 to the Desktop Editor; the Windows clipboard does not support these characters. However, I found that they are supported by the Desktop Editor's internal clipboard, so I made a little file, ASCIINOS.TXT, containing just the two ASCII characters needed for searching and replacing (I simply copied them from a new file that had not been cleaned up yet). Then I had the batch file open up this file and copy the characters into the editor's clipboard, and paste them into the search and replace statements.

The Norton ScriptMaker file is given below; the syntax is similar to that of other Windows batch languages, such as WinBatch, so you may be able to adapt it for use in other programs. However, if you do use another program, you will need to verify that your text editor supports the low-number ASCII characters in its internal clipboard, as the script relies on this feature to copy these characters. (*Note:* ASCII.SM and ASCIINOS.TXT are provided on the Companion Disc.)

```
' ' ASCII.SM
' ' Created by Helen Feddema 10/30/93
' ' Modified 5/2/94

' ' To open a ScriptMaker window, use the command line: c:\ndw\deskdedit /b

' ' To use this file, open the ASCII text file to be cleaned up in a
' ' Desktop Editor window and open this file in a ScriptMaker
' ' window, and click on the Run button on the toolbar; may also
' ' be run from a desktop tool.

Sub Main()
    ' substitute your NDW scripts path
    ndwpath$ = "c:\ndw\scripts\"

    ' activate editor window
    WinActivate ("Desktop Editor")
    ' move window
    WinMove 50, 40
    ' size window
    WinSize 450, 569, "Desktop Editor"
    ' tile window
```

```

Menu "Window.Tile"
' go to top of file
QueKeys "^{HOME}"
QueFlush TRUE
Menu "File.Open"
WinActivate "Desktop Editor|Open File"
SendKeys ndwpath
SendKeys "asciinos.txt~"
'copy ASCII #10 character to clipboard
SendKeys "+{end}^{insert}^{F6}"
' search for ASCII #10 character and replace it with nothing
SendKeys "%SR+{insert}{tab}{delete}%a~^{F6}"
'copy ASCII #12 character to clipboard
SendKeys "{home}{down}+{end}^{insert}^{F6}"
' search for ASCII #12 character and replace it with nothing
SendKeys "%SR+{insert}%a~"
' close ASCIINOS.TXT file
SendKeys "^{F6}^{F4}"
' search for @ (placeholder) and replace it with nothing
SendKeys "%SR@%a~"

```

End Sub

Figure 1. 9 shows the same report after cleanup.

**Figure 1. 9. An ASCII text report, after cleanup.**