

wIntegrate®

**PRINTING TIPS AND
TROUBLESHOOTING**

U2 Data Management Solutions

1099 18th St., Suite 2500

Denver, CO 80202

Phone: 303.294.0800

FAX: 303.293.8880

Contributors

Kenny Brunel, Alan Buckley, Trevor Grover, Tim Rasmussen, Ed Robb,
David Robertshaw, Susie Siegesmund

Contents

Page Breaks Ignored When Printing.....	1
Printing PC Files from wIntegrate	5
Using Slave Printing	11

WINTEGRATE TECH TIPS

PAGE BREAKS IGNORED WHEN PRINTING



wIntegrate®

When using an ANSI-based emulation such as VT100, printed reports don't always eject new pages correctly. This is because the ANSI 'new page' sequence is not understood by the printer. The solution is to convert the new page sequence to one the printer can understand, usually CHAR(12), or FormFeed. The following discussion will use examples using wIntegrate 4.1 but the same method is applicable to version 3.0.

PRINTERMAP

A common indication that an ANSI new page escape sequence is being sent by the host is the display of, apparently, 'garbage' characters such as "█[2J" and "█[2H" on the output. These escape sequences have no meaning to a printer so are printed as best they can and no page break occurs.

A new command, PrinterMap, introduced at versions 3.0.9 and 4.0.3, changes a sequence of characters sent by the host to a different sequence during printing. This has been implemented to allow ANSI-based emulations such as VT100 to eject a new page properly when an on-screen report is sent to the printer.

Syntax

PrinterMap [=] "Host sequence", "Printer sequence"

Example

The following might appear in a an emulation .WIT file:

```
PrinterIgnore = "\011"      (vertical tab)
PrinterIgnore = "\e[%aH"   (cursor position commands)
```

These lines prevent the character 11, vertical tab or any cursor position command from being printed

Adding the line:

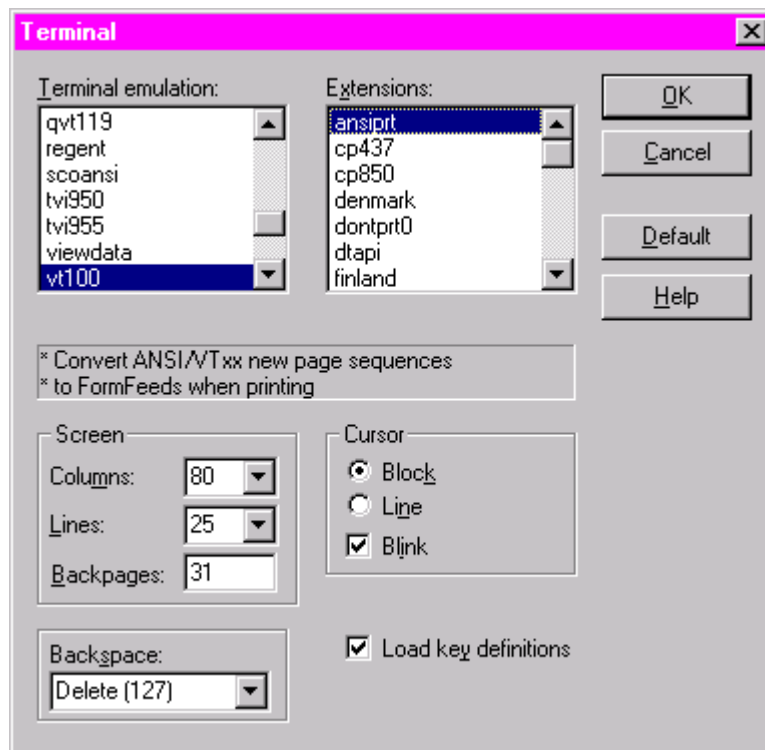
```
PrinterMap = "\e[%dJ", "\012"
```

will change the ANSI "new page" sequence "\e[J", "\e[0J" or "\e[2J" to ASCII character 12, which is interpreted by printers as a FormFeed, causing a page to be ejected

ANSIPRT.WIT

A new extension file, ANSIPRT.WIT, which uses the PrinterMap command, is included beginning at versions 3.0.9 and 4.0.3. This extension file converts ANSI/VTnnn (home cursor and clear to end of page) screen sequences to ASCII character 12, causing a proper FormFeed to occur.

To enable this extension, click **Setup | Terminal** on the wIntegrate menu then select **ANSIPRT.WIT** in the **Extensions:** listbox.

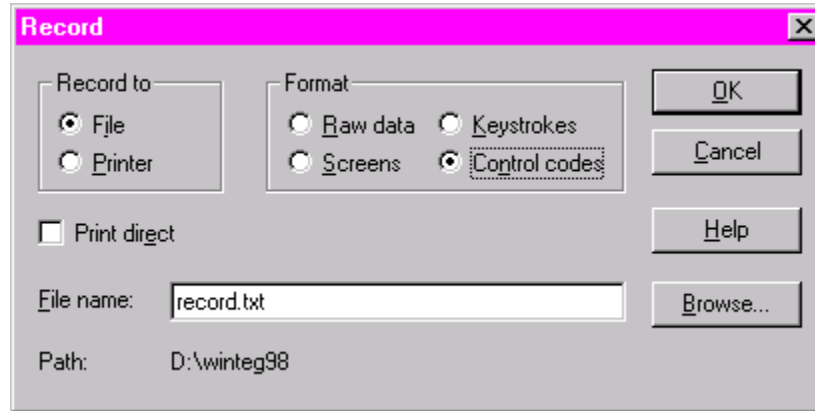


Be sure to click **File | Save** to retain the new setting.

CREATING YOUR OWN EXTENSION FILE

ANSIPRT.WIT works when you are using an ANSI emulation, particularly a VTxxx emulation. If you aren't using one of these emulations, and you find that page breaks are not occurring where you expected, you can create your own extension file to translate the ANSI escape sequence to CHAR(12).

The first step is to determine what escape sequence is being used by your emulation for a FormFeed. An easy way to do this is to record debug information while using Query Builder to print to the local printer. Click **Edit | Record** on the wIntegrate menu, select **Control codes**, then click **OK**.



Now create a simple query and print to the local printer. After the print job has completed, click **Edit | Record** again (there should be a check mark in front of **Record**) to discontinue recording.

Edit the file used to capture the control codes. By default it's Record.txt (in the wIntegrate directory). What you are looking for will vary depending on the host and emulation. In the following example, we used a Wyse50 emulation on a host running AIX.

Scan the file for the text 'Sending data to the local printer'. The escape sequence we're looking for may be the line directly above this one. In our example, the suspect escape sequence looks like it might be <27>;. (<27> is CHAR(27), or escape). To confirm this, we scrolled down to where we see the heading for each page of our printout. The line directly above the heading has the escape sequence we are looking for. It is <27>;, as we suspected and being ignored, causing the page to not be ejected.

```

78  1 clearallunprotected<27>;
79  1 Text          Sending data to the local printer...
80  1 cr            <13>
81  1 linefeed     <10>
82  1 Command      <27><1>Screen Off<13>
83  2 Command      <27><1>Capture On
ServerPrint,"Printer",,,CAP_ReadDirect<13>
84  1 ignore      <27>;
85  1 Text          SORT CUSTOMER NAME ADDRESS CITY STATE ZIP NO.PAGE
14:43:11 Jan 05 2000 1
:
269  2 ignore     <10>
270  3 ignore     <27>;

```

```

271  1 Text          SORT CUSTOMER NAME ADDRESS CITY STATE ZIP  NO.PAGE
14:43:11 Jan 05 2000 2
272  1 ignore       <13>
273  2 ignore       <10>

```

Now that we've identified which escape sequence is being used, we can create an extension file to translate that code to CHAR(12).

Using the wIntegrate editor (you can use any text editor), we created a new file. This file must be saved in the wIntegrate subdirectory wintsys\wit_ext and have a .WIT extension. We saved ours as \winteg\wintsys\wit_ext\eject.wit. We want to translate the CHAR(27); to a CHAR(12) using the PrinterMap command:

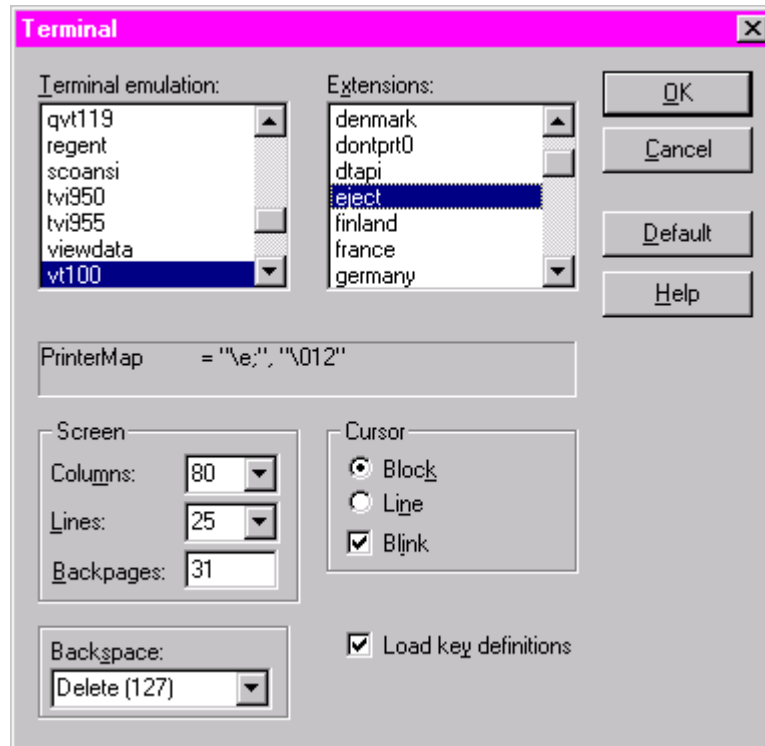
```

[Terminal]
PrinterMap = "\e;", "\012"

```

Note, the '\e' is shorthand for CHAR(27), the escape code.

Finally, to use this extension click **Setup | Terminal** then select **eject** in the **Extensions:** listbox. Be sure to click **File | Save** to save this in your configuration file. Now when you print, the pages will eject properly.



WINTEGRATE TECH TIPS

PRINTING PC FILES FROM WINTEGRATE



wIntegrate®

wIntegrate provides the capability to print PC files on a local printer or pool to a host printer. This is not to be confused with Slave printing. You can accomplish this by selecting the appropriate options on the wIntegrate menu, from a script, or from a BASIC program on your host by calling the appropriate host subroutine.

PRINTING A PC TEXT FILE LOCALLY

Using the wIntegrate menu

To print a PC file locally, select **File | Print** from the wIntegrate menu, then locate the file to be printed. This will print to the currently defined printer. To change the target printer, select **File | Printer Setup**. The printer must have been configured in Windows first.

Using a script

The wIntegrate menu is created using a script internal to the program. You can use the script language to run menu options and accomplish the same thing as manually selecting the menu option. Before you run a menu option you must know its name. See the Tech Tip, [How to Create Menus](#), for a list of these names. The name of the **File | Print** menu option is FilePrint.

To call the menu we use the **Dialog** script command, the menu name, and pass the name of the file to be printed. If the file is not in the wIntegrate directory you must specify the full path.

```
Dialog FilePrint
  Set Filename="d:\test\readme.txt"
Invoke
```

Using a BASIC program

To print a PC file on a local PC printer, call the subroutine WIN.PCPRINT. (WIN.PCPRINT is installed during the host routine installation process.)

```
CALL WIN.PCPRINT("file")
```


where "file" is the name of the PC file to be printed. If the file is not in the wIntegrate directory you must also specify the full path.

PRINTING A NON-TEXT PC FILE LOCALLY

The previous procedures allow you to print a text file on a local PC printer. You can also print non-text files locally but must do so using a script or BASIC program.

When you use Windows Explorer to open or print a file using its native application that file extension must have the application associated with it. For example, you can set up associations to open any BMP file using Windows Paint, or any XLS file using Excel. Once that's done when you click on a BMP file Paint starts up automatically; click on an XLS file and Excel starts automatically.

Before you can print a non-text file using wIntegrate you need to have an association set up in Windows. Further, if the application you have associated a file type to doesn't support printing directly, meaning you can't click on the name in Windows Explorer then select **File | Print**, you won't be able to print it from wIntegrate.

We will use the **Run** script command to print a non-text PC file to a local printer.

Using a script

The syntax for the Run command is:

```
Run "path", [params], [ret_var], [show_opt],[“verb”], [def_dir]
```

where,

path = The path to the program or file name.

params = Additional parameters if *path* is a program.

ret_var = Returns a value specifying whether the file or program ran. If the value is greater than 32, the program was launched correctly. If it was less than or equal to 32, an error occurs. The most common errors are:

- 31 — No association found for file type, or verb not supported.
- 0 — System was out of memory or executable file was corrupt.
- 2 — errmsg = “File was not found.”
- 3 — Path was not found.
- Others — Executable would not load or run under Windows.

show_opt The show option is a request. The applications will not necessarily start with the specified show state. The default is 0.

The [show_opt] can be one of the following:

- 0 — Show and activate
- 1 — Minimize
- 2 — Maximize
- 3 — Hide
- 4 — Show but do not activate

verb = Either “Open” to open a document or run an application, or “Print” to print a file (Print is not supported by all applications).

def_dir = Default directory (default none) for program to start in

As an example, we have a file called universe.bmp in c:\winnt which we wish to print from within wIntegrate. We create a script with the following command:

```
Run "c:\winnt\universe.bmp" ,,,, "Print"
```

When we run this script (**Run | Script**), because we have set up the association between BMP files and Windows Paint, Windows will use Paint to print the file. Paint will not start up during the process, only its print capabilities will be used.

Using a BASIC program

We can use the same **Run** command from within a BASIC program to print a non-text PC file locally. Using WIN.HSCRIPT or WIN.HSCRIPTC (installed during the host routine installation process), create a dynamic array containing the script command then call WIN.HSCRIPT or WIN.HSCRIPTC:

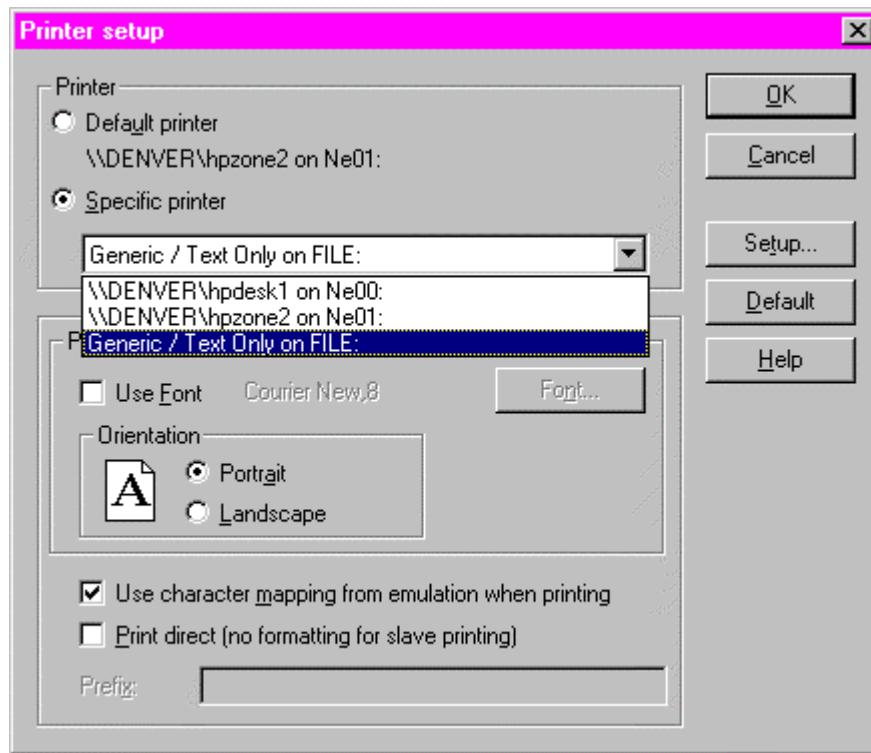
```
SCRIPT = ''  
SCRIPT<-1> = 'Run "c:\winnt\universe.bmp" ,,,, "Print" '  
CALL WIN.HSCRIPTC (SCRIPT)
```

SELECTING THE PRINTER FROM A SCRIPT OR BASIC PROGRAM

Usually, you would select the printer to be used by wIntegrate by selecting **File | Printer Setup**, then the specific Windows printer. This may not always be desirable. For example, you may have a script application or BASIC program and you want that application or program to select the printer automatically.

The printer name is a global variable within wIntegrate, which means it can be set without the use of a **Dialog** command. It also has to be EXACTLY the name Windows gives to it, including the case. For example, when you set up a generic text printer, the default name is 'Generic / Text Only'. If you wish to select this printer via a script you must specify it exactly as entered, including the spaces on either side of the '/'. If you want to select a network printer, you must use its UNC path as the name.

The easiest way to determine the name of a printer is to select **File | Printer Setup**. Then click on the pulldown box to display specific printer names. For example:



On this display, the valid printer names would be:

\\DENVER\hpdesk1
 \\DENVER\hpzone2
 Generic / Text Only

Notice the 'on...' is not used as part of the name.

Setting the printer in a script

To select a printer from within a script you set the value of the global variable, Printer, to the name of the new printer. Simply include the following command in your script.

```
Set Printer="printer name"
```

For example, using our list, above, to set the printer to hpzone2, the command would be:

```
Set Printer="\\DENVER\hpzone2"
```

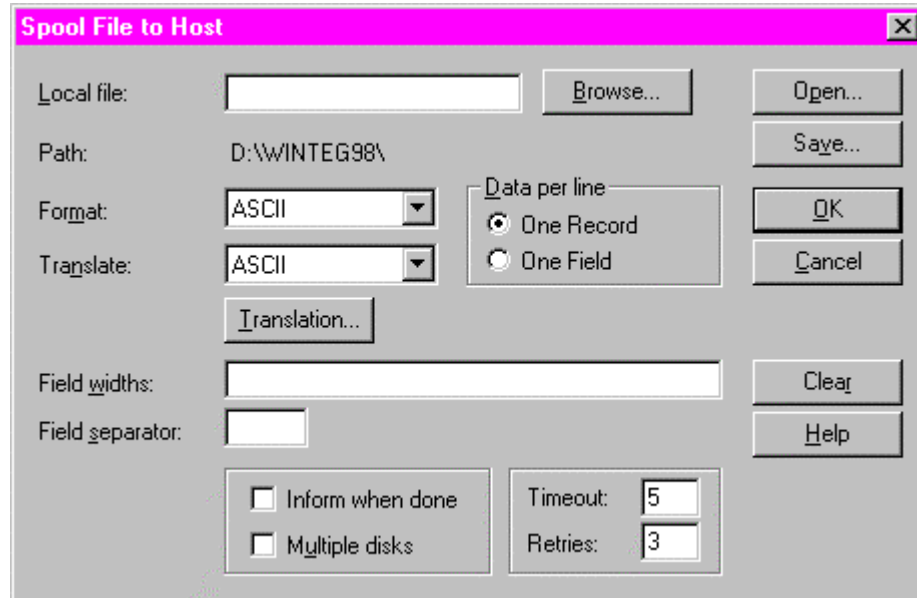
Setting the printer from a BASIC program

The command used previously can be executed from a BASIC program using the WIN.HSCRIPT or WIN.HSCRIPTC subroutines. These routines allow you to create a dynamic array holding the script then run that script on the PC. For example, the following will change the printer to 'Generic / Text Only'.

```
SCRIPT = ''
SCRIPT<-1> = 'Set Printer="Generic / Text Only"'
CALL WIN.HSCRIPTC (SCRIPT)
```

SPOOLING A PC FILE TO A HOST

If you don't have a locally attached printer, or perhaps need to print a file on a host printer, wIntegrate provides you with the option to use a host printer to print a PC file. Unlike **File | Print**, you can select from a variety of PC formats. Select **Run | Spool File** to display the Spool File to Host dialog box:



1. Select a file to spool. In the Local file box, enter the name of the local (PC) file you want to spool to the host. Otherwise, you can browse for a file to spool.
2. Select a format for the local file you are spooling.
3. Select a translate option. In the Translate box, click the scroll arrow to see a list of methods wIntegrate uses to change characters while spooling them. Click an option to select it.
 - None – wIntegrate does not translate characters while spooling them.
 - ASCII – wIntegrate translates characters only when spooling them from a local file in ASCII format.

- All – wIntegrate always translates characters regardless of the file format selected.
4. If you selected ASCII or All in the Translate box, click the Translation button. The Translation dialog box appears. Go to Setting Up Remote Character Translation for Spooling for instructions.
 5. Select a data per line option. This option controls whether each line in the host file contains one record or one field.
 6. In the Field widths box, enter the width for each field in the host file. Insert a space between each field width.
 7. In the Field separator box, enter the character wIntegrate inserts to separate each field in the host file.
 8. If you want a message box to inform you when the spooling process is complete, select the Inform when done check box. This feature is useful if you want to run another application full-screen while wIntegrate spools a large file.
 9. If the local file is stored on multiple floppy disks, select the Multiple disks check box. wIntegrate prompts you to insert successive disks during the spooling process.
 10. Specify a timeout value, in seconds (range 1-9999). A timeout occurs when wIntegrate sends a protocol message and it doesn't get a response within a specified length of time.
 11. Specify a retries value, or the maximum number of times wIntegrate sends a protocol message before a file transfer is abandoned.
 12. Save your changes. When you have finished making selections in the Spool File to Host dialog box, you can save the current settings for different purposes:

WINTEGRATE TECH TIPS

USING SLAVE PRINTING



wIntegrate®

wIntegrate supports printing to a local printer, somewhat like the functionality of some terminal emulations found with multi-user operating systems where output could be directed to a printer attached to a terminal AUX port. Some emulations support printing to a local printer, some may not. As such, success depends on the ability of the emulation to support this mode, and it is possible that a given emulation may not support slave printing at all. If a specific emulation does not have the necessary codes to enable/disable to local printer, you may add the appropriate lines in the emulation .WIT file. An example of these lines is shown in the next section. You will need to know the escape sequence used by your specific host.

HOW SLAVE PRINTING WORKS

Each wIntegrate emulation file (.WIT) contains the necessary code to enable/disable slave printing when specific escape sequences are received from the host. When these codes are received, wIntegrate will automatically run PRINTON.WIS or PRINTOFF.WIS, the scripts used to enable or disable slave printing.

For example, in VT100 emulation, if wIntegrate receives <esc>[5i from the host, it will run PRINTON.WIS, enabling slave printing. If <esc>[4i is received, wIntegrate will run PRINTOFF.WIS, disabling the slave printing. These codes will vary from emulation to emulation. You can determine what the specific codes are for your emulation by editing the specific .WIT file, found in \WINTEG\WINTSYS\WIT, and locating lines similar to the following:

```
TerminalCommand = "\e[4i", "Chain 'wintsys\\wit_scr\\printoff" ,P
TerminalCommand = "\e[5i", "Script 'wintsys\\wit_scr\\printon"
```

If you are writing an application, or have a custom application already in place, you will need to include the appropriate escape sequences for enabling/disabling slave printing in your code if you wish to use this feature. Without it, you will not easily be able to print to your local printer.

Once slave printing has been enabled, wIntegrate uses the printer driver installed through Windows to print host output according to its current settings. Note that wIntegrate can be set to have a different printer (perhaps a dot matrix) than the default printer for Windows (normally a laser printer).

TROUBLESHOOTING SLAVE PRINTING PROBLEMS

Normally, you should not have to do anything special in order for slave printing to work. If you do encounter problems, the following will help you identify the cause. Note: some of these techniques were not available prior to wIntegrate 3.0.01.

Before changing any settings, print a file from Notepad to verify the printer settings are correct. If you can't print from Notepad, the problem is with the printer settings in Windows, or the printer itself.

I don't get 80 or 132 columns across the page

Select File, Printer Setup. Under the section titled Print Formatting, select a font that will print the full 80 or 132 columns across the page. Be sure to select a fixed-pitch font, such as Courier New or Letter Gothic. Depending upon the selection of fonts you have available, either installed in Windows or in the printer, you may or may not be able to find a size that will fit the text on the page. As an alternative to changing the font, you can select landscape mode.

Graphic characters (e.g. lines) are not being displayed correctly

By default, 'Use character mapping from emulation when printing' is enabled. This means that wIntegrate will map some characters to display more appropriately in Windows. For example, users in Sweden would see "Ä" instead of "[". However, sometimes this feature must be turned off, for example some specific escape sequences sent from the host to the printer must not be changed by the remapping. Disabling this feature will prevent wIntegrate from remapping any characters. To disable character mapping, select File, Printer Setup. Click 'Use character mapping from emulation when printing' to disable this feature.

I get funny characters on my printout or it's not laid out properly

This is most likely caused by the host program sending a printer-specific escape sequence that isn't supported by the printer. Black boxes instead of characters are indicative of this. First, disable "Use character mapping from emulation when printing". If this doesn't resolve the problem, select 'Print Direct'. This will pass the data to the printer as a byte stream with no formatting by Windows Print Manager. Unfortunately, not all printers support this direct option. If you find this has not resolved the problem it's likely your printer is one that doesn't support direct printing. In this case, you will need to select the Generic Text Only printer in Windows.

If the unusual characters are the only thing wrong with the printout, you could determine what the character is, then edit the .WIT file for your emulation and add a line such as:

```
PrinterIgnore = "\038"
```

In this example, a CHAR(38) would not be passed to the printer.

To identify what this character is, prior to printing, select Edit, Record to make a recording of the session, select Control Codes. This will record the activity to the printer to the file you specify. You can then look at this file to see what the CHAR number of the problem character is.

At version 3.0, the NULL value is not being filtered out. This can be remedied by adding the following line to your emulation file (.WIT):

```
PrinterIgnore="^\z"
```

You may also be able to eliminate strange characters by first sending a printer reset, followed by the appropriate command to set the pitch. These escape sequences would be imbedded in your host application. For example, on an HP Deskjet 500C, after sending the code to turn slave printing on, send a reset, CHAR(27), then set the pitch to 10 pitch (or whatever pitch you need), <esc>(s10H. You will need to determine the appropriate codes from your printer's manual.

My margins are not what I want

Output options, such as setting margins or paper size, are controlled through the use of printer-specific escape sequences. Applications, such as word processing or spreadsheet programs, send the appropriate escape sequences necessary to change these settings. Unless your host application also does this, the margins will be whatever your printer defaults to.

You can, however, change the margins, or other settings, if you know the correct escape sequence. Assuming you know the sequence to change the margins or other settings, select Setup, Printer Setup, click on Direct Printing. This enables the Prefix box. You can now enter the escape sequence in this box. Note: control or escape characters are entered in the backslash format, e.g. \027 = Char(27), or escape.

You can find additional information about Direct Printing in the Help files README.HLP and OLDRDME.HLP, installed in the WINTEG directory.

Note: Direct printing, as used by wIntegrate, may not be supported by all printer drivers. It is possible that, after following the above suggestions, you may not be able to use slave printing with your specific printer driver or the behavior not be completely satisfactory.