The MathWorks

index | search | contact us

products consulting training/events support store

how do i...?

1206 Printing and Exporting Guide

Revision: 1.0 *Last Date Modified:* 06-February-2003

Introduction

Producing Publication Quality Graphics

Understanding Printing

Troubleshooting

Section 1: What is the purpose of this document?

This document deals with the printing and exporting of MATLAB figures. It explains the various settings that affect output and provides a guide to troubleshooting the problems that may occur during the process of printing or exporting MATLAB figures.

Besides MATLAB figure windows, there are several other user interface components that you can print (like the MATLAB desktop, editor or the help browser).

The MATLAB desktop, along with the Editor and the Help browser, are all Java-based components. They are all printed using an inherently different method than that used by the MATLAB figure window. All of the information contained herein pertains to printing MATLAB figure windows only.

Note: There are several options you can use when printing from the desktop, editor, or help browser. More detailed help on this can be found in the documentation.

Section 2: Producing Publication Quality Graphics

The MATLAB Digest article on Exporting Figures for Publication gets you started with creating presentation quality graphics quickly. There are two parts to this article spread over two editions of the Digest. The second part of this article can be found in the MATLAB Digest December 2000 issue .

Solution 1818 contains more information on printing to high quality graphics formats including PostScript, Windows Metafiles, and HPGL.

Section 3: Printing vs. exporting

It is necessary to make the distinction between the various terms used when discussing the creation of graphical output.

- **Printing:** The process of printing MATLAB figures directly to a printer or a file
- **Exporting:** The process of exporting MATLAB figures to various file formats that can then be imported into other applications for manipulation
- **Saving:** The process of saving MATLAB figures to a file that can be reopened in MATLAB later for further manipulation

For the purpose of this document, only the processes of printing and exporting will be discussed. All printing and exporting in MATLAB is handled through the `PRINT` command.

Note: There are several functions available for saving figures including `SAVEAS`, and `HGSAVE`.

Section 4: How do I control the destination of my printout?

After creating your figures, you can use one of the following mechanisms to print:

- Selecting the 'Print' menu item from the File menu
- Using the `PRINT` command from the MATLAB command line

There is a different sequence of events that is followed depending on which method you choose and the operating system on which you are running MATLAB.

On the PC: Printing from the menu

Selecting 'Print' from the File menu brings up the Windows print dialog. The destination and output format are determined through this dialog. The default value that is in the Printer Name field is the default Windows printer. If you want to print the figure to a file, then you can check the 'Print to File' checkbox, and Windows will prompt you for a file name to print. This file will then be created in a format that can be understood by the printer that was selected in the Windows Print dialog box and can be sent later with the system's copy command. For help with the system copy command, type `copy /h` at the system prompt

On UNIX systems: Printing from the menu

Selecting 'Print' from the File menu brings up a print dialog that enables you to select the printer and driver. You can instead specify a file name and have the output printed to a file. Additionally, you can select the driver that you want to use; the output format will use the selected device driver whether printing to a file or to a printer.

Note: On Windows, it is not possible to pick a driver from the Windows Print dialog box, as each printer has its own driver which will always be used.

All Platforms: Printing from the command line

Typing `PRINT` at the command line results in MATLAB selecting the default printer, printer driver, and print command for the operating system that MATLAB is running on. Typing `PRINT` followed by the device driver and then the file name will let you print using the specified driver to the specified file.

For example:

```
print -depsc2 myfile.eps
% Or you can use the functional form of print
filename = 'myfile.eps';
print('-depsc2',filename);
```

Note: As of MATLAB R13, the additional option of providing the Windows printer name is available via the -P option.

Thus far no mention of defaults has been made such as the defaults for printers, drivers, and print commands. These defaults are stored in the `PRINTOPT.M` file, and can be queried and set from within MATLAB. The `PRINTOPT` function is an M-file that stores the command line defaults for printing. To query the printing defaults for command line printing you can use the syntax

```
[pcmd, dev] = printopt;
```

To set your own options, you can edit the `PRINTOPT` file. If you do not set your own values, the values for `pcmd` and `dev` are set to default values. To view this file, or to set your own defaults, type

```
edit printopt
```

at the MATLAB command line. Within the file, there are two variables that are initialized, `pcmd` and `dev`. These variables contain strings, which are the operating system commands that MATLAB will use when it attempts to print. You can make modifications to these variables to reflect the commands that your particular operating system uses. This can be done by uncommenting the OS specific sections of the file. For example, on SGI you may want to use

```
pcmd = ['lp -c']; dev = ['-dps'];
```

Now the print command that would be used is the `lp -c` command rather than the default `lpr -s -r` command.

In order to confirm that these settings are in effect, you can use a debugging flag to get the information you are interested in. For example, if you type the following in MATLAB after making the necessary changes

```
surf(peaks)
print -DEBUG
```

you should receive an output that looks something like this:

```
Debugging in PRINT, PrintJob object is:

pj =

                Handles: {[1]}
                 Driver: 'win'
               FileName: ''
             PageNumber: 0
```

```
                 Active: 0
                 Return: 0
             AllFigures: 1
            AllPointers: 'arrow'
            PrinterName: ''
          SimWindowName: ''
               Renderer: ''
         rendererOption: 0
            PrintOutput: 0
                Verbose: 0
            Orientation: ''
                 hgdata: []
             PaperUnits: ''
                PrintUI: 1
             nouiOption: 0
                 UIData: []
                    DPI: -1
              DriverExt: ''
            DriverClass: 'MW'
           DriverExport: 0
            DriverColor: 0
         DriverColorSet: 0
            GhostDriver: ''
              GhostName: ''
             GhostImage: 0
            GhostExtent: []
       GhostTranslation: []
       PostScriptAppend: 0
       PostScriptLatin1: 1
         PostScriptCMYK: 0
    PostScriptTightBBox: 1
      PostScriptPreview: 0
            TiffPreview: 1
               PrintCmd: 'COPY /B %s \\PRINTERS\Nike'
          DefaultDevice: '-dwin'
                  Error: 0
              DebugMode: 1
              Validated: 1
          XTerminalMode: 0

Printing handle: 1
```

The following options are significant for users and technical support.

```
        PrintCmd: 'COPY /B %s \\PRINTERS\Nike'
          DefaultDevice: '-dwin'
```

You can see in this case that the command that is invoked on the operating system is `COPY /B %s \\PRINTERS\Nike` and the device is `-dwin`.

This way, you can confirm any changes that you have made to the `PRINTOPT.M` file.

Section 5: How do I control the format of my printout?

At this point, you should be able to print figures to the printer of your choice, using the device driver of your choice. However, you may want to change the format of the printed output. One of the first things that you may want to change is the size and location of the printed figure on the paper.

This information can be found in the documentation on printing. There are several detailed sections on controlling specific features of printed output. For example, the position of the figure on the page, the effect of print size on ticks and limits, and other settings and features.

After reading the above examples, you should have some understanding of properties, such as the figure's PaperPosition, PaperPositionMode, PaperOrientation, PaperUnits , and PaperSize properties.

These properties can be used to determine exactly how your figure will be printed on the page.

You can specify the print resolution in dots per inch by passing the -r option to the `PRINT` command. For a discussion on print resolutions, please consult the documentation on the `PRINT` command. The default resolution for bitmapped output is 150 dpi.

You can determine what resolution you have set for your monitor by typing the following at the MATLAB prompt.

```
set(0,'units', 'pixels')
Pix_SS = get(0,'screensize')
set(0,'units', 'inches')
Inch_SS = get(0,'screensize')
resolution = Pix_SS./Inch_SS
```

In general, when you view the same image on a machine running at 116dpi (dots/pixels per inch) and then on a machine running at 72dpi, the image on the higher resolution monitor appears much smaller than that on the lower resolution monitor. Some of the newer graphics software packages recognize this and are able to compensate and display an image at actual size rather than just showing all the pixels. Hence, you may find that your JPEG and TIFF files appear to be a different size depending on where you view them.

Recall that printing from the command line and printing from the File Print menu lead to different situations. This is also true for certain options that you can set from the Page Setup dialog that will only work when printing from the menu and not from the command line. For example, in the MATLAB documentation section Setting the Axes Ticks and Limits, the settings made in the Page Setup dialog box are honored only when printing from the File menu on the figure and not when printing from the command line.

Section 6: Which format should I use to export graphs from MATLAB into other Windows-based applications?

MATLAB provides several output formats that Windows-based applications can read. These formats are Windows Bitmaps (*.bmp), Enhanced Metafile (*.emf), HP Graphic Language (*.hgl), Encapsulated PostScript (*.eps), JPEG (*.jpg), and TIFF (*.tif). To save your figure window to a file in one of these formats, type one of the following:

```
print -deps my_file    % for Encapsulated PostScript format
print -dmeta my_file   % for Enhanced Metafile format
print -dhpgl my_file   % for HP Graphic Language format
print -dbitmap my_file % for Windows Bitmap format
print -djpeg my_file   % for JPEG graphics format
print -dtiff my_file   % for TIFF image format
```

When you go into your application, make sure that you select "insert --> picture" with the correct picture file format selected. The following is a brief description of the various formats.

Note:
Graphic images can usually be divided into two distinct categories, bitmap or vector graphics. Bitmap images (also called rastor images) are a collection pixels that form an image. The image consists of a matrix of individual dots (or pixels) that all have their own color data.

Vector graphics use mathematical relationships between points and the paths connecting them to describe an image. The area between the lines can then be filled with color.

The vector description procedure has major advantages, such as smaller file size and salability. A vector object can be scaled and resized without loss of quality, since only the scale is changed and the resulting image rendered at the time of display. Bitmaps can appear pixilated when they are resized.

PostScript is a very common graphical output format used for may presentations. To add a TIFF image preview of the MATLAB figure window, use the `-tiff` flag (available in MATLAB 5.1 and later). For example:

```
print -deps -tiff my_file_name
```

Drawbacks are:

- A PostScript-compatible printer must be used.
- The figures cannot be edited once the file is created, so all modifications must be made from within MATLAB.

PostScript format is used for many publications, presentations, and high quality documents. It is widely recognized by desktop publishing and word processing packages on all of the platforms that we support. **Enhanced Metafiles** use a vectorized approach to drawing graphics, similar in nature to PostScript, and are capable of producing near publication quality graphics.

Drawbacks are:

- Files can generally be read only by Windows-based programs.
- Reproduction is not as faithful as that available from PostScript.
- Some cross platform (and cross application) problems exist.

Tips:

- Make sure that "Copy options:Honor Figure Properties" is set to "off" (i.e., unchecked from the Preferences menu).
- Check the renderer that you are using by typing

  ```
  get(gcf,'Renderer')
  ```

  If you are using Zbuffer, your graphics will be copied as a bitmap. If you are using Painters, your figure will be copied in a vectorized graphics format. This note explains the difference in the two formats.

  To specify the renderer, use the `SET` command:

```
set(gcf,'Renderer','painters')
% Sets the renderer to Painter's Algorithm
set(gcf,'Renderer','zbuffer')
% Sets the renderer to Zbuffer
```

For a complete discussion of the different rendering algorithms in MATLAB, please refer to Technical Note 1201 as well as Section 11 of this Technical Note, entitled, "What are my rendering options when I print from MATLAB?"

HPGL uses a vectored approach as well, and is capable of producing near publication quality graphics. HPGL is a widely recognized file format and can be read by many non Windows-based programs.

Drawbacks are:

- Files are typically five times larger than a PostScript file of the same image.
- Minimal font support.

Bitmap creates a bitmap copy of the figure window. Since it is a bitmap, the resolution will not be as good as the other options, however, all the colors in the figure will be captured.

JPEG uses a lossy compression scheme that compresses files dramatically with relatively little loss of information. This scheme enables you to make tradeoffs between file size and quality, by specifying a quality level between 0 (minimum quality, maximum compression) and 100 (maximum quality, minimum compression). By default, `-djpeg` uses a quality level of 75. However, you can use a different level by appending the value to the device name. For example, this command produces a JPEG file with a quality level of 50:

```
print -djpeg50 -r100 newplot.jpg
```

Even at the highest quality level, JPEG files are often highly compressed. In fact, depending on the figure, a JPEG file with a quality level of 100 may be considerable smaller than a packbit-compressed TIFF file of the same figure.

TIFF output produced by `-dtiff` uses packbit compression, a lossless compression scheme that is supported by virtually all applications that can import TIFF graphics. If you need to import a TIFF file into an application that does not read packbit-compressed TIFF, use the `-dtiffnocompression` switch to produce an uncompressed TIFF file. (You can abbreviate this switch to `-dtiffn`.) For example,

```
print -dtiffn -r100 newplot.tif
```

An uncompressed TIFF file is often much larger than the same file compressed. For certain plots, the uncompressed file may be more than 10 times the size of the compressed file. The actual ratio will vary. The size of an uncompressed file depends only on the resolution and the width and height values in the PaperPosition figure property. The size of the compressed file also depends on the content of the figure.

You may also find the June 2000 MATLAB Digest article "Exporting Figures for Publication" helpful. Part 2 of this article is in the December 2000 issue.

Section 7: Can I change the default printer that MATLAB prints to from a script?

If no printer is specified, MATLAB uses your default printer. On Windows, you can set your default printer in the control panel. For information on how this is determined on UNIX, please see Solution 33377.

As of MATLAB R13, you can override the default printer with the `-P` option to specify the printer to which you want to print.

Summary of Default System PRINT Commands Used with MATLAB.

System	PRINT Command
UNIX	lp or lpr
PC	copy /B
MAC	Print (for 10.1) or lpr (for 10.2 and higher)

Section 8: Why are certain objects the wrong color when exported or printed?

Printing a figure having a background color (Color property) that is not white results in poor contrast between graphics objects and the figure background. It also consumes a lot of printer toner. Consequently, MATLAB changes the colors of some objects when printing. For more information on this effect, please see the documentation on the figure's InvertHardcopy property.

The following example illustrates the difference this may cause between on-screen and printed output:

```
figure
p(1) = patch([.1 .9 .9 .1 .1],[.1 .1 .9 .9 .1],'r')
p(2) = patch([.2 .8 .8 .2 .2],[.2 .2 .8 .8 .2],'g')
p(3) = patch([.3 .7 .7 .3 .3],[.3 .3 .7 .7 .3],'b')
p(4) = patch([.4 .6 .6 .4 .4],[.4 .4 .6 .6 .4],get(gcf,'color'))
axis off
print -djpeg test.jpg
```

In order to counter this effect, you may want to set the figure's Color property to white explicitly, and set the InvertHardcopy property of the figure to off. This will give you a more accurate picture of the colors as they were seen on-screen.

Section 9: What are my rendering options when I print from MATLAB? How do I choose a renderer?

MATLAB has three rendering methods available: Painters, ZBuffer, and OpenGL. Based on what is in your figure, MATLAB will choose a renderer for you and will print using that renderer. This is why your printout may not faithfully reproduce your original MATLAB figure.

If this happens, you can specify the renderer with which you want to print by adding the `-painters, -zbuffer, or -opengl` to your print command.

When you print with Painters, you will get a true rendering of your figure and you will get the results that you expect. However, this comes with an expense of speed. Printing with painters can be the slowest method if there are many graphics objects inside the figure.

Printing with Zbuffer or OpenGL can be faster because when you print using these renderers, you're getting a bitmap, rather than an actual re-rendering of the figure. This can cause resolution mismatch when rendering, printing or viewing. The more complex the figure, the more likely MATLAB is to choose Zbuffer or OpenGL as a renderer when printing.

If you need to print in Zbuffer or OpenGL (in order to preserve transparency, for example), then you can specify a resolution when printing, using the `-r` flag.

For more information on this, please see Solution 21560.

Section 10: What are the general steps I can take to debug printing problems?

There are several things to try to fix printing problems.

1. Change your printer drivers.

   There may be a conflict between MATLAB and your printer driver. To fix this problem, please contact the vendor of your printer for a different driver. You may want to try several versions of the same driver.

2. Change the renderer

   There are three rendering options available in MATLAB, Painters, Zbuffer, and OpenGL. For more information on which renderer to choose, please see the documentation.

   Sometimes specifying a different renderer can improve your printed output. To do this, pass the render as an option to your print command. The following example will use Painters to print the current figure to an EPS file and OpenGL to print the current figure to a tiff:

   ```
   print -painters -deps
   print -opengl -dtiff
   ```

3. Export to another application and print from there

   If you are not able get the results you are looking for by changing your printer driver or changing the renderer, you could try exporting your graphics to another application and printing from there. For a list of available formats you can print to, please see the documentation on the `PRINT` command.

   After you have created your output file, simply import it into your other application (like Microsoft Word or Paint) and then print from there.

4. Print to another format

   Sometimes, printing problems can occur if you are trying to print to a particular format. Printing to another format can solve these problems. For a list of available formats, please see the documentation on the PRINT command.

Section 11: Commonly asked Printing Questions

Why can't I print in MATLAB 6.5 (R13) on a MAC?

If you see errors about `PRINTDLG` when trying to print on the MAC, please see Solution 32500. Solution 32905 can help if you are receiving an error when you are printing from the drop down menu.

Do I have to print my entire figure window? How do I print a single set of axes?

There is no direct way to just print axes. If you have a figure with several sets of axis, you can copy the axis you want to print to a new figure and then print. For an example of this, please see Solution 30094.

What should I do if I get a blank or partial page when I try to print?

This is most likely due to a conflict between your printer driver and MATLAB. Detailed instructions on working around this problem, including changing your printer driver, changing your renderer, or printing to another format, please see Solution 24079.

When exporting a figure, MATLAB does not seem to honor my PaperPosition property.

An explanation of this circumstance is provided in the above section describing the PaperPosition property.

When printing from UNIX, I either get an error or no printout at all.

This issue is discussed in Solution 23310.

What should I do if noting happens when I try to print? What do I do if no output is produced from my printer?

This problem can often be fixed by changing the renderer that you are using. For more help on this problem, please see Solution 32075.

Why do I receive a black square when I try to print my figure window?

This problem can be fixed by using the painters renderer. For more help with this, please see Solution 29573.

Why do I have trouble printing when I am running MATLAB in the background (batch mode) under UNIX?

There are a few different causes for this problem. For more information please see Solution 31027 or Solution 30185.

Why do the JPEG and TIFF drivers output such big images?

For help with this, please see Solution 21704.

Section 12: What do I do if I need to contact technical support?

If you have gone through the above troubleshooting steps and are still unable to resolve the problem, please gather this information and send it to support@mathworks.com

Did this information help?	Yes ○	No ○	Didn't Apply ○
Is the level of technical detail appropriate?	Yes ○	Too Much ○	Not Enough ○

What did you expect to find on this page, that you want us to consider adding?

Additional Comments:

SUBMIT

related topics:

Demos | Search | Contact Support | Consulting | News & Notes | Usability