



print, printopt

Create hardcopy output

Syntax

```
print
print filename
print -ddriver
print -dformat
print -dformat filename
print -smodelname
print ... -options
[pcmd,dev] = printopt
```

Description

`print` and `printopt` produce hardcopy output. All arguments to the `print` command are optional. You can use them in any combination or order.

`print` sends the contents of the current figure, including bitmap representations of any user interface controls, to the printer using the device and system printing command defined by `printopt`.

`print filename` directs the output to the PostScript file designated by `filename`. If `filename` does not include an extension, `print` appends an appropriate extension.

`print -ddriver` prints the figure using the specified printer `driver`, (such as color PostScript). If you omit `-ddriver`, `print` uses the default value stored in `printopt.m`. The [Printer Driver](#) table lists all supported device types.

`print -dformat` copies the figure to the system clipboard (Windows only). A valid `format` for this operation is either `-dmeta` (Windows Enhanced Metafile) or `-dbitmap` (Windows Bitmap).

`print -dformat filename` exports the figure to the specified file using the specified graphics `format`, (such as TIFF). The [Graphics Format](#) table lists all supported graphics file formats.

`print -smodelname` prints the current Simulink model `modelname`.

`print -options` specifies print options that modify the action of the `print` command. (For example, the `-noui` option suppresses printing of user interface controls.) The [Options](#) section lists available options.

`print(...)` is the function form of `print`. It enables you to pass variables for any input arguments. This form is useful for passing filenames and handles. See [Batch Processing](#) for an example.

`[pcmd,dev] = printopt` returns strings containing the current system-dependent printing command and output device. `printopt` is an M-file used by `print` to produce the hardcopy output. You can edit the M-file `printopt.m` to set your default printer type and destination.

`pcmd` and `dev` are platform-dependent strings. `pcmd` contains the command that `print` uses to send a file to the printer. `dev` contains the printer driver or graphics format option for the `print` command. Their defaults are platform dependent.

Platform	System Printing Command	Driver or Format
UNIX	<code>lpr -r</code>	<code>-dps2</code>
Windows	<code>COPY /B %s LPT1:</code>	<code>-dwin</code>

Drivers

The table below shows the more widely used printer drivers supported by MATLAB. If you do not specify a driver, MATLAB uses the default setting shown in the previous table. For a list of all supported printer drivers, type

```
print -d
```

at the MATLAB prompt.

Some of the drivers are available from a product called Ghostscript, which is shipped with MATLAB. The last column indicates when Ghostscript is used.

Some drivers are not available on all platforms. This is noted in the first column of the table.

Printer Driver	PRINT Command Option String	Ghostscript
Canon BubbleJet BJ10e	<code>-dbj10e</code>	Yes
Canon BubbleJet BJ200 color	<code>-dbj200</code>	Yes
Canon Color BubbleJet BJC-70/BJC-600/BJC-4000	<code>-dbjc600</code>	Yes
Canon Color BubbleJet BJC-800	<code>-dbjc800</code>	Yes
DEC LN03	<code>-dln03</code>	Yes
Epson and compatible 9- or 24-pin dot matrix print drivers	<code>-depson</code>	Yes
Epson and compatible 9-pin with interleaved lines (triple resolution)	<code>-deps9high</code>	Yes
Epson LQ-2550 and compatible; color (not supported on HP-700)	<code>-depsonc</code>	Yes
Fujitsu 3400/2400/1200	<code>-depsonc</code>	Yes
HP DesignJet 650C color (not supported on Windows)	<code>-ddnj650c</code>	Yes
HP DeskJet 500	<code>-ddjet500</code>	Yes

HP DeskJet 500C (creates black and white output)	-dcdjmono	Yes
HP DeskJet 500C (with 24 bit/pixel color and high-quality Floyd–Steinberg color dithering) (not supported on Windows)	-dcdjcolor	Yes
HP DeskJet 500C/540C color (not supported on Windows)	-dcdj500	Yes
HP Deskjet 550C color (not supported on Windows)	-dcdj550	Yes
HP DeskJet and DeskJet Plus	-ddeskjet	Yes
HP LaserJet	-dlaserjet	Yes
HP LaserJet+	-dljetplus	Yes
HP LaserJet IIP	-dljet2p	Yes
HP LaserJet III	-dljet3	Yes
HP LaserJet 4.5L and 5P	-dljet4	Yes
HP LaserJet 5 and 6	-dpxlmono	Yes
HP PaintJet color	-dpaintjet	Yes
HP PaintJet XL color	-dpjxl	Yes
HP PaintJet XL color	-dpjetxl	Yes
HP PaintJet XL300 color (not supported on Windows)	-dpjxl300	Yes
HPGL for HP 7475A and other compatible plotters. (Renderer cannot be set to Z-buffer.)	-dhpgl	No
IBM 9-pin Proprinter	-dibmpro	Yes
PostScript black and white	-dps	No
PostScript color	-dpssc	No
PostScript Level 2 black and white	-dps2	No
PostScript Level 2 color	-dpssc2	No
Windows color (Windows only)	-dwinc	No
Windows monochrome (Windows only)	-dwin	No

Note Generally, Level 2 PostScript files are smaller and are rendered more quickly when printing than Level 1 PostScript files. However, not all PostScript printers support Level 2, so determine the capabilities of your printer before using those drivers. Level 2 PostScript is the default for UNIX. You can change this default by editing the `printopt.m` file. Likewise, if you want color PostScript to be the default instead of black-and-white PostScript, edit the line in the `printopt.m` file that reads `dev = '-dps2';` to be `dev = '-dpsc2';`

Graphics Format Files

To save your figure as a graphics-format file, specify a format switch and filename. To set the resolution of the output file for a built-in MATLAB format, use the `-r` switch. (For example, `-r300` sets the output resolution to 300 dots per inch.) The `-r` switch is also supported for Windows Enhanced Metafiles, JPEG, and PNG files, but is not supported for Ghostscript formats.

The table below shows the supported output formats for exporting from MATLAB and the switch settings to use. In some cases, a format is available both as a MATLAB output filter and as a Ghostscript output filter. All formats except for EMF are supported on both the PC and UNIX platforms.

Graphics Format	Bitmap or Vector	PRINT Command Option String	MATLAB or Ghostscript
BMP monochrome BMP	Bitmap	<code>-dbmpmono</code>	Ghostscript
BMP 16-bit BMP	Bitmap	<code>-dbmp16m</code>	Ghostscript
BMP 8-bit (256-color) BMP (this format uses a fixed colormap)	Bitmap	<code>-dbmp256</code>	Ghostscript
BMP 24-bit	Bitmap	<code>-dbmp</code>	MATLAB
EMF	Vector	<code>-dmeta</code>	MATLAB
EPS black and white	Vector	<code>-deps</code>	MATLAB
EPS color	Vector	<code>-depsc</code>	MATLAB
EPS Level 2 black and white	Vector	<code>-deps2</code>	MATLAB
EPS Level 2 color	Vector	<code>-depsc2</code>	MATLAB
HDF 24-bit	Bitmap	<code>-dhdf</code>	MATLAB
ILL (Adobe Illustrator)	Vector	<code>-dill</code>	MATLAB
JPEG 24-bit	Bitmap	<code>-djpeg</code>	MATLAB
PBM (plain format) 1-bit	Bitmap	<code>-dpbm</code>	Ghostscript
PBM (raw format) 1-bit	Bitmap	<code>-dpbmraw</code>	Ghostscript
PCX 1-bit	Bitmap	<code>-dpcxmono</code>	Ghostscript

PCX 24-bit color PCX file format, three 8-bit planes	Bitmap	-dpcx24b	Ghostscript
PCX 8-bit newer color PCX file format (256-color)	Bitmap	-dpcx256	Ghostscript
PCX Older color PCX file format (EGA/VGA, 16-color)	Bitmap	-dpcx16	Ghostscript
PCX 8-bit	Bitmap	-dpcx	MATLAB
PDF Color PDF file format	Vector	-dpdf	Ghostscript
PGM Portable Graymap (plain format)	Bitmap	-dpgm	Ghostscript
PGM Portable Graymap (raw format)	Bitmap	-dpgmraw	Ghostscript
PNG 24-bit	Bitmap	-dpng	MATLAB
PPM Portable Pixmap (plain format)	Bitmap	-dppm	Ghostscript
PPM Portable Pixmap (raw format)	Bitmap	-dppmraw	Ghostscript
TIFF 24-bit	Bitmap	-dtiff or -dtiffn	MATLAB
TIFF preview for EPS files	Bitmap	-tiff	

The TIFF image format is supported on all platforms by almost all word processors for importing images. JPEG is a lossy, highly compressed format that is supported on all platforms for image processing and for inclusion into HTML documents on the World Wide Web. To create these formats, MATLAB renders the figure using the Z-buffer rendering method and the resulting bitmap is then saved to the specified file.

Options

This table summarizes options that you can specify for `print`. The second column also shows which tutorial sections contain more detailed information. The sections listed are located under Printing and Exporting Figures with MATLAB.

Option	Description
-adobecset	PostScript only. Use PostScript default character set encoding. See Early PostScript 1 Printers .
-append	PostScript only. Append figure to existing PostScript file. See Settings That Are Driver Specific .
-cmyk	PostScript only. Print with CMYK colors instead of RGB. See Setting CMYK Color .

-ddriver	Printing only. Printer driver to use. See Drivers table.
-dformat	Exporting only. Graphics format to use. See Graphics Format Files table.
-dsetup	Display the Print Setup dialog.
-fhandle	Handle of figure to print. Note that you cannot specify both this option and the <code>-swindowtitle</code> option. See Which Figure Is Printed .
-loose	PostScript and Ghostscript only. Use loose bounding box for PostScript. See Producing Uncropped Figures .
-noui	Suppress printing of user interface controls. See Excluding User Interface Controls .
-opengl	Render using the OpenGL algorithm. Note that you cannot specify this method in conjunction with <code>-zbuffer</code> or <code>-painters</code> . See Selecting a Renderer .
-painters	Render using the Painter's algorithm. Note that you cannot specify this method in conjunction with <code>-zbuffer</code> or <code>-opengl</code> . See Selecting a Renderer .
-Pprinter	Specify name of printer to use. See Selecting Printer .
-rnumber	PostScript, JPEG, PNG, and Ghostscript only. Specify resolution in dots per inch. Defaults to 90 for Simulink, 150 for figures in image formats and when printing in Z-buffer or OpenGL mode, screen resolution for metafiles, and 864 otherwise. Use <code>-r0</code> to specify screen resolution. See Setting the Resolution .
-swindowtitle	Specify name of Simulink system window to print. Note that you cannot specify both this option and the <code>-fhandle</code> option. See Which Figure Is Printed .
-v	Windows only. Display the Windows Print dialog box. The <code>v</code> stands for "verbose mode."
-zbuffer	Render using the Z-buffer algorithm. Note that you cannot specify this method in conjunction with <code>-opengl</code> or <code>-painters</code> . See Selecting a Renderer .

Paper Sizes

MATLAB supports a number of standard paper sizes. You can select from the following list by setting the [PaperType](#) property of the figure or selecting a supported paper size from the Print dialog box.

Property Value	Size (Width by Height)
usletter	8.5 by 11 inches
uslegal	11 by 14 inches

tabloid	11 by 17 inches
A0	841 by 1189 mm
A1	594 by 841 mm
A2	420 by 594 mm
A3	297 by 420 mm
A4	210 by 297 mm
A5	148 by 210 mm
B0	1029 by 1456 mm
B1	728 by 1028 mm
B2	514 by 728 mm
B3	364 by 514 mm
B4	257 by 364 mm
B5	182 by 257 mm
arch-A	9 by 12 inches
arch-B	12 by 18 inches
arch-C	18 by 24 inches
arch-D	24 by 36 inches
arch-E	36 by 48 inches
A	8.5 by 11 inches
B	11 by 17 inches
C	17 by 22 inches
D	22 by 34 inches
E	34 by 43 inches

Printing Tips

This section includes information about specific printing issues.

Figures with Resize Functions

The `print` command produces a warning when you print a figure having a callback routine defined for the figure [ResizeFcn](#). To avoid the warning, set the figure [PaperPositionMode](#) property to `auto` or select **Match Figure Screen Size** in the **File->Page Setup** dialog box.

Troubleshooting MS Windows Printing

If you encounter problems such as segmentation violations, general protection faults, or application errors, or the output does not appear as you expect when

using MS–Windows printer drivers, try the following:

- If your printer is PostScript compatible, print with one of the MATLAB built–in PostScript drivers. There are various PostScript device options that you can use with the `print` command: they all start with `–dps`.
- The behavior you are experiencing might occur only with certain versions of the print driver. Contact the print driver vendor for information on how to obtain and install a different driver.
- Try printing with one of the MATLAB built–in Ghostscript devices. These devices use Ghostscript to convert PostScript files into other formats, such as HP LaserJet, PCX, Canon BubbleJet, and so on.
- Copy the figure as a Windows Enhanced Metafile using the **Edit→Copy Figure** menu item on the figure window menu or the `print –dmeta` option at the command line. You can then import the file into another application for printing.

You can set copy options in the figure's **File→Preferences→Copying Options** dialog box. The Windows Enhanced Metafile clipboard format produces a better quality image than Windows Bitmap.

Printing MATLAB GUIs

You can generally obtain better results when printing a figure window that contains MATLAB `uicontrols` by setting these key properties:

- Set the figure `PaperPositionMode` property to `auto`. This ensures that the printed version is the same size as the onscreen version. With `PaperPositionMode` set to `auto` MATLAB does not resize the figure to fit the current value of the `PaperPosition`. This is particularly important if you have specified a figure `ResizeFcn`, because if MATLAB resizes the figure during the print operation, `ResizeFcn` is automatically called.

To set `PaperPositionMode` on the current figure, use the command

```
set(gcf, 'PaperPositionMode', 'auto')
```

- Set the figure `InvertHardcopy` property to `off`. By default, MATLAB changes the figure `background` color of printed output to white, but does not change the color of `uicontrols`. If you have set the background color, for example, to match the gray of the GUI devices, you must set `InvertHardcopy` to `off` to preserve the color scheme.

To set `InvertHardcopy` on the current figure, use the command

```
set(gcf, 'InvertHardcopy', 'off')
```

- Use a color device if you want lines and text that are in color on the screen to be written to the output file as colored objects. Black and white devices convert colored lines and text to black or white to provide the best contrast with the background and to avoid dithering.
- Use the `print` command's `–loose` option to prevent MATLAB from using a bounding box that is tightly wrapped around objects contained in the figure. This is important if you have intentionally used space between `uicontrols` or axes and the edge of the figure and you want to maintain this appearance in the printed output.

Notes on Printing Interpolated Shading with PostScript Drivers

MATLAB can print [surface](#) objects (such as graphs created with [surf](#) or [mesh](#)) using interpolated colors. However, only [patch](#) objects that are composed of triangular faces can be printed using interpolated shading.

Printed output is always interpolated in RGB space, not in the colormap colors. This means that if you are using indexed color and interpolated face coloring, the printed output can look different from what is displayed on screen.

PostScript files generated for interpolated shading contain the color information of the graphics object's vertices and require the printer to perform the interpolation calculations. This can take an excessive amount of time and in some cases, printers might time out before finishing the print job. One solution to this problem is to interpolate the data and generate a greater number of faces, which can then be flat shaded.

To ensure that the printed output matches what you see on the screen, print using the `-zbuffer` option. To obtain higher resolution (for example, to make text look better), use the `-r` option to increase the resolution. There is, however, a tradeoff between the resolution and the size of the created PostScript file, which can be quite large at higher resolutions. The default resolution of 150 dpi generally produces good results. You can reduce the size of the output file by making the figure smaller before printing it and setting the figure `PaperPositionMode` to `auto`, or by just setting the `PaperPosition` property to a smaller size.

Examples

Specifying the Figure to Print

You can print a noncurrent figure by specifying the figure's handle. If a figure has the title "Figure 2", its handle is 2. The syntax is

```
print -fhandle
```

This example prints the figure whose handle is 2, regardless of which figure is the current figure.

```
print -f2
```

Note You must use the `-f` option if the figure's handle is hidden (i.e., its `HandleVisibility` property is set to `off`).

This example saves the figure with the handle `-f2` to a PostScript file named `Figure2`, which can be printed later.

```
print -f2 -dps 'Figure2.ps'
```

If the figure uses noninteger handles, use the `figure` command to get its value, and then pass it in as the first argument.

```
h = figure('IntegerHandle','off')
print h -depson
```

You can also pass a figure handle as a variable to the function form of `print`. For example,

```
h = figure; plot(1:4,5:8)
print(h)
```

This example uses the function form of `print` to enable a filename to be passed in as a variable.

```
filename = 'mydata';
print('-f3', '-dp3c', filename);
```

(Because a filename is specified, the figure will be printed to a file.)

Specifying the Model to Print

To print a noncurrent Simulink model, use the `-s` option with the title of the window. For example, this command prints the Simulink window titled `f14`.

```
print -sf14
```

If the window title includes any spaces, you must call the function form rather than the command form of `print`. For example, this command saves Simulink window title `Thruster Control`.

```
print('-sThruster Control')
```

To print the current system, use

```
print -s
```

For information about issues specific to printing Simulink windows, see the Simulink documentation.

Printing Figures at Screen Size

This example prints a surface plot with interpolated shading. Setting the current figure's (`gcf`) [PaperPositionMode](#) to `auto` enables you to resize the figure window and print it at the size you see on the screen. See [Options](#) and the previous section for information on the `-zbuffer` and `-r200` options.

```
surf(peaks)
shading interp
set(gcf,'PaperPositionMode','auto')
print -dp3c2 -zbuffer -r200
```

For additional details, see [Printing Images](#) in the MATLAB Graphics documentation.

Batch Processing

You can use the function form of `print` to pass variables containing file names. For example, this `for` loop uses filenames stored in a cell array to create a series of graphs and prints each one with a different file name.

```
fnames = {'file1', 'file2', 'file3'};
for k=1:length(fnames)
    surf(peaks)
    print('-dtiff', '-r200', fnames{k})
end
```

Tiff Preview

The command

```
print -depsc -tiff -r300 picture1
```

saves the current figure at 300 dpi, in a color Encapsulated PostScript file named `picture1.eps`. The `-tiff` option creates a 72 dpi TIFF preview, which many word processor applications can display on screen after you import the EPS file. This enables you to view the picture on screen within your word processor and print the document to a PostScript printer using a resolution of 300 dpi.

See Also

[orient](#), [figure](#)

◀ primes

printdlg ▶

© 1994–2005 The MathWorks, Inc. • [Terms of Use](#) • [Patents](#) • [Trademarks](#)