

guide2

COLLABORATORS

	<i>TITLE :</i> guide2		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		February 2, 2023	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	guide2	1
1.1	Startup Parameters, Parameter Files	1
1.2	Summary of all Parameters	2
1.3	Introduction to Parameters	5
1.4	using the dos command line	6
1.5	Setting Defaults (SSTOOLS.INI File)	6
1.6	Parameter Files and the <@> Command	7
1.7	general parameter syntax	8
1.8	Startup Parameters	8
1.9	calculation mode parameters	9
1.10	Fractal Type Parameters	10
1.11	Image Calculation Parameters	10
1.12	Color Parameters	12
1.13	Doodad Parameters	14
1.14	File Parameters	15
1.15	Video Parameters	16
1.16	Sound Parameters	19
1.17	Printer Parameters	19
1.18	postscript parameters	21
1.19	paintjet parameters	22
1.20	plotter parameters	23
1.21	3d parameters	24
1.22	Batch Mode	25
1.23	Video Adapter Notes	26
1.24	ega	27
1.25	tweaked vga	27
1.26	super-vga	28
1.27	8514/a	28
1.28	xga	29
1.29	targa	29
1.30	targa+	29
1.31	Disk-Video	30
1.32	Customized Video Modes, FRACTINT.CFG	31

Chapter 1

guide2

1.1 Startup Parameters, Parameter Files

Summary of all Parameters

Introduction to Parameters

Using the DOS Command Line

Setting Defaults (SSTOOLS.INI File)

Parameter Files and the <@> Command

General Parameter Syntax

Startup Parameters

Calculation Mode Parameters

Fractal Type Parameters

Image Calculation Parameters

Color Parameters

Doodad Parameters

File Parameters

Sound Parameters

Video Parameters

Printer Parameters

3D Parameters

1.2 Summary of all Parameters

Startup Parameters

@filename[/setname] Process commands from a file
 [filename=]filename Start with this saved file (one saved by FRACTINT
 or a generic GIF file [treated as a plasma cloud])
 ('filename=' is required except on command line)
 batch=yes Batch mode run (display image, save-to-disk, exit)
 autokey=play|record Playback or record keystrokes
 autokeyname=filename File for autokey mode, default AUTO.KEY
 fpu=387|iit|noit Assume 387 or IIT fpu is present or absent
 makedoc=filename Create Fractint documentation file

Calculation Mode Parameters

passes=1|2|g|b|t Select Single-Pass, Dual-Pass, Solid- ←
 Guessing,
 Boundary-Tracing, or the Tesseral drawing
 algorithms
 fillcolor=normal|<nnn> Sets a block fill color for use with Boundary
 Tracing and Tesseral options
 float=yes For most functions changes from integer math to fp
 symmetry=xxxx Force symmetry to None, Xaxis, Yaxis, XYaxis,
 Origin, or Pi symmetry. Useful for debugging.

Fractal Type Parameters

type=fractaltype Perform this Fractal Type (Default = mandel ←
)
 See Fractal Types for a full list
 params=xxx[/xxx[/... Begin with these extra Parameter values
 (Examples: params=4 params=-0.480/0.626)
 function=fn1/.../fn4 Allows specification of transcendental functions
 with types using variable functions. Values are
 sin, cos, tan, cotan, sinh, cosh, tanh, cotanh,
 exp, log, sqr, recip (1/z), ident (identity),
 conj, flip, zero, and cosxx (cos with bug)
 formulaname=name Formula name for 'type=formula' fractals
 lname=name Lsystem name for 'type=lsystem' fractals
 ifs=name IFS name for 'type=ifs' fractals
 3dmode=monocular|left|right|red-blue Sets the 3D mode used with Julibrot
 julibrot3d=nn[/nn[/nn[/nn[/nn[/nn]]]]) Sets Julibrot 3D parameters zdots,
 origin, depth, height, width, and distance
 julibroteyes=nn Distance between the virtual eyes for Julibrot
 julibrotfromto=nn/nn[/nn/nn] "From-to" parameters used for Julibrot
 miim=[depth|breadth|walk]/[left|right]/[xxx/yyy[/zzz]] Params for MIIM
 julias. xxx/yyy = julia constant, zzz = max hits.
 Eg. miim=depth/left/-.74543/.11301/3

Image Calculation Parameters

corners=xmin/xmax/ymin/ymax[/x3rd/y3rd]
 Begin with these Coordinates
 (Example: corners=-0.739/-0.736/0.288/0.291)
 center-mag=[Xctr/Yctr/Mag] An alternative method of entering corners.
 With no parameters causes command to output

'center-mag=' instead of corners.

maxiter=nnn Maximum number of iterations (default = 150)

bailout=nnnn Use this as the iteration bailout value (instead of the default (4.0 for most fractal types)

initorbit=nnn/nnn Sets the value used to initialize Mandelbrot orbits to the given complex number (real and imag parts)

orbitdelay=nn Slows up the display of orbits (by nn/10000 sec)

initorbit=pixel Sets the value used to initialize Mandelbrot orbits to the complex number corresponding to the screen pixel. This is the default for most types.

periodicity=[no|show|nnn] Controls periodicity checking. 'no' turns checking off; entering a number nnn controls the tightness of checking (default 1, higher is more stringent) 'show' or a neg value colors 'caught' points white.

rseed=nnnnn Random number seed, for reproducible Plasma Clouds

showdot=nn Colors the current dot being calculated color nn.

Color Parameters

inside=nnn|maxiter|zmag|bof60|bof61|epscr|star|per
Fractal interior color (inside=0 for black)

outside=nnn|iter|real|imag|mult|summ Fractal exterior color options

map=filename Use 'filename' as the default color map (vga/targa)

colors=@filename|colorspec Sets current image color map from file or spec, vga or higher only

cyclerrange=nnn/nnn Range of colors to cycle (default 1/255)

cyclelimit=nnn Color-cycler speed-limit (1 to 256, default = 55)

textcolors=aa/bb/cc/... Set text screen colors

textcolors=mono Set text screen colors to simple black and white

shortmap=yes Uses the compact .map file format

Doodad Parameters

logmap=yes|old|nn Yes maps logarithm of iteration to color. ←
Old uses

pre vsn 14 logic. >1 compresses, <-1 for quadratic.

ranges=nn/nn/nn/... Ranges of iteration values to map to colors

distest=nnn/nnn Distance Estimator Method

decomp=nn 'Decomposition' toggle, value 2 to 256.

biomorph=nnn Biomorph Coloring

potential=nn[/nn[/nn[/16bit]]] Continuous Potential

invert=nn/nn/nn Turns on inversion - turns images 'inside out'

finattract=yes Look for finite attractor in julia types

exitnoask=yes bypasses the final "are you sure?" exit screen

Sound Parameters

sound=off|x|y|z Nobody ever plays with fractals at work ←
, do they?
x|y|z can be used to add sound to attractor fractals, the orbits command, and reading GIFs.

hertz=nnn Base frequency for attractor sound effects

File Parameters

```

        savename=filename      Save files using this name (instead of ←
                               FRACT001)
overwrite=no|yes             Don't over-write existing files
savetime=nnn                 Autosave image every nnn minutes of calculation
gif87a=yes                   Save GIF files in the older GIF87a format (with
                               no FRACTINT extension blocks)
dither=yes                   Dither color GIFs read into a b/w display.
parmfile=filename            File for <@> and <b> commands, default FRACTINT.PAR
formulafile=filename         File for type=formula, default FRACTINT.FRM
lfile=filename               File for type=lsystem, default FRACTINT.L
ifsfile=filename             File for type=ifs, default FRACTINT.IFS
orbitsave=yes                Causes IFS and orbit fractals orbit points to be
                               saved in the file ORBITS.RAW

```

Video Parameters

```

        video=xxx             Begin with this video mode (Example: Video=F2)
askvideo=no                  Skip the prompt for video mode when restoring files
adapter=hgc|cga|ega|egamono|mcga|vga
                               Assume this (standard) video adapter is present
adapter=ATI|Everex|Trident|NCR|Video7|Genoa|Paradise|Chipstech|
                               Tseng3000|Tseng4000|AheadA|AheadB|Oaktech
                               Assume the named SuperVGA Chip set is present and
                               enable its non-standard SuperVGA modes.
afi=yes                       Disables the register-compatible 8514/A logic
                               and forces the use of the 8514/A API (HDILOAD)
textsafe=yes|no|bios|save    For use when images are not restored correctly on
                               return from a text display
exitmode=nn                  Sets the bios-supported videomode to use upon exit
                               (if not mode 3) - nn is the mode in hexadecimal
viewwindows=xx[/xx[/yes|no[/nn[/nn]]]]
                               Set the reduction factor, final media aspect ratio,
                               crop starting coordinates (y/n), explicit x size,
                               and explicit y size

```

Printer Parameters

```

        printer=type[/res[/lpt#[/ -1]]]
                               Set the printer type, dots/inch, and port#
types: IBM, EPSON, CO (Star Micronix),
        HP (LaserJet), PA (Paintjet),
        PS (PostScript portrait), PSL (landscape)
port# 1-3 LPTn, 11-14 COMn, 21-22 direct parallel,
        31-32 direct serial
linefeed=crlf|lf|cr         Control characters to emit at end of each line
title=yes                    Print a title with the output
printfile=filename           Print to specified file
epsf=1|2|3|...               Forces print to file; default filename fract001.eps,
                               forces PostScript mode
translate=yes|nnn            PostScript only; yes prints negative image;
                               >0 reduces image colors; <0 color reduce+negative
halftone=frq/angl/styl       PostScript: defines halftone screen
halftone=r/g/b               PaintJet: contrast adjustment
comport=port/baud/opt        COM port initialization. Port=1,2,3,etc.
        baud=115,150,300,600,1200,2400,4800,9600
        options 7,8 | 1,2 | e,n,o (any order)
        Example: comport=1/9600/n71

```

colorps=yes|no Enable or Disable the color postscript extensions
 rleps=yes|no Enable or Disable the postscript rle encoding

3D Parameters

3d=yes|overlay Resets 3D to defaults, starts 3D mode. If ←
 overlay
 specified, does not clear existing graphics screen
 preview=yes Turns on 3D 'preview' default mode
 showbox=yes Turns on 3D 'showbox' default mode
 sphere=yes Turns on 3D sphere mode
 coarse=nnn Sets Preview 'coarseness' default value
 stereo=nnn Sets Stereo (R/B 3D) option: 0 = none,
 1 = alternate, 2 = superimpose, 3 = photo
 ray=nnn selects raytrace output file format
 brief=yes selects brief or verbose file for DKB output

interocular=nnn Sets 3D Interocular distance default value
 converge=nnn Sets 3D Convergence default value
 crop=nnn/nnn/nnn/nnn Sets 3D red-left, red-right, blue-left,
 and blue-right cropping default values
 bright=nnn/nnn Sets 3D red and blue brightness defaults,

longitude=nn/nn Longitude minimum and maximum
 latitude=nn/nn Latitude minimum and maximum
 radius=nn Radius scale factor
 rotation=nn[/nn[/nn]] Rotation about x,y, and z axes
 scalexyz=nn/nn/nn X, Y, and Z scale factors
 roughness=nn Same as Z scale factor
 waterline=nn Colors this number and below will be 'inside' color
 filltype=nn 3D filltype
 perspective=nn Perspective viewer distance (100 is at the edge)
 xyshift=nn/nn Shift image in x & y directions (alters viewpoint)
 lightsource=nn/nn/nn The coordinates of the light source vector
 smoothing=nn Smooths rough images in light source mode
 transparent=mm/nn Sets colors 'mm' to 'nn as transparent
 xadjust=nnn/nnn Sets 3D X and Y adjustment defaults,
 randomize=nnn smoothes 3d color transitions between elevations
 fullcolor=yes allows creation of full color .TGA image with
 light source fill types
 ambient=nnn sets depth of shadows and contrast when using
 light source fill types
 haze=nnn sets haze for distant objects if fullcolor=1
 lightname=filename fullcolor output file name, default FRACT001.TGA

1.3 Introduction to Parameters

Fractint accepts command-line parameters that allow you to start it with a particular video mode, fractal type, starting coordinates, and just about every other parameter and option.

These parameters can also be specified in a SSTOOLS.INI file, to set them every time you run Fractint.

They can also be specified as named groups in a .PAR (parameter) file which you can then call up while running Fractint by using the <@> command.

In all three cases (DOS command line, SSTOOLS.INI, and parameter file) the parameters use the same syntax, usually a series of keyword=value commands like SOUND=OFF. Each parameter is described in detail in subsequent sections.

1.4 using the dos command line

You can specify parameters when you start Fractint from DOS by using a command like:

```
FRACTINT SOUND=OFF FILENAME=MYIMAGE.GIF
```

The individual parameters are separated by one or more spaces (an parameter itself may not include spaces). Upper or lower case may be used, and parameters can be in any order.

Since DOS commands are limited to 128 characters, Fractint has a special command you can use when you have a lot of startup parameters (or have a set of parameters you use frequently):

```
FRACTINT @MYFILE
```

When @filename is specified on the command line, Fractint reads parameters from the specified file as if they were keyed on the command line. You can create the file with a text editor, putting one "keyword=value" parameter on each line.

1.5 Setting Defaults (SSTOOLS.INI File)

Every time Fractint runs, it searches the current directory, and then the directories in your DOS PATH, for a file named SSTOOLS.INI. If it finds this file, it begins by reading parameters from it. This file is useful for setting parameters you always want, such as those defining your printer setup.

SSTOOLS.INI is divided into sections belonging to particular programs. Each section begins with a label in brackets. Fractint looks for the label [fractint], and ignores any lines it finds in the file belonging to any other label. If an SSTOOLS.INI file looks like this:

```
[fractint]
sound=off ; (for home use only)
printer=hp ; my printer is a LaserJet
inside=0 ; using "traditional" black
[startrek]
warp=9.5 ; Captain, I dinna think the engines can take it!
```

Fractint will use only the second, third, and fourth lines of the file.

(Why use a convention like that when Fractint is the only program you know of that uses an SSTOOLS.INI file? Because there are other programs (such as Lee Crocker's PICLAB) that now use the same file, and there may one day be other, sister programs to Fractint using that file.)

1.6 Parameter Files and the <@> Command

You can change parameters on-the-fly while running Fractint by using the <@> command and a parameter file. Parameter files contain named groups of parameters, looking something like this:

```
quickdraw {      ; a set of parameters named quickdraw
  maxiter=150
  float=no
}
slowdraw {       ; another set of parameters
  maxiter=2000
  float=yes
}
```

If you use the <@> command and select a parameter file containing the above example, Fractint will show two choices: quickdraw and slowdraw. You move the cursor to highlight one of the choices and press <Enter> to set the parameters specified in the file by that choice.

The default parameter file name is FRACTINT.PAR. A different file can be selected with the "parmfile=" option, or by using <@> and then hitting <F6>.

You can create parameter files with a text editor, or for some uses, by using the command. Parameter files can be used in a number of ways, some examples:

- o To save the parameters for a favorite image. Fractint can do this for you with the command.
- o To save favorite sets of 3D transformation parameters. Fractint can do this for you with the command.
- o To set up different sets of parameters you use occasionally. For instance, if you have two printers, you might want to set up a group of parameters describing each.
- o To save image parameters for later use in batch mode - see

Batch Mode

.

See Parameter Save/Restore Commands for details about the <@> and commands.

1.7 general parameter syntax

Parameters must be separated by one or more spaces.

Upper and lower case can be used in keywords and values.

Anything on a line following a ; (semi-colon) is ignored, i.e. is a comment.

In parameter files and SSTOOLS.INI:

- o Individual parameters can be entered on separate lines.
- o Long values can be split onto multiple lines by ending a line with a (backslash) - leading spaces on the following line are ignored, the information on the next line from the first non-blank character onward is appended to the prior line.

Some terminology:

- KEYWORD=nnn enter a number in place of "nnn"
- KEYWORD=[filename] you supply filename
- KEYWORD=yes|no|whatever choose one of "yes", "no", or "whatever"
- KEYWORD=1st[/2nd[/3rd]] the slash-separated parameters "2nd" and "3rd" are optional

1.8 Startup Parameters

 Causes Fractint to read "filename" for parameters. When it finishes, it resumes reading its own command line -- i.e., "FRACTINT MAXITER=250 line, as Fractint is not clever enough to deal with multiple indirection. ←

Like @FILENAME, but reads a named group of parameters from a parameter file. See

Parameter Files and the <@> Command

.

FILENAME=[name]

Causes Fractint to read the named file, which must either have been saved from an earlier Fractint session or be a generic GIF file, and use that as the starting point, bypassing the initial information screens. The filetype is optional and defaults to .GIF. Non-Fractint GIF files are restored as fractal type "plasma".

On the DOS command line you may omit FILENAME= and just give the file name.

BATCH=yes

See

Batch Mode

.

AUTOKEY=play|record

Specifying "play" runs Fractint in playback mode - keystrokes are read from the autokey file (see next parameter) and interpreted as if they're being entered from the keyboard.

Specifying "record" runs in recording mode - all keystrokes are recorded in the autokey file.

See also Autokey Mode.

AUTOKEYNAME=[filename]

Specifies the file name to be used in autokey mode. The default file name is AUTO.KEY.

FPU=387|IIT|NOIIT

This parameter is useful if you have an unusual coprocessor chip. If you have a 80287 replacement chip with full 80387 functionality use "FPU=387" to inform Fractint to take advantage of those extra 387 instructions. If you have the IIT fpu, but don't have IIT's 'f4x4int.com' TSR loaded, use "FPU=IIT" to force Fractint to use that chip's matrix multiplication routine automatically to speed up 3-D transformations (if you have an IIT fpu and have that TSR loaded, Fractint will auto-detect the presence of the fpu and TSR and use its extra capabilities automatically). Since all IIT chips support 80387 instructions, enabling the IIT code also enables Fractint's use of all 387 instructions. Setting "FPU=NOIIT" disables Fractint's IIT Auto-detect capability. Warning: multi-tasking operating systems such as Windows and DesQView don't automatically save the IIT chip extra registers, so running two programs at once that both use the IIT's matrix multiply feature but don't use the handshaking provided by that 'f4x4int.com' program, errors will result.

MAKEDOC[=filename]

Create Fractint documentation file (for printing or viewing with a text editor) and then return to DOS. Filename defaults to FRACTINT.DOC. There's also a function in Fractint's online help which can be used to produce the documentation file -

see Printing Fractint Documentation.

use "Printing Fractint Documentation" from the main help index.

1.9 calculation mode parameters

PASSES=1|2|guess|btm|tesseral

Selects single-pass, dual-pass, solid-Guessing mode, Boundary Tracing, or the Tesseral algorithm. See Drawing Method.

FILLCOLOR=normal|<nnn>

Sets a color to be used for block fill by Boundary Tracing and Tesseral algorithms. See Drawing Method.

FLOAT=yes

Most fractal types have both a fast integer math and a floating point version. The faster, but possibly less accurate, integer version is the default. If you have a new 80486 or other fast machine with a math coprocessor, or if you are using the continuous potential option (which looks best with high bailout values not possible with our integer math implementation), you may prefer to use floating point. Just add "float=yes" to the command line to do so.

Also see Limitations of Integer Math .

SYMMETRY=xxx

Forces symmetry to None, Xaxis, Yaxis, XYaxis, Origin, or Pi symmetry.
Useful for debugging.

1.10 Fractal Type Parameters

TYPE=[name]

Selects the fractal type to calculate. The default is type "mandel".

PARAMS=n/n/n/n...

Set optional (required, for some fractal types) values used in the calculations. These numbers typically represent the real and imaginary portions of some startup value, and are described in detail as needed in Fractal Types.

(Example: FRACTINT TYPE=julia PARAMS=-0.48/0.626 would wait at the opening screen for you to select a video mode, but then proceed straight to the Julia set for the stated x (real) and y (imaginary) coordinates.)

FUNCTION=[fn1[/fn2[/fn3[/fn4]]]]

Allows setting variable functions found in some fractal type formulae. Possible values are sin, cos, tan, cotan, sinh, cosh, tanh, cotanh, exp, log, sqr, recip (i.e. 1/z), ident (i.e. identity), and cosxx (cos with a pre version 16 bug).

FORMULANAME=[formulaname]

Specifies the default formula name for type=formula fractals. (I.e. the name of a formula defined in the FORMULAFILE.) Required if you want to generate one of these fractal types in batch mode, as this is the only way to specify a formula name in that case.

LNAME=[lssystemname]

Specifies the default L-System name. (I.e. the name of an entry in the LFILE.) Required if you want to generate one of these fractal types in batch mode, as this is the only way to specify an L-System name in that case.

IFS=[ifsname]

Specifies the default IFS name. (I.e. the name of an entry in the IFSFILE.) Required if you want to generate one of these fractal types in batch mode, as this is the only way to specify an IFS name in that case.

1.11 Image Calculation Parameters

MAXITER=nnn

Reset the iteration maximum (the number of iterations at which the program gives up and says 'OK, this point seems to be part of the set in question and should be colored [insidecolor]') from the default 150. Values range from 10 to 32000 (super-high iteration limits like 30000 are useful when using logarithmic palettes). See The Mandelbrot Set for a description of the iteration method of calculating fractals.

"maxiter=" can also be used to adjust the number of orbits plotted for 3D "attractor" fractal types such as lorenz3d and kamtorus.

CORNERS=xmin/xmax/ymin/ymax[/x3rd/y3rd]

Example: corners=-0.739/-0.736/0.288/0.291

Begin with these coordinates as the range of x and y coordinates, rather than the default values of (for type=mandel) -2.0/2.0/-1.5/1.5. When you specify four values (the usual case), this defines a rectangle: x-coordinates are mapped to the screen, left to right, from xmin to xmax, y-coordinates are mapped to the screen, bottom to top, from ymin to ymax. Six parameters can be used to describe any rotated or stretched parallelogram: (xmin,ymax) are the coordinates used for the top-left corner of the screen, (xmax,ymin) for the bottom-right corner, and (x3rd,y3rd) for the bottom-left.

CENTER-MAG=[Xctr/Yctr/Mag]

This is an alternative way to enter corners as a center point and a magnification that is popular with some fractal programs and publications. Entering just "CENTER-MAG=" tells Fractint whether to use this form rather than corners when saving parameters with the command. The <TAB> status display shows the "corners" in both forms. Note that an aspect ratio of 1.3333 is assumed; if you have altered the zoom box proportions or rotated the zoom box, this form can no longer be used.

BAILOUT=nnn

Over-rides the default bailout criterion for escape-time fractals. Can also be set from the parameters screen after selecting a fractal type. See description of bailout in The Mandelbrot Set.

RESET

Causes Fractint to reset all calculation related parameters to their default values. Non-calculation parameters such as "printer=", "sound=", and "savename=" are not affected. RESET should be specified at the start of each parameter file entry (used with the <@> command) which defines an image, so that the entry need not describe every possible parameter - when invoked, all parameters not specifically set by the entry will have predictable values (the defaults).

INITORBIT=pixel

INITORBIT=nnn/nnn

Allows control over the value used to begin each Mandelbrot-type orbit. "initorbit=pixel" is the default for most types; this command initializes the orbit to the complex number corresponding to the screen pixel. The command "initorbit=nnn/nnn" uses the entered value as the initializer. See the discussion of the Mandellambda Sets for more on this topic.

ORBITDELAY=<nn>

Slows up the display of orbits using the <o> command for folks with hot new computers. Units are in 1/10000 seconds per orbit point. ORBITDELAY=10 therefore allows you to see each pixel's orbit point for about one millisecond. For best display of orbits, try passes=1 and a moderate resolution such as 320x200. Note that the first time you press the 'o' key with the 'orbitdelay' function active, your computer will pause for a half-second or so to calibrate a high-resolution timer.

PERIODICITY=no|show|nnn

Controls periodicity checking (see Periodicity Logic).

"no" turns it off, "show" lets

you see which pixels were painted as "inside" due to being caught by periodicity. Specifying a number causes a more conservative periodicity test (each increase of 1 divides test tolerance by 2). Entering a negative number lets you turn on "show" with that number. Type `lambdafn function=exp` needs periodicity turned off to be accurate -- there may be other cases.

`RSEED=nnnn`

The initial random-number "seed" for plasma clouds is taken from your PC's internal clock-timer. This argument forces a value (which you can see in the <Tab> display), and allows you to reproduce plasma clouds. A detailed discussion of why a TRULY random number may be impossible to define, let alone generate, will have to wait for "FRACTINT: The 3-MB Doc File."

`SHOWDOT=<nn>`

Colors the pixel being calculated color <nn>. Useful for very slow fractals for showing you the calculation status.

1.12 Color Parameters

`INSIDE=nnn|bof60|bof61|zmag|attractor|epscross|startrail|period`

Set the color of the interior: for

example, "inside=0" makes the M-set "lake" a stylish basic black. A setting of -1 makes `inside=maxiter`. Three more options reveal hidden structure inside the lake. `Inside=bof60` and `inside=bof61`, are named after the figures on pages 60 and 61 of "Beauty of Fractals". See `Inside=bof60|bof61|zmag|period` for a brilliant explanation of what these do!

`Inside=zmag` is a method of coloring based on the magnitude of Z after the maximum iterations have been reached. The affect along the edges of the Mandelbrot is like thin-metal welded sculpture.

`Inside=epscross` colors pixels green or yellow according to whether their orbits swing close to the Y-axis or X-axis, respectively. `Inside=starcross` has a coloring scheme based on clusters of points in the orbits. Best with `outside=<nnn>`. For more information, see `Inside=epscross|startrail`.

`Inside=period` colors pixels according to the period of their eventual orbit.

Note that the "Look for finite attractor" option on the <Y> options screen will override the selected inside option if an attractor is found - see Finite Attractors.

`OUTSIDE=nnn|iter|real|imag|summ|mult`

The classic method of coloring outside

the fractal is to color according to how many iterations were required before Z reached the bailout value, usually 4. This is the method used when

`OUTSIDE=iter`.

However, when Z reaches bailout the real and imaginary components can be at very different values. `OUTSIDE=real` and `OUTSIDE=imag` color using the iteration value plus the real or imaginary values. `OUTSIDE=summ` uses the sum of all these values. These options can give a startling 3d quality to otherwise flat images and can change some boring images to wonderful ones. `OUTSIDE=mult` colors by multiplying the iteration by real divided by imaginary. There was no

mathematical reason for this, it just seemed like a good idea.

Outside=nnn sets the color of the exterior to some number of your choosing: for example, "OUTSIDE=1" makes all points not INSIDE the fractal set to color 1 (blue). Note that defining an OUTSIDE color forces any image to be a two-color one: either a point is INSIDE the set, or it's OUTSIDE it.

MAP=[filename]

Reads in a replacement color map from [filename]. This map replaces the default color map of your video adapter. Requires a VGA or higher adapter. The difference between this argument and an alternate map read in via <L> in color-command mode is that this one applies to the entire run. See Palette Maps.

COLORS=@filename|colorspecification

Sets colors for the current image, like the <L> function in color cycling and palette editing modes. Unlike the MAP= parameter, colors set with COLORS= do not replace the default - when you next select a new video mode, colors will revert to their defaults.

COLORS=@filename tells Fractint to use a color map file named "filename". See Palette Maps.

COLORS=colorspecification specifies the colors directly. The value of "colorspecification" is rather long (768 characters for 256 color modes), and its syntax is not documented here. This form of the COLORS= command is not intended for manual use - it exists for use by the command when saving the description of a nice image.

CYCLERANGE=nnn/nnn

Sets the range of color numbers to be animated during color cycling. The default is 1/255, i.e. just color number 0 (usually black) is not cycled.

CYCLELIMIT=nnn

Sets the speed of color cycling. Technically, the number of DAC registers updated during a single vertical refresh cycle. Legal values are 1 - 256, default is 55.

TEXTCOLORS=mono

Set text screen colors to simple black and white.

TEXTCOLORS=aa/bb/cc/...

Set text screen colors. Omit any value to use the default (e.g. textcolors=///50 to set just the 5th value). Each value is a 2 digit hexadecimal value; 1st digit is background color (from 0 to 7), 2nd digit is foreground color (from 0 to F).

Color values are:

0	black	8	gray
1	blue	9	light blue
2	green	A	light green
3	cyan	B	light cyan
4	red	C	light red
5	magenta	D	light magenta
6	brown	E	yellow
7	white	F	bright white

31 colors can be specified, their meanings are as follows:

```

heading:
  1 Fractint version info
  2 heading line development info (not used in released version)
help:
  3 sub-heading
  4 main text
  5 instructions at bottom of screen
  6 hotlink field
  7 highlighted (current) hotlink
menu, selection boxes, parameter input boxes:
  8 background around box and instructions at bottom
  9 emphasized text outside box
 10 low intensity information in box
 11 medium intensity information in box
 12 high intensity information in box (e.g. heading)
 13 current keyin field
 14 current keyin field when it is limited to one of n values
 15 current choice in multiple choice list
 16 speed key prompt in multiple choice list
 17 speed key keyin in multiple choice list
general (tab key display, IFS parameters, "thinking" display):
 18 high intensity information
 19 medium intensity information
 20 low intensity information
 21 current keyin field
disk video:
 22 background around box
 23 high intensity information
 24 low intensity information
diagnostic messages:
 25 error
 26 information
credits screen:
 27 bottom lines
 28 high intensity divider line
 29 low intensity divider line
 30 primary authors
 31 contributing authors
The default is
  textcolors=1F/1A/2E/70/28/71/31/78/70/17/
              1F/1E/2F/3F/5F/07/0D/71/70/78/0F/
              70/0E/0F/4F/20/17/20/28/0F/07
(In a real command file, all values must be on one line.)

```

1.13 Doodad Parameters

LOGMAP=yes|old|n

Selects a compressed relationship between escape-time iterations and palette colors. See Logarithmic Palettes and Color Ranges for details.

RANGES=nn/nn/nn/...

Specifies ranges of escape-time iteration counts to be mapped to each color number. See Logarithmic Palettes and Color Ranges for details.

DISTEST=nnn/nnn

A nonzero value in the first parameter enables the distance estimator method. The second parameter specifies the "width factor", defaults to 71. See Distance Estimator Method for details.

DECOMP=2|4|8|16|32|64|128|256

Invokes the corresponding decomposition coloring scheme. See Decomposition for details.

BIOMORPH=nnn

Turn on biomorph option; set affected pixels to color nnn. See Biomorphs for details.

POTENTIAL=maxcolor[/slope[/modulus[/16bit]]]

Enables the "continuous potential" coloring mode for all fractal types except plasma clouds, attractor types such as lorenz, and IFS. The four arguments define the maximum color value, the slope of the potential curve, the modulus "bailout" value, and whether 16 bit values are to be calculated. Example: "POTENTIAL=240/2000/40/16bit". The Mandelbrot and Julia types ignore the modulus bailout value and use their own hardwired value of 4.0 instead. See Continuous Potential for details.

INVERT=nn/nn/nn

Turns on inversion. The parameters are radius of inversion, x-coordinate of center, and y-coordinate of center. -1 as the first parameter sets the radius to 1/6 the smaller screen dimension; no x/y parameters defaults to center of screen. The values are displayed with the <Tab> command. See Inversion for details.

FINATTRACT=no|yes

Another option to show coloring inside some Julia "lakes" to show escape time to finite attractors. Works with lambda, magnet types, and possibly others. See Finite Attractors for more information.

EXITNOASK=yes

This option forces Fractint to bypass the final "are you sure?" exit screen when the ESCAPE key is pressed from the main image-generation screen. Added at the request of Ward Christensen. It's his funeral <grin>.

1.14 File Parameters

SAVENAME=[name]

Set the filename to use when you <S>ave a screen. The default filename is FRACT001. The .GIF extension is optional (Example: SAVENAME=myfile)

OVERWRITE=no|yes

Sets the savename overwrite flag (default is 'no'). If 'yes', saved files will over-write existing files from previous sessions; otherwise the automatic incrementing of FRACTnnn.GIF will find the first unused filename.

SAVETIME=nnn

Tells Fractint to automatically do a save every nnn minutes while a calculation is in progress. This is mainly useful with long batches - see

Batch Mode

GIF87a=YES

Backward-compatibility switch to force creation of GIF files in the GIF87a format. As of version 14, Fractint defaults to the new GIF89a format which permits storage of fractal information within the format. GIF87a=YES is only needed if you wish to view Fractint images with a GIF decoder that cannot accept the newer format. See GIF Save File Format.

DITHER=YES

Dither a color file into two colors for display on a b/w display. This gives a poor-quality display of gray levels. Note that if you have a 2-color display, you can create a 256-color gif with disk video and then read it back in dithered.

PARMFILE=[parmfilename]

Specifies the default parameter file to be used by the <@> and commands. If not specified, the default is FRACTINT.PAR.

FORMULAFILE=[formulafilename]

Specifies the formula file for type=formula fractals (default is FRACTINT.FRM). Handy if you want to generate one of these fractal types in batch mode.

LFILE=[lssystemfile]

Specifies the default L-System file for type=lssystem fractals (if not FRACTINT.L).

IFSFILE=[ifsfilename]

Specifies the default file for type=ifs fractals (default is FRACTINT.IFS).

FILENAME=[.suffix]

Sets the default file extension used for the <r> command.

When this parameter is omitted, the default file mask shows .GIF and .POT files. You might want to specify this parameter and the SAVENAME= parameter in your SSTOOLS.INI file if you keep your fractal images separate from other .GIF files by using a different suffix for them.

ORBITSAVE=yes

Causes the file ORBITS.RAW to be opened and the points generated by orbit fractals or IFS fractals to be saved in a raw format. This file can be read by the Acrospin program which can rotate and scale the image rapidly in response to cursor-key commands. The filename ORBITS.RAW is fixed and will be overwritten each time a new fractal is generated with this option. (see Barnsley IFS Fractals Orbit Fractals Acrospin);

1.15 Video Parameters

VIDEO=xxx

Set the initial video mode (and bypass the informational screens). Handy for batch runs. (Example: VIDEO=F4 for IBM 16-color VGA.)

You can obtain the current VIDEO= values (key assignments) from the "select video mode" screens inside Fractint. If you want to do a batch run

with a video mode which isn't currently assigned to a key, you'll have to modify the key assignments - see Video Mode Function Keys.

ASKVIDEO=yes|no

If "no," this eliminates the prompt asking you if a file to be restored is OK for your current video hardware.

WARNING: every version of Fractint so far has had a bigger, better, but shuffled-around video table. Since calling for a mode your hardware doesn't support can leave your system in limbo, be careful about leaving the above two parameters in a command file to be used with future versions of Fractint, particularly for the super-VGA modes.

ADAPTER=hgc|cga|ega|egamono|mcga|vga|ATI|Everex|Trident|NCR|Video7|Genoa|Paradise|Chipstech|Tseng3000|Tseng4000|AheadA|AheadB|Oaktech
Bypasses Fractint's internal video autodetect logic and assumes that the specified kind of adapter is present. Use this parameter only if you encounter video problems without it. Specifying adapter=vga with an SVGA adapter will make its extended modes unusable with Fractint. All of the options after the "VGA" option specify specific SuperVGA chipsets which are capable of video resolutions higher than that of a "vanilla" VGA adapter. Note that Fractint cares about the Chipset your adapter uses internally, not the name of the company that sold it to you.

VESADETECT=yes|no

Specify no to bypass VESA video detection logic. Try this if you encounter video problems with a VESA compliant video adapter or driver.

AFI=yes|8514|no

Normally, when you attempt to use an 8514/A-specific video mode, Fractint first attempts to detect the presence of an 8514/A register-compatible adapter. If it fails to find one, it then attempts to detect the presence of an 8514/A-compatible API (IE, IBM's HDILOAD or its equivalent). Fractint then uses either its register-compatible or its API-compatible video logic based on the results of those tests. If you have an "8514/A-compatible" video adapter that passes Fractint's register-compatible detection logic but doesn't work correctly with Fractint's register-compatible video logic, setting "afi=yes" will force Fractint to bypass the register-compatible code and look only for the API interface.

TEXTSAFE=yes|no|bios|save

When you switch from a graphics image to text mode (e.g. when you use <F1> while a fractal is on display), Fractint remembers the graphics image, and restores it when you return from the text mode.

This should be no big deal - there are a number of well-defined ways Fractint could do this which *should* work on any video adapter. They don't - every fast approach we've tried runs into a bug on one video adapter or another. So, we've implemented a fast way which works on most adapters in most modes as the default, and added this parameter for use when the default approach doesn't work.

If you experience the following problems, please fool around with this parameter to try to fix the problem:

- o Garbled image, or lines or dashes on image, when returning to image after going to menu, <tab> display, or help.
- o Blank screen when starting Fractint.

The problems most often occur in higher resolution modes. We have not encountered them at all in modes under 320x200x256 - for those modes

Fractint always uses a fast image save/restore approach.

Textsafe options:

- yes: This is the default. When switching to/from graphics, Fractint saves just that part of video memory which EGA/VGA adapters are supposed to modify during the mode changes.
- no: This forces use of monochrome 640x200x2 mode for text displays (when there is a high resolution graphics image to be saved.) This choice is fast but uses chunky and colorless characters. If it turns out to be the best choice for you, you might want to also specify "textcolors=mono" for a more consistent appearance in text screens.
- bios: This saves memory in the same way as textsafe=yes, but uses the adapter's BIOS routines to save/restore the graphics state. This approach is fast and ought to work on all adapters. Sadly, we've found that very few adapters implement this function perfectly.
- save: This is the last choice to try. It should work on all adapters in all modes but it is slow. It tells Fractint to save/restore the entire image. Expanded or extended memory is used for the save if you have enough available; otherwise a temporary disk file is used. The speed of textsafe=save will be acceptable on some machines but not others. The speed depends on:
 - o Cpu and video adapter speed.
 - o Whether enough expanded or extended memory is available.
 - o Video mode of image being remembered. A few special modes are *very* slow compared to the rest. The slow ones are: 2 and 4 color modes with resolution higher than 640x480; custom modes for ATI EGA Wonder, Paradise EGA-480, STB, Compaq portable 386, AT&T 6300, and roll-your-own video modes implemented with customized "yourvid.c" code.

If you want to tune Fractint to use different "textsafe" options for different video modes, see

Customized Video Modes, FRACTINT.CFG

(E.g. you might

want to use the slower textsafe=save approach just for a few high-resolution modes which have problems with textsafe=yes.)

EXITMODE=nn

Sets the bios-supported videomode to use upon exit to the specified value. nn is in hexadecimal. The default is 3, which resets to 80x25 color text mode on exit. With Hercules Graphics Cards, and with monochrome EGA systems, the exit mode is always 7 and is unaffected by this parameter.

TPLUS=yes|no

For TARGA+ adapters. Setting this to 'no' pretends a TARGA+ is NOT installed.

NONINTERLACED=yes|no

For TARGA+ adapters. Setting this to 'yes' will configure the adapter to a non-interlaced mode whenever possible. It should only be used with a multisynch monitor. The default is no, i.e. interlaced.

MAXCOLORRES=8|16|24

For TARGA+ adapters. This determines the number of bits to use for color resolution. 8 bit color is equivalent to VGA color resolution. The 16 and 24 bit color resolutions are true color video modes which are not yet supported by Fractint but are hopefully coming soon.

PIXELZOOM=0|1|2|3

For TARGA+ adapters. Lowers the video mode resolution by powers of 2. For example, the 320x200 video resolution on the TARGA+ is actually the 640x400 video mode with a pixel zoom of 1. Using the 640x400 video mode with a zoom of 3 would lower the resolution by 8, which is 2 raised to the 3rd power, for a full screen resolution of 80x50 pixels.

VIEWWINDOWS=xx[/xx[/yes|no[/nn[/nn]]]]

Set the reduction factor, final media aspect ratio, crop starting coordinates (y/n), explicit x size, and explicit y size, see View Window.

1.16 Sound Parameters

SOUND=off|x|y|z

We're all MUCH too busy to waste time with Fractint at work, and no doubt you are too, so "sound=off" is included only for use at home, to avoid waking the kids or your Significant Other, late at night. (By the way, didn't you tell yourself "just one more zoom on LambdaSine" an hour ago?) Suggestions for a "boss" hot-key will be cheerfully ignored, as this sucker is getting big enough without including a spreadsheet screen too. The "sound=x/y/z" options are for the "attractor" fractals, like the Lorenz fractals - they play with the sound on your PC speaker as they are generating an image, based on the X or Y or Z co-ordinate they are displaying at the moment. At the moment, "sound=x" (or y or z) really doesn't work very well when using an integer algorithm - try it with the floating-point toggle set, instead.

The scope of the sound command has been extended. You can now hear the sound of fractal orbits--just turn on sound from the command line or the <X> menu, fire up a fractal, and try the <O>rbits command. Use the orbitdelay=<nnn> command (also on the <X> menu) to dramatically alter the effect, which ranges from an unearthly scream to a series of discrete tones. Not recommended when people you have to live with are nearby! Remember, we don't promise that it will sound beautiful!

You can also "hear" any image that Fractint can decode; turn on sound before using <R> to read in a GIF file. We have no idea if this feature is useful. It was inspired by the comments of an on-line friend who is blind. We solicit feedback and suggestions from anyone who finds these sound features interesting or useful. The orbitdelay command also affects the sound of decoding images.

HERTZ=nnn

Adjusts the sound produced by the "sound=x/y/z" option. Legal values are 200 through 10000.

1.17 Printer Parameters

General printer parameters are described below. Additional parameters for specific types of printers are described in:

PostScript Parameters

PaintJet Parameters

Plotter Parameters

```
PRINTER=type[/resolution[/port#]]
```

Defines your printer setup. The SSTOOLS.INI file is a REAL handy place to put this option, so that it's available whenever you have that sudden, irresistible urge for hard copy.

Printer types:

```
IB  IBM-compatible (default)
EP  Epson-compatible
HP  LaserJet
CO  Star Micronics Color printer, supposedly Epson-color-compatible
PA  Paintjet
PS  PostScript
PSL Postscript, landscape mode
PL  Plotter using HP-GL
```

Resolution:

```
In dots per inch.
Epson/IBM: 60, 120, 240
LaserJet: 75, 150, 300
PaintJet: 90, 180
PostScript: 10 through 600, or special value 0 to print full page to
within about .4" of the edges (in portrait mode, width is full page and
height is adjusted to 3:4 aspect ratio)
Plotter: 1 to 10 for 1/Nth of page (e.g. 2 for 1/2 page)
```

Port:

```
1, 2, 3 for LPT1-3 via BIOS
11, 12, 13, 14 for COM1-4 via BIOS
21, 22 for LPT1 or LPT2 using direct port access (faster when it works)
31, 32 for COM1 or COM2 using direct port access
```

```
COMPORT=port/baud/options
```

Serial printer port initialization.

```
Port=1,2,3,etc.
```

```
Baud=115,150,300,600,1200,2400,4800,9600
```

```
Options: 7,8 | 1,2 | e,n,o (any order).
```

Example: comport=1/9600/n81 for COM1 set to 9600, no parity, 8 bits per character, 1 stop bit.

```
LINEFEED=crlf|lf|cr
```

Specifies the control characters to emit at end of each line: carriage return and linefeed, just linefeed, or just carriage return. The default is crlf.

```
TITLE=yes
```

If specified, title information is added to printouts.

```
PRINTFILE=filename
```

Causes output data for the printer to be written to the named file on disk instead of to a printer port. The filename is incremented by 1 each time an image is printed - e.g. if the name is FRAC01.PRN, the second print operation writes to FRAC02.PRN, etc. Existing files are not overwritten - if the file exists, the filename is incremented to a new name.

1.18 postscript parameters

EPSF=1|2|3

Forces print-to-file and PostScript. If PRINTFILE is not specified, the default filename is FRACT001.EPS. The number determines how 'well-behaved' a .EPS file is. 1 means by-the-book. 2 allows some EPS 'no-nos' like settransfer and setscreen - BUT includes code that should make the code still work without affecting the rest of the non-EPS document. 3 is a free-for-all.

COLORPS=YES|NO - Enable or disable the color extensions.

RLEPS=YES|NO

Enable or disable run length encoding of the PostScript file. Run length encoding will make the PostScript file much smaller, but it may take longer to print. The run length encoding code is based on pnmtops, which is copyright (C) 1989 by Jef Poskanzer, and carries the following notice: "Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation. This software is provided "as is" without express or implied warranty."

TRANSLATE=yes|-n|n

Translate=yes prints the negative image of the fractal.

Translate=n reduces the image to that many colors. A negative value causes a color reduction as well as a negative image.

HALFTONE=frq/ang/sty[/f/a/s/f/a/s/f/a/s]

Tells the PostScript printer how to define its halftone screen. The first value, frequency, defines the number of halftone lines per inch. The second chooses the angle (in degrees) that the screen lies at. The third option chooses the halftone 'spot' style. Good default frequencies are between 60 and 80; Good default angles are 45 and 0; the default style is 0. If the halftone= option is not specified, Fractint will print using the printer's default halftone screen, which should have been already set to do a fine job on the printer.

These are the only three used when colorps=no. When color PS printing is being used, the other nine options specify the red, green, then blue screens. A negative number in any of these places will cause it to use the previous (or default) value for that parameter. NOTE: Especially when using color, the built-in screens in the printer's ROM may be the best choice for printing.

The default values are as follows: halftone=45/45/1/45/75/1/45/15/1/45/0/1 and these will be used if Fractint's halftone is chosen over the printer's built-in screen.

Current halftone styles:

- 0 Dot
 - 1 Dot (Smoother)
 - 2 Dot (Inverted)
 - 3 Ring (Black)
 - 4 Ring (White)
-

- 5 Triangle (Right)
- 6 Triangle (Isosceles)
- 7 Grid
- 8 Diamond
- 9 Line
- 10 Microwaves
- 11 Ellipse
- 12 Rounded Box
- 13 Custom
- 14 Star
- 15 Random
- 16 Line (slightly different)

A note on device-resolution black and white printing

This mode of printing can now be done much more quickly, and takes a lot less file space. Just set `EPSF=0 PRINTER=PSx/nnn COLORPS=NO RLEPS=YES TRANSLATE=m`, where `x` is `P` or `L` for portrait/landscape, `nnn` is your printer's resolution, `m` is `2` or `-2` for positive or negative printing respectively. This combination of parameters will print exactly one printer pixel per each image pixel and it will keep the proportions of the picture, if both your screen and printer have square pixels (or the same pixel-aspect). Choose a proper (read large) window size to fill as much of the paper as possible for the most spectacular results. 2048 by 2048 is barely enough to fill the width of a letter size page with 300 dpi printer resolution. For higher resolution printers, you will wish fractint supported larger window sizes (hint, hint...). Bug reports and/or suggestions should be forwarded to Yavuz Onder (post to sci.fractals, no e-mail yet).

A word from the author (Scott Taylor)

Color PostScript printing is new to me. I don't even have a color printer to test it on. (Don't want money. Want a Color PostScript printer!) The initial tests seem to have worked. I am still testing and don't know whether or not some sort of gamma correction will be needed. I'll have to wait and see about that one.

1.19 paintjet parameters

Note that the pixels printed by the PaintJet are square. Thus, a printout of an image created in a video mode with a 4:3 pixel ratio (such as 640x480 or 800x600) will come out matching the screen; other modes (such as 320x200) will come out stretched.

Black and white images, or images using the 8 high resolution PaintJet colors, come out very nicely. Some images using the full spectrum of PaintJet colors are very nice, some are disappointing.

When 180 dots per inch is selected (in `PRINTER=` command), high resolution 8 color printing is done. When 90 dpi is selected, low resolution printing using the full 330 dithered color palette is done. In both

cases, Fractint starts by finding the nearest color supported by the PaintJet for each color in your image. The translation is then displayed (unless the current display mode is disk video). This display *should* be a fairly good match to what will be printed - it won't be perfect most of the time but should give some idea of how the output will look. At this point you can <Enter> to go ahead and print, <Esc> to cancel, or <k> to cancel and keep the adjusted colors.

Note that you can use the color map PAINTJET.MAP to create images which use the 8 high resolution colors available on the PaintJet. Also, two high-resolution disk video modes are available for creating full page images.

If you find that the preview image seems very wrong (doesn't match what actually gets printed) or think that Fractint could be doing a better job of picking PaintJet colors to match your image's colors, you can try playing with the following parameter. Fair warning: this is a very tricky business and you may find it a very frustrating business trying to get it right.

HALFTONE=r/g/b

(The parameter name is not appropriate - we appropriated a PostScript parameter for double duty here.)

This separately sets the "gamma" adjustment for each of the red, green, and blue color components. Think of "gamma" as being like the contrast adjustment on your screen. Higher gamma values for all three components results in colors with more contrast being produced on the printer. Since each color component can have its gamma separately adjusted, you can change the resulting color mix subtly (or drastically!) Each gamma value entered has one implied decimal digit. The default is "halftone=21/19/16", for red 2.1, green 1.9, and blue 1.6. (A note from Pieter Branderhorst: I wrote this stuff to come out reasonably on my monitor/printer. I'm a bit suspicious of the guns on my monitor; if the colors seem ridiculously wrong on your system you might start by trying halftone=17/17/17.)

1.20 plotter parameters

Plotters which understand HP-GL commands are supported. To use a plotter, draw a SMALL image (32x20 or 64x40) using the <v>iew screen options. Put a red pen in the first holder in the plotter, green in the second, blue in the third. Now press <P> to start plotting. Now get a cup of coffee... or two... or three. It'll take a while to plot. Experiment with different resolutions, plot areas, plotstyles, and even change pens to create weird-colored images.

PLOTSTYLE=0|1|2

0: 3 parallel lines (red/green/blue) are drawn for each pixel, arranged like "///". Each bar is scaled according to the intensity of the corresponding color in the pixel. Using different pen colors (e.g. blue, green, violet) can come out nicely. The trick is to not tell anyone what color the bars are supposed to represent and they will accept these plotted colors because they do look nice...

1: Same as 0, but the lines are also twisted. This removes some of the

'order' of the image which is a nice effect. It also leaves more whitespace making the image much lighter, but colors such as yellow are actually visible.

2: Color lines are at the same angle and overlap each other. This type has the most whitespace. Quality improves as you increase the number of pixels squeezed into the same size on the plotter.

1.21 3d parameters

To stay out of trouble, specify all the 3D parameters, even if you want to use what you think are the default values. It takes a little practice to learn what the default values really are. The best way to create a set of parameters is to use the command on an image you like and then use an editor to modify the resulting parameter file.

3D=Yes

3D=Overlay

Resets all 3d parameters to default values. If FILENAME= is given, forces a restore to be performed in 3D mode (handy when used with 'batch=yes' for batch-mode 3D images). If specified, 3D=Yes should come before any other 3d parameters on the command line or in a parameter file entry. The form 3D=Overlay is identical except that the previous graphics screen is not cleared, as with the <#> (<shift-3> on some keyboards) overlay command. Useful for building parameter files that use the 3D overlay feature.

The options below override the 3D defaults:

PREVIEW=yes	Turns on 3D 'preview' default mode
SHOWBOX=yes	Turns on 3D 'showbox' default mode
COARSE=nn	Sets Preview 'coarseness' default value
SPHERE=yes	Turns on spherical projection mode
STEREO=n	Selects the type of stereo image creation
RAY=nnn	selects raytrace output file format
BRIEF=yes	selects brief or verbose file for DKB output
INTEROCULAR=nn	Sets the interocular distance for stereo
CONVERGE=nn	Determines the overall image separation
CROP=nn/nn/nn/nn	Trims the edges off stereo pairs
BRIGHT=nn/nn	Compensates funny glasses filter parameters
LONGITUDE=nn/nn	Longitude minimum and maximum
LATITUDE=nn/nn	Latitude minimum and maximum
RADIUS=nn	Radius scale factor
ROTATION=nn[/nn[/nn]]	Rotation about x,y, and z axes
SCALEXYZ=nn/nn/nn	X,y, and z scale factors
ROUGHNESS=nn	Same as z scale factor
WATERLINE=nn	Colors nn and below will be "inside" color
FILLTYPE=nn	3D filltype
PERSPECTIVE=nn	Perspective distance
XYSHIFT=nn/nn	Shift image in x and y directions with perspective
LIGHTSOURCE=nn/nn/nn	Coordinates for light-source vector
SMOOTHING=nn	Smooths images in light-source fill modes
TRANSPARENT=min/max	Defines a range of colors to be treated as "transparent" when <#>Overlaying 3D images.
XYADJUST=nn/nn	This shifts the image in the x/y dir without

perspective

Below are new commands as of version 14 that support Marc Reinig's terrain features.

RANDOMIZE=nnn (0 - 100)

This feature randomly varies the color of a pixel to near by colors. Useful to minimize map banding in 3d transformations. Usable with all FILLTYPES. 0 disables, max values is 7. Try 3 - 5.

AMBIENT=nnn (0 - 100)

Set the depth of the shadows when using full color and light source filltypes. "0" disables the function, higher values lower the contrast.

FULLCOLOR=yes

Valid with any light source FILLTYPE. Allows you to create a Targa-24 file which uses the color of the image being transformed or the map you select and shades it as you would see it in real life. Well, its better than B&W. A good map file to use is topo

HAZE=nnn (0 - 100)

Gives more realistic terrains by setting the amount of haze for distant objects when using full color in light source FILLTYPES. Works only in the "y" direction currently, so don't use it with much y rotation. Try "rotation=85/0/0". 0 disables.

LIGHTNAME=<filename>

The name of the Targa-24 file to be created when using full color with light source. Default is light001.tga. If overwrite=no (the default), the file name will be incremented until an unused filename is found. Background in this file will be sky blue.

1.22 Batch Mode

It IS possible, believe it or not, to become so jaded with the screen drawing process, so familiar with the types and options, that you just want to hit a key and do something else until the final images are safe on disk. To do this, start Fractint with the BATCH=yes parameter. To set up a batch run with the parameters required for a particular image you might:

- o Find an interesting area. Note the parameters from the <Tab> display. Then use an editor to write a batch file.
- o Find an interesting area. Set all the options you'll want in the batch run. Use the command to store the parameters in a file. Then use an editor to add the additional required batch mode parameters (such as VIDEO=) to the generated parameter file entry. Then run the batch using "fractint @myname.par/myentry" (if you told the command to use file "myname" and to name the entry "myentry").

Another approach to batch mode calculations, using "FILENAME=" and resume, is described later.

When modifying a parameter file entry generated by the command, the only parameters you must add for a batch mode run are "BATCH=yes", and "VIDEO=xxx" to select a video mode. You might want to also add "SAVENAME=[name]" to name the result as something other than the default

FRACT001.GIF. Or, you might find it easier to leave the generated parameter file unchanged and add these parameters by using a command like:

```
fractint @myname.par/myentry batch=y video=AF3 savename=mygif
```

"BATCH=yes" tells Fractint to run in batch mode -- that is, Fractint draws the image using whatever other parameters you specified, then acts as if you had hit <S> to save the image, then exits to DOS.

"FILENAME=" can be used with "BATCH=yes" to resume calculation of an incomplete image. For instance, you might interactively find an image you like; then select some slow options (a high resolution disk video mode, distance estimator method, high maxiter, or whatever); start the calculation; then interrupt immediately with a <S>ave. Rename the save file (fract001.gif if it is the first in the session and you didn't name it with the <X> options or "savename=") to xxx.gif. Later you can run Fractint in batch mode to finish the job:

```
fractint batch=yes filename=xxx savename=xxx
```

"SAVETIME=nnn" is useful with long batch calculations, to store a checkpoint every nnn minutes. If you start a many hour calculation with say "savetime=60", and a power failure occurs during the calculation, you'll have lost at most an hour of work on the image. You can resume calculation from the save file as above. Automatic saves triggered by SAVETIME do not increment the save file name. The same file is overwritten by each auto save until the image completes. But note that Fractint does not directly over-write save files. Instead, each save operation writes a temporary file FRACTINT.TMP, then deletes the prior save file, then renames FRACTINT.TMP to be the new save file. This protects against power failures which occur during a save operation - if such a power failure occurs, the prior save file is intact and there's a harmless incomplete FRACTINT.TMP on your disk.

If you want to spread a many-hour image over multiple bits of free machine time you could use a command like:

```
fractint batch=yes filename=xxx savename=xxx savetime=60 video=F3
```

While this batch is running, hit <S> (almost any key actually) to tell fractint to save what it has done so far and give your machine back. A status code of 2 is returned by fractint to the batch file. Kick off the batch again when you have another time slice for it.

While running a batch file, pressing any key will cause Fractint to exit with an errorlevel = 2. Any error that interrupts an image save to disk will cause an exit with errorlevel = 2. Any error that prevents an image from being generated will cause an exit with errorlevel = 1.

The SAVETIME= parameter, and batch resumes of partial calculations, only work with fractal types which can be resumed. See Interrupting and Resuming for information about non-resumable types.

1.23 Video Adapter Notes

True to the spirit of public-domain programming, Fractint makes only a limited attempt to verify that your video adapter can run in the mode you specify, or even that an adapter is present, before writing to it. ↵

So if you use the "video=" command line parameter, check it before using a new version of Fractint - the old key combo may now call an ultraviolet holographic mode.

Comments about some particular video adapters:

EGA

Tweaked VGA

Super-VGA
8514/A

XGA

Targa

Targa+

Also see

Customized Video Modes, FRACTINT.CFG

.

1.24 ega

EGA

Fractint assumes that every EGA adapter has a full 256K of memory (and can therefore display 640 x 350 x 16 colors), but does nothing to verify that fact before slinging pixels.

1.25 tweaked vga

"TWEAKED" VGA MODES

The IBM VGA adapter is a highly programmable device, and can be set up to display many video-mode combinations beyond those "officially" supported by the IBM BIOS. E.g. 320x400x256 and 360x480x256 (the latter is one of our favorites).

These video modes are perfectly legal, but temporarily reprogram the adapter (IBM or fully register-compatible) in a non-standard manner that the BIOS does not recognize.

Fractint also contains code that sets up the IBM (or any truly register-compatible) VGA adapter for several extended modes such as 704x528, 736x552, 768x576, and 800x600. It does this by programming the VGA controller to use the fastest dot-clock on the IBM adapter (28.322 MHz), throwing more pixels, and reducing the refresh rate to make up for it.

These modes push many monitors beyond their rated specs, in terms of both resolution and refresh rate. Signs that your monitor is having problems with a particular "tweaked" mode include:

- o vertical or horizontal overscan (displaying dots beyond the edges of your visible CRT area)

- o flickering (caused by a too-slow refresh rate)
- o vertical roll or total garbage on the screen (your monitor simply can't keep up, or is attempting to "force" the image into a pre-set mode that doesn't fit).

We have successfully tested the modes up to 768x576 on an IBM PS/2 Model 80 connected to IBM 8513, IBM 8514, NEC Multisync II, and Zenith 1490 monitors (all of which exhibit some overscan and flicker at the highest rates), and have tested 800x600 mode on the NEC Multisync II (although it took some twiddling of the vertical-size control).

1.26 super-vga

SUPER-EGA AND SUPER-VGA MODES

Since version 12.0, we've used both John Bridges' SuperVGA Autodetecting logic *and* VESA adapter detection, so that many brand-specific SuperVGA modes have been combined into single video mode selection entries. There is now exactly one entry for SuperVGA 640x480x256 mode, for instance.

If Fractint's automatic SuperVGA/VESA detection logic guesses wrong, and you know which SuperVGA chipset your video adapter uses, you can use the "adapter=" command-line option to force Fractint to assume the presence of a specific SuperVGA Chipset - see
Video Parameters
for details.

1.27 8514/a

8514/A MODES

The IBM 8514/A modes (640x480 and 1024x768) default to using the hardware registers. If an error occurs when trying to open the adapter, an attempt will be made to use IBM's software interface, and requires the preloading of IBM's HDILOAD TSR utility.

The Adex 1280x1024 modes were written for and tested on an Adex Corporation 8514/A using a Brooktree DAC. The ATI GU 800x600x256 and 1280x1024x16 modes require a ROM bios version of 1.3 or higher for 800x600 and 1.4 or higher for 1280x1024.

There are two sets of 8514/A modes: full sets (640x480, 800x600, 1024x768, 1280x1024) which cover the entire screen and do NOT have a border color (so that you cannot tell when you are "paused" in a color-cycling mode), and partial sets (632x474, 792x594, 1016x762, 1272x1018) with small border areas which do turn white when you are paused in color-cycling mode. Also, while these modes are declared to be 256-color, if you do not have your 8514/A adapter loaded with its full complement of memory you will actually be in 16-color mode. The hardware register 16-color modes have not been tested.

If your 8514/A adapter is not truly register compatible and Fractint does

not detect this, use of the adapter interface can be forced by using `afi=y` or `afi=8514` in your `SSTOOLS.INI` file.

Finally, because IBM's adapter interface does not handle drawing single pixels very well (we have to draw a 1x1 pixel "box"), generating the zoom box when using the interface is excruciatingly slow. Still, it works!

1.28 xga

XGA MODES

The XGA adapter is supported using the VESA/SuperVGA Autodetect modes - the XGA looks like just another SuperVGA adapter to Fractint. The supported XGA modes are 640x480x256, 1024x768x16, 1024x768x256, 800x600x16, and 800x600x256. Note that the 1024x768x256 mode requires a full 1MB of adapter memory, the 1024x768 modes require a high-rez monitor, and the 800x600 modes require a multisyncing monitor such as the NEC 2A.

1.29 targa

TARGA MODES

TARGA support for Fractint is provided courtesy of Joe McLain and has been enhanced with the help of Bruce Goren and Richard Biddle. To use a TARGA board with Fractint, you must define two DOS environment variables, "TARGA" and "TARGASET". The definition of these variables is standardized by Truevision; if you have a TARGA board you probably already have added "SET" statements for these variables to your `AUTOEXEC.BAT` file.

Be aware that there are a LOT of possible TARGA configurations, and a LOT of opportunities for a TARGA board and a VGA or EGA board to interfere with each other, and we may not have all of them smoothed away yet. Also, the TARGA boards have an entirely different color-map scheme than the VGA cards, and at the moment they cannot be run through the color-cycling menu. The "MAP=" argument (see

Color Parameters

),

however, works with both TARGA and VGA boards and enables you to redefine the default color maps with either board.

1.30 targa+

TARGA+ MODES

To use the special modes supported for TARGA+ adapters, the `TARGAP.SYS` device driver has to be loaded, and the `TPLUS.DAT` file (included with Fractint) must be in the same directory as Fractint. The video modes with names containing "True Color Autodetect" can be used with the Targa+. You might want to use the command line parameters `"tplus="`, `"noninterlaced="`,

"maxcolorres=", and "pixelzoom=" (see
Video Parameters
)

in your SSTOOLS.INI file to modify Fractint's use of the adapter.

1.31 Disk-Video

These "video modes" do not involve a video adapter at all. They use (in order or preference) your expanded memory, your extended memory, or your disk drive (as file FRACTINT.\$\$\$) to store the fractal image. These modes are useful for creating images beyond the capacity of your video adapter right up to the current internal limit of 2048 x 2048 x 256, e.g. for subsequent printing.

They're also useful for

background processing under multi-tasking DOS managers - create an image in a disk-video mode, save it, then restore it in a real video mode.

While you are using a disk-video mode, your screen will display text information indicating whether memory or your disk drive is being used, and what portion of the "screen" is being read from or written to. A "Cache size" figure is also displayed. 64K is the maximum cache size. If you see a number less than this, it means that you don't have a lot of memory free, and that performance will be less than optimum. With a very low cache size such as 4 or 6k, performance gets considerably worse in cases using solid guessing, boundary tracing, plasma, or anything else which paints the screen non-linearly. If you have this problem, all we can suggest is having fewer TSR utilities loaded before starting Fractint, or changing in your config.sys file, such as reducing a very high BUFFERS value.

The zoom box is disabled during disk-video modes (you couldn't see where it is anyway). So is the orbit display feature.

Color Cycling can be used during disk-video modes, but only to load or save a color palette.

When using real disk for your disk-video, Fractint will not generate some "attractor" types (e.g. lorenz) nor "IFS" images. These would kill your disk drive. Boundary tracing is allowed - it may give your drive a bit of a workout, but is generally tolerable.

When using a real disk, and you are not directing the file to a RAM disk, and you aren't using a disk caching program on your machine, specifying BUFFERS=10 (or more) in your config.sys file is best for performance. BUFFERS=10,2 or even BUFFERS=10,4 is also good. It is also best to keep your disk relatively "compressed" (or "defragmented") if you have a utility to do this.

In order to use extended memory, you must have HIMEM.SYS or an equivalent that supports the XMS 2.0 standard or higher. Also, you can't have a VDISK installed in extended memory. Himem.sys is distributed with Microsoft Windows 286/386 and 3.0. If you have problems using the extended memory, try rebooting with just himem.sys loaded and see if that clears up the problem.

If you are running background disk-video fractals under Windows 3, and you don't have a lot of real memory (over 2Mb), you might find it best to force Fractint to use real disk for disk-video modes. (Force this by using a .pif file with extended memory and expanded memory set to zero.) Try this if your disk goes crazy when generating background images, which are supposedly using extended or expanded memory. This problem can occur because, to multi-task, sometimes Windows must page an application's expanded or extended memory to disk, in big chunks. Fractint's own cached disk access may be faster in such cases.

1.32 Customized Video Modes, FRACTINT.CFG

If you have a favorite adapter/video mode that you would like to add to Fractint... if you want some new sizes of disk-video modes... if you want to remove table entries that do not apply to your system... if you want to specify different "textsafe=" options for different video modes... relief is here, and without even learning "C"!

You can do these things by modifying the FRACTINT.CFG file with your text editor. Saving a backup copy of FRACTINT.CFG first is of course highly recommended!

Fractint uses a video adapter table for most of what it needs to know about any particular adapter/mode combination. The table is loaded from FRACTINT.CFG each time Fractint is run. It can contain information for up to 300 adapter/mode combinations. The table entries, and the function keys they are tied to, are displayed in the "select video mode" screen.

This table makes adding support for various third-party video cards and their modes much easier, at least for the ones that pretend to be standard with extra dots and/or colors. There is even a special "roll-your-own" video mode (mode 19) enabling those of you with "C" compilers and a copy of the Fractint source to generate video modes supporting whatever adapter you may have.

The table as currently distributed begins with nine standard and several non-standard IBM video modes that have been exercised successfully with a PS/2 model 80. These entries, coupled with the descriptive comments in the table definition and the information supplied (or that should have been supplied!) with your video adapter, should be all you need to add your own entries.

After the IBM and quasi-pseudo-demi-IBM modes, the table contains an ever-increasing number of entries for other adapters. Almost all of these entries have been added because someone like you sent us spec sheets, or modified Fractint to support them and then informed us about it.

Lines in FRACTINT.CFG which begin with a semi-colon are treated as comments. The rest of the lines must have eleven fields separated by commas. The fields are defined as:

1. Key assignment. F2 to F10, SF1 to SF10, CF1 to CF10, or AF1 to AF10. Blank if no key is assigned to the mode.
2. The name of the adapter/video mode (25 chars max, no leading blanks). The adapter is set up for that mode via INT 10H, with:

3. AX = this,
4. BX = this,
5. CX = this, and
6. DX = this (hey, having all these registers wasn't OUR idea!)
7. An encoded value describing how to write to your video memory in that mode. Currently available codes are:
 - 1) Use the BIOS (INT 10H, AH=12/13, AL=color) (last resort - SLOW!)
 - 2) Pretend it's a (perhaps super-res) EGA/VGA
 - 3) Pretend it's an MCGA
 - 4) SuperVGA 256-Color mode using the Tseng Labs chipset
 - 5) SuperVGA 256-Color mode using the Paradise chipset
 - 6) SuperVGA 256-Color mode using the Video-7 chipset
 - 7) Non-Standard IBM VGA 360 x 480 x 256-Color mode
 - 8) SuperVGA 1024x768x16 mode for the Everex chipset
 - 9) TARGA video modes
 - 10) HERCULES video mode
 - 11) Non-Video, i.e. "disk-video"
 - 12) 8514/A video modes
 - 13) CGA 320x200x4-color and 640x200x2-color modes
 - 14) Reserved for Tandy 1000 video modes
 - 15) SuperVGA 256-Color mode using the Trident chipset
 - 16) SuperVGA 256-Color mode using the Chips & Tech chipset
 - 17) SuperVGA 256-Color mode using the ATI VGA Wonder chipset
 - 18) SuperVGA 256-Color mode using the EVEREX chipset
 - 19) Roll-your-own video mode (as you've defined it in YOURVID.C)
 - 20) SuperVGA 1024x768x16 mode for the ATI VGA Wonder chipset
 - 21) SuperVGA 1024x768x16 mode for the Tseng Labs chipset
 - 22) SuperVGA 1024x768x16 mode for the Trident chipset
 - 23) SuperVGA 1024x768x16 mode for the Video 7 chipset
 - 24) SuperVGA 1024x768x16 mode for the Paradise chipset
 - 25) SuperVGA 1024x768x16 mode for the Chips & Tech chipset
 - 26) SuperVGA 1024x768x16 mode for the Everex Chipset
 - 27) SuperVGA Auto-Detect mode (we poke around looking for your adapter)
 - 28) VESA modes
 - 29) True Color Auto-Detect (currently only Targa+ supported)
- Add 100, 200, 300, or 400 to this code to specify an over-ride "textsafe=" option to be used with the mode. 100=yes, 200=no, 300=bios, 400=save. E.g. 428 for a VESA mode with textsafe=save forced.
8. The number of pixels across the screen (X - 160 to 2048)
9. The number of pixels down the screen (Y - 160 to 2048)
10. The number of available colors (2, 4, 16, or 256)
11. A comment describing the mode (25 chars max, leading blanks are OK)

NOTE that the AX, BX, CX, and DX fields use hexadecimal notation (fifteen ==> 'f', sixteen ==> '10'), because that's the way most adapter documentation describes it. The other fields use standard decimal notation.

If you look closely at the default entries, you will notice that the IBM VGA entries labeled "tweaked" and "non standard" have entries in the table with AX = BX = CX = 0, and DX = some other number. Those are special flags that we used to tell the program to custom-program the VGA adapter, and are NOT undocumented BIOS calls. Maybe they should be, but they aren't.

If you have a fancy adapter and a new video mode that works on it, and it is not currently supported, PLEASE GET THAT INFORMATION TO US! We will add the video mode to the list on our next release, and give you credit

for it. Which brings up another point: If you can confirm that a particular video adapter/mode works (or that it doesn't), and the program says it is UNTESTED, please get that information to us also. Thanks in advance!
