Xfig User Manual

Version 3.2.4

December 2002

Table of Contents

---

written by us

Xfig User Manual

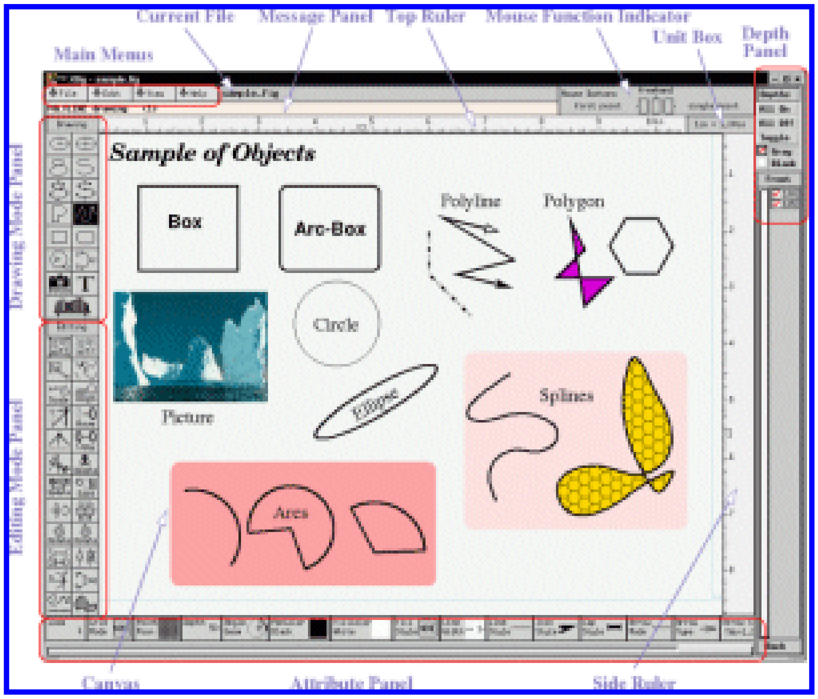
Version 3.2.4 December 2002

Introduction

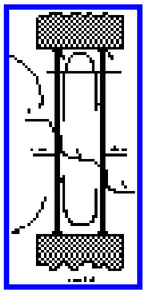
Xfig is an interactive drawing tool which runs under X Window System Version 11 Release 4 (X11R4) or later, on most UNIX-compatible platforms. It is freeware, and available via anonymous ftp.

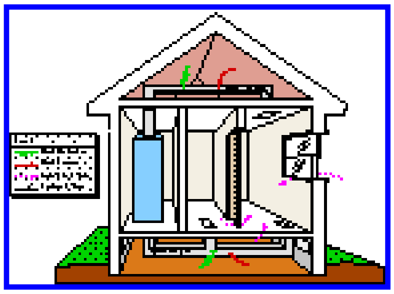
In **xfig**, figures may be drawn using objects such as circles, boxes, lines, spline curves, text, etc. It is also possible to import images in formats such as GIF, JPEG, EPSF (PostScript), etc. Those objects can be created, deleted, moved or modified. Attributes such as colors or line styles can be selected in various ways. For text, 35 fonts are available. Text can also include Latin-1 characters such as ``ä" or ``ç".

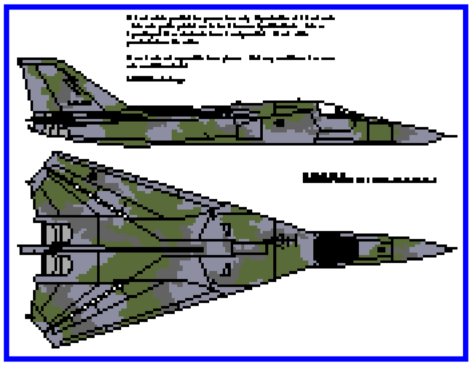
Here is a screen image of **xfig**. Click on the image below for larger version (96k).



And here are some example figures extracted from the **xfig** distribution. Click on them to see a larger version.







Xfig saves figures in its native *Fig format*, but they may be converted into various formats such as PostScript, GIF, JPEG, HP-GL, etc. **xfig** has facility to print figures to a PostScript printer, too.

There are some applications which can produce output in the *Fig format*. For example, **xfig** doesn't have a facility to create graphs, but tools such as **gnuplot** or **xgraph** can create graphs and export them in *Fig format*. Even if your favorite application can't generate output for **xfig**, tools such as **pstoedit** or **hp2xx** may allow you to read and edit those figures with **xfig**. If you want to import images into the figure but you don't need to edit the image itself (like this example), it is also possible to import images in formats such as GIF, JPEG, EPSF (PostScript), etc.

Most operation are performed using the mouse, but some operations may also be performed using keyboard accelerators (shortcuts). Use of a three-button mouse is recommended, but it is also possible to use a two-button mouse (if you have a two-button mouse and your X server doesn't emulate a three-button mouse, press the Meta (or Alt) key and right mouse button together to simulate mouse button 2). Normally, mouse buttons 1 to 3 are assigned to the left, middle, and right buttons respectively.

xfig 3.2.X and **fig2dev 3.2.X** include code for internationalization, it allows use of local characters (e.g., Japanese) in **xfig**. It uses standard internationalization (I18N) mechanism of X11R5, and known to work for Japanese and Korean at this time. See Internationalization about this.

Starting xfig

xfig is started by the `xfig` command.

```
    xfig  [ options... ] [ filename ]
```

options are command line options which may be used to customize **xfig**. It is also possible to use X resources instead of specifying command line options each time when starting `xfig`.

If *filename* is given, the file will be loaded when **xfig** is started.

Display Windows

The following components comprise the **xfig** window:

- Main Menus:
  Has buttons for global operations; such as load/save file, print or export figures, quit **xfig**, etc.
- Drawing Mode Panel:
  Has buttons for drawing operations; such as circle, box, polyline, text, etc.
- Editing Mode Panel:
  Has buttons for editing operations; such as move, copy, delete, scale, edit attributes, etc.
- Attribute Panel:
  Has buttons to set attributes of objects; such as color, line width, line style, text font, text justification, etc. There are also buttons for global settings such as zoom scale or grid mode.
- Mouse Function Indicator:
  The function of each mouse button is displayed here. This changes with the mode of the operation (drawing, editing, etc.) to reflect the function of each mouse button.
- Rulers:
  Graduations in the selected units (e.g. inches or cm) are displayed on top (horizontal) and side (vertical) ruler. The rulers are also used for scrolling the canvas.
- Depth Panel:
  This panel shows the depths of all objects on the canvas. The user may hide or show any depth by clicking on the checkboxes next to the depth number.
- Units:
  The scale of the drawing is displayed here, e.g. 1cm = 3m. It is also used to change the units and/or scale.
- Message Panel:

Various messages are displayed here. For example, the size of objects will be displayed here when entering objects.

- Canvas:

  Area to draw figures. The canvas may be scrolled with the top and side rulers or the arrow keys on the keyboard.

[Contents | Introduction | Credits]

Xfig User Manual

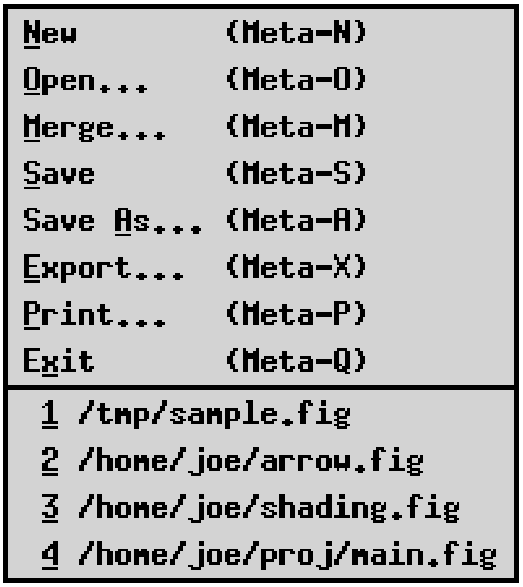
Version 3.2.4 December 2002

Main Menus

The following menus are at the top of the **xfig** window.

- File: File menu
- Edit: Edit menu
- View: View menu
- Help: Help menu

File menu



- **New**

  Delete all objects from the canvas and erase the current file name to make a new figure. The accelerator `Meta-N` will also perform this function. This operation may be undone by Undo.
- **Open...**

  Popup panel to open a Fig file. See File Panels for details.
- **Merge...**

  Popup panel to merge one or more Fig files with the current figure. See File Panels for details.
- **Save**

  Save current figure in current filename.
- **Save As...**

  Popup panel to save current figure in new filename. See File Panels for details.
- **Export...**

  Popup panel to export current figure to various formats such as PostScript, GIF, etc. See Exporting for details.
- **Print...**

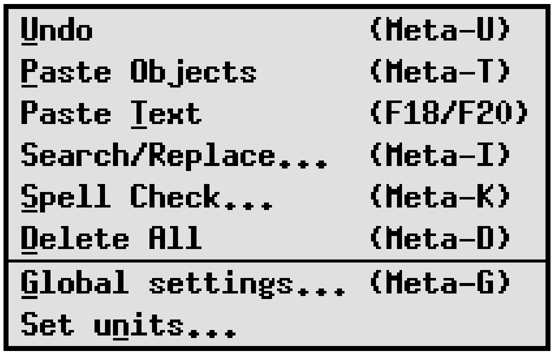
  Popup panel to print current figure to PostScript printer. See Printing for details.
- **Exit**

  Exit from **xfig**. The accelerator `Meta-Q` will also perform this function.

If the figure has been modified and not saved, a popup panel will appear to ask the user if he wants to save the figure first and then quit, quit without saving, or cancel the quit altogether.
If **xfig** is terminated with signals such as *HUP* or *INT* (for example, by the `kill` command or an internal error of **xfig**), **xfig** will try to save the figure in the **SAVE.fig** file.

- Recent file list (**1 2 3 4**)
  Below the **Exit** menu entry is a list of recently loaded **Fig** files. The number of files in this list may be set in the **.xfigrc** file in the user's login directory, or from the Global Settings panel. The maximum number of file names saved is 9.

Edit Menu



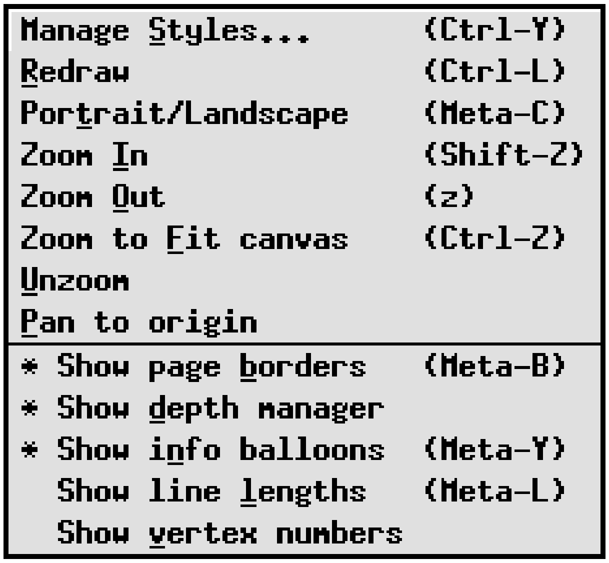
- **Undo**
  Undo the last operation such as object creation, deletion or modification. The accelerator `Meta-U` will also perform this function.
  Multi-level undo is not supported. If *Undo* is clicked twice, it will revert the state before the first *undo*.
- **Paste Objects**
  Paste the **Fig** object previously copied into the **xfig** *cut buffer* (see also COPY and DELETE) into the current figure. The accelerator `Meta-T` will also perform this function.
  The object will appear on the canvas under or near the mouse where it may then be moved and placed by pressing mouse button 1 (`place object'). If you want to place it where it originally came from press mouse button 2 (`place at orig posn').
  This function can be used to copy part of another figure into the figure being edited. The *cut buffer* can be shared between **xfig**s if a user runs two or more **xfig** programs at the same time, and it is possible to copy objects between those **xfig** programs. If you want to copy an entire figure from another file, you can merge the file using File panel.
  Normally, the ``.xfig'' file in user's home directory is used as the *cut buffer*.
- **Paste Text**
  Paste text from the **X11** cut buffer onto the canvas where a text object has been started. Note that you must already have started a text object by clicking the Text mode and clicking on the canvas where you want the text pasted. The accelerators `F18` and `F20` will also perform this function.
- **Search/Replace...**
  Popup panel to search and/or replace strings in Fig text objects. The accelerator `Meta-I` will also perform this function. See Search and Replace Text for details.
- **Spell Check...**
  Popup panel to check Fig text objects for spelling errors. The accelerator `Meta-K` will also perform this function. See Spell Check for details.
- **Delete all**
  Delete all objects on the canvas. Can be undone with `Meta-U` (Undo).

- **Global settings...**

  Popup panel showing global settings such as the HTML browser, spelling checker, mouse tracking in rulers, etc. The accelerator `Meta-G` will also perform this function. See Global Settings for details.
- **Set units...**

  Popup the unit panel to change drawing/scaling units. Right-clicking on the **unit** box in the upper-right corner of **xfig** will also perform this operation.

View Menu


Manage Styles... (Ctrl-Y)
Redraw (Ctrl-L)
Portrait/Landscape (Meta-C)
Zoom In (Shift-Z)
Zoom Out (z)
Zoom to Fit canvas (Ctrl-Z)
Unzoom
Pan to origin
* Show page borders (Meta-B)
* Show depth manager
* Show info balloons (Meta-Y)
Show line lengths (Meta-L)
Show vertex numbers

- **Manage Styles...**

  Popup the Style Manager. These are user-defined settings which can be named and saved. For example you may have a group of settings called "special lines" which have predefined line styles and colors from which you can choose. See Manage Styles... for details.
- **Redraw**

  Redraw the canvas. The accelerators `Meta-R` and `Ctrl-L` will also perform this function.
- **Portrait/Landscape**

  Change the orientation of the **xfig** canvas between portrait and landscape. The orientation will also be used as the default orientation when Exporting or Printing a figure.

  The default orientation may be specified by command line options. The orientation is automatically changed when loading a fig file to correspond to the setting in the file.
- **Zoom in**

  Zoom in on the canvas by a factor of two. `Shift-Z` will also perform this function. See also Zooming.
- **Zoom out**

  Zoom in on the canvas by a factor of two. A lowercase `z` will also perform this function. See also Zooming.
- **Zoom to Fit canvas**

  Zoom the figure so that it just fits in the canvas. `Ctrl-Z` will also perform this function. See also Zooming.
- **Pan to origin**

  Pan the canvas to the origin (0,0). See also Panning the Canvas.

For the following, an asterisk (*) in the menu indicates that the setting is turned on.

- **Show page borders**

  Turn on or off the drawing of the page borders on the canvas. If on, **xfig** draws a lightblue line (default) on the canvas where the page boundaries are and the name of the page size, for the currently selected page size The accelerator `Meta-B` will also perform this function. (see Exporting or Printing for page sizes supported).

- **Show depth manager**

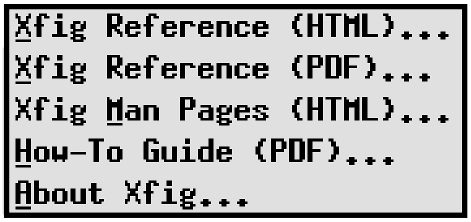
  Show or hide the depth manager panel to the right of the side ruler. See Layers (Depths) for details.
- **Show info balloons**

  Turn on or off displaying of information balloons. These balloons popup after a small delay when the mouse is moved over a button or other areas of **xfig** with information about that item. The accelerator `Meta-Y` will also perform this function.
  The popup delay may be set in the Global Settings panel.
- **Show line lengths**

  Turn on or off the display of line lengths as lines/circles/etc. are drawn. If on, **xfig** shows the lengths of lines as they are being drawn in red, near the lines themselves. The accelerator `Meta-L` will also perform this function.
- **Show vertex numbers**

  Turn on or off labelling of vertices of Fig objects. If on, **xfig** will label each vertex of objects on the canvas with their drawing order. This is only useful for diagnostic purposes.

Help Menu



- **Xfig Reference (HTML)...**

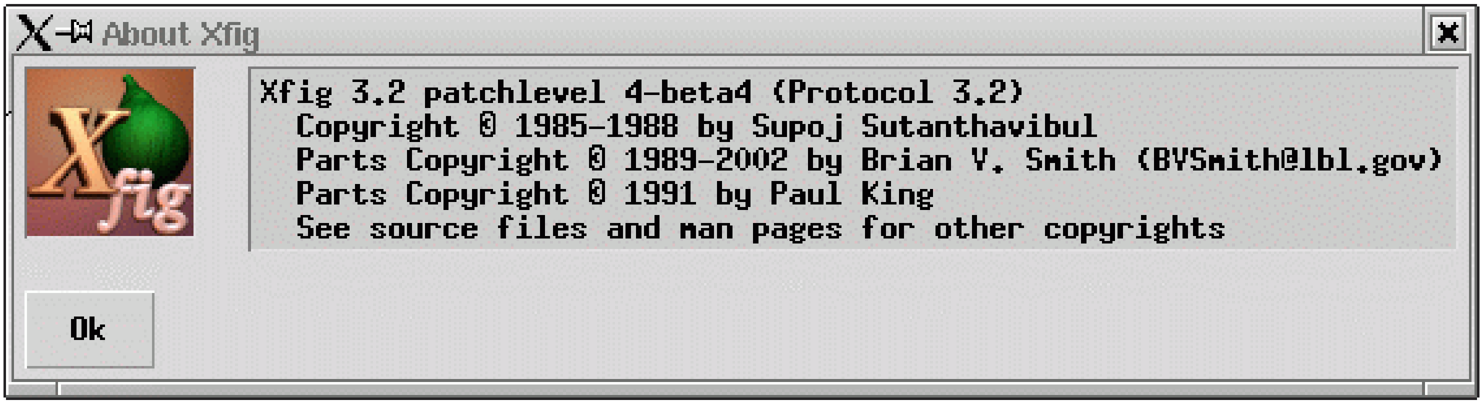
  This starts up your web browser (resource `Fig.browser`, default *netscape*) to view the main **xfig** reference manual.
- **Xfig Reference (PDF)...**

  These start up your PDF viewer (resource `Fig.pdfviewer`, to view the PDF version of the reference manual.
- **How-To Guide (PDF)...**

  This starts up your PDF viewer (resource `Fig.pdfviewer`, default *acroread*) to view the **xfig** tutorial, written by Peter Hiscocks.
- **Man pages (PDF)...**

  This starts up your PDF viewer (resource `Fig.pdfviewer`, default *acroread*) to view the **xfig** man pages.
- **About Xfig...**

  This pops up a window with the version and copyright information about **xfig**.



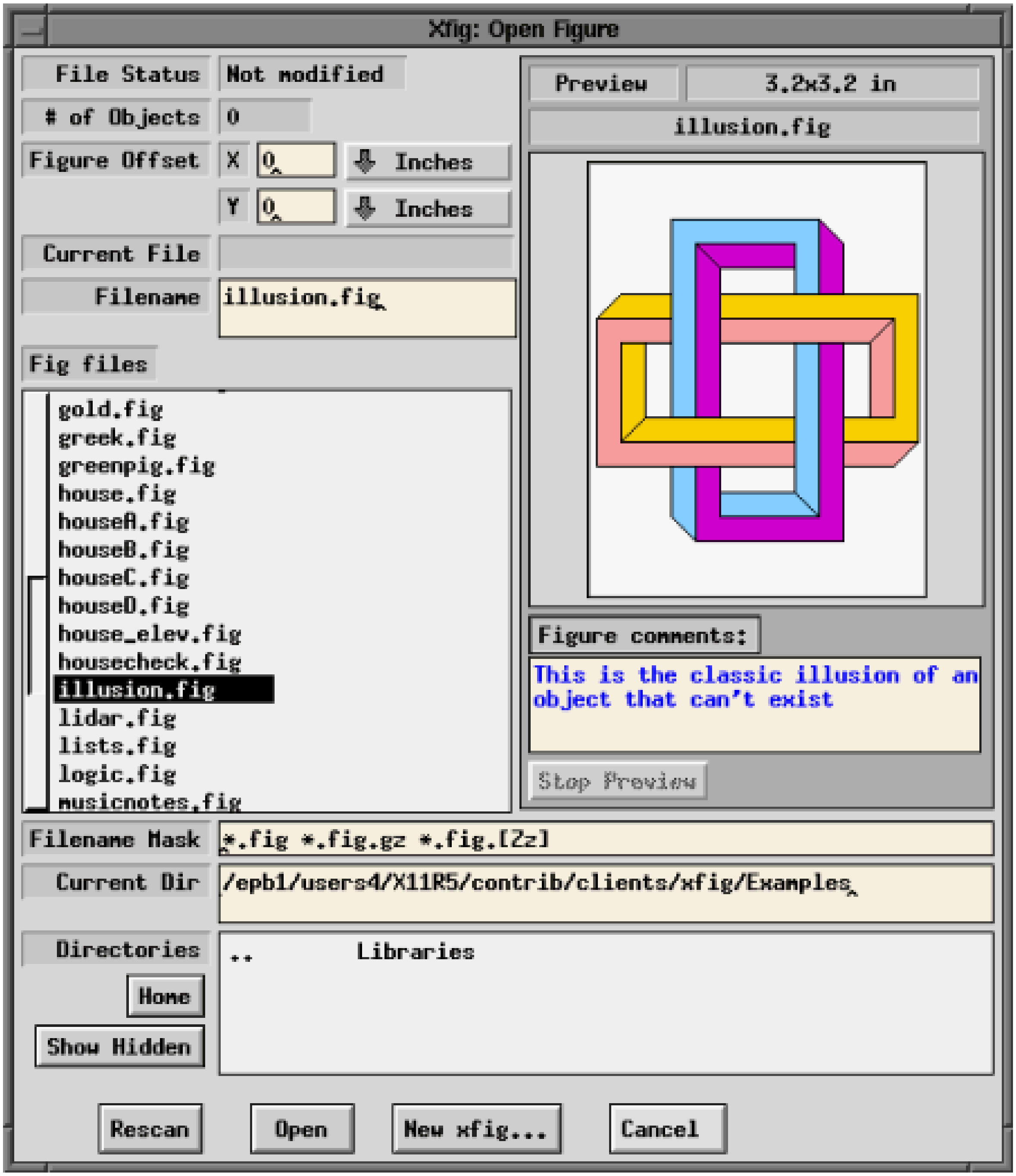
Xfig 3.2 patchlevel 4-beta4 (Protocol 3.2)
  Copyright © 1985-1988 by Supoj Sutanthavibul
  Parts Copyright © 1989-2002 by Brian V. Smith (BVSmith@lbl.gov)
  Parts Copyright © 1991 by Paul King
  See source files and man pages for other copyrights

File Panels

The file panels provide the facility to Load, SaveAs, or Merge Fig files. Use Exporting if you want to output figure with a format other than a **Fig** file, and use IMPORTING PICTURE OBJECTS if you want to insert image files into the figure.

- Choosing *Open...* from the *File* menu will pop up up the **Open Figure** panel. The accelerator `Meta-O` will also perform this function.
- Choosing *Merge...* from the *File* menu will pop up up the **Merge Figure** panel. The accelerator `Meta-M` will also perform this function.
- Choosing *Save As...* from the *File* menu will pop up up the **SaveAs** panel. The accelerator `Meta-A` will also perform this function.

The panels look almost the same except for the bottom row of buttons, which reflect the operations like "Open", "Merge" and "Save":



Xfig: Open Figure

File Status: Not modified

# of Objects: 0

Figure Offset: X 0 Inches; Y 0 Inches

Current File:

Filename: illusion.fig

Fig files:

gold.fig
greek.fig
greenpig.fig
house.fig
houseA.fig
houseB.fig
houseC.fig
houseD.fig
house_elev.fig
housecheck.fig
illusion.fig
lidar.fig
lists.fig
logic.fig
musicnotes.fig

Filename Mask: *.fig *.fig.gz *.fig.[Zz]

Current Dir: /epb1/users4/X11R5/contrib/clients/xfig/Examples

Directories: .. Libraries

Preview: 3.2x3.2 in

illusion.fig

Figure comments: This is the classic illusion of an object that can't exist

- *File Status*
  This shows whether the current figure has been modified or not.
- *# of Objects*
  This shows the number of Fig objects in the drawing. A compound object with several objects inside is considered one object.
- *Figure Offset*
  Specify amounts of offset the figure should be shifted when loaded or merged. The figure will be shifted to right or down by the amounts specified here. It is also possible to shift figure to left or up by specifying negative values here. Unit of the amounts may be *Inches*, *Centimeters*, or *Fig Units (1/1200 inch in version 3.x)*.
- *Current File*

The name of the current file is displayed here. This is updated if a different filename is entered in the *Filename* field, and that file is either loaded or saved.

- *Filename*

  Specify the name of file to be loaded or saved. If this field is empty, the file indicated at *Current File* will be accessed.
  The file name in the *Filename* field may be changed by clicking a file name in *Fig Files* list, or by typing the file name on the keyboard . If *Return* is typed after the file name is entered, the file will be opened for the *Open* panel, or saved for the *SaveAs* panel, or merged for the *Merge* panel.
- *Fig Files*

  The files in the current directory matching the pattern specified by *Filename Mask* are displayed here, and users may select a file name from the list.
  Clicking a file name in this list with mouse button 1 will copy the file name to *Filename* field.
  Double-clicking a file name in this list with mouse button 1 will open the file for the *Open* panel, or or save the file for the *SaveAs* panel, or merge the file for the *Merge* panel.
- *Filename Mask*

  Only the files which match this pattern will be listed in the *Fig Files* list. The pattern is similar to the one which is used in the UNIX shell, so it is possible to use meta-characters like ``*" or ``?". In addition, multiple patterns separated by a space are allowed.
  Typing *Return* in this field will cause a rescan of the current directory as if the *Rescan* button was clicked. The initial value of this string is ``*.fig *.fig.gz *.fig.[Zz]", which will show all **Fig** files including compressed files and backups, but it may be changed by the `Fig*file_panel*mask*string` resource if necessary.
- *Current Dir*

  This shows the current directory, and files in the directory will be displayed in the *Fig files* list.
  The directory name in the *Current Dir* field may be changed by clicking a directory name in *Directories* list, or by typing the directory name on the keyboard. If *Return* is typed after the directory name is entered, the directory will scanned as if the *Rescan* button was clicked and the contents of *Fig Files* list will be updated.
- *Directories*

  List of directories in the current directory is displayed here, and clicking any item in this list with mouse button 1 will cause a move to the directory. Normally, *hidden directories* are not displayed here, but that may be toggled with the *Show Hidden* button.
  ``..'' indicates the parent directory. Moving to the parent directory may also be performed by clicking mouse button 3 on the *File Alternatives* list or the *Directories* list.
- *Home*

  Clicking this button will move to the home directory of the user.
- *Show Hidden*

  This button controls whether *hidden directories* (directories whose name starts with ``.") are displayed or not. Clicking this button will toggle the state. By default, hidden directories are not displayed.
- *Rescan*

  Clicking this button will scan the files in the current directory and update the *Fig Files* list with those files whose names match the *Filename Mask*. The accelerator `Meta-R` will also perform this function.

- *Cancel*
  Clicking this button will close the File panel. The accelerator `Meta-C` or the `Escape` key will also perform this function.

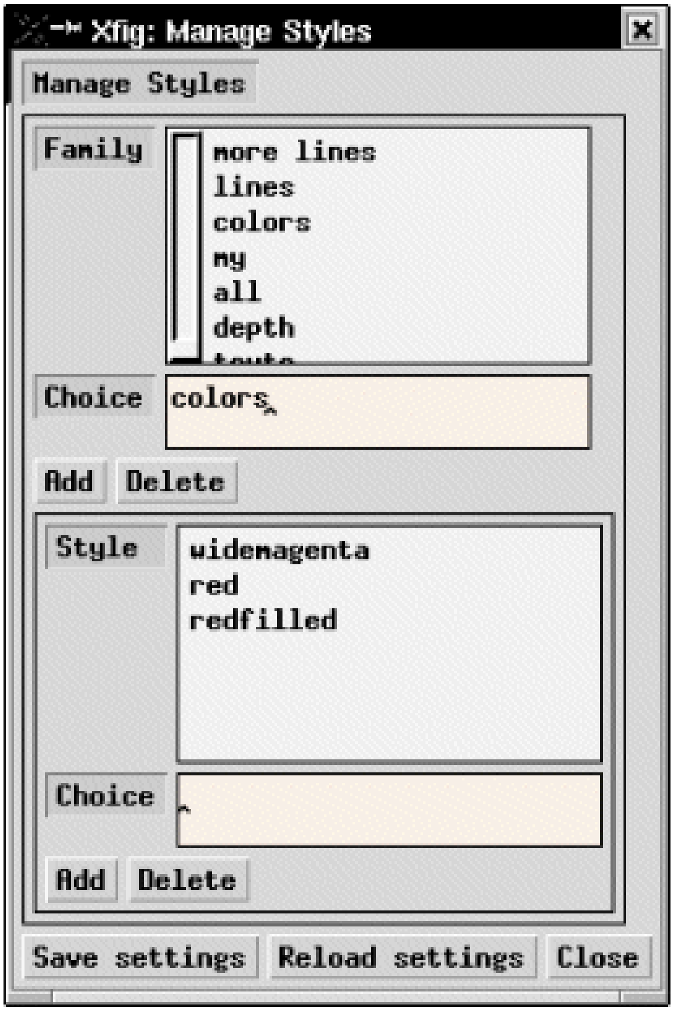
Only one of the following three buttons will appear, depending on the operation (Open, Merge, or Save):

- *Open*
  Clicking this button will open the file specified by the *Filename* field.
  Before opening the file, the figure on the canvas is cleared. If the figure on the canvas has been modified and not saved, a popup panel will appear and the user will be asked to save the current figure, discard it and open the new file or cancel the open of the new file altogether.
  The figure will be shifted by the amounts specified by *Figure Offset*.
  xfig can also read files compressed by **compress** or **gzip** (files which have suffix `.Z`, `.z`, or `.gz`).
  Before opening a compressed file, the file will be uncompressed by **uncompress** or **gunzip**.
  A newer version of **xfig** can always open *Fig* files created by older version of **xfig**, but the reverse is not true. When opening files of older format, **xfig** will notify the user in a popup message panel and will convert the file to the current version. There is no way to save a **Fig** file in an older version format.
  In addition to the **Open** button, a button labeled **New xfig...** allows the user to start another copy of **xfig** running on any file selected. This is a completely separate Unix process, but the two programs may communicate through the cut/paste feature.

- *Merge*
  Clicking this button will read the figure from the file specified by the *Filename* field and merge it with the figure on the canvas.
  As in *Open*, the figure will be shifted at the amounts specified by *Figure Offset*.

- *Save*
  Clicking this button will save the figure to the file specified in the *Filename* field if any, or the file name in *Current File* otherwise.
  When trying to save to an existing file other than *Current File*, a popup panel will appear and the user will be asked if he wants to save over that file.
  Also, *Current File* is updated to reflect the new file name.
  Before saving the file, the original file (if any) will be automatically renamed to the same name but with the suffix ``` ``.bak" ``` to leave it as a backup file.

Manage Styles...



The style manager lets you save and load groups of attributes (e.g. line color, thickness, font size etc.) by name. For example you may want a thick, dashed, red line style often, so using the style manager you could choose those attributes and save and recall them in a style called "thickred".

When adding styles, only the attributes which are selected for UPDATE will be saved.

Also, you may have "families" of styles. For example a family called "lines" may have a collection of named styles for lines (e.g. color, thickness etc.) and a family called "fonts" may have a collection of named styles for fonts (e.g. font, size, color, etc.)

To use styles, when you popup the style manager (View/Manage Styles... or `Control-Y`), **xfig** automatically goes into update mode. For styles already defined, choose the style family in the upper list and the style in that family from the lower list. The update buttons will automatically be turned on for those attributes affected by the style and off for those not affected. At this point, if you click on objects on the canvas they will be updated with those attributes. Creating new objects will use those attribtes and the others shown in the attribute panel.

To create a new family, enter the name in the **Choice** entry under **Family** and press the **Add** button.

Now select the attributes that you want to be affected by your style (e.g. line color, thickness etc.) by turning on or off the red buttons in the upper-right corner of the attribute buttons (see Update), and setting the value of the attribute. Finally, enter the name of the style you want to create in the **Choice** entry under **Style** and press the **Add** button.

There is a limit of 16 families and 30 styles in each.

- **Family**

  Choose style family from this list. A style family may have one or more styles (described below).

  - **Choice**

    This shows the current family selection, or the user may enter a family name here and press enter to load it.

  - **Add**

    After entering a name in the **Choice** field above, press this button to add it to the list of style families.

  - **Delete**

Delete the selected style family.

- **Style**

  This shows the styles available in the above-chosen family. Click on a style entry to select it.

  - **Choice**

    This shows the current style selection, or the user may enter a style name here and press enter to load it.

  - **Add**

    After entering a name in the **Choice** field above, press this button to add it to the list of styles.

  - **Delete**

    Delete the selected style.

- **Save settings**

  Save all the families and styles currently defined. To change an existing named style, first select it and delete it, then enter the name again, make the changes to the attribute panel and press "Save settings".

- **Reload settings**

  Reload the families and their styles, discarding any changes to the current listts.

- **Close**

  Close the Style Manager panel.

[Contents | Introduction | Credits]

Xfig User Manual

Version 3.2.4 December 2002

Drawing Objects

Buttons to create new objects are in the drawing mode panel, and users can create new objects using operations on the canvas after selecting a drawing mode.

Circles and Ellipses

Spline Curves

Polylines and Dimension lines NEW

Polygons

Regular Polygons

Rectangular Boxes

Arc Boxes (Rectangular Boxes with rounded corners)

Arcs

Picture Objects (Imported image files)

Texts

Object Libraries

The drawing mode is selected by clicking on the corresponding button. The image on the button will turn negative, indicating that it is selected. Settings in the Attribute Panel will be applied to newly created objects, so users should set those attributes before creating new objects.

NEW Predefined styles may be created by the user by using the style manager.

It is also possible to modify the attributes of an object using EDIT or UPDATE after the object has been created.

The function of each mouse button will change depending on the selected drawing mode or situation. This is displayed in the Mouse Function Indicator. Mouse button 1 is mainly used for creating objects, but mouse button 2 is also used in some cases. In most cases, mouse button 3 (`cancel') is used to cancel an operation.

By default, there is a point positioning grid (snap-to grid) which forces the points of the object you draw to a discrete grid. This grid may be turned off or set to various sizes. See POINT POSITION

(CIRCLE AND ELLIPSE)

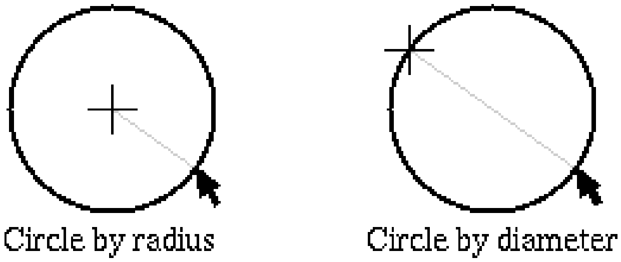
Create circles and ellipses. To draw a circle, mouse button 1 is used. For an ellipse, mouse button 2 is used.

Circles

Users can specify the center point and either the radius or diameter of the circle.

In the former mode, users should click mouse button 1 (`Circle center') at the center point of the circle first, and then click mouse button 1 (`set radius') again at any point on the circle.

In the diameter mode, users should click mouse button 1 (`Circle diameter') at any point on the circle, and then click mouse button 1 (`set radius') again at the opposite point on the circle.



If area fill is specified by FILL STYLE, the inside of the circle will be filled according to the setting. If you want to create tiny filled circle, you can also use POLYLINE.

When creating a circle, the size (radius or diameter) of the circle is displayed on the Message Panel dynamically.

Related Attributes:

depth, pen color, fill color, fill style, line width, line style

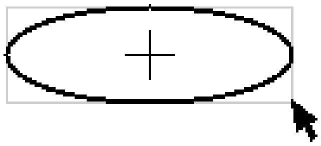
---

Ellipses

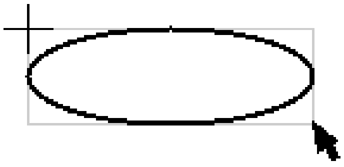
As in CIRCLE, users can specify the center point and radius or diameter of the ellipse. But because it is necessary to specify vertical and horizontal length, the operation is somewhat different from CIRCLE.

In the former mode, users should click mouse button 2 (`Ellipse center') at the center point of the ellipse first, and then click mouse button 2 (`first corner') again at any corner point of the imaginary rectangle which encloses the ellipse.

In the diameter mode, users should click mouse button 2 (`Ellipse corner') at any corner point of the imaginary rectangle which encloses the ellipse, and then click mouse button 2 (`final corner') again at the opposite corner point of the rectangle.



Ellipse by radius



Ellipse by diameter

If TEXT/ELLIPSE ANGLE is set to non-zero, the ellipse will be rotated by the angle, positive being counter-clockwise.

Related Attributes:
depth, pen color, fill color, fill style, line width, line style, text/ellipse angle

(CLOSED APPROXIMATED SPLINE)
Create a closed approximated spline curve (smooth closed curve which approximates the specified points). The operation is similar to POLYGON or CLOSED INTERPORATED SPLINE.

To make a freehand line for any SPLINE, POLYGON, or POLYLINE, click mouse button 2 (`freehand') at the first point, then drag the mouse to make the line. As you move the mouse points will be added to the line. Click mouse button 2 (`final point') again to finish.

See also About Spline Curves.

Related Attributes:
depth, pen color, fill color, fill style, line width, line style

(APPROXIMATED SPLINE)
Create an open approximated spline curve (smooth curve which passes through two end-points and approximates the specified points). The operation is similar to POLYLINE or INTERPORATED SPLINE.

See also About Spline Curves.

Related Attributes:
depth, pen color, fill color, fill style, line width, line style, cap style, arrow mode, arrow type

(CLOSED INTERPOLATED SPLINE)
Create a closed interpolated spline curve (smooth closed curve which passes through the specified points). The operation is similar to POLYGON or CLOSED APPROXIMATED SPLINE.

See also About Spline Curves.

Related Attributes:
depth, pen color, fill color, fill style, line width, line style

---

(INTERPOLATED SPLINE)
Create an open interpolated spline curve (smooth curve which passes through the specified points). The operation is similar to POLYLINE or APPROXIMATED SPLINE.

See also About Spline Curves.

Related Attributes:
depth, pen color, fill color, fill style, line width, line style, cap style, arrow mode, arrow type

---

(POLYLINES and Dimension Lines NEW)

Polylines

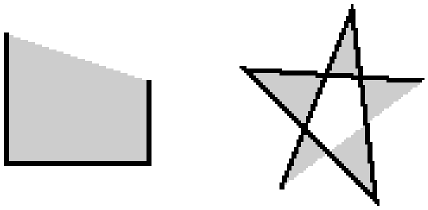
Create polylines (line segments connecting a sequence of points). A simple line segment or point may also be created with this function.

To create a polyline or line segment, specify the first point by clicking mouse button 1 (`first point') at the point, and then specify intermediate points by clicking mouse button 1 (`next point') repeatedly. Finally, click mouse button 2 (`final point') at the final point.

To make a freehand line for any SPLINE, POLYGON, or POLYLINE, click mouse button 2 (`freehand') at the first point, then drag the mouse to make the line. As you move the mouse points will be added to the line. Click mouse button 2 (`final point') again to finish.

To enter a single point click mouse button 3 (`single point'). The shape of the point will be a filled circle if CAP STYLE is *ROUND*, or a filled square otherwise. The size of the point will be equal to the line width.

If area fill is specified by FILL STYLE, the region enclosed by the polyline (first point and last point are considered connected) will be filled according to the setting. When filling a complex region, the `even-odd rule' will be used to decide the area to be filled.



Filled Polylines

For all objects, clicking mouse button 3 (`cancel') will cancel all input. For objects that may have more than two points, clicking mouse button 1 while pressing the SHIFT key (`del point') will cancel entered points one by one.

When entering lines, splines or polygons, the number of points entered and the distance from last point (length, dx and dy) is displayed on the Message Panel dynamically.

For open-ended objects, if ARROW MODE is not *None*, an arrow head will be added to one or both end-points.

If *Show line lengths* is set ON (see *Global Settings* panel and View menu), more information about the lengths of lines being drawn is shown. This is true for most **Fig** objects being drawn.

Related Attributes:

depth, pen color, fill color, fill style, line width, line style, join style, cap style, arrow mode, arrow type

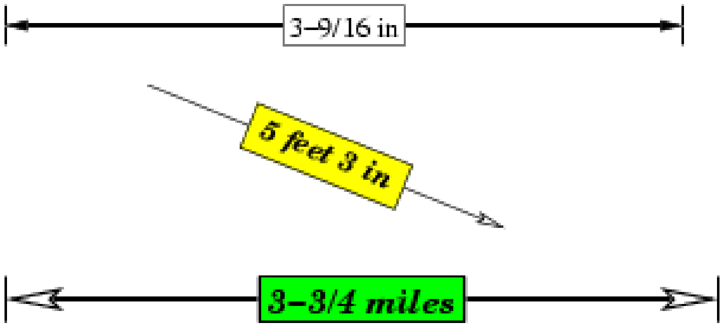
Dimension Lines NEW

Dimension lines are lines that show length and have one or more of the following:

- Text surrounded by a box showing the length of the line
- End "ticks" perpendicular to the line
- Arrowheads at either or both ends of the line

To create a dimension line, hold down the **Shift** key and press mouse button 1 on the first endpoint of the line and again at the other endpoint. **xfig** will automatically calculate the length of the line and label it *in the units currently selected for the rulers*.

By default, end "ticks", arrowheads and a box around the text are automatically included in the dimension line, but these are controlled in the attributes listed below. Here are several examples of dimension lines:



Related Attributes:

dimension line settings

(POLYGON)
Create polygons. This is identical to POLYLINE except that a line is drawn between the first point and the last point.

Related Attributes:

depth, pen color, fill color, fill style, line width, line style, join style

(REGULAR POLYGON)
Create regular polygons.

Click mouse button 1 (`center point') at the center of the polygon, and then click mouse button 1 (`final point') again at the any vertex of the regular polygon. The number of sides is specified by NUM SIDES in advance.

This will create a simple POLYGON object, and each point of the object may be moved independently after it is created (and result of the edit will no longer be a regular polygon).

Related Attributes:

depth, pen color, fill color, fill style, line width, line style, join style, num sides

(BOX)
Create rectangular boxes.

Click mouse button 1 (`corner point') at any corner of the rectangular box, and click mouse button 1 (`final point') again at the opposite corner.

If area fill is specified by FILL STYLE, the box will be filled according to the setting.

A box created with this function can't be rotated by an arbitrary angle, only by multiples of 90 degrees. To create an arbitrarily rotatable rectangular box, you should use POLYGON (it may be easier to create nonrotated rectangle first and then rotate it by ROTATE). You may also use REGULAR POLYGON to create a rotated square.

Related Attributes:

depth, pen color, fill color, fill style, line width, line style, join style

---

(ARC-BOX)
Create rectangular boxes with rounded corners. This is identical to BOX except that the corners of the box will be arcs with the radius specified by BOX CURVE.

It is also possible to convert objects between BOX and ARC-BOX with BOX <-> ARC-BOX.

Related Attributes:

depth, pen color, fill color, fill style, line width, line style, join style, box curve

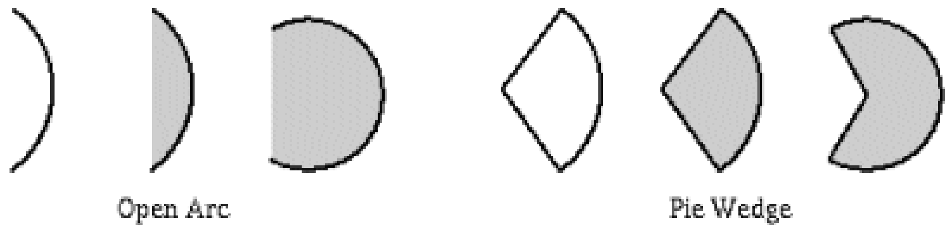
---

(ARC)
Create arcs.

There are two ways to draw arcs in **xfig**:

- The "Original" Way
  1. Click mouse button 1 (`first point') at the one end-point of the arc
  2. Click mouse button 1 (`mid point') at any intermediate point on the arc
  3. Click mouse button 1 (`final point') again at the end-point
- The "New and Improved" Way
  1. Click mouse button 2 (`center point') at the center of the arc radius
  2. Click mouse button 1 (`first point') to define one end-point of the arc
  3. Click mouse button 1 (`mid angle') again to define direction of the arc
  4. Click mouse button 1 (`final angle') a final time to define the ending **angle** of the arc

There are two types of arcs, open-arc and pie-wedge, which may be selected by ARC TYPE.



Related Attributes:

depth, pen color, fill color, fill style, line width, line style, cap style*, arrow mode*, arrow type*, arc type

(* These attributes are only effective for OPEN ARC)

Picture (IMPORTING PICTURE OBJECTS)

Import image files and create PICTURE objects.

Click mouse button 1 (`corner point') at the corner where the upper-left corner of the imported image should be placed, then click mouse button 1 (`final point') again at the opposite corner. The imported image will be rotated automatically according to the order of the corners entered, e.g. if you start with the lower-right and end with the upper-left the image will be rotated 180 degrees. Similarly, starting with the lower-left and ending with the upper-right will rotate the image 90 degrees counter-clockwise, and starting with the upper-right and ending with the lower-left will rotate the image 90 degrees clockwise. It is not necessary to try to set the exact size here because it is possible to modify size or height/width ratio of the image later.
After this, the Edit Panel will open, where you enter the file name of the image file. It is also possible to capture an image from the screen using the *Screen Capture* button.

It is possible to read image files in the following image formats:

- EPSF (Encapsulated PostScript)
- GIF with transparency
  - See also Important Note about GIF
- JPEG (JFIF)
- PCX (PC Paintbrush)
- PNG (Portable Network Graphics)
  with alpha transparency blended with xfig's background
- PPM (Portable Pixmap)
- TIFF (Tagged Image File Format)
- XBM (X11 bitmap)
- XPM (X11 pixmap)

Although it is possible to move, scale and rotate an imported image, it is not possible to edit the image itself using **xfig**. To edit images, use another application such as **xpaint** or **GIMP**. The *Edit Image* button on the Edit panel will start an external image editor (default is `xv` and it may be changed to your favorite image editor, e.g. `xpaint` by changing the `Fig.image_editor` resource or starting **xfig** with the `-image_editor` option followed by the name of your image editor).

If you want edit figures in formats such as PostScript or HP-GL using **xfig**, try **pstoedit** or **hp2xx** to first convert them to **Fig** files then Load or Merge them into **xfig**.

To import EPSF files, **xfig** invokes **GhostScript** to render the image. **GhostScript** must be compiled with `pcx256' and `pbmraw' drivers. The `pcx256' driver will used to generate color images, and `pbmraw' driver will used to generate monochrome images (used when **xfig** is running on a monochrome X server or with the `-mono` option).

JPEG and XPM support may be included or excluded depending on the configuration when compiling **xfig**.

xfig can also read files compressed with **compress** (`.Z` suffix) or **gzip** (`.z` or `.gz` suffix). To do this, the

uncompress and **gunzip** command must be available for **xfig** to invoke them.

See also Edit Panel about PICTURE objects.

Related Attributes:
depth, pen color (only for XBM images)

T (TEXT)
Create text strings.

Click mouse button 1 (`posn cursor') at the desired position on the canvas to specify the position where the text should be placed, then enter text from the keyboard, or to paste text already selected from another application (the *PRIMARY cut buffer*), press the shift key and mouse button 2 NEW, or F18 (the `Paste' key on Sun Workstations), or F20. Selecting *Paste Text* from the *Edit* menu will also paste text from the *PRIMARY cut buffer*.

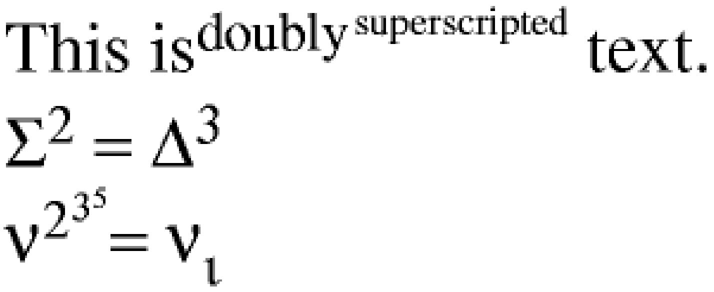
It is also possible to edit existing text by clicking mouse button 1 (`posn cursor') at any point on the existing text. See Character Editing for editing facilities.

If TEXT/ELLIPSE ANGLE is set to non-zero, the text will rotated by the angle, positive being counter-clockwise.

It is possible to enter Latin-1 glyphs such as `ä' or `¿' (Compose characters) using the COMPOSE (META) key. See Latin-1 Characters about this. These tables also show the keyboard mapping for glyphs from the Symbol font.

i18n In international-**xfig**, it is possible to enter local text such as Japanese with this.

A simplistic superscript/subscript facility is provided. While entering text, if **Ctrl-^** (control caret) is pressed, a new, smaller string is started above the baseline of the current string, which acts as a superscript. Conversely, if **Ctrl-_** (control underscore) is typed, a new, smaller string is started below the baseline of the current string. These may be nested to create super-superscripted or sub-subscripted text to four levels.
Here are some examples:



This is$^{\text{doubly}^{\text{superscripted}}}$ text.

$$\Sigma^2 = \Delta^3$$

$$\nu^{2^{3^5}} = \nu_\iota$$

There are several caveats:

- Since separate strings are created each time a super- or sub-script is initiated, if you want to move them together, you must glue all the strings into a compound first.

- Editing one of the strings may be problematic because the other strings will not move if characters are added or deleted to/from the string being edited.
- This doesn't work for right justified or centered text.

Character Editing

The following editing facilities are available:

`Control-H`	[Backspace]	Delete a character to the left of the cursor.
`Control-D`	[Delete]	Delete a character to the right of the cursor.
`Control-X`		Delete all the characters to the left of the cursor.
`Control-K`		Delete all the characters to the right of the cursor.
`Control-A`	[Home]	Move cursor of the start of the text.
`Control-E`	[End]	Move cursor of the end of the text.
`Control-B`	[<-]	Move cursor left one character.
`Control-F`	[->]	Move cursor right one character.
`Control-M`	[Return]	Finish entering the text and start entering text on the next line. See also TEXT STEP.
`F18` or **`F20`**	[Paste]	Paste text from the *PRIMARY cut buffer* (cut/paste buffer of **xterm**). Selecting *Paste Text* from the *Edit* menu will also perform this function, as will **Shift+Mouse Button 2**.
`Alt` or **`Meta`**)	[Compose]	This key is used to enter Latin-1 character such as \`ä' or \`Ç'. See also Latin-1 Characters.

It is also possible to search, replace, and spell-check all the text in the figure.

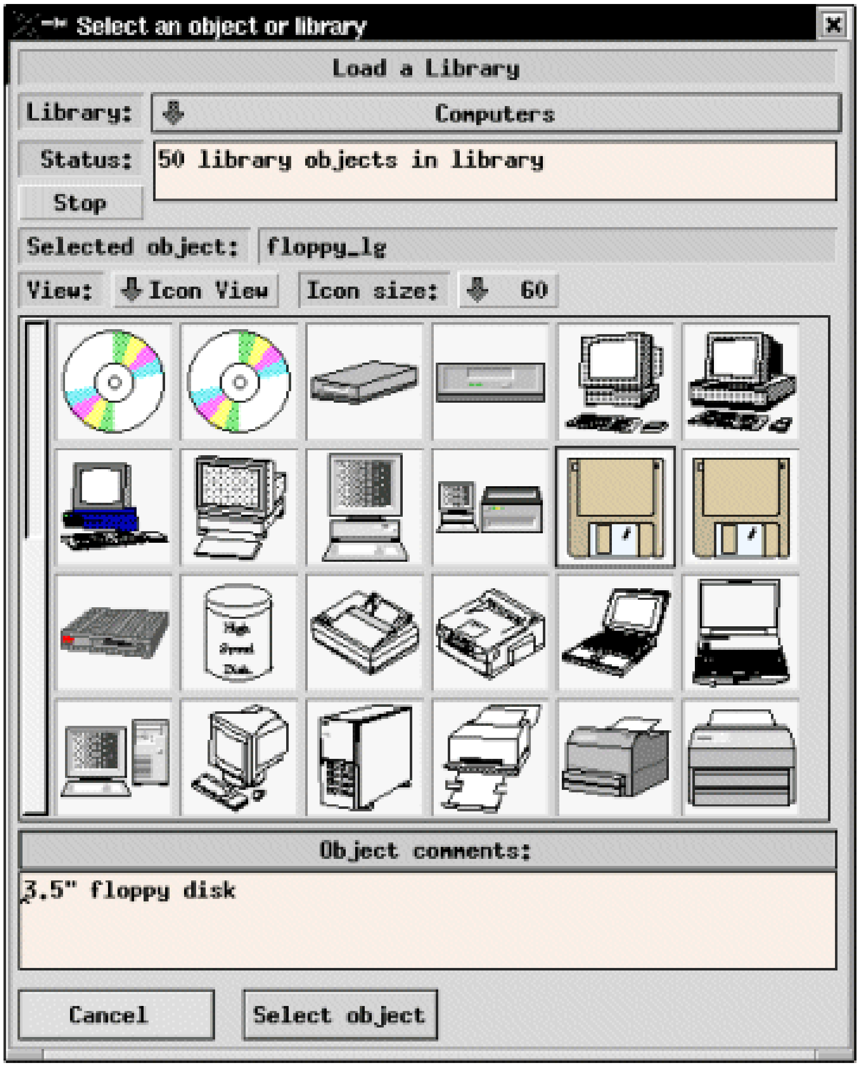
Related Attributes:

depth, pen color, text font, text size, text step, text flags, text/ellipse angle, text justification

---

(OBJECT LIBRARIES)
Pick objects from libraries to place on the canvas.

Clicking on this button will pop up a panel from which you may choose a library of **Fig** objects. From there, you may select an object and place copies of it on the canvas by clicking on the canvas where you want it placed.



Library: Computers

Status: 50 library objects in library

Selected object: floppy_lg

View: Icon View

Icon size: 60

Object comments:

3.5" floppy disk

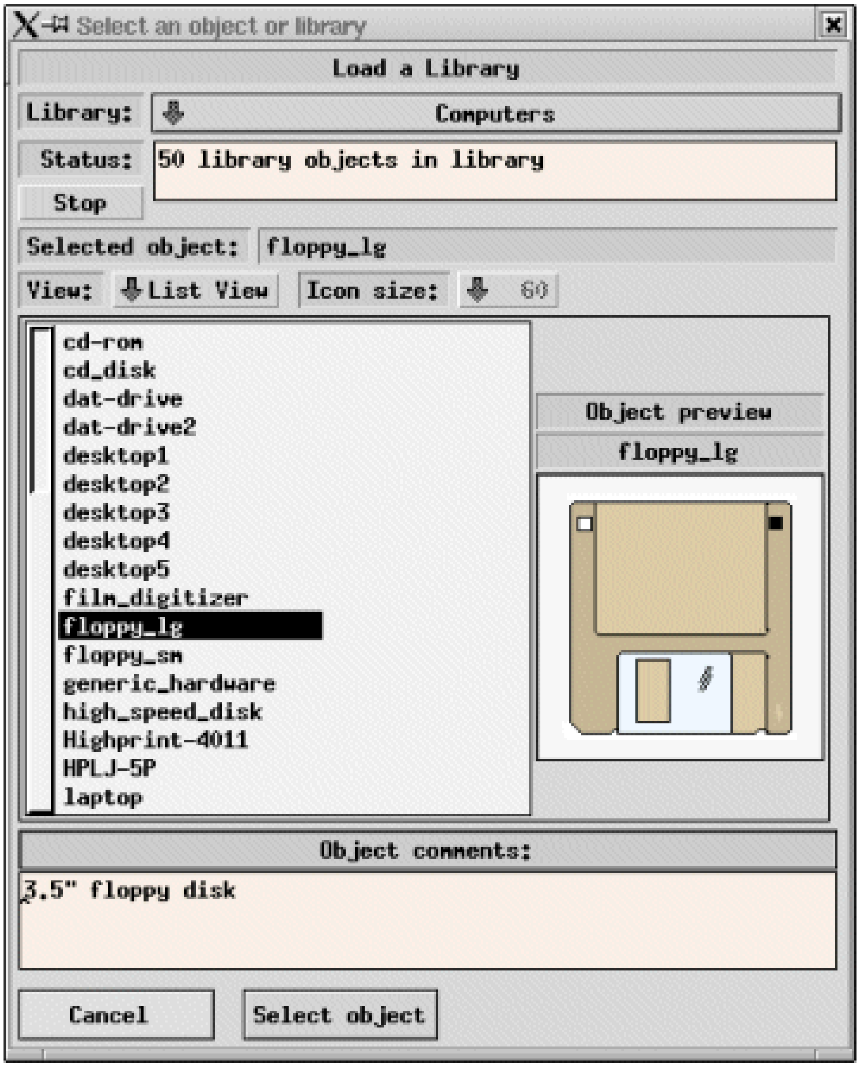
In the panel there is a pull-down menu of the available libraries. Select one and the library will automatically be loaded. The status window will show how many objects it contains after it loads the library, and the icons of the library objects will appear in a view box. Double clicking on an icon or single clicking and pressing the **Select Object** button will select that object and pop down the panel.

The size of the icons may be chosen from a pulldown menu. The options are 40, 60, 80, 100 and 120 pixels.

As you move the mouse around the canvas you will see the object you chose from the library appear under the curser. Simply click mouse button 1 (`place object') to place it anywhere on the canvas. This may be repeated for the same object.

NEW If you wish to place the object at the position it was originally created, e.g. for laser printer labels, hold down the **Shift** key and press mouse button 3 (`place at orign posn') to place it.

There is an alternate list view of the library objects, which may be selected by choosing **List View** from the pulldown menu next the **View:** label.



Select an object or library

Library: Computers

Status: 50 library objects in library

Selected object: floppy_lg

View: List View

Icon size: 60

cd-rom
cd_disk
dat-drive
dat-drive2
desktop1
desktop2
desktop3
desktop4
desktop5
film_digitizer
floppy_lg
floppy_sm
generic_hardware
high_speed_disk
Highprint-4011
HPLJ-5P
laptop

Object comments:

3.5" floppy disk

In this mode, you will see the object names in a list area. Single clicking on an object name will generate a preview in the small canvas on the right. After selecting the desired object, press `Select object' and the panel will pop down, or double-click on the object name to perform the select and popdown in one step.

If the shift key is pressed when mouse button 1 (`place and edit') is clicked to place the object, the Edit panel is popped up so you may change any comments in the object. This allows customizing of comments in library objects.

The object may be rotated, flipped or scaled before placing it on the canvas by pressing **l** to rotate left, **r** to rotate right, **h** to flip horizonally, **v** to flip vertically, **s** to scale down by 0.9 and **S** to scale up by 1.1.

To choose another object from the library either click mouse button 2 (`new object') or press the library button in the drawing mode panel again.

If you have a slow display or if the library object is complex, you may elect to have **xfig** show only the outline rectangle as you move the mouse around before placing the object. To switch to this mode, press SHIFT and mouse button 2 (`change draw mode'). You may toggle between drawing modes with this sequence.

To cancel the library placing mode, press mouse button 3 (`cancel library') or any other drawing/editing mode button.

Available Libraries of Fig Objects

The **xfig** distribution comes with several libraries of **Fig** objects which you may use:

- Various 3D arrows
- Audiovisual components
- Some buildings
- Some charts (polar, log-log)
- Various computers
- DSP (Digital Signal Processor) components
- Electrical - electrical objects, e.g. connectors
- Electronic Physical - symbols for creating electronic wiring diagrams or layouts
- Electronic Schematic - symbols for creating electronic schematics
- ERD (Chen's model of Entity Relations)
- Examples - various examples of xfig drawings contributed by users
- Flags of various countries and some U.S. states
- Flowchart symbols
- GUI widgets (buttons, sliders, etc.)
- Hospital equipment
- Knitting symbols
- Various printer labels (CD, Avery address labels)
- Logic symbols
- Maps - outline maps of countries, U.S. states and Canadian provinces
- DIN Mechanical symbols
- Miscellaneous - various fun items here
- Music symbols
- Networking devices
- Office equipment - filing cabinet, telephone, overhead projector
- Optics symbols
- Process Flowsheet - chemical process flow symbols
- Structural Analysis symbols
- Unified Modelling Language (UML) symbols
- Welding - common welding patterns and errors

It is hoped that users will contribute Fig objects to the libraries to expand and enhance them for all **xfig** users. If you wish to contribute any Fig objects to existing or new libraries, please send them to me at xfigmail@epb.lbl.gov. You may restrict their usage by users if you wish, but all will benefit if you allow free usage, i.e. copying, using them in published reports, etc.

Making Your Own Libraries of Fig Objects

A library is simply a directory containing **Fig** files. There are two ways to create your own libraries in **xfig**; simply put all libraries in a directory, or use a text file which contains list of libraries.

Method 1

In the first method, you put all libraries (directories containing **Fig** files) in a directory. If you want use additional libraries in other directories (for example, libraries of somebody else), simply make a symbolic link to those directories in your library directory.

Then you can start **xfig** specifying your library directory with `-library_dir` option or `Fig.library_dir` resource. **xfig** scans the library directory and all libraries in the directory and any subdirectories will become available.

The default value of `library_dir` is ``$XFIGLIBDIR/Libraries", where libraries comes with **xfig** will be installed automatically when installing **xfig**. So, you don't have to specify `library_dir` if you want to use libraries comes with **xfig**.

This method is more versatile than the second method because you can add more directories in the top-level directory (or sub-directories) and there is no need to edit the file which contains list of libraries.

Using this method, the names of the libraries are simply the names of the directories; e.g. if the resource ``Fig.library_dir: /xfig/Libraries" is defined, the `/xfig/Libraries/Flowchart` directory will result in the library name ``Flowchart", and the `/xfig/Libraries/Electrical/Physical` directory will result in the library name ``Electrical - Physical".

Method 2

In the second method, you create a text file which contains list of libraries. Each line of the file contains the absolute path of the library (a directory containing **Fig** files) followed by the name of the library, which may be multiple words with whitespace.

And then you can start **xfig** specifying the file with `-library_dir` option or `Fig.library_dir` resource.

Here is an example of the file:

```
  /xfig/Libraries/Electrical/Logic    Electrical Parts (Logic)
  /xfig/Libraries/Electrical/Analog   Electrical Parts (Analog)
  /home/tim/lib/Flowchart             Tim's Flowchart Elements
  /home/chim/xfig/Buildings           Computer Clipart
```

The advantage of this method is flexibility; you can easily put libraries in various directories into your list of libraries, and you can use any name for each library independent of the filenames.

N.B.: In both methods, any sub-directories (and their sub-directories) found with `*.fig` files in them will be included in the list.

Limits

There are some limits with the libraries. These limits are independent of the two methods used.

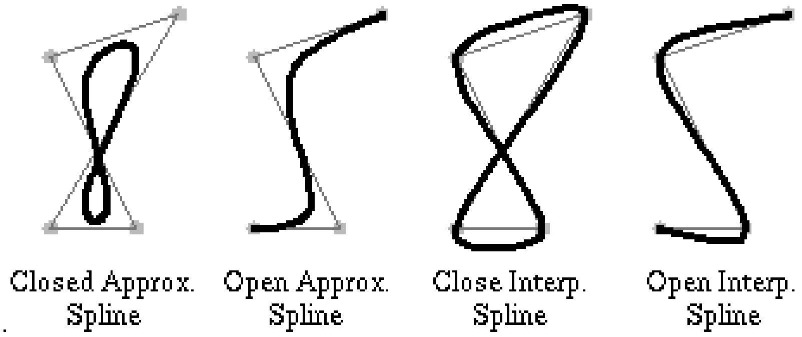
- Number of libraries: at most 100
- Length of each library name: at most 40 characters
- Number of objects in a library: at most 400 (for each library)
- Length of each object name (filename of **Fig** files): at most 40 characters

About Spline Curves

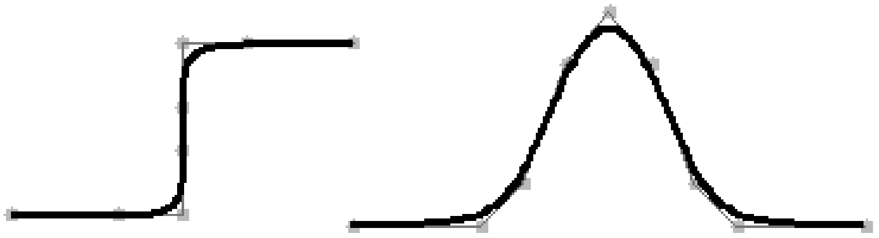
A *Spline curve* is a smooth curve controlled by specified points.

Users may select from four types of spline curves on the Drawing Mode Panel of **xfig**. It is also possible to change the type of spline curves using SPLINE <-> LINE, or adjust the shape of the curve by adjusting ``shape factor'' after curve has been created.

- CLOSED APPROXIMATED SPLINE
  Smooth closed curve which approximates specified points.
- OPEN APPROXIMATED SPLINE
  Smooth curve which approximates specified points.
- CLOSED INTERPORATED SPLINE
  Smooth closed curve which passes through specified points.
- OPEN INTERPORATED SPLINE
  Smooth curve which passes through specified points.

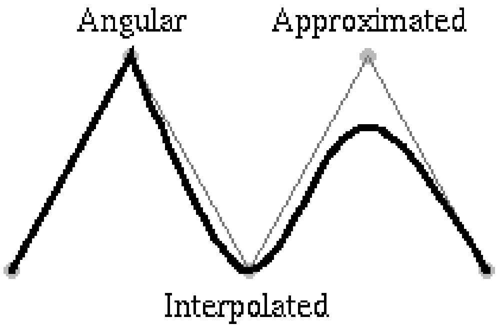


Using splines, curves such as the following may be easily drawn.



About X-Splines

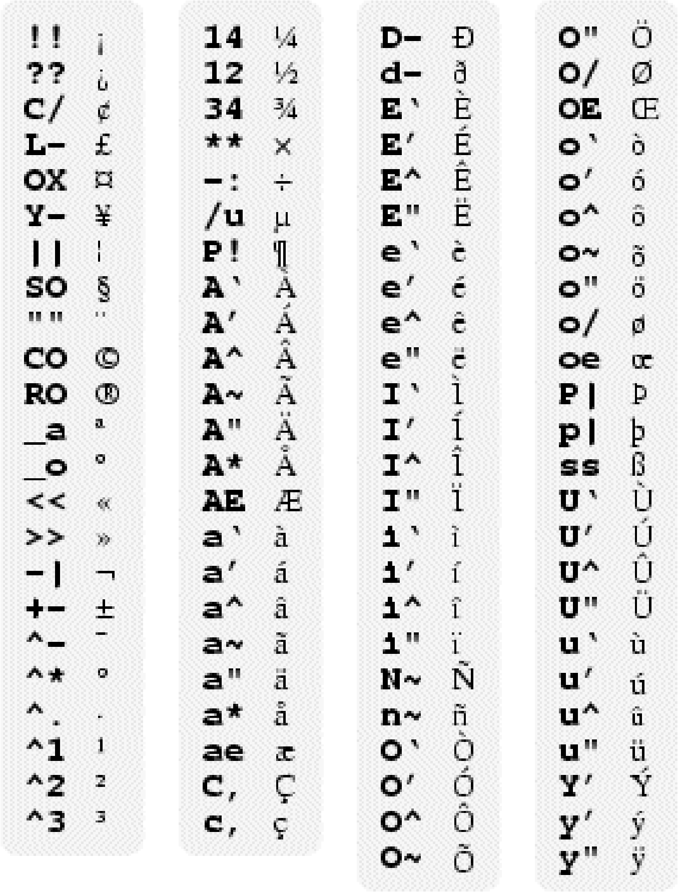
xfig's new X-Spline is bit more flexible. Each points (*control points*) of spline has `*shape factor*' and value of the *shape factor* controls behavior of the curve near the point. Value of *shape factor* can be changed independently using EDIT POINT, and it is possible to mix approximated spline, interpolated spline and polyline in one continued curve.



Caveat: Because previous spline models (quadratic B-splines and Bezier with hidden points) are no longer supported per se, curves that are present in **xfig** version 3.1 and older files are automatically converted to X-splines. This translation is only an approximation process. It means that the converted curves may not be exactly the same as the original ones. Though they are usually very close, some hand-fitting may be needed in some pathological cases.

Latin-1 and Symbol Characters (Compose Characters)

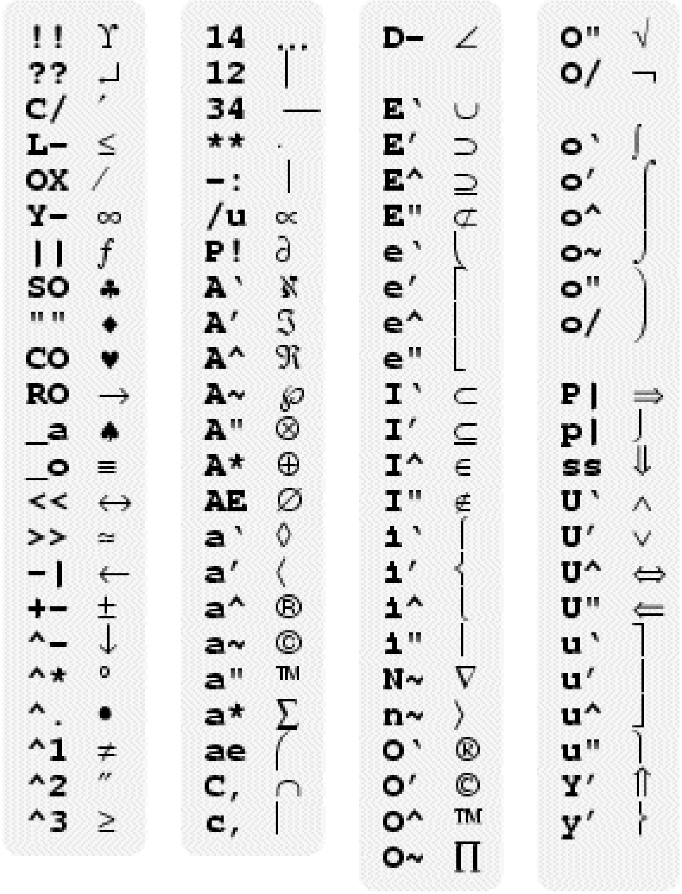
In TEXT mode, the following Latin-1 glyphs may be entered using the COMPOSE (META) key. For example to enter `ä', press and release the COMPOSE key (or META or ALT key), enter the character `a', and then the character `"' (double quote). To enter `ç', enter COMPOSE, `c' and `,'.



!!	¡	14	¼	D-	Ð	O"	Ö
??	¿	12	½	d-	ð	O/	Ø
C/	¢	34	¾	E`	È	OE	Œ
L-	£	**	×	E'	É	o`	ò
OX	¤	-:	÷	E^	Ê	o'	ó
Y-	¥	/u	µ	E"	Ë	o^	ô
||	¦	P!	¶	e`	è	o~	õ
SO	§	A`	À	e'	é	o"	ö
""	¨	A'	Á	e^	ê	o/	ø
CO	©	A^	Â	e"	ë	oe	œ
RO	®	A~	Ã	I`	Ì	P|	Þ
_a	ª	A"	Ä	I'	Í	p|	þ
_o	º	A*	Å	I^	Î	ss	ß
<<	«	AE	Æ	I"	Ï	U`	Ù
>>	»	a`	à	i`	ì	U'	Ú
-|	¬	a'	á	i'	í	U^	Û
+-	±	a^	â	i^	î	U"	Ü
^-	¯	a~	ã	i"	ï	u`	ù
^*	°	a"	ä	N~	Ñ	u'	ú
^.	·	a*	å	n~	ñ	u^	û
^1	¹	ae	æ	O`	Ò	u"	ü
^2	²	C,	Ç	O'	Ó	Y'	Ý
^3	³	c,	ç	O^	Ô	y'	ý
				O~	Õ	y"	ÿ

Note: glyphs **OE** and **oe** will not be displayed on the screen, because they are not in the X11 font sets, but they will export and print to PostScript, EPS, PDF and the bitmap formats.

The following glyphs from the Symbol font may be entered in the same way:



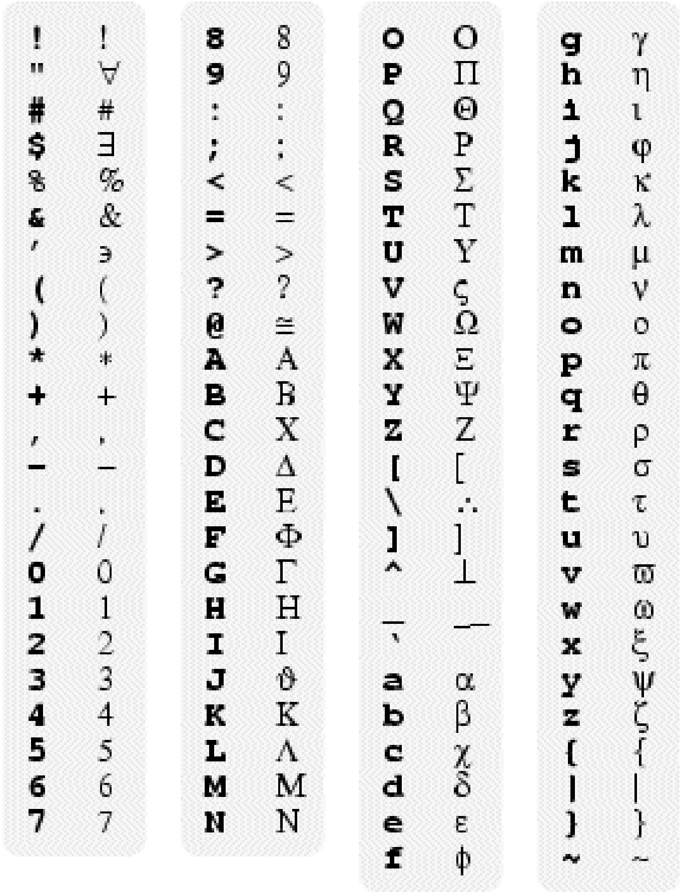
!!	ϒ
??	↵
C/	′
L-	≤
OX	⁄
Y-	∞
||	ƒ
SO	♣
""	♦
CO	♥
RO	→
_a	♠
_o	≡
<<	↔
>>	≃
-|	←
+-	±
^-	↓
^*	°
^.	•
^1	≠
^2	″
^3	≥

14	…
12	⏐
34	⎯
**	.
-:	⏐
/u	∝
P!	∂
A`	ℵ
A′	ℑ
A^	ℜ
A~	℘
A"	⊗
A*	⊕
AE	∅
a`	◊
a′	〈
a^	®
a~	©
a"	™
a*	∑
ae	⎛
C,	∩
c,	⎜

D-	∠
E`	∪
E′	⊃
E^	⊇
E"	⊄
e`	⎝
e′	⎡
e^	⎢
e"	⎣
I`	⊂
I′	⊆
I^	∈
I"	∉
i`	⎧
i′	⎨
i^	⎩
i"	⎪
N~	∇
n~	〉
O`	®
O′	©
O^	™
O~	∏

O"	√
O/	¬
o`	∫
o′	⌠
o^	⎮
o~	⌡
o"	⎞
o/	⎠
P|	⇒
p|	⎠
ss	⇓
U`	∧
U′	∨
U^	⇔
U"	⇐
u`	⎤
u′	⎥
u^	⎦
u"	⎫
Y′	⇑
y′	⎬

Here is the keyboard mapping for the lower set of glyphs from the Symbol font:



!	!
"	∀
#	#
$	∃
%	%
&	&
′	∍
(	(
)	)
*	∗
+	+
,	,
-	−
.	.
/	/
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7

8	8
9	9
:	:
;	;
<	<
=	=
>	>
?	?
@	≅
A	Α
B	Β
C	Χ
D	Δ
E	Ε
F	Φ
G	Γ
H	Η
I	Ι
J	ϑ
K	Κ
L	Λ
M	Μ
N	Ν

O	Ο
P	Π
Q	Θ
R	Ρ
S	Σ
T	Τ
U	Υ
V	ς
W	Ω
X	Ξ
Y	Ψ
Z	Ζ
[	[
\	∴
]	]
^	⊥
_	‾
`	
a	α
b	β
c	χ
d	δ
e	ε
f	ϕ

g	γ
h	η
i	ι
j	φ
k	κ
l	λ
m	μ
n	ν
o	ο
p	π
q	θ
r	ρ
s	σ
t	τ
u	υ
v	ϖ
w	ω
x	ξ
y	ψ
z	ζ
{	{
|	|
}	}
~	∼

Not all COMPOSE, META and ALT keys can be used on all systems. On some systems such as Sun Workstations, the META key may be labeled as ◆.

i18n If international-**xfig** is used in environments which use character set other than Latin-1 (Japanese or Korean, for example), it is not possible to use these Latin-1 glyphs when `*Times-Roman + Mincho*' or `*Times-Bold + Gothic*' is selected in TEXT FONT.

[Contents | Introduction | Credits]

Xfig User Manual

Version 3.2.4 December 2002

Editing Objects

The editing Mode Panel is used to perform edit operations on existing objects, See also Selecting Objects.

Glue COMPOUND

Break COMPOUND

Open COMPOUND temporarily

Join/Split Lines/Splines/etc

Scale Object

Align Objects

Move Point

Add Point

Delete Point

Move Object

Copy Object

Delete Object

Update Attributes of Object

Edit Attributes of Object

Flip Object Vertically or Horizontally

Rotate Rotate Rotate Object

Convert SPLINE to/from LINE or BOX to/from ARC-BOX

Add/Delete Arrows

Add Tangent or Normal to Curve NEW

Measure Angle NEW

Measure Length of Lines and Arcs, and Perimeters of Circles and Ellipses NEW

Measure Area of Polygons and Ellipses NEW

Glue selected objects and make them a COMPOUND object.

This is useful to move or copy some objects together at the same time. It is also possible to scale part of a figure by scaling a COMPOUND object after making part of the figure COMPOUND.

Clicking on an object with mouse button 1 (`tag object') will tag the object. Clicking one corner of a rectangular region with mouse button 2 (`tag region') and then clicking mouse button 2 (`final corner') again at the opposite corner of the rectangular region will tag all objects inside the region. After tagging all objects to be glued into a COMPOUND object, clicking mouse button 3 (`compound tagged') will glue those objects and make them a COMPOUND object.

If an already tagged object is selected by mouse button 1 or mouse button 2, that object will be untagged. If you want to remove all tags, change the mode temporarily to any other mode (e.g. "move object") except BREAK COMPOUND.

The COMPOUND object may be separated into component objects by BREAK COMPOUND.

Break a COMPOUND object into separate component objects.

Clicking a COMPOUND object with mouse button 1 (`break compound') will break the COMPOUND into separate component objects. Clicking with mouse button 2 (`break and tag') will also achieve the same effect, but it will tag those component objects for re-gluing with GLUE COMPOUND again.

If you want to edit component objects without breaking the COMPOUND object, you may open the COMPOUND object temporarily using OPEN COMPOUND.

Open a compound object temporarily for editing the component objects.

Clicking a COMPOUND object with mouse button 1 (`open compound') will open the COMPOUND object, and then the component objects may be edited. All other objects which were not in the COMPOUND object will become invisible temporarily, but they will reappear after all COMPOUND objects are closed.

A popup panel with two buttons to close the COMPOUND objects (*Close This Compound* and *Close All Compounds*) will appear when a COMPOUND object is opened, and the open COMPOUND may be closed again using those buttons. Clicking *Close This Compound* will close the COMPOUND object last opened, and clicking *Close All Compounds* will close all nested, opened COMPOUND objects.

If you want to break a COMPOUND object into component objects, use BREAK COMPOUND.

If the File/Save is attempted while a compound is open, **xfig** will ask if you want to save only the open COMPOUND object (Save Part) or the whole figure (Save All).

Note: If you open a COMPOUND object, delete an object in that COMPOUND object, close the COMPOUND object and Undo the delete operation, the object you deleted will reappear but it will outside of the COMPOUND object. Conversely, if you delete an object from the canvas, open a compound then undo the delete of the original object, it will now be part of the open compound. In this way you may add or remove selected objects from a COMPOUND.

Join or split lines

With this feature you may join lines or splines together or split a line or spline into two objects. Also, you may close an open line (make it a polygon) or spline (closed spline) or open a box, polygon or closed spline.

To join two lines or two splines together click mouse button 1 (`Join lines/splines') to select the first endpoint and again (`Choose next point') to select the second endpoint to be joined. If the endpoints are on the same line or spline, it will make a closed object.

To split a line or spline into two separate objects or to convert a box or polygon to a line, or to open a closed spline, click mouse button 2 (`Spline line/spline') between two the points on the line/spline etc.

Scale object

Clicking any **corner** of BOX, ARC-BOX, or COMPOUND object with mouse button 1 (`scale box') and dragging the mouse will start scaling the object proportionally to its horizontal/vertical ratio. Clicking any **edge** of that object with mouse button 1 (`scale box') will start scaling the object only in one dimension (width or height). The scaling operation is finished by clicking mouse button 1 (`new position') again.

Clicking on an object with mouse button 2 (`scale about center') will scale the object proportionally about its center. The scaling operation is finished by clicking mouse button 2 (`new position') again.

In any case, The scaling operation may be canceled by clicking mouse button 3 (`cancel').

It is not possible to scale TEXT objects directly with this function. To scale TEXT objects, use EDIT OBJECT or UPDATE and change the size of the text (see TEXT SIZE). However, if a COMPOUND object that includes TEXT objects is scaled, the TEXT objects will also be scaled if the *Rigid* flag is not set.

You may use also MOVE POINT if you want to scale an object such as a BOX without keeping its horizontal/vertical ratio.

Align objects

How objects are aligned is set by VERTICAL ALIGN and HORIZONTAL ALIGN.

Clicking on a COMPOUND object with mouse button 1 (`align compound') will align objects within the COMPOUND object. Clicking mouse button 2 (`align canvas') anywhere on the canvas will align all objects to the canvas.

Move a point of an object

Select the point to be moved by clicking the point with mouse button 1 (`move point'), and finish the move by click mouse button 1 (`new position') again after moving mouse to the new point. If you want to restrict the move to horizontal or vertical, use mouse button 2 (`horiz/vert move') instead of mouse button 1. The operation may be canceled by clicking mouse button 3 (`cancel') in either case.

This may not be used for TEXT objects. For BOX, ARC-BOX, and COMPOUND objects, not only the selected point but the edges connected to the point will be moved. The object created by REGULAR POLYGON is an ordinary POLYGON object, and only the selected point will be moved.

Points may also be added or deleted. If you want to move entire object, use MOVE instead.

Add a new point to a POLYLINE, POLYGON, or SPLINE object

Clicking mouse button 1 (`break/add here') between two points of an object will add a new point at that location, and the point will move following mouse cursor. The operation is finished by clicking mouse button 1 (`place new point') again.

It is not possible to add a point to BOX or ARC-BOX objects. The object created by REGULAR POLYGON is an ordinary POLYGON object, and it is possible to add new points anywhere in it.

Points may also be moved or deleted.

Delete point from POLYLINE, POLYGON, or SPLINE object

Clicking a point to be deleted with mouse button 1 (`delete point') will delete the point, and the points on either side of the point will connected directly.

It is not possible to delete points from a BOX or ARC-BOX object. The object created by REGULAR POLYGON is ordinary POLYGON object, and it is possible to delete points from it.

Points may also be moved or added.

Move object

Select the object to be moved by clicking the object with mouse button 1 (`move object'), and finish the move operation by clicking mouse button 1 (`place object') again at the new position. If you want to restrict the movement to horizontal or vertical, use mouse button 2 (`horiz/vert move') instead of mouse button 1. The operation may be canceled by clicking mouse button 3 (`cancel') in either case.

If you want to move several objects at one time, you may put them into a COMPOUND object using GLUE COMPOUND.

By setting SMART-LINKS MODE, it is possible to move the endpoints of lines which touch the object as you move the object itself. Additionally, endpoints of lines which fall inside a COMPOUND object will also be moved. The SMART-LINKS feature only works with POLYLINES (not splines) touching a BOX, ARC-BOX, or COMPOUND.

Objects may also be copied or deleted. If you want to move a point of the object, use MOVE POINT instead.

Copy object

Select the object to be copied by clicking the object with mouse button 1 (`copy object'), and finish the copy operation by clicking mouse button 1 (`place object') again at the position to place the copy. If you want to restrict the movement to horizontal or vertical, use mouse button 2 (`horiz/vert copy') instead of mouse button 1. The operation may be canceled by clicking mouse button 3 (`cancel') in either case.

If you want to copy several objects at the same time, you may put them a COMPOUND object using GLUE COMPOUND.

By setting SMART-LINKS MODE, it is possible to copy the lines which touch the object as you move the object itself. This only works with POLYLINES (not splines) touching a BOX or ARC-BOX.

It is also possible to make many copies in this mode, using *ARRAY PLACEMENT*. To do this, first set the number of copies to be created by setting NUMBER OF X COPIES and NUMBER OF Y COPIES. Then select the object to be copied by clicking the object with mouse button 1 (`copy object') or 2 (`constrained copy'), and specify the direction and distance to place the copies by clicking mouse button 2 (`array placement') after moving the mouse. Normally, objects will be placed on the array of NUMBER OF X COPIES and NUMBER OF Y COPIES, but when either of them is set 0 or 1, copies of specified number will be generated and they can be placed obliquely. For example, if the number of X copies is 3 and the number of Y copies is 0 or 1, the copied objects will be placed along the X direction at the same Y position as the original object.

Clicking an object with mouse button 3 (`copy to cut buf') will copy the object to the *xfig cut buffer*. The object copied into the *xfig cut buffer* may be inserted into the figure on the canvas using Paste Objects. This allows you to copy part of a figure to another figure. Any object in the *xfig cut buffer* will be overwritten when the new object is copied into the *xfig cut buffer*. To copy multiple objects into the *xfig cut buffer*, you must first put them into a COMPOUND object using GLUE COMPOUND.

Objects may also be moved or deleted.

Delete object

Clicking an object with mouse button 1 (`delete object') will remove the object. It is also possible to remove objects in a rectangular region by clicking mouse button 2 (`delete region') at one corner of the region and then again (`final corner') again at the opposite corner.

Clicking an object with mouse button 3 (`del to cut buf') will remove the object from the canvas and move it to the *xfig cut buffer*. The object moved into the *xfig cut buffer* may be inserted into the figure on the canvas using Paste Objects later. See also COPY.

If you want to delete all objects from the canvas to make a new figure, you may use New.

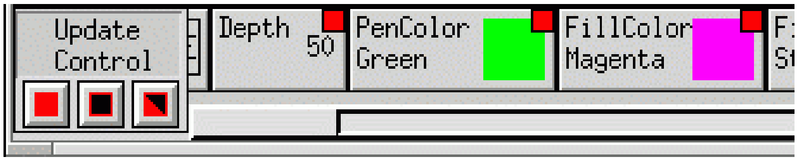
Update object or attribute panel

Update the attributes (such as line width or colors) of object with the current settings in the Attribute Panel. Conversely, it is also possible to copy the attributes of an object to the attribute panel.

Clicking an object with mouse button 1 (`update object') will apply the current settings of the attribute buttons selected for UPDATE to the object. If a COMPOUND object is clicked with mouse button 1 (`update object'), the attributes of all the objects in the COMPOUND object will be updated.

Clicking an object (except a COMPOUND object) with mouse button 2 (`update settings') will copy the attributes of the object to the attribute buttons which are selected for UPDATE. This allows you to copy attributes from one object to another object.

In UPDATE mode, a small toggle button appears in the upper-right corner of each attribute button, and only the attribute buttons whose toggle button is set *ON* will be used for update. Clicking the toggle button at the upper-right corner of an attribute button will toggle the button *ON/OFF*. It is also possible to select, unselect, or toggle all attribute buttons, using *Update Control* buttons at the left side of the attribute panel.



Depth 50

PenColor Green

FillColor Magenta

The attributes of objects may also be modified using EDIT OBJECT. Some attributes can't be modified by UPDATE but may be modified by EDIT OBJECT.

Edit

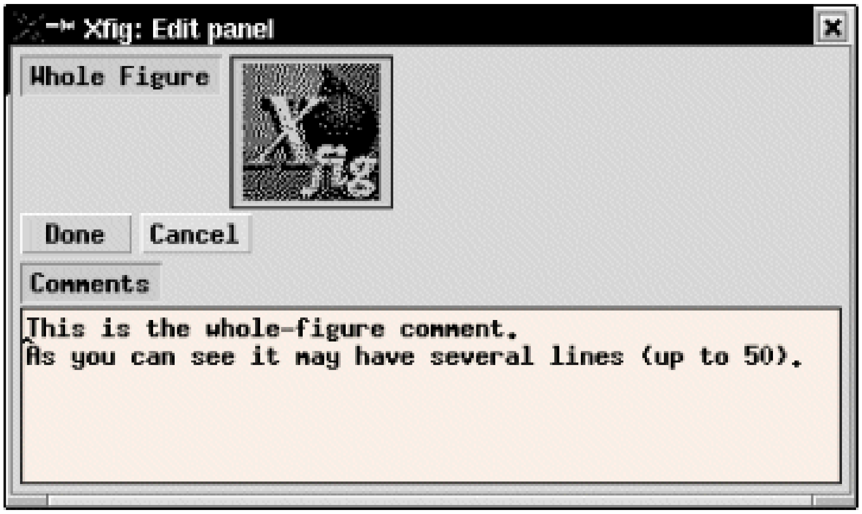
Edit object attributes

Clicking an object with mouse button 1 (`edit object') will popup the Edit Panel to edit the attributes. The items in the panel depend on the object being edited. See Edit Panel for more information.

Clicking a control point of a SPLINE object with mouse button 3 (`edit point') will popup EDIT POINT panel to modify the *shape factor* of the spline curve.

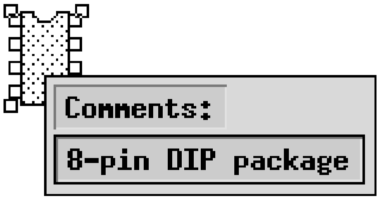
The attributes of objects may also be modified using UPDATE, but some attributes can't be modified with UPDATE.

If mouse button 2 (`edit Fig comments') is clicked on the canvas, the edit panel will popup to allow the user to change the comments associated with the whole figure:



This is the whole-figure comment.
As you can see it may have several lines (up to 50).

If the **Shift** key is held down while mouse button 2 (`show comments') is clicked on an object, any comments in the object are displayed in a popup until the mouse button is released:



8-pin DIP package

Flip object vertically or horizontally

Clicking the object with mouse button 1 (`flip') will flip the object vertically/horizontally about that point. Clicking the object with mouse button 2 (`copy & flip') will make **copy** of the object flipped vertically/horizontally about that point, leaving original object as it was.

It is also possible to set the *anchor point* about which the object will be flipped, by clicking mouse button 3 (`set anchor') at the point. A crosshair (+) indicates the anchor point on the canvas. The anchor point may be removed by clicking mouse button 3 (`set anchor') again.

To rotate an object, use ROTATE.

Rotate Rotate Rotate object

Clicking an object with mouse button 1 (`rotate object') will rotate the object to clockwise or counter-clockwise by the angle specified by ROTATION ANGLE about the point that is clicked.

It is also possible to make many copies in this mode, using *COPY & ROTATE*. Clicking object with mouse button 2 (`copy & rotate') may make many copies rotated by the angle specified by the ROTATION ANGLE. The number of copies to be created may be set by NUMBER OF COPIES.

As in FLIP VERTICALLY/HORIZONTALLY, it is possible to set the *anchor point* about which the object will be rotated, by clicking mouse button 3 (`set anchor').

BOX, ARC-BOX and PICTURE objects, and COMPOUND objects containing any of those objects may only be rotated by multiples of 90 degrees. To flip object vertically or horizontally, use FLIP VERTICALLY/HORIZONTALLY.

Convert object type

Clicking an object with mouse button 1 (`spline <-> line') will perform a conversion thus:

- polyline --> open approximated spline --> open interpolated spline --> polyline
- polygon --> closed approximated spline --> closed interpolated spline --> polygon
- box --> arc-box --> box

Clicking an object with mouse button 3 (`open <-> close') will perform a conversion thus:

- polyline <-> polygon
- open approximated spline <-> closed approximated spline
- open interpolated spline <-> closed interpolated spline

Finer control is possible for spline curves. See EDIT POINT and About Spline Curves for more information.

Add/delete arrow head

Clicking the end point of object with mouse button 1 (`add arrow') will add an arrow head to that end. Clicking an arrow head with mouse button 2 (`delete arrow') will remove the arrow head.

This operation is only valid for POLYLINE, OPEN SPLINE, or OPEN ARC objects.

The shape of the arrow head may be set by ARROW TYPE. The size of arrow head may be set by ARROW SIZE. It is also possible to automatically create arrow heads when creating new objects by setting ARROW MODE.

Add Tangent or Normal to Curve NEW

Clicking mouse button 1 near a curve of an arc or spline will create a 2cm line **tangent** to the curve, with the current line attributes such as thickness, color, etc. The line has three points, with the middle point touching the original curve.

Clicking mouse button 2 near a curve will create a 2cm line **normal** to the curve.

Measure Angle NEW

Clicking mouse button 1 on a vertex of a polyline or polygon will report (in the message window below the main menus) the angle between the two lines meeting at the vertex.

Clicking mouse button 1 on any three points in the canvas will report the angle between the two lines formed by the three points.

Clicking mouse button 2 on the vertex of a polyline or polygon will report the angle between the two lines meeting at the vertex, **and** save the angle in the current rotation angle setting in the attributes panel.

Measure Length of Lines and Arcs, and Perimeters of Circles and Ellipses NEW

Clicking mouse button 1 on any polyline, polygon, box, arc or circle/ellipse will report (in the message window below the main menus) the length of the lines forming the object, or the perimeter for a circle or ellipse.

Clicking mouse button 2 on such an object will add the length to a running sum and report the individual and accumulated lengths.

Clicking mouse button 3 will reset the accumulated length to 0.

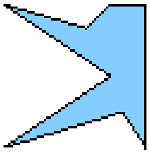
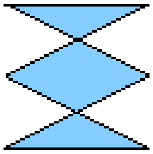
Measure Area of Polygons and Ellipses NEW

Clicking mouse button 1 on any polyline, polygon, box or circle/ellipse will report (in the message window below the main menus) the area of the object.

Clicking mouse button 2 on such an object will add the area to a running sum and report the individual and accumulated areas.

Clicking mouse button 3 will reset the accumulated area to 0.

The area for polygons that have overlapping parts will not be calculated correctly. This should be fixed in a future version.

Will work correctly:  Will NOT work correctly: 

Selecting Objects

Objects may be selected by clicking the *corner markers* (*handles*) of the object. The corner markers are small square marks which are displayed at the corner point of objects which may be selected for any particular mode.

Most objects may also be selected by clicking on the line connecting corner markers, but it may have a different meaning in some editing modes. For example in the SCALE OBJECT mode, clicking on a corner marker will scale the object proportionally to its horizontal/vertical ratio, but clicking on a line connecting corner markers (edge of the rectangle) will scale the object only in that direction (i.e. horizontally or vertically).

If POINT POSITION is set to other than *Any*, objects which are not on the virtual grid may not be be selected.

When object corners or edges are coincident, clicking on the object may not select the desired object but the other object instead. In this case, the desired object may be selected with following operations:

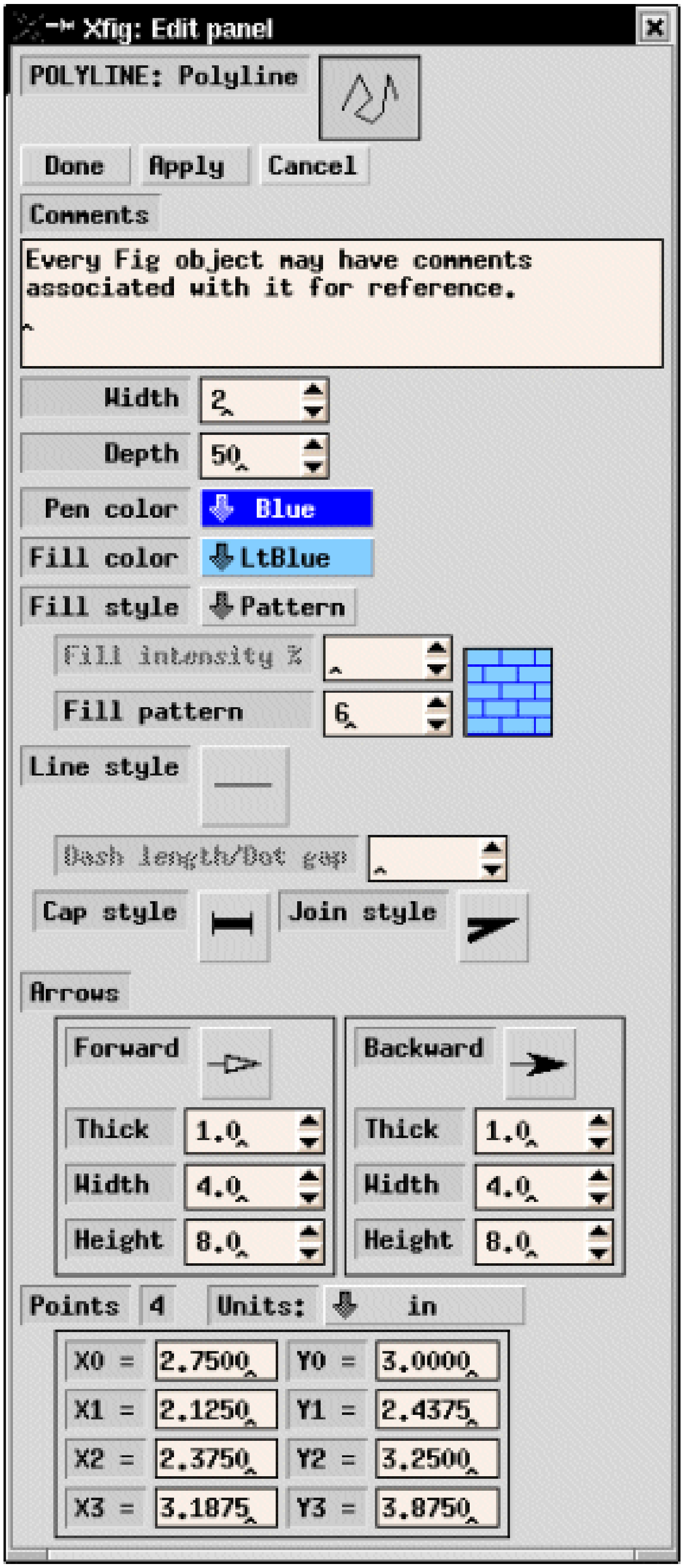
1. Press (and hold) the SHIFT key and click on the object to be selected.
2. If the corner marker of the other object is highlighted, keep the SHIFT key down and click the object again.
3. When corner marker of the desired object is highlighted, release the SHIFT key and click the object again.

Note that you shouldn't move the mouse during this operation.

EDIT Panel

The Edit Panel will popup when an object is selected in EDIT OBJECT mode, and it shows all of the attributes of the object. The Contents of the Edit Panel varies depending one the type of object. This shows a few of those Edit Panels.

General



POLYLINE: Polyline

Comments

Every Fig object may have comments associated with it for reference.

Width	2
Depth	50
Pen color	Blue
Fill color	LtBlue
Fill style	Pattern
Fill intensity %	
Fill pattern	6
Line style	
Dash length/Dot gap	
Cap style	
Join style	

Arrows

	Forward	Backward
Thick	1.0	1.0
Width	4.0	4.0
Height	8.0	8.0

Points 4 Units: in

X0 =	2.7500	Y0 =	3.0000
X1 =	2.1250	Y1 =	2.4375
X2 =	2.3750	Y2 =	3.2500
X3 =	3.1875	Y3 =	3.8750

- *Comments*

  Every object may have comments associated with it for informational purposes. This is an ASCII text widget which may hold up to 50 lines of 200 characters each.
  The whole figure may also have comments. To edit these, click mouse button 2 (`edit Fig comments') on the canvas after choosing the Edit mode.
- *Width*

  The width of the line. (see LINE WIDTH)
- *Pen Color*

The color used to draw the outline of the object. (see PEN COLOR)

- *Fill Color*

  The color used to fill region inside the object. (see FILL COLOR)
- *Depth*

  The `*depth*' of the object. (see DEPTH)
- *Fill Style*

  The fill intensity (darkness) or fill pattern. The *Fill Intensity* is effective if *Fill Style* is *Filled*, and *Fill Pattern* is effective if *Fill Style* is *Pattern*. *Fill Intensity* is an integer value from 0 to 200; 0 for black, 100 for pure *Fill Color*, and 200 for white.

  The *Fill Pattern* is an integer value from 0 to 21. (see FILL STYLE for the actual patterns)
- *Line Style*

  The line style. (see LINE STYLE)
- *Cap Style*

  The shape of the end-points of the line. (see CAP STYLE)
- *Join Style*

  The shape of the join point (corner) of lines. (see JOIN STYLE)
- *Arrows*

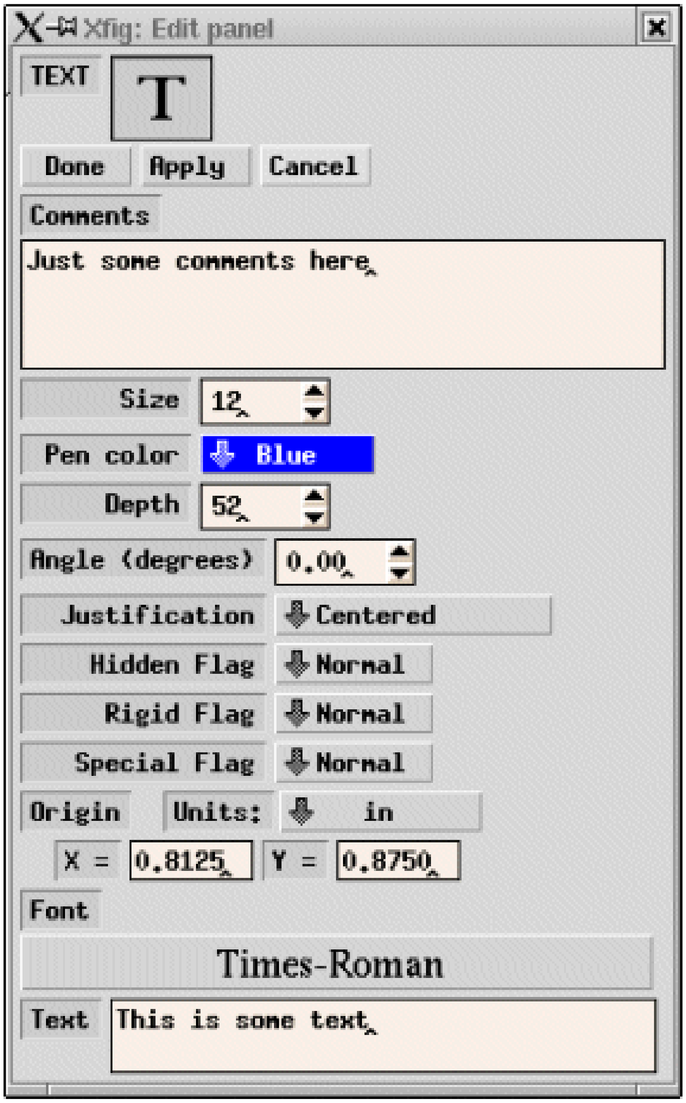
  Indicates whether the object has arrow heads at one or both ends. The shape and size of the arrow heads may be modified here. (see ARROW TYPE and ARROW SIZE)
- *Corner Radius*

  The radius of the corners for ARC-BOX objects. (see BOX CURVE)
- *Center, Radius, Points, etc.*

  These fields indicate the position and size of the object, and the coordinates of any points, in **Fig** units (1/1200 inch).

  ⓘ Note: For these entries, you may enter values followed by either ``c'' or ``i'' for centimeters or inches respectively.

TEXT object



Comments
Just some comments here

Size 12

Pen color Blue

Depth 52

Angle (degrees) 0.00

Justification Centered

Hidden Flag Normal

Rigid Flag Normal

Special Flag Normal

Origin Units: in

X = 0.8125 Y = 0.8750

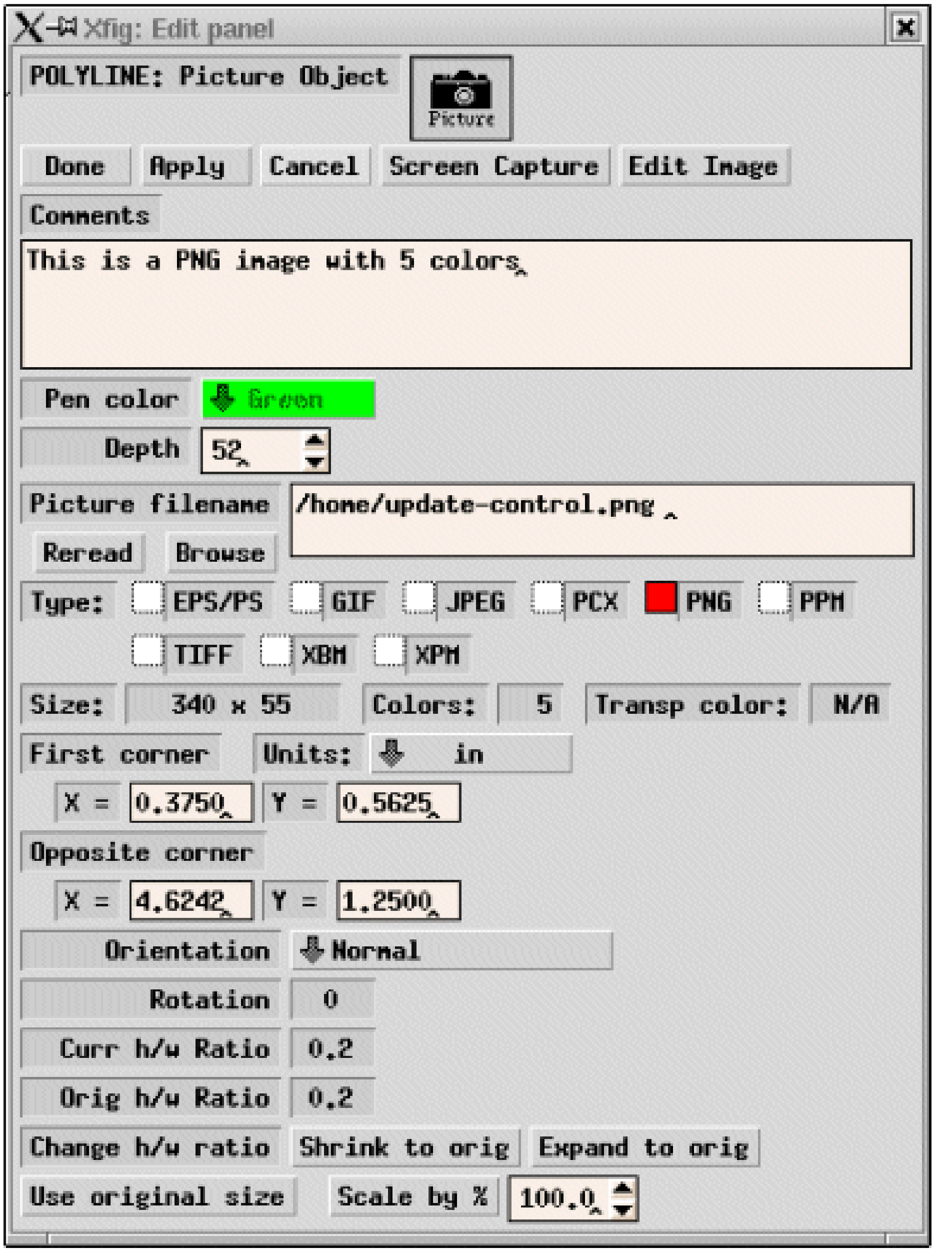
Font
Times-Roman

Text This is some text

- *Size*
  The size of the text, in points. (see TEXT SIZE)
- *Pen Color*
  The color of the text. (see PEN COLOR)
- *Depth*
  The `*depth*' of the object. (see DEPTH)
- *Angle*
  The angle of the text, in degrees. Angles > 0 are counterclockwise. (see TEXT/ELLIPSE ANGLE)
- *Justification*
  Indicates which part of the text is placed at the specified point. (see TEXT JUSTIFICATION)
- *Hidden Flag, Rigid Flag, and Special Flag*
  The state of those flags. (see TEXT FLAGS)
- *Origin*
  The position of the text.
- *Font*
  The font used for the text. The size of the font is displayed in the *Size* field. (see TEXT FONT)
- *Text*
  This field contains the text itself, and it is possible to edit the text here. Pressing *Ctrl-Return* in the text box will pop down the edit panel. It is also possible to edit text without popping up the Edit Panel; in TEXT mode,

simply click any point of the text to be edit.

PICTURE object



POLYLINE: Picture Object

Comments

This is a PNG image with 5 colors

Pen color: Green

Depth: 52

Picture filename: /home/update-control.png

Type: □ EPS/PS □ GIF □ JPEG □ PCX 🟥 PNG □ PPM □ TIFF □ XBM □ XPM

Size: 340 x 55 Colors: 5 Transp color: N/A

First corner Units: in

X = 0.3750 Y = 0.5625

Opposite corner

X = 4.6242 Y = 1.2500

Orientation: Normal

Rotation: 0

Curr h/w Ratio: 0.2

Orig h/w Ratio: 0.2

Scale by %: 100.0

- *Screen Capture*

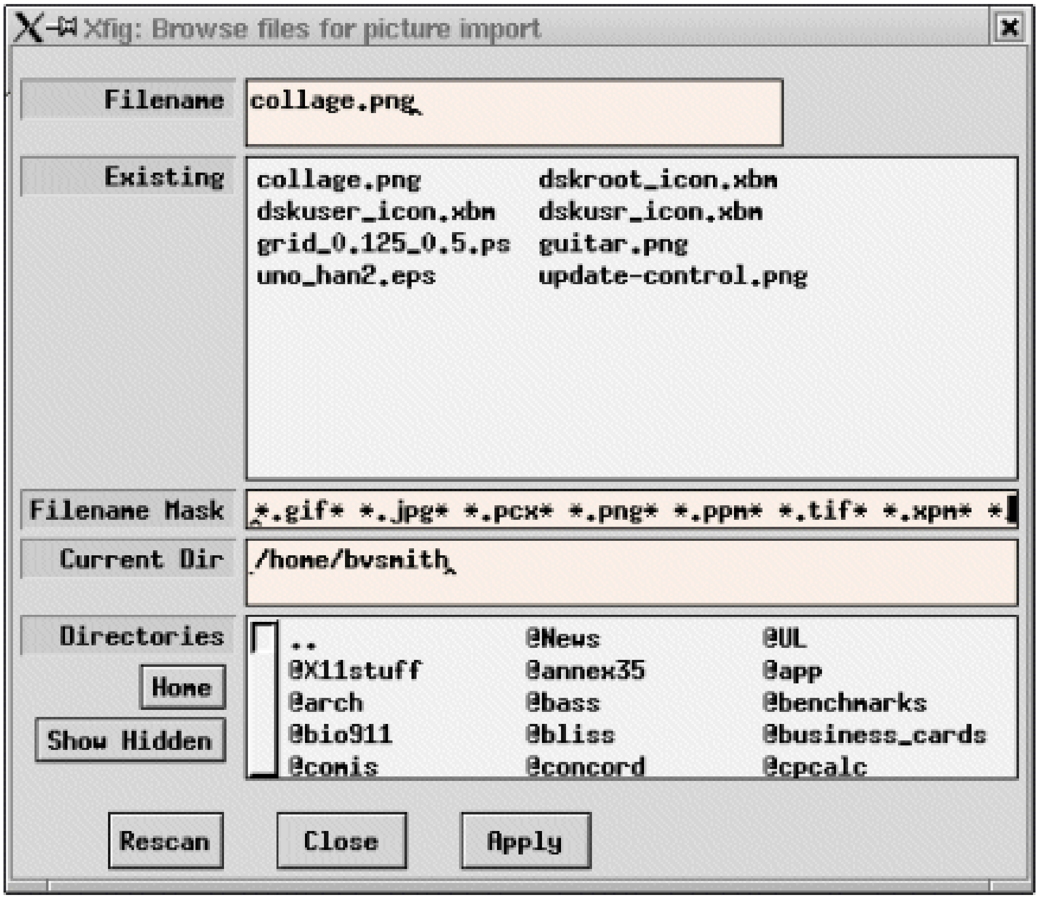
  This is used to capture an image displayed on the screen. After clicking *Screen Capture* button, you can perform one of following actions:

  1. click mouse button 1 on any window to capture the whole window
  2. click mouse button 1 on the root window (background of the screen) to capture the whole screen
  3. click mouse button 2 on one corner of a rectangle on the screen, drag the mouse to the other corner and click mouse button 2 again, to capture image in the rectangle
  4. click mouse button 3 to cancel capture operation

  During the capture operation, **xfig** unmaps all its windows from the screen. Those windows reappear on the screen when completing or cancelling the capture. The captured image will be displayed on the canvas in the picture object area. During the capture, the **xfig** window(s) will be hidden (unmapped). The captured image will be stored in a file such as `figure1_828065129.pcx`, which includes the current figure name and the time in seconds since 00:00:00 GMT, Jan 1, 1970. The file format will be PCX, and will be indicated in the picture type area.

You can't use this facility if visual of your display is TrueColor or it is deeper than 8 bits.

- *Edit Image*
  Clicking this button will invoke an external bitmap editor (the default is **xv**) to edit the image. Program that will be invoked can be checked on the Global Settings panel. *Edit Image* will invoke **xv** by default, but you can change it to invoke your favorite bitmap editor (**xpaint** or **GIMP**, for example) with the resource `Fig.image_editor` or command-line argument `-image_editor`.
- *Pen Color*
  This is effective only when the image file is XBM format and will be used to color the bits of the image.
- *Depth*
  The `*depth*' of the object. (see DEPTH)
- *Picture Filename*
  Specify the file name of the image file.
- *Type, Size, and Number of colors*
  Information such as file format or size of the image is displayed here. They may not be modified.
- *Reread*
  Clicking this button will cause **xfig** to re-read the image file. This may be necessary when the image file has been modified after the file has been read by **xfig**.
- *Browse*
  Clicking this button will popup a Browse Panel to select an image file to be imported. Click on an image file and press **Apply** to preview it, or double-click on it to choose it and close the browse panel. You may change directories in the same manner as in the file panels. The **Filename Mask** by default will show all image types that **xfig** can import.



Filename: collage.png

Existing:

```
collage.png          dskroot_icon.xbm
dskuser_icon.xbm     dskusr_icon.xbm
grid_0.125_0.5.ps    guitar.png
uno_han2.eps         update-control.png
```

Filename Mask: *.gif* *.jpg* *.pcx* *.png* *.ppm* *.tif* *.xpm* *

Current Dir: /home/bvsmith

Directories:

```
..            @News          @UL
@X11stuff     @annex35       @app
@arch         @bass          @benchmarks
@bio911       @bliss         @business_cards
@comis        @concord       @cpcalc
```

- *First Corner, and Opposite Corner*

  The position of two corner points of the rectangular region where the image is pasted.

- *Orientation*

  x and *y* coordinates of the image will be swapped if *Flipped about diag* is selected here.

- *Rotation*

  The rotation angle of the image. It may not be modified here. (see ROTATE)

- *Cur H/W Ratio and Orig H/W Ratio*

  The current and original horizontal/vertical ratio of the image.

- *Change H/W Ratio*

  These buttons may be used to set horizontal/vertical ratio of the image.

  Shrink to orig

  revert to horizontal/vertical ratio without expanding the image.

  Expand to orig

  revert to horizontal/vertical ratio without shrinking the image.

  Use original size

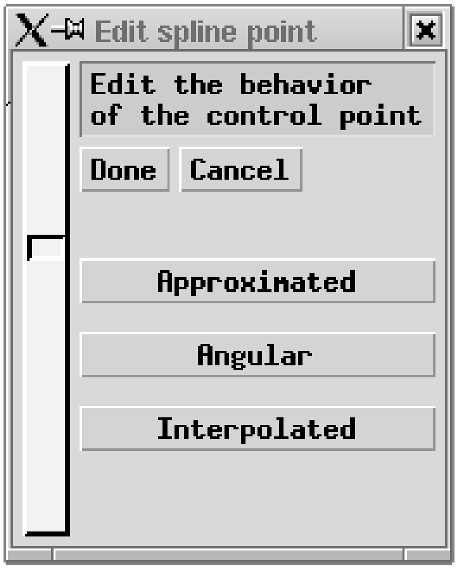
  Clicking *Use original size* will set size of the image to the original size in the picture file.

  Scale by %

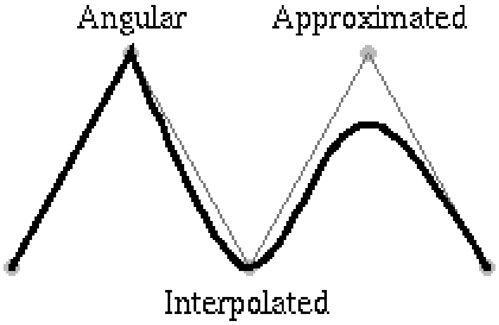
  Clicking *Scale by %* will scale the image by a percentage of the original size in the picture file.

EDIT POINT

The Edit Point panel is used to modify the *shape factor* of control points of spline curves, and it will popup by clicking on a control point of spline curve with mouse button 3 (`edit point') in EDIT OBJECT mode. The *shape factor* will control the shape of the curve.



Clicking *Approximated*, *Angular*, or *Interpolated* button will set the shape factor to a preset value. The shape factor may also be set to arbitrary values by moving the slider on the left side of the panel. The modified spline curve will be displayed interactively in either case.



The shape factor of a control point may also be modified without popping up the Edit Point panel by clicking the control point while pressing both of SHIFT key and CONTROL key. Clicking a control point with mouse button 1 (`More approx') will decrease the shape factor, and clicking a control point with mouse button 3 (`More interp') will increase the shape factor. Clicking a control point with mouse button 2 (`Cycle shape') will cycle through three preset values: Approximated, Angular, and Interpolated.

[Contents | Introduction | Credits]

Xfig User Manual

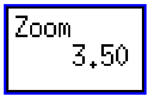
Version 3.2.4 December 2002

Object Attributes

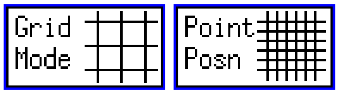
Buttons to set various drawing parameters (called *attribute buttons*), such as line width or color, are placed on the attribute panel. Buttons to set global parameters such as the zoom scale or grid mode are also placed on the attribute panel. Current settings are displayed in each attribute button.

Settings in the attribute panel are applied when creating new objects. It is also possible to modify attributes of existing objects using UPDATE or EDIT.

The first three buttons do not control object attributes but are placed here for convenience:


Zoom 3.50

Zoom (magnify) figure in canvas

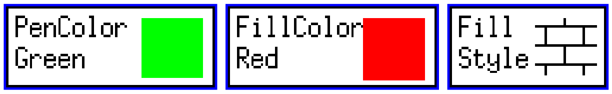


Grids

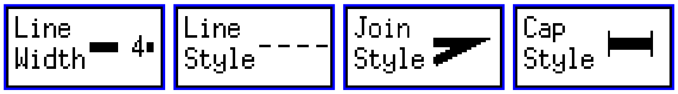
These are the attribute buttons:

Depth 100

Object depths

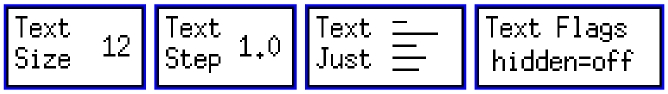

PenColor Green
FillColor Red

Colors and patterns


Line Width 4

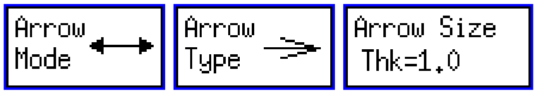
Width and shape of lines

Text Font Times-Roman


Text Size 12
Text Step 1.0

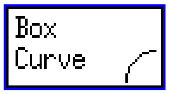
Text Flags hidden=off

Font and other attributes of TEXT objects

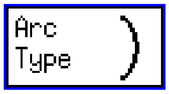


Arrow Size Thk=1.0

Arrows



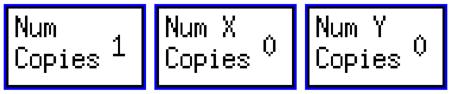
Corner radius of ARC-BOX objects



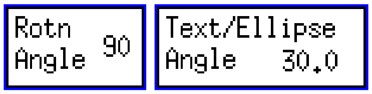
Shape of ARC objects

Num Sides 6

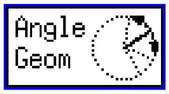
Number of sides of REGULAR POLYGON



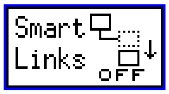
Number of copies to be created



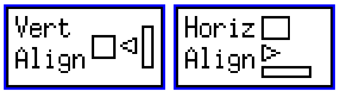
Angle of rotation



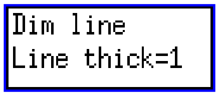
Restriction of the angle of lines



Smart Links



Alignment of objects



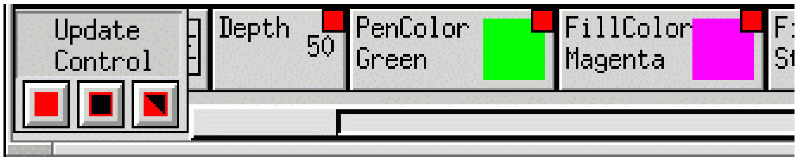
Dimension Lines

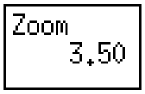
Normally, only the attribute buttons which are significant to the selected mode are placed on the attribute panel. When it is not possible to display all the attribute buttons at once, a scrollbar will be added to the attribute panel.

These settings may be changed by clicking those attribute buttons. Each mouse button has the following function:

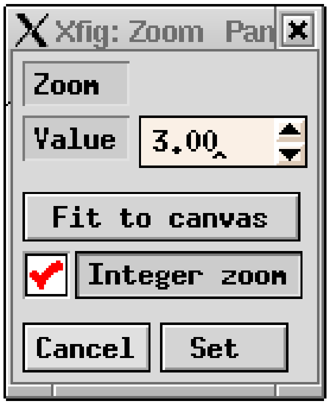
- Mouse button 1 (`Menu')
  Clicking an attribute button with mouse button 1 will popup a panel to change setting of the attribute button, and the user may select a new value by clicking any button on the panel (in the case of *grid mode* or *text font*, etc.) or entering a new value from the keyboard (in the case of *zoom scale* or *line width*, etc.)
- Mouse button 2 (`Dec/Prev')
  If the attribute button contains a numeric value (*zoom scale* or *line width*, for example), clicking the attribute button with mouse button 2 will decrease the value. If the attribute button is of another type (*grid mode* or *text font*, for example), the selection will be chosen from the previous item in the list. The list is considered circular.
- Mouse button 3 (`Inc/Next')
  This mouse button is the reverse of mouse button 2. If the attribute button contains a numeric value, clicking the attribute button with mouse button 3 will increase the value. If the attribute button is of another type, the selection will be chosen from the next item in the list. The list is considered circular.

In UPDATE mode, a small toggle button appears in the upper-right corner of each attribute button. This controls whether the particular attribute should be applied when using the UPDATE mode. Also, an Update Control Button will be placed at the left of the attribute panel.





(ZOOM SCALE)



Value 3.00

☑ Integer zoom

Set the zoom scale of display on the canvas. The zoom scale may be set in the range of 0.01 to 50. On many systems, the display on the screen will be almost actual-size when this is set to 1.

Clicking this button with mouse button 1 will popup a panel to enter new value of zoom scale from keyboard. Clicking this button with mouse button 2 or mouse button 3 will decrease or increase the zoom scale by a factor of 1.5 respectively if the zoom is > 1, or by 0.1 if zoom scale is less than 1.0 and 0.01 if the zoom is less than 0.1.

Zooming of the canvas may also be performed by specifying the rectangle to be zoomed on the canvas (`area zoom'). To do this, specify the rectangle to be zoomed by clicking mouse button 1 (`zoom area') while pressing CONTROL key and specify one corner of the rectangle, and then clicking mouse button 1 (`final point') again at the opposite corner of the rectangle.

If xfig was compiled with Wheel mouse support (WHEELMOUSE in the Imakefile) then rotating the wheel on the canvas while the **Shift** key is pressed will zoom/unzoom the canvas around the pointer and rotating the wheel while the **Control** key is pressed will scroll (pan) the canvas up or down.

It is also possible to set zoom scale to 1 by clicking mouse button 3 (`unzoom') while pressing CONTROL key on the canvas.

To force the area-zoom to zoom using integer values check the box in the zoom popup panel that says *Integer zoom*.

To zoom the figure so that it just fills the canvas, press the *Fit to canvas* button. *Integer zoom* is ignored if the zoom needed to fit the figure on the canvas is less than 1.0.

See also Zooming.

(DEPTH)

Set the `*depth*' of objects.

The *Depth* is used to decide which object should be hidden when objects overlap. The object with larger *depth* will be hidden by object with smaller *depth*. *Depth* is an integer value of the range 0 to 999.

If objects with the same *depth* overlap, it is unknown which object will be hidden, and the display on the screen may be different from a printout or export.

Normally, all depths are displayed on the canvas, but the user may control which depths are displayed and editable. See the Depth Panel section.

If the depth of a compound object is changed with UPDATE, the relative depths of the objects contained will be maintained. For example, consider a compound object containing three objects at depths 3, 8 and 11 respectively. If the depth of the compound is changed to 2, then the objects contained will have depths of 2, 7 and 10 respectively.

Grid Mode (GRID MODE)

Set the size of the grid displayed on the canvas. If this is set to *None*, no grid will be displayed.

The size of the grid may be selected from 1/8inch, 1/4inch, 1/2inch and 1inch if unit is set to *Imperial (inches)*, and from 2mm, 5mm, 10mm and 20mm if unit is set to *Metric*.

This grid will not restrict positioning of objects. If you want to set a restriction of positioning of objects, use POINT POSITION.

Point Posn (POINT POSITION)

Set the size of virtual grid (it is not visible) to control the restriction of positioning of objects. If this is not set to *Any*, objects may only be placed on the virtual grid of the selected size. If this is set to *Any*, objects can be placed anywhere.

The size of the virtual grid may be selected from 1/16inch, 1/8inch, 1/4inch, 1/2inch and 1inch if unit is set to *Imperial (inches)*, and from 1mm, 2mm, 5mm, 10mm and 20mm if unit is set to *Metric*.

This will also restrict which objects may be `picked up' when editing. If an object is not on the virtual grid specified here, the object can't picked up.

This virtual grid will not displayed on the canvas. The visible grid can be set by GRID MODE.

Rotn Angle 90 (ROTATION ANGLE)

Set the angle for rotating objects with ROTATE. Clicking this button with mouse button 2 or mouse button 3 will decrease or increase the angle by 15 degrees respectively. Clicking this button with mouse button 1 will popup a panel to enter the angle.

The angle of TEXT or ELLIPSE objects may also be set by TEXT / ELLIPSE ANGLE.

Note that some objects may only be rotated by 90 degrees (e.g. BOX)

(TEXT / ELLIPSE ANGLE)

Set the angle of TEXT or ELLIPSE objects. Positive values indicate counter-clockwise rotation, and negative values indicate clockwise rotation.

Clicking this button with mouse button 2 or mouse button 3 will decrease or increase angle by 15 degrees respectively. Clicking this button with mouse button 1 will popup a panel to enter the angle.

To rotate exisiting objects, use ROTATE. See also ROTATION ANGLE.

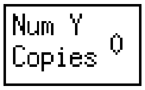
(NUMBER OF POLYGON SIDES)

Set the number of sides used in creating a regular polygon with REGULAR POLYGON.

(NUMBER OF COPIES)

Set the number of copies to be made with *COPY & ROTATE* (see ROTATE).

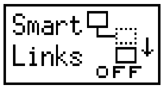
The number of copies to be made with *ARRAY PLACEMENT* should be set with NUMBER OF X COPIES and NUMBER OF Y COPIES.



(NUMBER OF X COPIES / NUMBER OF Y COPIES)

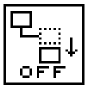
Set the number of copies to be made in the X and Y direction when copying objects with *ARRAY PLACEMENT* (see COPY).

The number of copies to be made with *COPY & ROTATE* should be set with NUMBER OF COPIES.



(SMART-LINKS MODE)

This button controls the smart-link mode. If the smart-link mode is enabled, lines with an end point on the perimeter of objects such as a BOX or inside a COMPOUND object (henceforth called *links*) will be treated specially when those objects are moved or copied.



OFF

Disable the smart-link mode. When moving or copying an object, only the object will moved or copied.

MOVE

When moving an object, the end point of the *link* will be moved with the object so that the *link* will remain linked. When copying an object, the *link* is also copied.

SLIDE

When moving an object, the end segment of the *link* (the segment which touches the object) will slide so that the angle of the segment is maintained. When copying an object, the *link* is also copied.

Lines which touch objects such as circles or ellipses are not treated as *links*. If you want to use smart-link facility for such lines, make the object into a COMPOUND object.

(VERTICAL ALIGN)

Set the vertical alignment mode for ALIGN.

Do not align vertically.

Align to top of objects.

Align center of objects.

Align to bottom of objects.

Vertically distribute objects so that the distance between center of each object is equal.

Vertically distribute objects so that the distance between the edge of each object is equal.

Vertically distribute objects so that the edge of each object touch.

(HORIZONTAL ALIGN)

Set the horizontal alignment mode for ALIGN

Do not align horizontally.

Align to left of objects.

Align center of objects.

Align to right of objects.

Horizontally distribute objects so that the distance between center of each object is equal.

Horizontally distribute objects so that the distance between the edge of each object is equal.

Horizontally distribute objects so that the edge of each object touch.

Angle Geom (ANGLE GEOMETRY)

Set the restriction of the angle of lines when creating POLYLINE, POLYGON, SPLINE objects, etc. This will also restrict the angle of line segment when moving its end point with MOVE POINT.

Unrestricted
Don't restrict the angle of lines. With this setting, lines of any angle may be created. This is the default.

LaTeX Line
Allow lines to be drawn only at slopes which can be handled by the *line* command of the LaTeX picture environment. With this setting, lines with slope *x* / *y* (here, *x* and *y* are integers in the range -6 to 6) may be created.

LaTeX Vector
Allow lines to be drawn only at slopes which can be handled by the *vector* command of the LaTeX picture environment. With this setting, lines with slope *x* / *y* (here, *x* and *y* are integers in the range -4 to 4) may be created.

Manhattan-Mountain
Allow lines to be drawn in the horizontal, vertical or diagonal (45 degrees) direction only.

Manhattan
Allow lines to be drawn in the horizontal or vertical direction only.

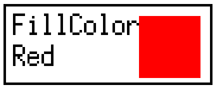
Mountain
Allow lines to be drawn in the diagonal (45 degrees) direction only.

PenColor Green 🟩 (PEN COLOR)

Set the color used to draw objects. The color may be selected from 32 standard colors and other arbitrary colors defined by user. See Colors Panel for more information.

Clicking this button with mouse button 1 will popup the Colors Panel.

The color to fill the region inside object is set by FILL COLOR and FILL STYLE. If you want to fill the region without drawing the outline of the object, set LINE WIDTH to zero.

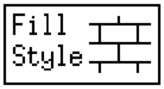


(FILL COLOR)

Set the color used to fill the region inside objects. As in PEN COLOR, the color may be selected from 32 standard colors and other arbitrary colors defined by user. See Colors Panel for more information.

Clicking this button with mouse button 1 will popup the Colors Panel.

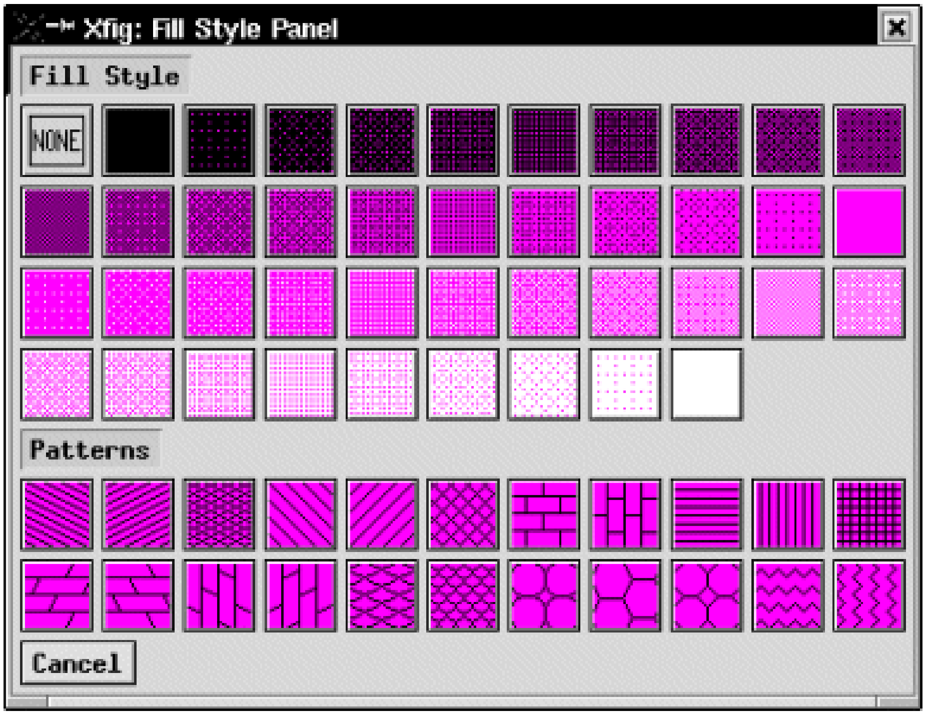
The color to draw the outline of object is set by PEN COLOR. The FILL COLOR is only effective when the FILL STYLE is set to other than *None*. If you don't want to fill the region inside objects, set the FILL STYLE to *None*.

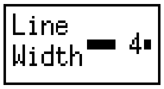


(FILL STYLE)

Set fill darkness or fill pattern. Fill color is set by FILL COLOR. If *None* is selected here, the region inside object will not be filled.

Clicking this button with mouse button 1 will popup this panel:

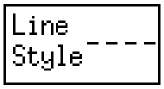




(LINE WIDTH)

Set the width of lines to draw the outline of objects. The unit of the line width is exactly 1 screen pixel (about 1/80 inch).

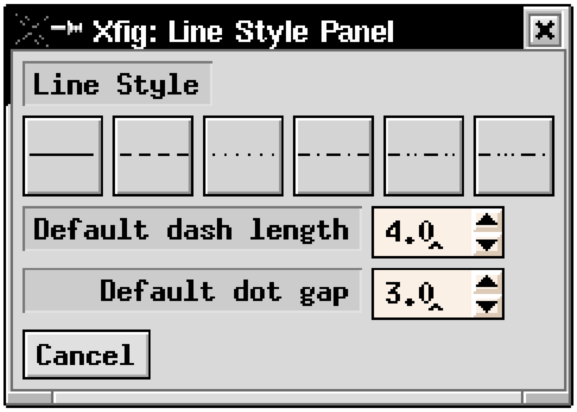
If this is set to zero, the outline of objects will not be drawn. The area fill will still be used if area fill is enabled by FILL STYLE.



(LINE STYLE)

Set the line style from *solid*, *dashed*, *dotted*, *dash-dot*, *dash-dot-dot* and *dash-dot-dot-dot*. It is also possible to set the dash length or dot distance. The shape (ends) of each dash and dot may be set by CAP STYLE.

Clicking this button with mouse button 1 will popup this panel:



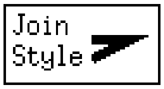
Xfig: Line Style Panel

Line Style

Default dash length 4.0

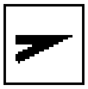
Default dot gap 3.0

Cancel



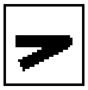
(JOIN STYLE)

Set the shape of the join point (vertex) of lines of objects such as POLYLINE or POLYGON. This setting does not have a remarkable effect when the line is very thin.



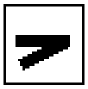
MITER

Extend the outer edge of the lines until they touch. Automatically uses *BEVEL* instead if the angle is too acute. This is the default.



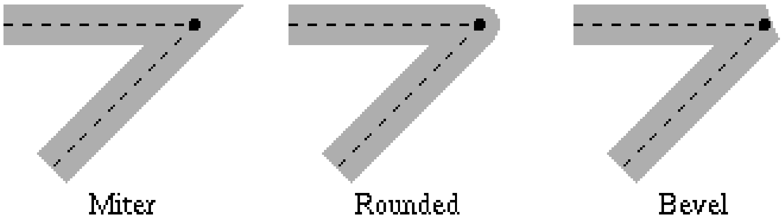
ROUNDED

Join lines with an arc whose center is at the join point and whose diameter is equal to the line width.



BEVEL

Join the lines with *butt* end caps and fill the resulting triangular notch at the join position.



The shape of the end points of lines will set by CAP STYLE.

Cap Style (CAP STYLE)

Set the shape of end points of lines of objects such as POLYLINE. This setting does not have a remarkable effect when the line is very thin.

BUTT

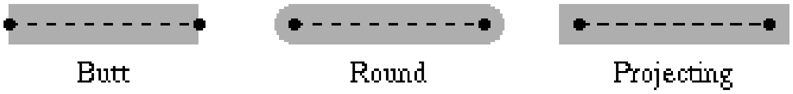
The ends of the line don't extend beyond the end points. This is the default.

ROUND

Terminate lines with a circle whose diameter is equal to the line width.

PROJECTING

Similar to *BUTT*, but the ends of the line extend by half of line width beyond the end points.



This setting also affects the shape of dashes or dots of dashed-lines or dotted-lines. The shape of the join point (vertex) of lines is set by JOIN STYLE.

Arrow Mode (ARROW MODE)

Determines whether an arrow head should be automatically added at either or both ends of a line when creating a POLYLINE, OPEN ARC, or OPEN SPLINE object.

Do not add any arrow heads.

Add an arrow head at the last point of the line.

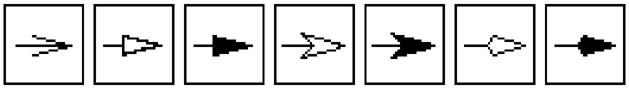
↔ Add an arrow head at each end of the line.

← Add an arrow head at the first point of the line.

The shape of the arrow head may be set by ARROW TYPE and the size by ARROW SIZE. It is also possible to add (or remove) arrow heads to existing objects using ADD / DELETE ARROW.

Arrow Type (ARROW TYPE)

Sets the shape of arrow heads.

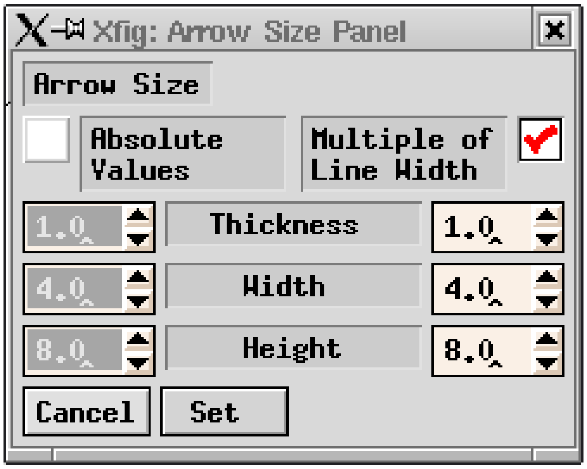


The size of arrow head may be set using ARROW SIZE.

Arrow Size Thk=1.0 (ARROW SIZE)

Set the size of arrow head.

Clicking this button with mouse button 1 will popup this panel:



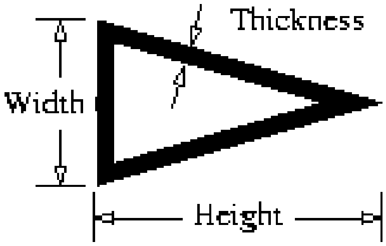
Xfig: Arrow Size Panel

Arrow Size

□ Absolute Values

Multiple of Line Width ☑

1.0	Thickness	1.0
4.0	Width	4.0
8.0	Height	8.0

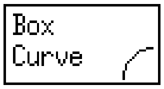


The *Thickness* controls the thickness of the line forming the arrow head.
The *Width* controls the width at the back of the arrow head.
The *Height* controls the length or height of the arrow head.

All these units are the same as for LINE WIDTH. This is a change from previous versions of **xfig**, where the width and height were in *Fig* units (1/1200 inch).

- If the *Absolute Values* box is checked, the values from the left side entries are used and are absolute as just described.

- If the *Multiple of Line Width* box is checked, the respective value from the right side entry is multiplied by the current line width setting to arrive at the sizes.

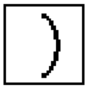


(BOX CURVE)

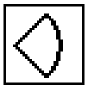
Set the corner radius of ARC-BOX objects. The units are the same as for LINE WIDTH.

(ARC TYPE)

Set the style of ARC object.


OPEN ARC

Draw only the arc.


PIE-WEDGE

Adding to the arc, draw lines which connect each end of the arc to the center.

(TEXT FONT)

Set the font used to draw TEXT objects. The size of the font may be set by TEXT SIZE.

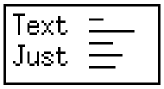
Clicking this button with mouse button 1 will popup the Font Panel to select the font.

(TEXT SIZE)

Set the size (point size) of TEXT object. The default is 12 points. The font itself is set by TEXT FONT.

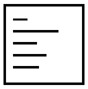
(TEXT STEP)

Set the vertical distance by which lines of text will be spaced when entering multiple lines in TEXT mode. The distance is the product of the height of the font (approximately TEXT SIZE) and this value.

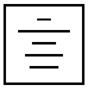


(TEXT JUSTIFICATION)

Set which part of the text should be placed at the specified point when entering TEXT object.


LEFT JUSTIFIED

Place the left edge of the text at the specified point. This is the default.


CENTERED

Place the center of the text at the specified point.

RIGHT JUSTIFIED

Place the right edge of the text at the specified point.

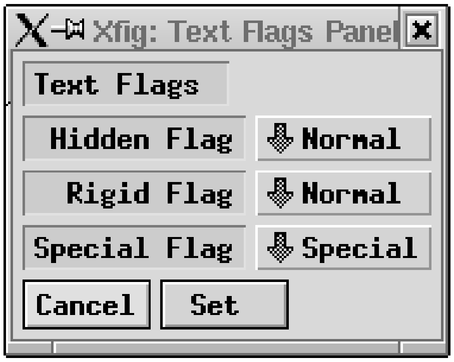
Text Flags
hidden=off

(TEXT FLAGS)

Set the *Hidden*, *Rigid* and *Special* flags of TEXT objects. These flags are used for special situations, and they are all *OFF* by default.

Clicking this button with mouse button 2 or mouse button 3 will switch which flag's state should be displayed on the button.

Clicking this button with mouse button 1 will popup this panel:



Xfig: Text Flags Panel

Text Flags

Hidden Flag: Normal

Rigid Flag: Normal

Special Flag: Special

If the *Hidden* flag is set *ON*, the string `<<>>' is displayed on the canvas instead of the text itself. The text will output as usual when Printing or Exporting. This is useful to avoid text with long sequences of LaTeX commands which makes the display hard to read.

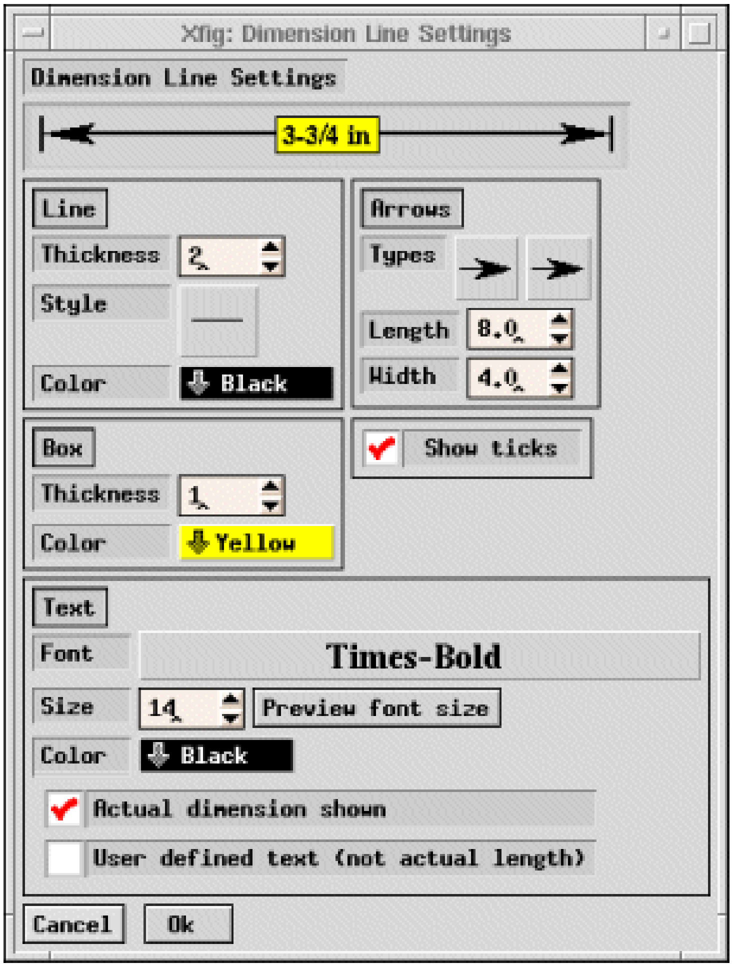
If the *Rigid* flag is *ON*, the size of the font of the TEXT object will not be changed when a COMPOUND object which includes the TEXT object is scaled.

If the *Special* flag is *ON*, special characters (such as `\', for example) in text will not be specially processed but passed to the output `as is' when Exporting. This may be used to put LaTeX commands in text. The use of LaTeX commands is especially helpful to put complex mathematical formulas in the figure, for example. If the *Special* flag is *OFF*, the text will be processed before output.

Dim line
Line thick=1

Dimension Lines

Set the attributes of dimension lines. One may have end-ticks on the lines, arrowheads, box around the text with or without a border, and the actual length of the line for the text or a user-defined text.



Xfig: Dimension Line Settings

Dimension Line Settings

3-3/4 in

Line

Thickness 2

Style

Color Black

Arrows

Types

Length 8.0

Width 4.0

☑ Show ticks

Box

Thickness 1

Color Yellow

Text

Font Times-Bold

Size 14

Color Black

☑ Actual dimension shown

□ User defined text (not actual length)

As you make changes to the settings, the image at the top of the panel shows what the dimension line would look like.

- *Line - This is the main line*
  - *Thickness*

    This is the thickness of the main dimension line. It may be 0, in which case no line is drawn.
  - *Style*

    The style of the line (solid, dash, etc.). The standard line style pulldown menu appears when this is pressed.
  - *Color*

    The color of the line. A color panel pops up when this is pressed, to choose the color.
- *Arrows*
  - *Types*

    When pressed a pulldown menu appears from which the user may select the arrowhead type (or none) for each end of the line.
  - *Length*

    The length of the arrowheads as a multiple of the main line thickness
  - *Width*

    The width of the arrowheads as a multiple of the main line thickness
- *Box - This is the box drawn around the dimension text*
  - *Thickness*

    The thickness of the line drawn around the box
  - *Color*

    The color used to fill the box. A color panel pops up.

- *Show ticks*

  Checkbox to show or hide end ticks on main line.
- *Text - The text showing the length of the line*
  - *Font*

    When pressed the standard font menu pops up to choose the font.
  - *Size*

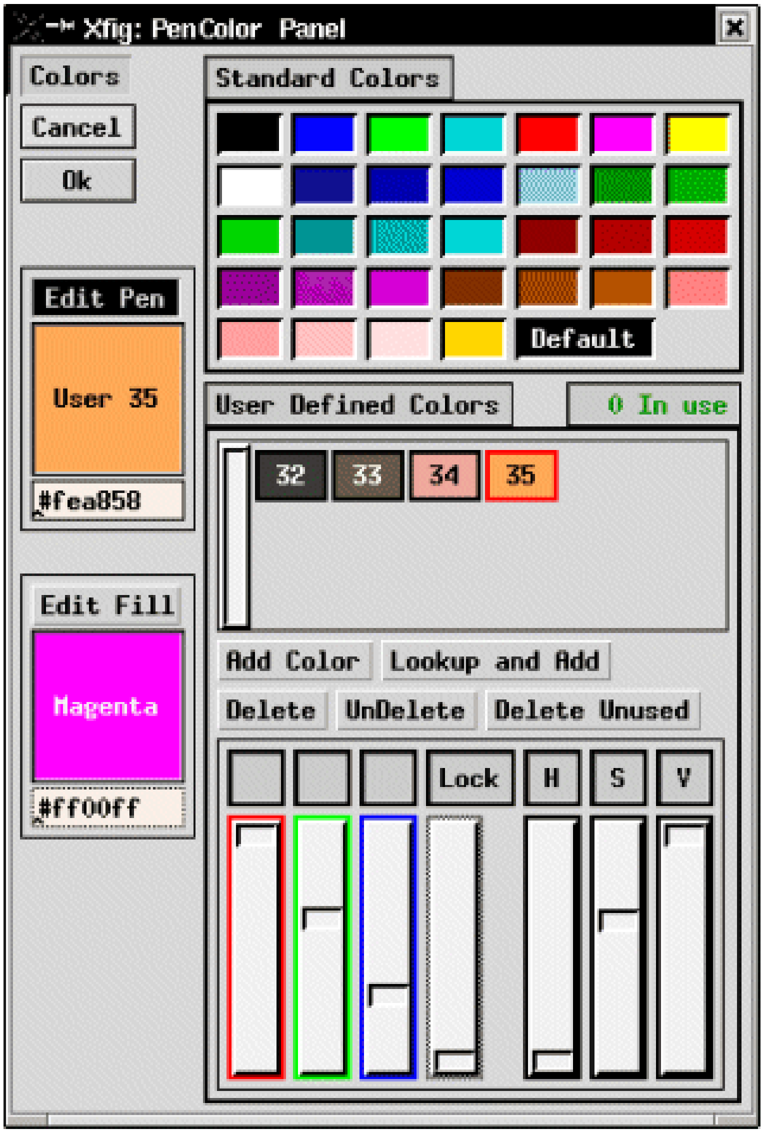
    The font size in points.
  - *Color*

    Text color popup menu.
  - *Text type*

    Checkboxes - show actual dimension of line even when the line is resized, or show user-entered text.

Colors Panel

The Colors Panel is used to choose or define colors for the outline of objects or the fill area inside objects, and it will popup when either PEN COLOR button or FILL COLOR button is clicked.



Edit Pen: User 35, #fea858

Edit Fill: Magenta, #ff00ff

0 In use

- *Edit Pen*

  If this button is selected, the PEN COLOR (color to draw outline of object) may be selected. If this panel is popped up by clicking PEN COLOR button, this button is selected initially.

- *Edit Fill*

  If this button is selected, the FILL COLOR (color to fill region inside of object) may be selected. If this panel is popped up by clicking FILL COLOR button, this button is selected initially.

- *Standard Colors*

  This is a panel of 32 standard color buttons. Any of these colors may be selected by clicking on the color. The standard colors are black, yellow, white, gold, five shade of blue, four shade of green, four shade of cyan, four shade of red, five shade of magenta, four shade of brown, and four shade of pink. Additional colors may be defined by *User Defined Colors*.

- *User Defined Colors*
  - *Add Color*

    This button will add a new user-defined color in the box under *User Defined Colors*. Its initial color is black. The color sliders may be used to change the color.
  - *Lookup Color*

    This button will also add new user-defined color, by getting a color from the display. After clicking *Lookup Color*, click on any color on the display and that color will be used for the new user-defined color.
  - *Delete*

    This button will delete a user-defined color. A user-defined color may be deleted by clicking this button after selecting a color cell. However, the color may not be deleted if the color is in use by any objects in the current figure.
  - *UnDelete*

    This button will undo the last delete of user-defined color.
  - *Delete Unused*

    This button will delete any unused user-defined colors. Notice that the number of colors actually in use are displayed next to the "User Defined Colors" title. Also, the borders of the colors in use are drawn in green, while the borders of unused colors are black. The border of the selected color is red.
  - *RGB and HSV Sliders*

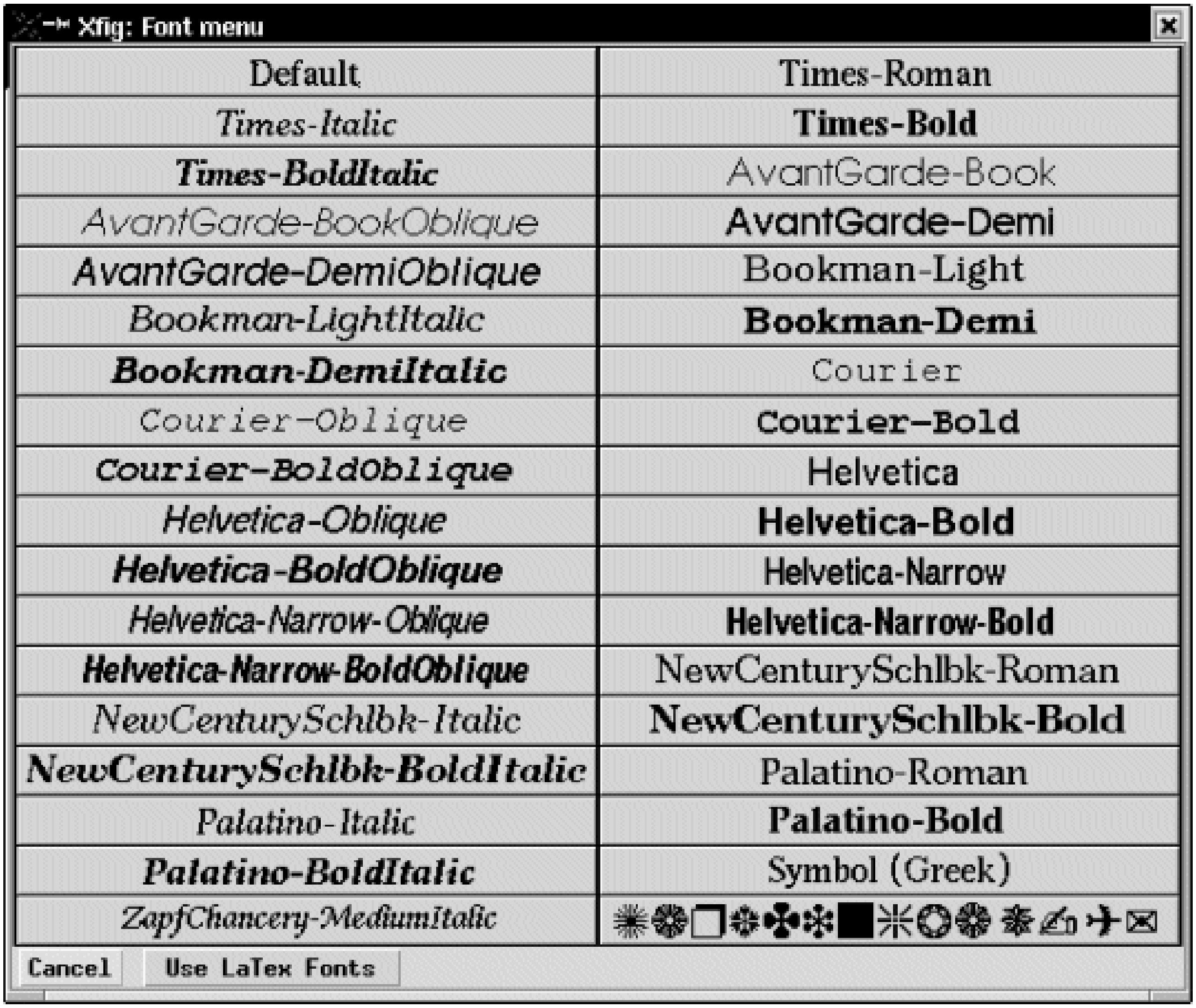
    The user-defined colors may be modified using RGB (red/green/blue) or HSV (hue/saturation/value) sliders. It is also possible to enter hexadecimal values to indicate the intensity of red, green and blue (each value is 00 to FF) into the fields under *Edit Pen* and *Edit Fill*.

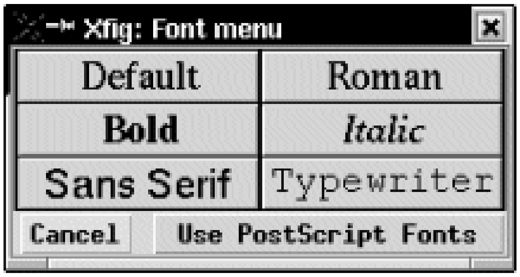
    If two or three of the buttons above the red, green and blue sliders are checked (the figure shows the red and green locked), those sliders will be `locked' together, so that moving the *Lock* slider will move those sliders together.

Font Panel

The Font Panel is used to select the font to draw text, and it will popup when the TEXT FONT button is clicked. Only the font name may be selected here; the size of the font must be set using TEXT SIZE.

There are two Font Panels; one for PostScript fonts and one for LaTeX fonts.

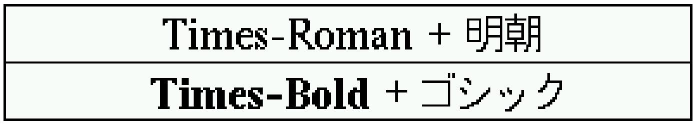




Clicking a font button (the labels of the buttons are drawn using the font) will select that font. Clicking the *Use LaTeX Fonts* button will switch to the panel for LaTeX fonts, and clicking *Use PostScript Fonts* button will switch to the panel for PostScript fonts.

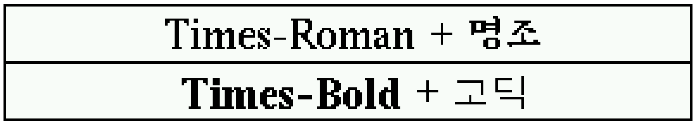
It is not guaranteed that all fonts in the panel of PostScript fonts are available on all PostScript printers. Also, different fonts may be used on the screen. This is dependent on which fonts the X server has. But at least *Times-Roman*, *Times-Italic*, *Times-Bold*, *Times-BoldItalic*, *Courier*, *Courier-Oblique*, *Courier-Bold*, *Courier-BoldOblique*, *Helvetica*, *Helvetica-Oblique*, *Helvetica-Bold* and *Helvetica-BoldOblique* will be available in most environments.

i18n In international-**xfig**, ``*Times-Roman*'' and ``*Times-Bold*'' will be replaced with



Times-Roman + 明朝
Times-Bold + ゴシック

in the Japanese environment, and



Times-Roman + 명조
Times-Bold + 고딕

in the Korean environment. It is possible to put Japanese or Korean in text when these fonts are selected. See Internationalization about this.

[Contents | Introduction | Credits]

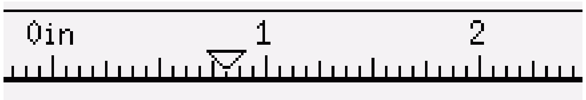
Xfig User Manual

Version 3.2.4 December 2002

Panning, Zoom, Ruler and Units

Panning the Canvas

Rulers



The rulers are placed at the top and right (left, if the drawing mode panel is *right*) of the canvas, and the scale selected by Units is displayed on them (i.e. inches or cm). Normally, triangular markers which show the mouse cursor position are also displayed on the rulers. The rulers may also be used to scroll the canvas.

To scroll the canvas, press mouse button 2 (`Drag') on the ruler and move cursor left/right or up/down;

the ruler will be moved following the cursor, and the contents of canvas will be moved when mouse button is released. If the SHIFT key was pressed during this operation, the canvas will scroll five times further than the mouse movement.

Clicking on the ruler with mouse button 1 (`Pan Left' or `Pan Up') or mouse button 3 (`Pan Right' or `Pan Down') will scroll the canvas by 1/2 inch (1cm, if unit is cm). The arrow keys have the same effect, too. If the SHIFT key was pressed during those operations, amount of scroll will be five times further.

NEW If **xfig** is compiled with wheel mouse support and you have a wheel-mouse, then rotating the wheel **backward** on the canvas will scroll the canvas up, and rotating the wheel **forward** will scroll the canvas down.

Panning with Keyboard

The canvas may be panned **left**, **right**, **up**, or **down** by pressing the respective arrow keys while the mouse pointer is in the canvas. If the SHIFT key was pressed during those operations, amount of scroll will be five times further.

Panning may be done by clicking rulers with mouse button 1 or 3, too.

Pan to the Origin

The canvas view will be moved to the origin if mouse button 1 (`Pan to Origin') is clicked on the UNIT box, or mouse button 2 (`Pan to Origin') is clicked on the canvas while the CONTROL key is down. Selecting *Pan to origin* from the View menu will achieve the same action.

Negative Coordinate

By default, **xfig** will allow panning into negative coordinates.

This may be turned off by either starting **xfig** with the `-dontallownegcoords` option, or unchecking the box in the Global Settings panel.

Zooming

The canvas may be zoomed/unzoomed by setting the zoom scale with the Zoom Scale button in the Attribute panel. Here are description of some shortcuts to zooming and unzooming

quickly.

Zoom to Fit Canvas

- To zoom the figure just fill the canvas, choose *Zoom to Fit canvas* from the View menu, or use accelerator `Ctrl-Z`.

- *Fit to canvas* button on the Zoom dialog has the same function.

Zoom In

- Click Zoom Scale in the Attribute panel with mouse button 3 (`Inc/Next').

- Choose *Zoom in* from the View menu, or use accelerator `Shift-Z`.

- While holding the **control key** down click mouse button 1 (`zoom area') on one corner of a rectangle, drag the mouse to the opposite corner and click mouse button 1 (`final point') again. The area in the rectangle will be zoomed to fit the canvas.

- NEW If **xfig** is compiled with wheel mouse support and you have a wheel-mouse, then rotating the wheel **backward** on the canvas while the **control key** is pressed will zoom the canvas in around the pointer.

Zoom Out

- Click Zoom Scale in the Attribute panel with mouse button 2 (`Dec/Prev').

- Choose *Zoom out* from the View menu, or use accelerator `Z`.

- NEW If **xfig** is compiled with wheel mouse support and you have a wheel-mouse, then rotating the wheel **forward** on the canvas while the **control key** is pressed will zoom the canvas out.

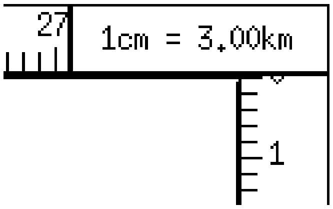
Unzoom

- Choose *Unzoom* from the View menu.

- Click mouse button 3 (`Unzoom') on the canvas while pressing the CONTROL key.

Units

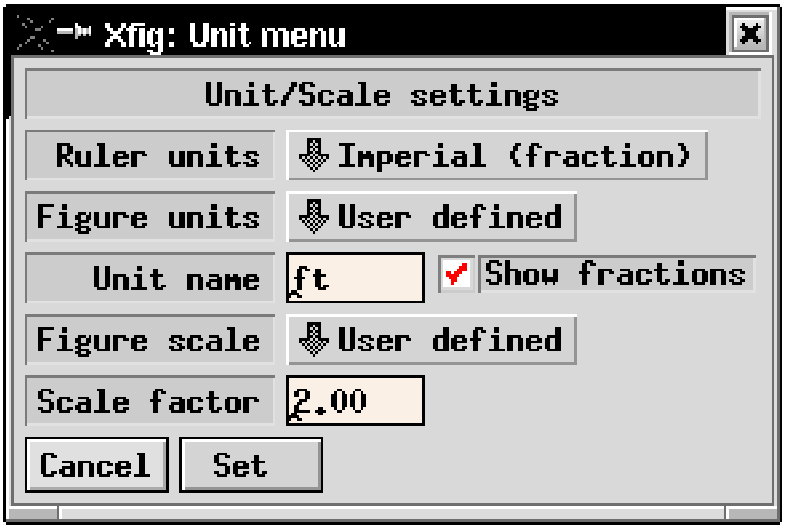
UNIT box



The Units box is placed where the top and side rulers meet. The units of the ruler (*in* or *cm*) and the scale of the drawing (e.g. 1cm=3km) will be displayed here.

Set Units

Clicking on the Units box with mouse button 3 (`Set Units/Scale') will pop up a panel to set the units. This panel is also available from the Edit Menu



Ruler units: Imperial (fraction)
Figure units: User defined
Unit name: ft
☑ Show fractions
Figure scale: User defined
Scale factor: 2.00

- *Ruler units*

  Select ruler unit from *Imperial (fraction)*, *Imperial (decimal)* or *Metric (cm)*. When *Imperial (fraction)* is chosen, then fractions of the units will be shown in measurement messages, e.g. **3-5/8 in**, or **3 ft 7-1/2 in** For the other choices, decimal values are shown, e.g. **3.3242 cm**.
- *Figure units*

  Select unit which is used to display dimensions when entering or editing object, from *Ruler units* and *User defined*. If *User defined* is selected here, the string entered in *Unit name* will be displayed as the unit of dimension.
- *Unit name*

Set the name of the unit. This is effective only if *User defined* is selected in *Figure units*.

- *Use fractions* NEW

  If the Ruler units is Imperial (in), you may have **xfig** display measurements using fractions where possible. If a measurement can be shown as an integer multiple of 64ths of an inch (or 32nds, 16ths, etc.) it will show it as such (e.g. 3-5/64 in), otherwise a decimal value will be shown (e.g. 3.085 in).

- *Figure scale*

  Choose the scale which is used to display dimensions when entering or editing objects, from *Unity* and *User defined*. If *User defined* is selected here, the value specified by *Scale factor* will be multiplied by the dimensions displayed when drawing objects.

- *Scale factor*

  Set the value which will be multiplied by the dimensions when displaying them. This is effective only if *User defined* is selected as *Figure scale*.

As an exmple, in the above panel *Ruler units* are *Imperial (fraction)*, *Figure units* are *User defined*, *Unit name* is "ft" (feet), *Figure scale* is *User defined* and *Scale factor* is 2.0, so when drawing, 1 inch will be reported as 2 ft., and 1-7/16 inch will be reported as 2 ft 10-1/2 in.

The default values for this panel can be set by command line options and resources.

Xfig User Manual

Version 3.2.4 December 2002

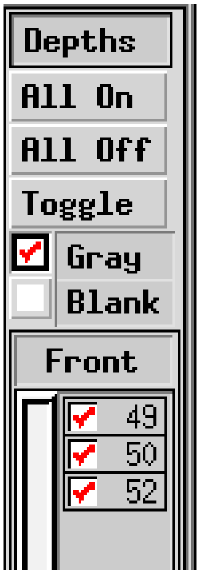
Layers (Depths)

Overview

Xfig uses *depths* (a simple form of layers) to determine the drawing order of overlapping objects. The depth is one of the attributes of an object, just as color is. If you have two or more overlapping objects and the order in which they are drawn is important you must give each a different depth value.

Depth 0 is the top of the drawing order, with increasing depths being underneath objects with smaller depth. The maximum depth is 999.

Depth Panel



While the depth feature has been a part of **xfig** for a long time, there is now a panel to the right of the canvas which allows the user to enable or disable editing and displaying depth levels. This panel may be hidden or shown by selecting *Show depth manager* from the View menu.

Clicking on the checkbox area of a depth indicator will turn on or off the display and editing of objects at that depth. By clicking on one checkbox and holding down the mouse button while dragging the mouse up or down over other depth checkboxes, multiple depths may be enabled or disabled at one time.

If you click on a checkbox that is ***ON*** and drag the mouse over other checkboxes, they will be turned ***OFF*** regardless of their initial state. Conversely, if you click on a checkbox that is ***OFF*** and drag the mouse over other checkboxes, they will be turned ***ON***.

Controls

- All On

  Pressing this button will turn **on** all depths.

- All Off

  Pressing this button will turn **off** all depths.
- Toggle

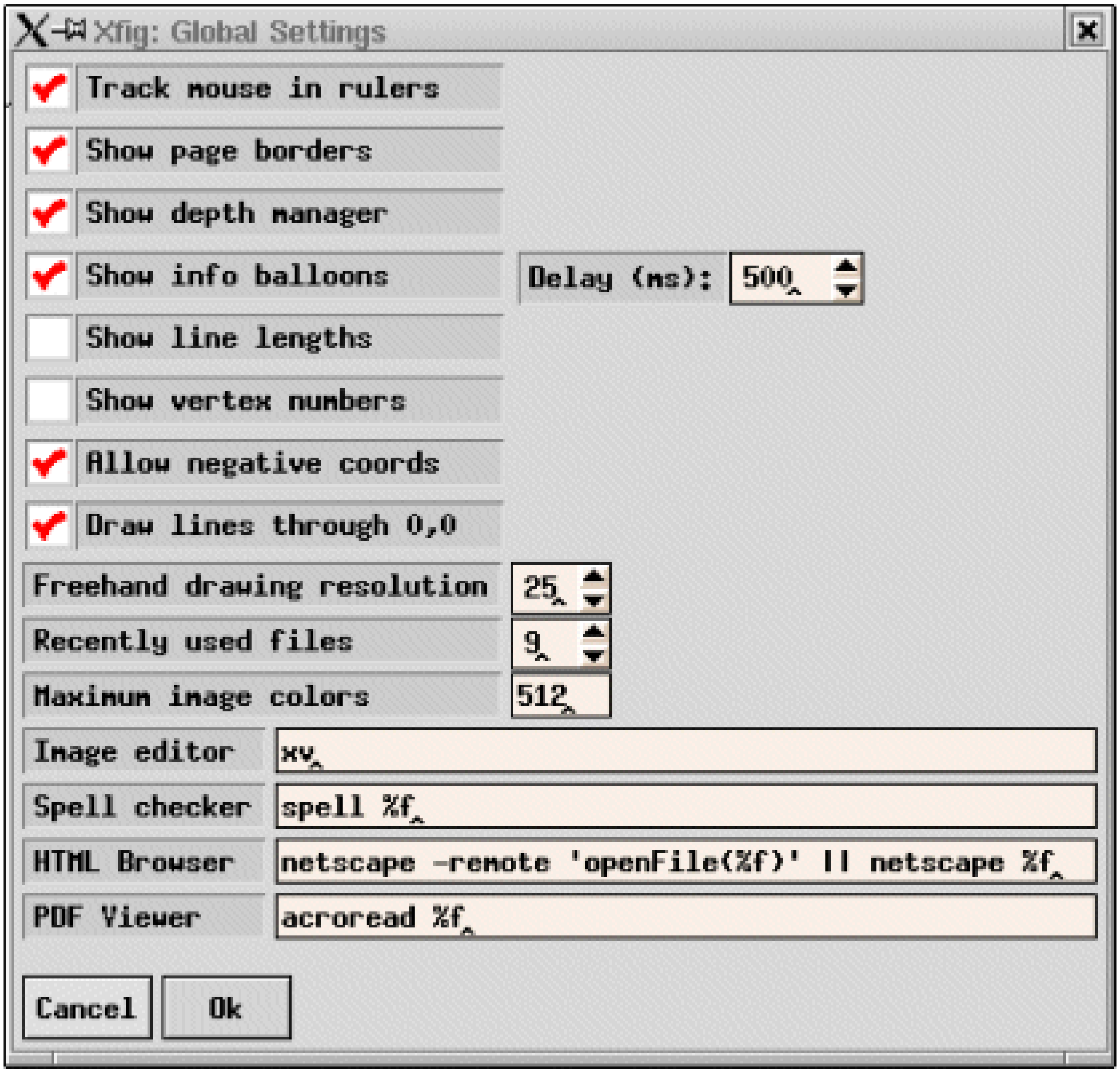
  Pressing this button will toggle all depths, i.e. turn **on** all depths that are **off** and turn **off** all depths that are **on**.
- Gray NEW

  Turning on this checkbox will make **xfig** display layers that that are **off** in gray.
- Blank NEW

  Turning on this checkbox will make **xfig** not display layers that that are **off**.

[Contents | Introduction | Credits]

Global Settings



Xfig: Global Settings

☑ Track mouse in rulers

☑ Show page borders

☑ Show depth manager

☑ Show info balloons   Delay (ms): 500

☐ Show line lengths

☐ Show vertex numbers

☑ Allow negative coords

☑ Draw lines through 0,0

Freehand drawing resolution: 25

Recently used files: 9

Maximum image colors: 512

Image editor: xv

Spell checker: spell %f

HTML Browser: netscape -remote 'openFile(%f)' || netscape %f

PDF Viewer: acroread %f

This panel, which may be popped up with the *Global Settings* menu item in the Edit menu or the keyboard accelerator `Meta-G`, allows the user to change various settings in **xfig**.

- Track mouse in rulers

  This option will cause **xfig** to show triangular pointers in the rulers which indicate where the mouse is pointing on the canvas.
  The initial setting can be set by -track / -notrack option or trackCursor resource.

- Show depth manager

  Display or hide the depth manager panel on the right-side of the xfig window. See Depth Panel for details.

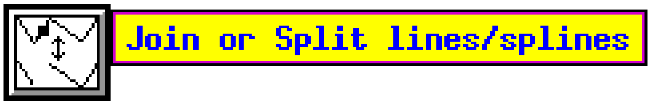
- Show page borders

  Xfig will draw a border on the canvas showing the size of the current page setting (see Export or Print) if this option is checked.

  The initial setting can be set by -showpageborder / -dontshowpageborder option or showpageborder resource.

- Show info balloons. delay(ms)

  If this option is checked, **xfig** pops up a ``balloon'' with information as the mouse passes over buttons or other areas of **xfig**. E.g.:

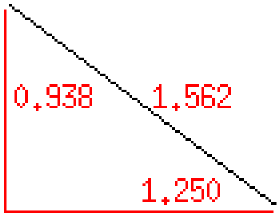


  The delay tells **xfig** how long to wait (in milliseconds) after the mouse is positioned on an area before popping up the balloon. The default is 500 ms.

  The initial setting of *Show info balloon* can be set by showballoons resource, and initial setting of *Delay (ms)* can be set by -balloons_delay option or balloon_delay resource.

- Show lengths on lines

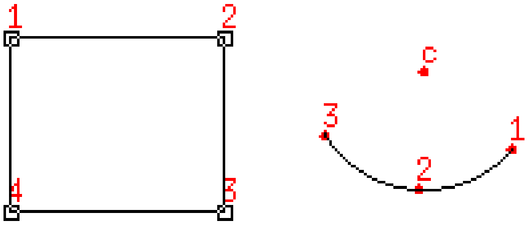
  If this is checked, **xfig** will show the lengths of lines as they are being drawn, thus:



  This option may also be turned on or off with the -showlength/-dontshowlength command-line option or showlength resource.

- Show vertex numbers

  If this is checked, **xfig** will show the number of each vertex of objects on the canvas thus:



  For arcs, it also shows the center point. This is probably only useful for debugging of Fig files. This option may also be turned on or off with the -shownums/-dontshownums command-line option or shownums resource.

- Allow negative coords

  If checked (the default), **xfig** will allow panning to negative x and y coordinates. This also turns off the query about shifting a figure with negative coordinates when loading a figure.

The initial setting can be set by -allownegcoords / -dontallownegcoords option or allownegcoords resouce.

- Draw lines through 0,0

  If checked (the default), **xfig** will draw X and Y axis lines through 0,0 on the canvas
  The initial setting can be set by -drawzerolines / -dontdrawzerolines option or drawzerolines resource.
- Freehand drawing resolution

  When drawing either a polygon or polyline in freehand mode (middle mouse button), **xfig** uses this resolution to decide when the mouse has moved enough to be considered a new point in the polygon/line. The units are Fig units (1/1200 inch) and the default is 25 (roughly 0.02 inches or 0.53 mm).
- Recently used files

  This spinner entry lets the user set the number of recently used files that are kept in the **File** menu. This is also updated in the user's **.xfigrc** file in the user's home directory, as are the file names themselves.
- Max image colors

  This sets the maximum number of colors **xfig** will attempt to use for imported images.

  The default is **64** and may also be set with the -max_image_colors command-line option or max_image_colors resource.
- Image editor

  This is the external program used when the *Edit Image* button is pressed in the import picture editor. The default is **xv** . You may use this program to crop/rotate/change colors in etc. your imported image. When you exit from this program **xfig** will re-import the image.
  The initial setting can be set by -image_editor option or image_editor resource.
- Spelling checker

  This is the external spelling check program which will be called when you use the spell check feature in **xfig** (*Spell Check...* in the Edit menu, or `Meta-K` accelerator).

  The default is ``spell %f", but it can be changed with -spellcheckcommand option or spellcheckcommand resource. The **%f** is necessary to tell **xfig** to insert the filename into the spell command.
- HTML Browser

  This is the external web (HTML) browser that **xfig** will call to view help files.
  The default is ``netscape -remote 'openFile(%f)' || netscape %f", it says to look for a running **Netscape** and open the pages in that. If none is running, a new one will start.
  The **%f** is necessary to tell **xfig** to insert the filename into the browser command. This can be set by browser resource.
- PDF Viewer

  This is the external PDF (Portable Document Format) viewer that **xfig** will call to view the **xfig** tutorial.
  The default is ``acroread %f", which will invoke **Acrobat Reader**. The **%f** is necessary to tell **xfig** to insert the filename into the viewer command. This can be set by pdfviewer resource.

[Contents | Introduction | Credits]

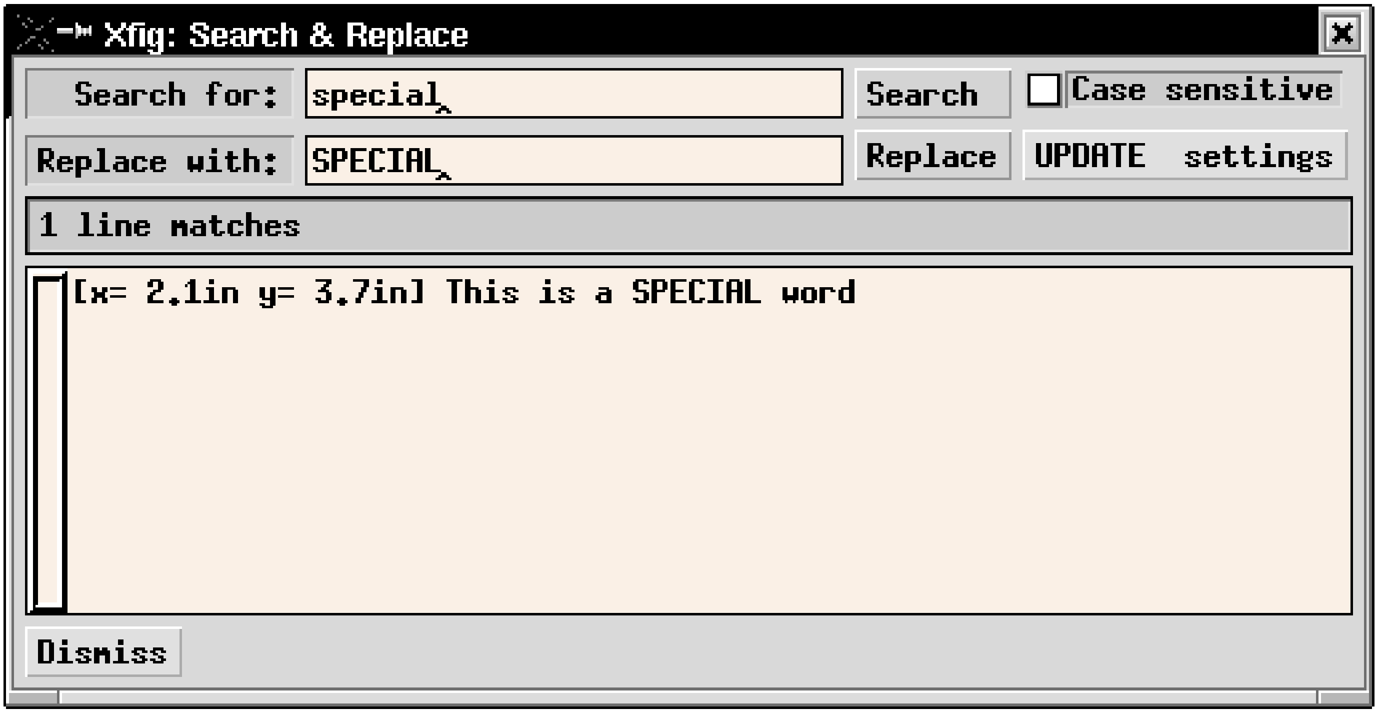
Xfig User Manual

Version 3.2.4 December 2002

Miscellaneous

Search and Replace Text

The **Search/Replace** entry in the **Edit** menu will popup the Search & Replace Panel. The keyboard accelerator `Meta-I` will do the same. Using the Search & Replace Panel, users can search and optionally replace text in the figure. It is also possible to apply UPDATE to searched texts.



Search for: special

Replace with: SPECIAL

1 line matches

[x= 2.1in y= 3.7in] This is a SPECIAL word

- *Search for*

  Specify the string to be searched. Pressing *Return* in this window or pressing the *Search* button will start the search.
- *Search*

  Pressing this button will start the search for the text in the *Search for* entry. The results of the search will be displayed in the bottom window, with the coordinates of each text object containing the word or words.
- *Case Sensitive*

  If this box is checked, upper-case and lower-case letters will be considered significant when searching.
- *Replace with*

  Specify the string to replace the string entered in the *Search* field.
- *Replace*

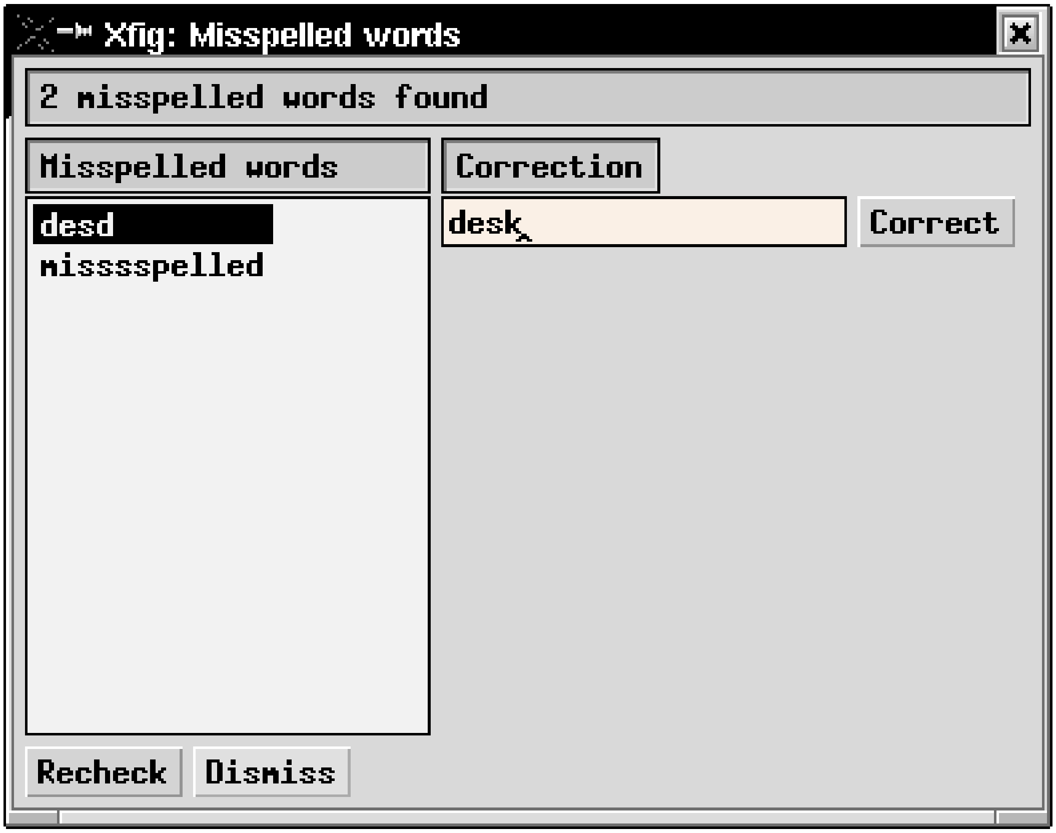
  If this button is clicked after the search, the string specified in *Search for* field will be replaced with the string specified in the *Replace with* field. *Search* must be clicked once before clicking this button.
- *UPDATE Settings*

  If this button is clicked after the search, the UPDATE function will be performed on the matching texts, applying settings in the Attribute Panel to the matched text objects.

N.B.: The *Undo* function is not available for this feature.

Spell Check

Spell Check... in the *Edit* menu or keyboard accelerator `Meta-K` will popup the Spell Check Panel. Xfig will run the external spelling checker (resource `Fig.spellcheckcommand`) on text in the figure, and shows any misspelled words in the text.



- *Status window*
  This window shows the status of the spelling checker, and how many words (if any) are misspelled.
- *Misspelled words*
  This listbox shows the misspelled words, with a scrollbar if necessary If you click on a word it is copied to the *Correction* window where you may edit it to correct it. Pressing *Return* in this window or the *Correct* button will replace **all** occurrances of this word in all text objects in the figure.
- *Recheck*
  Pressing this button will do a re-check of spelling. This is useful after correcting words to verify the corrections.

N.B.: The *Undo* function is not available for this feature.

2- or 3-Button Mouse

Use of a three-button mouse is recommended, but it is also possible to use a two-button mouse (if you have a two-button mouse and your X server doesn't emulate a three-button mouse, press the Meta (or Alt) key and right mouse button together to simulate mouse button 2). Normally, mouse buttons 1 to 3 are assigned to the left, middle, and right buttons respectively.

Wheel Mouse

If xfig is compiled with `` `#define WHEELMOUSE' `` defined in the Imakefile and you have a wheel-mouse, then the following functions are available:

- Attribute Buttons

  Rotating the wheel on attribute buttons will increase/decrease setting of the button as if mouse button 2 or 3 is clicked. This can be useful to alter parameters such as Line Width.
- Spinners

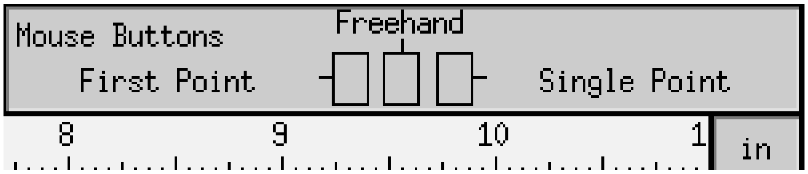
  Rotating the wheel on a spinner will increase/decrease the value.
- Rulers and Canvas

  Rotating the wheel on the rulers or canvas will scroll the canvas.
- Zooming

  Rotating the wheel on the canvas while the **Control** key is pressed will zoom/unzoom the canvas around the pointer.
- Changing Modes

  Rotating the wheel on the canvas while the **Shift** key is pressed will popup a panel to select drawing modes (upward) and editing modes (downward). The panel will close when any item is selected, or when the wheel is rotated in the reverse direction.

Mouse Function Indicator



Mouse Function Indicator shows which functions are assigned to each mouse button.

Although the function of each mouse button changes depending on the state (selected mode, position of the mouse cursor, state of SHIFT key or CONTROL key, etc) at the time, the mouse function indicator shows the current function at all times.

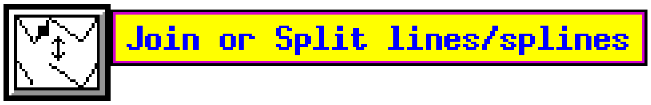
If the TEXT mode is selected and it is ready to enter text from keyboard, the mouse function indicator shows that as a keyboard icon.

If left and right mouse button are switched, message on the mouse function indicator can be switched with `-flipvisualhints` option or `Fig.flipvisualhints` resource.

See also Information Balloons.

Information Balloons

If this mode is *ON* (see Global Settings Panel), `balloon' messages corresponding to the object the mouse is positioned over will popup after a predetermined time (in milliseconds; resource `Fig.balloon_delay`). Here is an example:



Year 2000 Compliancy

Both **Xfig** and **TransFig/fig2dev** are Year 2000 compliant. They don't rely on any dates for their operation.

[Contents | Introduction | Credits]

LaTeX and Xfig

- Importing xfig figures into your LaTeX Files" by Eric Masson (ericm@kirk.ee.mcgill.ca),
- Changing the Size of Pictures in pstex_t by Stephen Eglen (stephene@cogs.susx.ac.uk)
- Xfig and PDFLaTeX by Josselin Mouette (jmouette@ens-lyon.fr)
- Miscellaneous Notes

How to Import Xfig Figures in Your LaTeX Files

When you call xfig use the following command line:

```
xfig -specialtext -latexfonts -startlatexFont default
```

If you want ALL of your figures to be started with special text and latex fonts, you can set the following resources in your .Xresources or whatever resource file you use:

```
Fig.latexfonts: true
Fig.specialtext: true
```

There are several formats to which xfig can generate output and latex can read. I will only cover three cases:
(A) Export Fig format directly into latex form
(B) Export Fig in encapsulated postscript and import the postscript in latex.
(C) Save the figure partly in postscript and partly in latex form and superimpose them in your document.

All three methods have their advantages and are equally simple to handle. In method (A) the advantage is that all your work is in tex form and that your .dvi files will hold all the necessary information. In (B) you have all the power and fonts of postscript at your disposal. In (C) you get the drawing power of postscript and the typesetting of latex for your strings.

In your latex preamble (the part that preceeds your \begin{document} statement) place the following lines:

```
\input{psfig}
```

So your preamble could look like this:

```
\documentstyle[12pt,bezier,amstex]{article}  % include bezier curves
\renewcommand\baselinestretch{1.0}           % single space
\pagestyle{empty}                            % no headers and page numbers
```

```
\oddsidemargin -10 true pt      % Left margin on odd-numbered pages.
\evensidemargin 10 true pt      % Left margin on even-numbered pages.
\marginparwidth 0.75 true in    % Width of marginal notes.
\oddsidemargin  0 true in       % Note that \oddsidemargin=\evensidemargin
\evensidemargin 0 true in
\topmargin -0.75 true in        % Nominal distance from top of page to top of
\textheight 9.5 true in         % Height of text (including footnotes and figures)
\textwidth 6.375 true in        % Width of text line.
\parindent=0pt                  % Do not indent paragraphs
\parskip=0.15 true in
\usepackage{color}              % Need the color package
\usepackage{epsfig}

\input{psfig}           % Capability to place postscript drawings
```

And your document beginning could look like this:

```
\begin{document}
\end{document}
```

TYPE A - Exporting directly to latex form

In terms of drawing capabilities this is the weakest form you can use. Lines in latex can only be drawn at multiples of 30 and 45 degrees. And lines with arrows can only be drawn at multiples of 45 degrees. Several features such as ellipses, splines, etc. are not supported (xfig does not take advantage of available LaTeX macro packages such as bezier). When drawing lines for type A drawings make sure you restrict yourself to the proper angle geometry in xfig. Otherwise when you export your figures to latex format, xfig will approximate your lines to the nearest angle available in latex. Usually this has unpleasant results.

In this mode, you can type any LaTeX string on your figure. Once imported to LaTeX, the string will be interpreted properly. For example:

```
                $\int_0^9 f(x) dx$
```

would result in a integration from 0 to 9 of the function f(x).

To create your LaTeX file just choose the export option off the xfig main menu. And then select LaTeX picture as the language to export. This will create a file with a .latex extension which you can then call directly into your latex document. For example this code would import the file yourfile.latex directly into latex format:

```
\begin{figure}[htbp]
\begin{center}

\input{yourfile.latex}
```

```
\caption{The caption on your figure}
\label{figure:yourreferencename}
\end{center}
\end{figure}
```

TYPE B - Exporting to Encapsulated postscript

There are no limitations in drawing figures of this type. Except that one cannot use latex command strings in this format. However all of the many fonts of postscript are available when this format is selected. Once you are done drawing your figure simply choose the export menu off of the xfig main menu and select encapsulated postscript as your output language. This will create a .eps file which you can then include into you latex ducument in the following way:

```
\begin{figure}[htbp]
\begin{center}
\ \psfig{file=yourfile.eps}
\end{center}
\caption{Your caption}
\label{figure:yourreference}
\end{figure}
```

TYPE C - Postscript/Latex format

You can draw any lines or curves when using this format. In this type of export, latex strings are permitted you also have the postscript fonts available to you. Therefore you can type in strings such as

```
                $\int_0^9 f(x) dx$
```

and they will be processed by latex. You will need to export your file twice when using this format: once to the combined ps/latex (ps part) language and once to the combined ps/latex (latex part) language. The first will create a .pstex file and the second will create a .pstex_t file. The .pstex_t file automatically calls the .pstex file and you do not need to include it explicitely in your tex file (users of the previous version please take note of this.) To include your figure just use something similar to this:

```
\begin{figure}[htbp]
\begin{center}

\input{yourfigure.pstex_t}

\caption{Your figure}
\label{figure:example}
\end{center}
\end{figure}
```

N.B. You might want to edit the .pstex_t files created by xfig. When it refers to the other file (.pstex) it automatically gives the path specification to the .pstex file. This can be an inconvenience if you move your files to another directory because your latex processing will fail. I personally prefer to remove the full path specification and only keep the filename.

Changing the Size of Pictures in pstex_t

If you just include the picture using `\input{file.psttext_t}` you have no way of specifying the size of the picture. There are two solutions to this.

1. Draw it the right size in xfig to start with. Or, when you are exporting the figure, change the magnification of the picture by using the magnification box in the export window. Either way you have to go back into xfig if you dont like the size of the image in your latex document.

2. Get LaTeX to change the size of the picture, using either `\scalebox` or `\resizebox`. These are general functions for scaling text or pictures from the graphics package:

   A. `\scalebox{factor}{object}`
   Will scale the object by any factor. Factor is just a number (< 1 = reduction; > 1 = enlargement) Object is normally some text or graphics

   B. `\scalebox{2}{ \input{file.pstex_t} }`
   will scale the picture by 2, dependent on driver (.ps works, but xdvi wont). Scaling bitmap fonts may produce ugly results, so try and avoid them!

   C. `\resizebox{width}{ht} {stuff}`
   will resize "stuff" to be of size width x ht. Using "!" as an argument retains the aspect ratio of the box. eg `\resizebox{5cm}{!}{fat cat}` will make "fat cat" appear 5 cm wide, and suitably high. (From p129, Lamport)

xfig and PDFLaTeX

written by Josselin Mouette (jmouette@ens-lyon.fr)

1. A STANDARD PDF FILE
In xfig, select the "PDF" export filter, which will generate a `foo.pdf` file. In your document, put in the preamble : \usepackage[pdftex]{graphicx} and insert your picture with `\includegraphics{foo.pdf}` You may use all the includegraphics options as well. Pros: Very easy to use. Cons: The text in your figure will appear as is on your document, using postscript fonts; you cannot put some TeX code in it.

2. A COMBINED PDF/LaTeX FILE
This is the method I would recommend in most cases. It may be difficult for the beginner at the first time, but it is really powerful. If you choose this method, you'll have to set the xfig fonts to LaTeX ones, and to set the "special" attribute of your text boxes. To do this automatically, you can add these 2 lines in your .Xresources or .Xdefaults (depends on your system):

```
Fig.latexfonts: true
Fig.specialtext: true
```

Then, when exporting, select the "Combined PDF/LaTeX" format. Then, in your LaTeX file, put in the preamble:
`\usepackage[pdftex]{graphicx,color}`
The color package is required whenever you put some text in colors. Then include the picture with:
`\input foo.pdftex`
You can also resize it:
`\resizebox{3cm}{!}{\input foo.pdftex} % sets the width to 3cm`
Pros: Whatever is put in your text boxes is treated just as your document's code; that means you can use your own macros, which is really cool.
Cons: When putting big formulas on your figure, it is sometimes difficult to predict what place they will take because xfig and fig2dev don't know the bounding area of the text because they contain LaTeX directives that aren't printed.

3. METAPOST
   There is nothing special to do in xfig to use MetaPost. All the text you type will be treated as plain TeX code - note, this will be not compiled within your document, so you don't have acess to packages like AMS-TeX, neither have you to your macros. In xfig, export your file with the MetaPost filter, it creates `foo.mp`. Then, type mpost foo.mp, it will generate `foo.0` (or `foo.1`, sometimes). In your document, put this in the preamble:

   ```
   \input supp-pdf.tex
   \usepackage[pdftex]{graphicx}
   ```

   And to include your figure :

   ```
   \convertMPtoPDF{foo.0}
   ```

   That's it. Quite simple, and you can put a bit TeX inside.
   Pros: Can be easily included in a dual-output (pdf/dvi) file: for including it as PS, just put a `\includegraphics{foo.0}` in the document.
   Cons: Not adapted to big formulas, as AMS-LaTeX is not accessible. Long phrases may look bad as well, if your document is not in English (babel cannot be used).

4. MULTI-METAPOST
   This method is designed to be used in PDF presentations. Using the `\pause` command, it will display step by step the layers of your figure as you click on the button, which can look very nice (and can even be useful sometimes). All that have been told about MetaPost inclusions is true, but there are a few extra things to know:

   A. When creating your figure, be careful with the depth of your objects. When exporting your figure in the MultiMetaPost format, transfig will treat the consecutive depth levels where is an object as a single layer, for example:

      ```
      Circle at depth 51 \__first displayed layer
      Text at depth 50   /
      *** Nothing at depth 49
      ```

```
Square at depth 48 \
Text at depth 48    > Second displayed layer
Curve at depth 47  /
... and so on.
```

B. After exporting, mpost `foo.mmp` will create a set of files named `foo.0, foo.1...` To include them in the document, you will need the mpmulti.sty provided with the latest version of PPower4 (still in Beta stage at the time of writing). The preamble of your document should look like this:

```
\input supp-pdf.tex
\usepackage[pdftex]{graphicx}
\usepackage{pause,mpmulti}
```

And to include your animation, just put:

```
\multiinclude{foo}
```

You can adjust it to a defined size by using:

```
\multiinclude[graphics={width=5cm}]{foo}
```

Compile your document, then ppower4 it. Nifty, isn't it?
Pros: The only way to insert automatically animations. Benefit of the existing xfig's depth system.
Cons: Are there any?

Some Other Notes

Sometimes one may get "mn" in your LaTeX text. Here is what one user did to work around that problem:

The problem was in the `\smash` part of a command that was generated in the latex part of the export. If the `\mddefault` and `\updefault` are not set properly on your system you need to define them to do nothing.

[Contents | Introduction | Credits]

Xfig User Manual
Version 3.2.4
December 2002

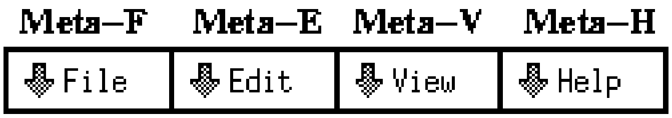
Keyboard Accelerators

The following keyboard accelerators (shortcuts) are available in **xfig**.

It is also possible to use accelerators on some of the popup panels. See the description about each popup panel for more information.

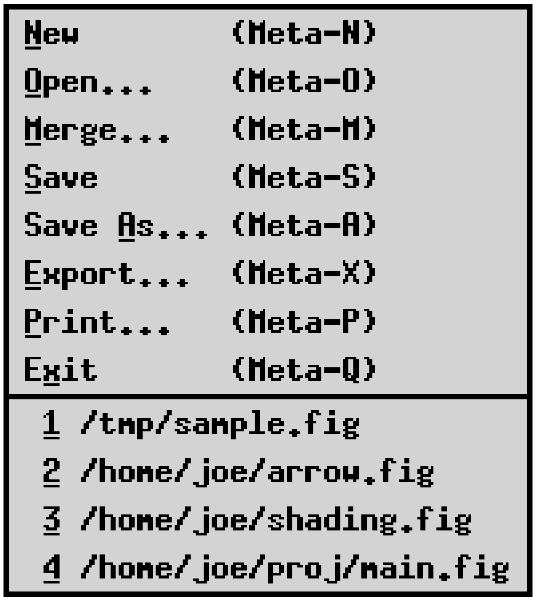
- Menu Accelerators
- Drawing/Editing Accelerators

Menu Accelerators:



File Menu Accelerators:

(Meta-F)

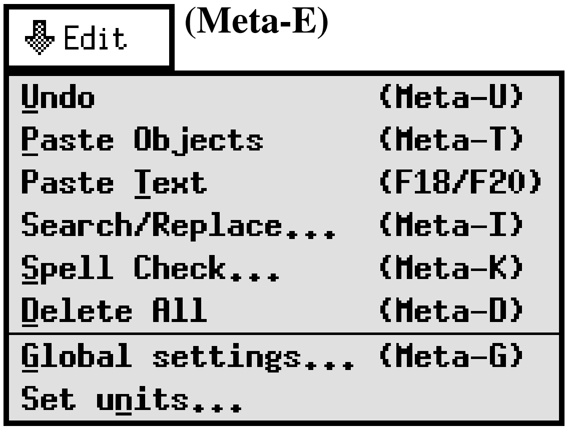


The underlined letter or number shows the shortcut that will perform the indicated action ***once the menu is popped up***. The shortcuts in parentheses (e.g. Meta-X) show the shortcut available without popping up the menu.

Note that pressing the number **1** while the file menu is up will load the file **/tmp/sample.fig** in the above example.

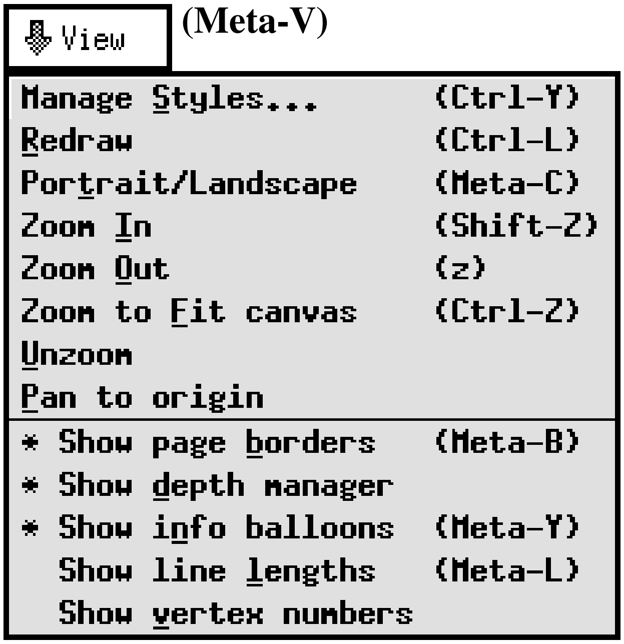
The accelerator **`Shift-Meta-P`** will print using the previously set parameters (i.e. the print panel isn't popped up, but the figure is printed).
The accelerator **`Shift-Meta-X`** will export using the previously set paramters (i.e. the export panel isn't popped up, but the figure is exported).

Edit Menu Accelerators:



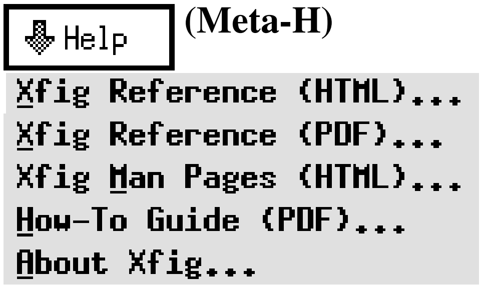
(Meta-E)

View Menu Accelerators:


View
Manage Styles... (Ctrl-Y)
Redraw (Ctrl-L)
Portrait/Landscape (Meta-C)
Zoom In (Shift-Z)
Zoom Out (z)
Zoom to Fit canvas (Ctrl-Z)
Unzoom
Pan to origin
* Show page borders (Meta-B)
* Show depth manager
* Show info balloons (Meta-Y)
Show line lengths (Meta-L)
Show vertex numbers

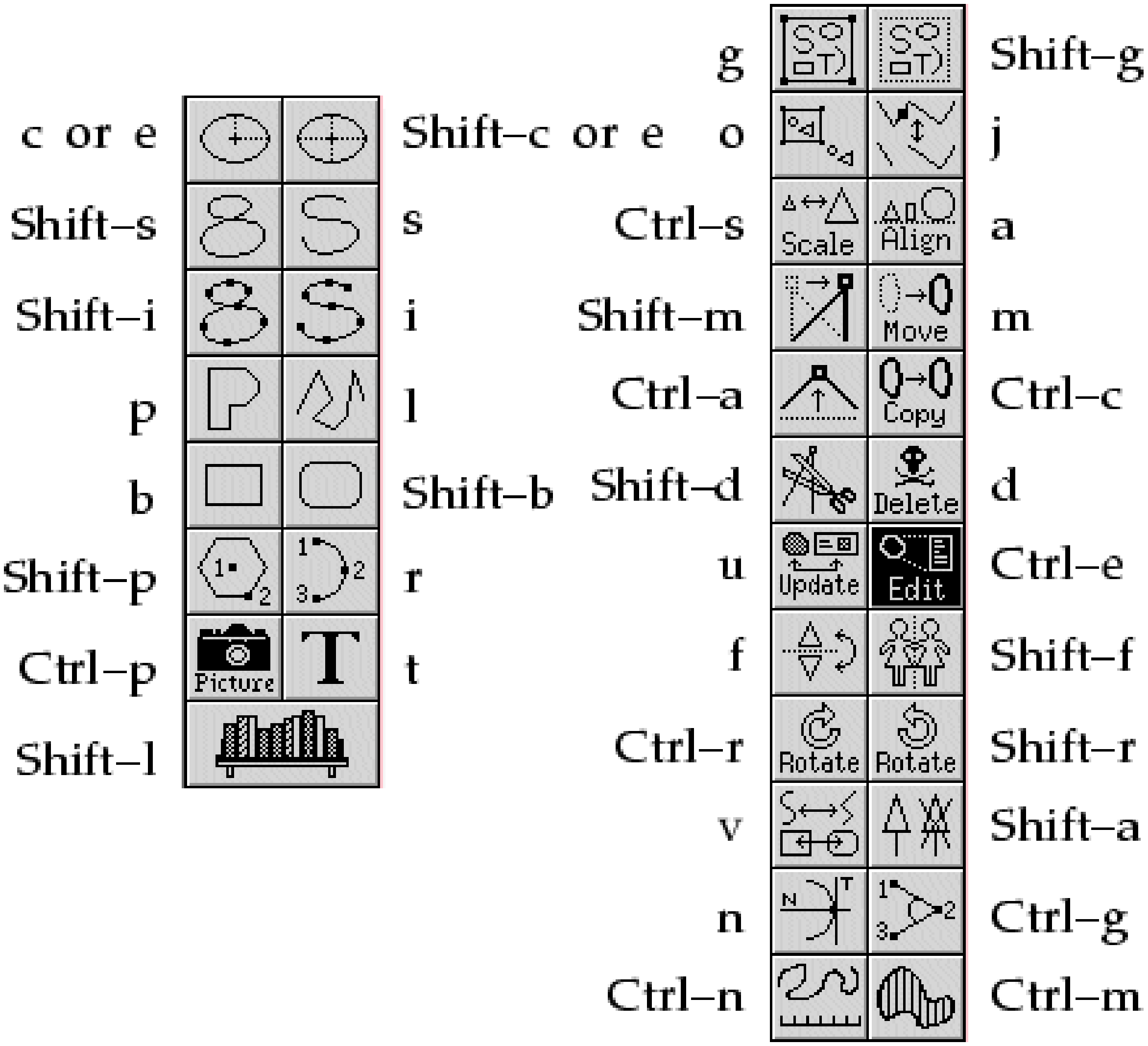
(Meta-V)

Help Menu Accelerators:



(Meta-H)

Drawing and Editing Mode Accelerators:



[Contents | Introduction | Credits]

Xfig User Manual

Version 3.2.4 December 2002

Command Line Options and Resources

Command-line Options

X Resources

Environment Variables

Command-line Options

xfig will accept the following command-line options. They may be specified such as `` `xfig -portrait -metric' `` when starting **xfig**. A short form of the option name is shown in parentheses. This is the miminum number of letters needed to distinguish it from another option.

It is also possible to use X resources instead of, or in addition to command-line options. There are also some **xfig**-related environment variables.

Mode Settings

`-allownegcoords (-allow)`
: Allow panning the canvas into the negative region. This is the default. Also, in this mode the user is **not** asked if he/she wants to shift the figure when reading in a file with objects that have negative coordinates.

`-dontallownegcoords (-dontallow)`
: Don't allow panning the canvas into the negative region. Also, when in this mode the user **is** asked if he/she wants to shift the figure when reading in a file with objects that have negative coordinates.

`-dontshowzerolines (-dontshowz)`
: Don't draw the lines crossing through 0,0 on the canvas. The page border may still be drawn unless -dontshowpageborder is selected.

`-freehand_resolution` *`resolution`* `(-free)`
: Set the resolution of the freehand drawing mode to *resolution* pixels. This is the number of pixels the mouse must move before **xfig** adds another point to the object being drawn.

`-grid_unit` *unit* `(-grid_u)`
: Set the *unit* of the grid and rulers when in inches mode. If *unit* is "1/10", "ten", "tenth", "1/10" or "10", (e.g. -grid_unit ten) then the grid and point positioning is in decimal inches. Otherwise they are in 1/16 inch units (as before). When in this mode, the grid step sizes are 0.1, 0.2, 0.5 and 1 inch and the positioning steps are 0.05, 0.10, 0.2, 0.5, and 1.0 inch. The default is *1/16 inch*.

`-hiddentext (-hi)`
: Set the *Hidden* flag of TEXT FLAGS to *ON*. The default is *off*.

`-inches (-inc, -imperial)`
: Make *inches* the unit of choice. This is the default. See also the `-metric` option and Set Units.

`-landscape (-lan)`
: Start **xfig** in *landscape* (11inch x 8.5inch) mode. This is the default. See also the `-portrait` option.
It is also possible to change the orientation using the View/Portrait/Landscape menu entry after starting **xfig**.
Fig files of version 3.0 and later contain the orientation information, and the mode will be changed as those files are loaded.

`-latexfonts (-lat)`
: Select LaTeX fonts to start. Normally, PostScript fonts are selected.

`-metric (-me, -centimeters)`
: Make *centimeters* the unit of choice. The default is *inches*. See also the `-inches` option and Set Units.

`-portrait (-po)`
: Start **xfig** in *portrait* (8.5inch x 9inch) mode. The default is *landscape* mode. See also the `-landscape` option.

`-rigidtext (-rigi)`
: Set the *Rigid* flag of TEXT FLAGS to *ON*. The default is *off*.

`-rulerthick` *`thick`* `(-ru)`
Set the thickness of top and side rulers in pixels. The default (and minimum) is *24*.

`-scale_factor` *`scale`* `(-scale_factor)`
Scale all drawings read in by *scale* factor. This is useful in conjunction with the update -update option to do a batch scaling of figures.

`-showzerolines (-showz)`
Draw the lines crossing through 0,0 on the canvas. See also zerolines option for the color of the zero lines.

`-specialtext (-spec)`
Set the *Special* flag of TEXT FLAGS to *ON*. The default is *off*.

`-startfontsize` *pointsize* `(-startfo)`
Set the initial value of TEXT SIZE. The default is 12pt.

`-startpsFont` *font* `(-startps)`
Select the initial PostScript font as TEXT FONT. The default is *Times-Roman*.

`-startlatexFont` *font* `(-startla)`
Select the initial LaTeX font as TEXT FONT. This is effective only if `-latexfonts` option is given. The default is *Default*.

`-starttextstep` *stepsize* `(-startt)`
Set the initial value of TEXT STEP. The default is 1.2.

`-startfillstyle` *stylenumber* `(-startfi)`
Set the initial setting (-1 to 21) of FILL STYLE. The default is -1 (*No fill*).

`-startgridmode` *modenumber* `(-startg)`
Set the initial GRID MODE. Mode 0 is *None*. When *unit* is inches, mode 1 1/8 inch, mode 2 is 1/4 inch, mode 3 is 1/2 inch, and mode 4 is 1 inch. When *unit* is centimeters, mode 1 2mm, mode 2 is 5mm, mode 3 is 1cm, and mode 4 is 2cm. The default is mode 0 (*no grid*).

`-startposnmode` *modenumber* `(-startpo)`
Set the initial POINT POSITION. Mode 0 is *Any*. When *unit* is inches, mode 1 is 1/16 inch, mode 2 is 1/8 inch, mode 3 is 1/4 inch, mode 4 is 1/2 inch, and mode 5 is 1 inch. When *unit* is centimeters, mode 1 is 1mm, mode 2 is 2mm, mode 3 is 5mm, mode 4 is 1cm, and mode 5 is 2cm. The default is mode 1 (1/16inch or 1mm).

`-startlinewidth` *width* `(-startli)`
Set the initial value of LINE WIDTH. The default is 1.

`-update` *file [file ...]* `(-up)`
Run **xfig** in an "update" mode, where it will read each Fig file specified on the command line and write it out to the original file, in the current file format for the version of **xfig** being run. The original Fig file will be

preserved with the suffix *.bak* attached to the name.
In this mode, **xfig** doesn't connect the X server, so no window is opened, and it exits when finished.

`-userscale` *scale* (`-users`)
Set the multiplier used when displaying dimensions. See also Set Units.

`-userunit` *units* (`-useru`)
Set the unit name used when displaying dimensions. See also Set Units.

`-zerolines` *color* (`-zerol`)
Draw the lines crossing through 0,0 on the canvas in this color. The page border lines will overwrite the zero lines in the positive quadrant unless it is turned off.

`-zoom` *zoomscale*
Set the initial value of ZOOM SCALE. The default is 1.0.

PRINT and EXPORT Settings

`-center (-cente)`
Select *Center* as *Justification* when Exporting or Printing. This the the default.

`-exportLanguage` *language* (`-exportL`)
Set the initial export language when Exporting. *Language* may be one of the following:

Vector formats:

box	LaTeX box (figure boundary)
latex	LaTeX picture
epic	LaTeX picture + epic macros
eepic	LaTeX picture + eepic macros
eepicemu	LaTeX picture + eepicemu macros
pictex	PiCTeX macros
ibmgl	IBMGL (or HPGL)
eps	Encapsulated PostScript
ps	PostScript
pdf	Portable Document Format
pstex	Combined PS/LaTeX (both PS and LaTeX parts)
textyl	Textyl \special commands
tpic	TPIC
pic	PIC
html	HTML Image map
mf	MF (MetaFont)
mp	MP (MetaPost)
mmp	MMP (Multi-MetaPost)
cgm	CGM (Computer Graphics Metafile)
binary cgm	

```
          emf          EMF (Enhanced Metafile Format)
          tk           Tk (ala Tcl/Tk)
     Bitmap formats:
          gif          GIF
          jpg          JPEG (JFIF)
          pcx          PCX
          png          PNG (Portable Network Graphics)
          ppm          PPM (Portable Pixmap)
          sld          AutoCAD Slide format
          tiff         TIFF (no compression)
          xbm          X11 Bitmap
          xpm          X11 Pixmap (XPM3 package)
```

The default is **eps**.

`-export_margin (-export_m)`

Set the size of the border margin around the figure for exporting to bitmap, PostScript, Encapsulated PostScript, or HTML MAP formats. This is in units of pixels (1/80th inch). The default is 0.

`-flushleft (-flu)`

Select *Flush Left* as *Justification* when Exporting or Printing. The default is *Centered*.

`-jpeg_quality` *quality* `(-j)`

Set the quality used when Exporting to the JPEG (JFIF) image format.

`-magnification` *mag* `(-mag)`

Set *Magnification* when Exporting or Printing. *mag* is magnification in %. The default is 100%.

`-multiple (-mu)`

Set *Pages* to *Multiple* when Exporting or Printing. The default is *Single*.

`-paper_size` *size* `(-pap)`

Set the paper size when Exporting or Printing. *size* can be one of follows:

Paper	Size	Option name
Letter	8.5in x 11in	Letter
Legal	8.5in x 14in	Legal
Tabloid	17in x 11in	Tabloid
ANSI A	8.5in x 11in	A
ANSI B	11in x 17in	B
ANSI C	17in x 22in	C
ANSI D	22in x 34in	D
ANSI E	34in x 44in	E
ISO A9	37mm x 52mm	A9

ISO A8	52mm x 74mm	A8
ISO A7	74mm x 105mm	A7
ISO A6	105mm x 148mm	A6
ISO A5	148mm x 210mm	A5
ISO A4	210mm x 297mm	A4
ISO A3	297mm x 420mm	A3
ISO A2	420mm x 594mm	A2
ISO A1	594mm x 841mm	A1
ISO A0	841mm x 1189mm	A0
JIS B10	32mm x 45mm	B10
JIS B9	45mm x 64mm	B9
JIS B8	64mm x 91mm	B8
JIS B7	91mm x 128mm	B7
JIS B6	128mm x 182mm	B6
JIS B5	182mm x 257mm	B5
JIS B4	257mm x 364mm	B4
JIS B3	364mm x 515mm	B3
JIS B2	515mm x 728mm	B2
JIS B1	728mm x 1030mm	B1
JIS B0	1030mm x 1456mm	B0

`-single (-si)`

Set *Pages* to *Single* when Exporting or Printing. This is the default.

`-transparent_color` *xfig color number* `(-tran)`

Set the "transparent" color when Exporting to the GIF image format. This is one of the **xfig** colors, which are numbered from -3 (Background), -2 (None), -1 (Default), 0 (Black), 1 (Blue) etc. See Colors.

Setting of Fonts or Colors, etc.

`-boldFont` *font* `(-bol)`

Set the font used to display file names or confirmation messages, etc. The default is `8x13bold`.

`-buttonFont` *font* `(-butt)`

Set the font used to display labels of most buttons. The default is `6x13`.

`-cbg` *color*

Set the background color of the canvas. If you want to set the background color of everything in **xfig** (menus, etc), use the general `-bg` option.

`-cfg` *color*

Set the default color of objects. If you want to set the foreground color of everything in **xfig** (menus, etc), use the general `-fg` option.

`-depth (-dep)`

Choose depth of visual desired.
Use `xdpyinfo` to see what visuals and depths are supported. See also `-visual` option.

`-dontswitchcmap (-dontsw)`

Normaly, **xfig** will switch to a private colormap when not enough colors are available in the default colormap, but this option will inhibit the switch. See also `-max_image_colors`.

`-max_image_colors` *numcols* `(-max)`

Limit the number of colors used for imported images to *numcols*. The default is 64. **Xfig** uses a The Kohonen neural network to map the colors in imported pictures to the limited set available to it.

`-monochrome (-mo)`

Use black and white only for display.

`-normalFont` *font* `(-nor)`

Set the font used in message windows, etc. This font will also be used on the canvas when the desired font is not available. The default is `6x13`.

`-noscalablefonts (-nos)`

Disable use of the X11R5 or OpenWindows scalable fonts. You might want to use this for debugging.

`-scalablefonts (-sc)`

Allows use of the X11R5 or OpenWindows scalable fonts. This is the default. If the scalable fonts aren't available, **xfig** will automatically switch to non-scaling fonts.

`-visual` *visualname*

Use *visualname* as the visual. *visualname* is one of TrueColor, StaticColor, DirectColor, StaticGray, GrayScale, and PseudoColor. **xfig** uses the default visual unless this is specified.
Use the `xdpyinfo` command to see which visuals and depths are supported. See also `-depth` option.

Settings for the Size of Windows, etc.

`-but_per_row` *number* `(-but_)`

Specify how many buttons wide the mode panel should be. This option is not usually needed since **xfig** automatically adjusts the number of buttons per row in the mode panel to fit the canvas size chosen (see -pheight). The default is 2, but **xfig** will automatically increase this if necessary to fit on the user's screen, if this option is not explicitly specified. Also, **xfig** may be compiled to use smaller icons for the mode panel.

`-geometry [`*W*x*H*`]`+*x*+*y* `(-geom)`

Set the **size** and/or **position** of the **xfig** window. You may use this to set the **size** of the main **xfig** window, or you may instead set the size of the drawing canvas in inches or cm using the `-pwidth` and `-pheight` options.

`-iconGeometry` *+x+y* (`-iconG`)

Set the position for the **xfig** icon on the display.

`-internalBW` *width*

Set the width of lines between all buttons and panels. The default is 2.

`-left (-le)`

Put the drawing and editing mode panel on the left side of the canvas. This is the default.

`-pheight` *height* (`-ph`)

Set the height of the **xfig** canvas to *height*. *height* is either cm or inches, depending on the selected unit.

`-pwidth` *width* (`-pw`)

Set the width of **xfig** canvas to *width*. *width* is either cm or inches, depending on the selected unit.

`-right (-righ)`

Put the drawing and editing mode panel on the right of the canvas. The default is left.

`-showallbuttons (-showa)`

Always show all the attribute buttons instead of only those relevant to the current drawing or editing mode. This takes up more screen real estate, but allows the user to see all settable parameters. Normally, each attribute button will only appear when it is relevant to the current drawing mode.

Other Settings

`-help` (`-h`)

Print all command-line options for **xfig** and quit.

`-balloon_delay` *msec* (`-bal`)

Specify the delay time in milliseconds before the information balloons popup. The default is 500 milliseconds.

`-debug (-deb)`

Turn on debugging mode. Print various debugging messages such as font names, etc.

`-flipvisualhints (-fli)`

Flip left/right mouse indicator messages for mice whose buttons have been switched. The default is to *not* flip the messages.

`-grid_color` *color* (`-grid`)

Draw the grid on the canvas in this color (default black).

`-icon_view (-icon_)`
Show the icon view of the library objects. This is the default. See also -list_view.

`-image_editor (-ima)`
Specify the program to be invoked when the *Edit Image* button on the Edit Panel for PICTURE object is clicked.

`-inputStyle` *style* i18n
Set the input style when entering international text with *input method*. The style may be selected from `OffTheSpot`, `OverTheSpot` and `Root`. If this is set to `None`, input method will not used. This is effective only if internationalization facility is enabled.

`-international` i18n
Turn on internationalization facility. **Xfig** must be compiled with the internationalization facility. Languages such as Japanese or Korean can't be used in international-**xfig** unless this option is specified when invoking **xfig** or the `international` resource is set to *ON*.

`-keyFile` *compose-key-file* `(-k)`
Use *compose-key-file* instead of `CompKeyDB` for compose key database. If there are no \``/`'s in the name, the file must reside in the **xfig** library directory *$XFIGLIBDIR* (normally `/usr/local/lib/X11/xfig`). If there are any \``/`'s in the name it is taken as is (not relative to *$XFIGLIBDIR*). If the name is start with \``~/`', it will replaced to the user's home directory.

`-library_dir (-li)`
Specify the directory where **Fig** object (drawing) directories are stored. See available libraries.

`-library_icon_size` *size* `(-library_i)`
Set the size of the library icons to *size* pixels square. See also -icon_view.

`-list_view (-list_)`
Show the list view of the library objects. The default is to show the icon view. See also -icon_view.

`-nosplash` NEW
Don't show the startup screen on startup. The default is to show the splash screen

`-pageborder` *color* `(-pag)`
Draw the page border on the canvas in this color (default light blue). The page border is turned on by the `-showpageborder` (resource Fig.showpageborder) option, and shows the edges of the current *export* paper size.

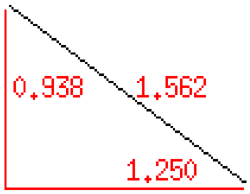
`-showballoons (-showb)`
`-dontshowballoons (-dontshowb)`
Popup/don't popup balloon messages when the cursor passes over buttons etc. The default is to show the balloons.

`-showdepthmanager (-showd)`
`-dontshowdepthmanager (-dontshowd)`

Show or hide the depth manager This will allow more room for the canvas. The default is to *show* the depth manager.

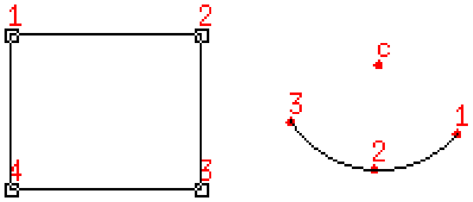
`-showlengths (-showl)`
`-dontshowlengths (-dontshowl)`

Makes **xfig** show the lengths of lines being drawn, in red text near the line itself. Also, when points are moved or added. In addition, imagine a triangle formed with the line segment being drawn as the hypotenuse, and a vertical and horizontal line forming the other two sides. These lines and their lengths are also drawn in red as the point is moved or added.



`-shownums (-shown)`
`-dontshownums (-dontshown)`

Makes **xfig** show the number of each vertex of objects on the canvas. For arcs, it also shows the center point. This is probably only useful for debugging of Fig files.



`-showpageborder (-showp)`
`-dontshowpageborder (-dontshowp)`

Show or don't show the border of the current paper size in the color specified by the `-pageborder` option (default light blue).

`-spellcheckcommand` *spell-check-command* `(-spel)`

Specify the external spell check program to use when checking spelling of text objects. The default is `spell'.

`-spinner_delay` *msec* `(-spinner_d)`

Specify the delay time in milliseconds before a spinner counts automatically, while the up- or down-arrow is being pressed. The default is 500 milliseconds.

`-spinner_rate` *msec* `(-spinner_r)`

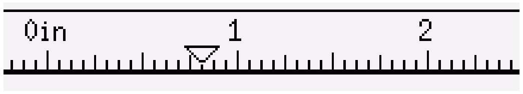
Specify the rate in milliseconds at which a spinner counts automatically. The default is 100 milliseconds.

`-tablet (-ta)`

Specifies that **xfig** should use the input tablet instead of the mouse for drawing. You must have the XInputExtension in your X server and an input tablet for this to work, and **xfig** must be compiled with the `TABLIB` and `USETAB` variables set in the `Imakefile` for it to use the tablet.
Use `xdpyinfo` to see if your server supports the XInputExtension.

`-track (-trac)`
`-notrack (-not)`

Turn on or off cursor (mouse) tracking arrows. The default is to track the cursor position by displaying arrows on the top and side ruler.



X Resources

Here are X resources which may be used to customize **xfig**. They are usually specified by putting lines into the resource database file of **xfig**, such as `/usr/X11/lib/app-defaults/Fig` or the user's own resource file which must be read by `xrdb`.

A note about xfig color resources:

Generally, the user should have a common resource `` `*customization: -color' `` set if they have a color display. This signals the X system (version 4 and later) to look for the `XXX-color` app-defaults file in the applications defaults directory. In that file (e.g. "`Fig-color`") the color definitions are used and it is responsible for including the main app-defaults file `Fig`.

Mode Resources

`allownegcoords (NegativeCoordinates)`

Allow panning the canvas into the negative region. This is the default. If this is **false** the user is asked if he/she wants to shift the figure when reading in a file with objects that have negative coordinates.
Option: `-allownegcoords`, `-dontallownegcoords`
Default: on

`hiddentext (HiddenText)`

Set initial value of the *Hidden* flag of the TEXT FLAGS.
Option: `-hiddentext`
Default: off

`inches (Inches)`

inches will be selected as the unit if this is set *ON*, *cm* otherwise.
Option: `-inches (-imperial)`, `-metric (-centimeters)`
Default: on (inches)

`landscape (Orientation)`

landscape orientation will be selected if this is set *ON*, *portrait* mode otherwise.
Option: `-landscape`, `-portrait`
Default: on (landscape)

`latexfonts (Latexfonts)`

LaTeX fonts will be used initially if this is set *ON*. Otherwise PostScript fonts will be used.
Option: `-latexfonts`
Default: off

`rigidtext (RigidText)`

Set initial value of the *Rigid* flag of the TEXT FLAGS.
Option: `-rigidtext`

Default: off

`specialtext (SpecialText)`

Set initial value of the *Special* flag of the TEXT FLAGS.

Option: `-specialtext`
Default: off

`startpsFont (StartpsFont)`

Select the specified PostScript font as the initial TEXT FONT.

Option: `-startpsFont`
Default: `Times-Roman`

`startlatexFont (StartlatexFont)`

Select the specified LaTeX font as the initial TEXT FONT.

Option: `-startlatexFont`
Default: `Default`

`startfontsize (StartFontSize)`

Set initial value of TEXT SIZE.

Option: `-startfontsize`
Default: 12

`starttextstep (StartTextStep)`

Set initial value of TEXT STEP.

Option: `-starttextstep`
Default: 1.2

`startfillstyle (StartFillStyle)`

Set the initial value (-1 to 21) of FILL STYLE.

Option: `-startfillstyle`
Default: -1 (None)

`startgridmode (StartGridMode)`

Set the initial value (0 to 4) of GRID MODE.

Option: `-startgridmode`
Default: 0 (None)

`startposnmode (StartPosnMode)`

Set the initial value (0 to 5) of POINT POSITION.

Option: `-startposnmode`
Default: 1 (1/16inch, or 1mm)

`startlinewidth (StartLineWidth)`

Set the initial value of LINE WIDTH.

Option: `-startlinewidth`
Default: 1

`userscale (UserScale)`

Set the multiplier used when displaying dimensions.
Option: `-userscale`
Default: 1.0

`userunit (UserUnit)`

Set the unit name used when displaying dimensions.
Option: `-userunit`
Default: `in` (or `cm`)

`zoom (Zoom)`

Set the initial value of ZOOM SCALE.

Option: `-zoom`
Default: 1.0

PRINT and EXPORT Resources

`export_margin (Width)`

Set the size of the border margin around the figure for exporting to bitmap, PostScript, Encapsulated PostScript, or HTML MAP formats. This is in units of pixels (1/80th inch).
Option: `-export_margin`
Default: `0`

`exportLanguage (ExportLanguage)`

Set the initial export *Language* when Exporting.

Option: `-exportLanguage`
Default: `eps` (Encapsulated PostScript)

`flushleft (FlushLeft)`

Set the initial *Justification* when Exporting or Printing.

Option: `-flushleft`
Default: off (Centered)

`jpeg_quality` *quality* `(Quality)`

Set the quality used when Exporting to the JPEG (JFIF) image format.

Option: `jpeg_quality`
Default: 75

`magnification (Magnification)`

Set the initial *Magnification* when Exporting or Printing.

Option: `-magnification`
Default: 100

`multiple (Multiple)`

Set the initial selection of *Pages* when Exporting or Printing.
Option: -multiple
Default: off

paper_size (Papersize)

Set the initial *Paper Size* when Exporting or Printing. See paper sizes.
Option: -paper_size
Default: Letter (when unit is inches), or A4 (when unit is cm)

transparent *xfig color number* (Transparent)

Set the "transparent" color when Exporting to the GIF image format. This is one of the **xfig** colors, which are numbered from -3 (Background), -2 (None), -1 (Default), 0 (Black), 1 (Blue) etc. See Colors.
Option: -transparent_color
Default: -2 (None)

Resources for Fonts or Colors, etc.

boldFont (Font)

Set the font to use for file names or confirmation messages, etc.
Option: -bold
Default: 8x13bold

buttonFont (Font)

Set the font to use for the labels of most buttons.
Option: -button
Default: 6x13

normalFont (Font)

Set the font used in message windows, etc.
Option: -normal
Default: 6x13

canvasbackground (Background)

Set the background color of the canvas.
Option: -cbg
Default: ivory

canvasforeground (Foreground)

Set the default color of objects.
Option: -cfg
Default: black

depth (Depth)

Set depth of the visual.
Option: -depth

Default: default screen depth

`dontswitchcmap (Dontswitchcmap)`

Setting this *ON* will inhibit switching to a private colormap even if not enough colors are available in the default colormap.
Option: `-dontswitchcmap`
Default: off

`jpeg_quality` *quality* `(Quality)`

Set the quality used when Exporting to the JPEG (JFIF) image format.
Option: `jpeg_quality`
Default: 75

`monochrome (Monochrome)`

Setting this *ON* will make **xfig** use black and white only for display.
Option: `-monochrome`
Default: off

`visual (Visual)`

Use *visualname* as visual.
Option: `-visual`
Default: default screen visual

Resources for the Size of Windows, etc.

`but_per_row (But_per_row)`

Specify the number of buttons wide the mode panel should be.
Option: `-but_per_row`
Default: 2

`internalborderwidth (InternalBorderWidth)`

Set the width of lines between all buttons and panels.
Option: `-internalBW`
Default: 2 (pixels)

`justify (Justify)`

If this is *ON*, drawing and editing mode panel will put to the right of the canvas.
Option: `-left`, `-right`
Default: off (left)

`pheight (Height)`

Set the height of the canvas.
Option: `-pheight`
Default: 8.5 (inches) (9 inches in portrait mode)

`pwidth (Width)`

Set the width of the canvas.
Option: `-pwidth`
Default: 11 (inches) (8.5 inches in portrait mode)

`showallbuttons (ShowAllButtons)`

If this is *ON*, **xfig** will always show all the attribute buttons instead of only those relevant to the current drawing mode.
Option: `-showallbuttons`
Default: off

`rulerthick (RulerThick)`

Set the thickness of top and side rulers.
Option: `-rulerthick`
Default: 24 (pixels)
Minimum: `24`

Other Resources

`balloon_delay (Balloon_Delay)`

Specify the delay time in milliseconds before the information balloons popup.
Option: `-balloon_delay`
Default: 500 milliseconds

`browser (Browser)`

Specify the web browser to use to view the HTML help files.
Default: `netscape -remote 'openFile(%f)' || netscape %f`

`debug (Debug)`

Setting this *ON* will turn on debugging mode.
Option: `-debug`
Default: off

`ghostscript` *name* `(Ghostscript)`

The name of the ghostscript program (default gs).
Option: `-ghostscript`
Default: gs

`flipvisualhints (Hints)`

Setting this *ON* will flip the left/right mouse indicator messages for mice whose buttons have been switched.
Option: `-flipvisualhints`
Default: off

`grid_color` *color* `(Color)`

Draw the grid on the canvas in this color (default black).
Option: `-grid_color`

Default: black

`icon_view (View)`

Show the icon view of the library objects. See also library_icon_size.

Option: `-icon_view`

Default: True

`image_editor (ImageEditor)`

Specify the program to be invoked when the *Edit Image* button on the Edit Panel for PICTURE object is clicked.

Option: `-image_editor`

Default: `xv`

`keyFile (KeyFile)`

Specify the file to be used as the compose key (international characters) database.

Option: `-keyFile`

Default: `$XFIGLIBDIR/CompKeyDB`

(where $XFIGLIBDIR is the variable in the Imakefile for **xfig**)

`library_dir (libraryDirectory)`

Specify the directory where **Fig** object directories are stored. When the Library button is pressed this path will be parsed for **Fig** files and sub-directories. Each sub-directory found will be used as an object library and will appear in the pull-down menu in the library popup panel.

Option: `-library_dir`

Default: `~/xfiglib`

`library_icon_size (Dimension)`

Set the size of the library icons in pixels. See also icon_view.

Option: (`-library_icon_size`)

Default: 60

`max_image_colors (Max_image_colors)`

Limit the number of colors used for imported images.

Option: `-max_image_colors`

Default: 64

`pageborder` *color* `(Color)`

Draw the page border on the canvas in this color. The page border is turned on by the `-showpageborder` option (resource Fig.showpageborder), and shows the edges of the current *export* paper size.

Option: `-pageborder`

Default: light blue

`pdfviewer (Viewer)`

Specify the external program to use to view the PDF help files.

Default: `acroread %f`

`showballoons (showBalloons)`

Popup popup balloon messages when the cursor passes over buttons etc.
Option: `-showballoons / -dontshowballoons`
Default: on (show balloons)

`showdepthmanager (Hints)`
Show the depth manager to the right of the side ruler.
Option: `-showdepthmanager / -dontshowdepthmanager`
Default: on (show depth manager)

`showlengths (Debug)`
Makes **xfig** show the lengths of lines being drawn, in red text near the line itself. Also, when points are moved or added. In addition, imagine a triangle formed with the line segment as the hypotenuse, and a vertical and horizontal line forming the other two sides. These lines and their lengths are also drawn in red as the point is moved or added.
Option: `-showlengths / -dontshowlengths`
Default: off

`shownums (Debug)`
Makes **xfig** show the number of each vertex of objects on the canvas. This is probably only useful for debugging of Fig files.
Option: `-shownums`, `-dontshownums`
Default: off (don't show numbers)

`showpageborder (Debug)`
Makes **xfig** show the border of the current *export* paper size in the color specified by the `-pageborder` option (resource Fig.pageborder).
Option: `-showpageborder`
Default: on

`spellcheckcommand (spellCheckCommand)`
Specify the external spell check program to use when checking spelling of text objects.
Option: `-spellcheckcommand`
Default: spell

`splash (View)` NEW
Show or don't show the startup screen on startup.
Option: `-nosplash`
Default: True

`spinner_delay (spinnerDelay)`
Specify the delay time in milliseconds before a spinner counts automatically, while the up- or down-arrow is being pressed.
Option: `-spinner_delay`
Default: 500 milliseconds

`spinner_rate (spinnerRate)`
Specify the rate in milliseconds at which a spinner counts automatically.
Option: `-spinner_rate`

Default: 100 milliseconds

trackCursor (Track)

Setting this *ON* will turn on cursor (mouse) tracking arrows.
Option: -track and -notrack
Default: on

Fig*browse_panel*mask*string

Set the initial value of *Filename Mask* of Browse Panel. There may be multiple wildcard strings separated by blanks or tabs
Default: *.gif* *.jpg* *.pcx* *.png* *.xpm* *.xbm* *.ps* *.eps*

Fig*file_panel*mask*string

Set the initial value of *Filename Mask* of File Panel. There may be multiple wildcard strings separated by blanks or tabs
Default: *.fig *.fig.gz *.fig.[Zz]
(this will match compressed, gzipped and normal .fig files)

Fig*job_params*string

Set the initial value of *Print Job Params* of Print Panel. This resource may be used to set any options which should given to lpr (lp on System V system) when printing.
Default: ""

Fig*printer*string

Set the initial value of *PostScript Printer* of Print Panel. If this resource is not set and the environment variable $PRINTER is set, it is used as the initial value.
Default: ""

Resources for Internationalization i18n

These resources are effective only for **xfig** with internationalization facility (I18N Imakefile variable enabled).

international (International)

Set if the internationalization facility should be enabled.
Option: -international
Default: off

boldFontSet (BoldFontSet)

Set the fontset used for drawing international text when *Times-Bold* is selected. This is effective only if internationalization facility is enabled. See also normalFontSet.
Default: -*-times-bold-r-normal--16-*-*-*-*-*-*-*, -*-*-bold-r-normal--16-*-*-*-*-*-*-*, -*-*-*-r-*--16-*-*-*-*-*-*-*

eucEncoding

If this is *ON*, it is assumed that EUC encoding is used for international text. Normally, EUC should be used. This is effective only if internationalization facility is enabled.
Default: on

`fig2devLocalizeOption`
Specify the option to be added when invoking **fig2dev**. This is effective only if internationalization facility is enabled, and the string will added as an option when invoking **fig2dev**.
Default: `-j`

`fixedFontSet (FontSet)`
Set the fontset used for displaying text in conversion or status when entering text with *Off the Spot* or *Over the Spot*. This is effective only if internationalization facility is enabled.
Default: `-*-times-medium-r-normal--16-*-*-*-*-*-*-*, -*-*-medium-r-normal--16-*-*-*-*-*-*-*, -*-*-*-r-*--16-*-*-*-*-*-*-*`

`fontMenuLanguage (Language)`
Specify the language to be used for label of TEXT FONT. `japanese` and `korean` are supported in the current release, and label of *Times-Roman* and *Times-Bold* will be replaced by those for the language. Normally, you don't have to set this because **xfig** will choose it from the locale if this is not explicitly specified. This is effective only if internationalization facility is enabled.
Default: ""

`inputStyle`
Set the input style when entering international text with *input method*. The style may be selected from `OffTheSpot`, `OverTheSpot` and `Root`. If this is set to `None`, input method will not used. This is effective only if internationalization facility is enabled.
Option: `-inputStyle`
Default: `OffTheSpot`

`latinKeyboard (LatinKeyboard)`
If this is set ON, **xfig** will accept input via input-method even if a font other than *Times-Roman* and *Times-Bold* is selected in TEXT FONT. This may be useful when European keyboard which can enter Latin-1 characters is used, for example.
Default: off

`normalFontSet (NormalFontSet)`
Set the fontset used for drawing international text when *Times-Roman* is selected. This is effective only if internationalization facility is enabled. See also `boldFontSet`.
Default: `-*-times-medium-r-normal--16-*-*-*-*-*-*-*, -*-*-medium-r-normal--16-*-*-*-*-*-*-*, -*-*-*-r-*--16-*-*-*-*-*-*-*`

Environment Variables

`XFIGTMPDIR`

Specify the directory where temporary files should stored. If this is not set, the `/tmp` directory will be used.

`FIG2DEV_DIR`

Specify the path where the **fig2dev** executable resides. If this is not set, **xfig** relies on the user's normal command search path to run **fig2dev**.

`PRINTER`

Specify the default printer when printing figures with Print. If the `Fig*printer*string` resource is set, this variable will ignored.

`LANG` i18n

Set the language to be used. For Japanese, a locale name like `japanese` or `ja_JP.eucJP` may be used.

`XMODIFIERS` i18n

Specify the *input method* to be used for entering text. If this is not set, the default input method will be used.

`FIG2DEV_LIBDIR` i18n

Specify the directory where the files needed by international-**fig2dev** (`japanese.ps`, etc.) are placed. If this is not set, the setting when compiling **fig2dev** will be used.

[Contents | Introduction | Credits]

Xfig User Manual

Version 3.2.4 December 2002

Printing and Exporting

- Print
- Export
- LaTeX and Xfig
- Generating an HTML Image Map

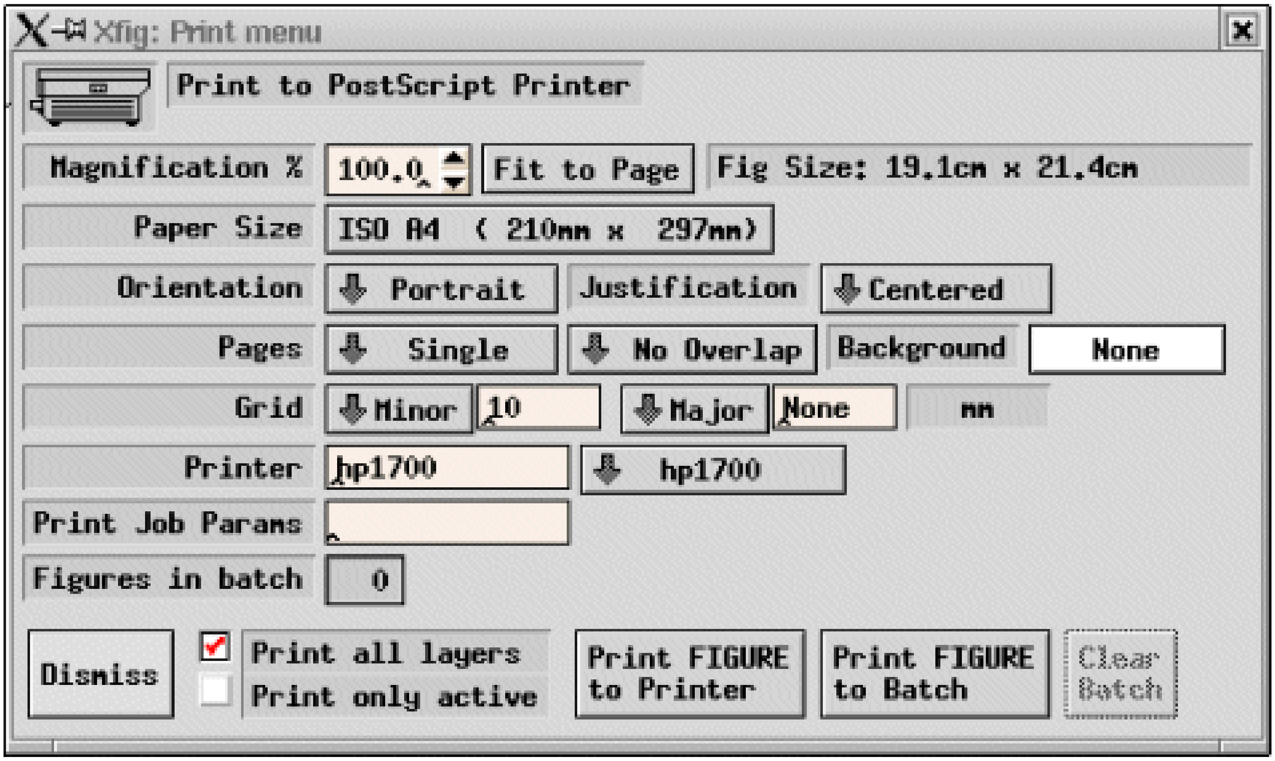
To export or print a **Fig** file, **xfig** calls **fig2dev**, the post-processor which converts the **Fig** file to the desired output language. For printing, this is PostScript. For exporting, there are a dozen or so languages, including PostScript, EPS, LaTeX, MetaFont, and bitmap formats such as GIF, JPEG, PPM and several others. See the *Language* description of the Exporting section.

This means that you must have **fig2dev** installed on your system along with **xfig**. **Fig2dev**, which is part of the **TransFig** distribution is always available where you find **xfig**. See Getting and Installing Xfig for details.

Printing

The *Print...* entry in the *File* menu (or accelerator `Meta-P`) provides the facility to print figures to PostScript printers. Use File/Export if you want to store the output to a file.

The accelerator `Shift-Meta-P` will print using the previously set paramters (i.e. the print panel isn't popped up, but the figure is printed).



Xfig: Print menu

Print to PostScript Printer

Magnification % 100.0

Fig Size: 19.1cm x 21.4cm

Paper Size ISO A4 (210mm x 297mm)

Orientation Portrait

Justification Centered

Pages Single

No Overlap

Background None

Grid Minor 10 Major None mm

Printer hp1700 hp1700

Print Job Params

Figures in batch 0

☑ Print all layers

□ Print only active

- *Magnification*

  Specify the magnification when printing figure in percent of full size (%). The default is 100% and may be set by the `Fig.magnification` resource.

- *Fit to Page*

  Clicking this button will set the *Magnification* automatically so that figure size will just fill current *Paper Size* with at least 1/2 inch margin all around.

- *Orientation*

  Specify the orientation of the output as *Landscape* (horizontal) or *Portrait* (vertical). The default orientation is same as the orientation of the canvas which may be changed by *Portrait/Landscape* in the *View* menu.

- *Justification*

  Specify if the figure should be *Flush Left* or *Centered* on the paper of size selected by *Paper Size*.

- *Paper Size*

  Specify the size of the paper. The following paper sizes are available:
  - Letter (8.5in x 11in)
  - Legal (8.5in x 14in)
  - Tabloid (17in x 11in)
  - ANSI A (8.5in x 11in)
  - ANSI B (11in x 17in)
  - ANSI C (17in x 22in)

  - ANSI D (22in x 34in)
  - ANSI E (34in x 44in)
  - ISO A9 (37mm x 52mm)
  - ISO A8 (52mm x 74mm)
  - ISO A7 (74mm x 105mm)
  - ISO A6 (105mm x 148mm)
  - ISO A5 (148mm x 210mm)
  - ISO A4 (210mm x 297mm)
  - ISO A3 (297mm x 420mm)
  - ISO A2 (420mm x 594mm)
  - ISO A1 (594mm x 840mm)
  - ISO A0 (840mm x 1189mm)
  - JIS B10 (32mm x 45mm)
  - JIS B9 (45mm x 64mm)
  - JIS B8 (64mm x 91mm)
  - JIS B7 (91mm x 128mm)
  - JIS B6 (128mm x 182mm)
  - JIS B5 (182mm x 257mm)
  - JIS B4 (257mm x 364mm)
  - JIS B3 (364mm x 515mm)
  - JIS B2 (515mm x 728mm)
  - JIS B1 (728mm x 1030mm)
  - JIS B0 (1030mm x 1456mm)

- *Pages*

  If *Multiple* is selected here, the figure will be split into multiple pages if the size of the figure is larger than *Paper Size*. It allows the user to output a figure larger than paper size by pasting those papers together (to make it easier, **xfig** generates output so that the parts of the figure will be overlap).

  If *Single* is selected here, this facility will not be used; any part of the figure outside the paper boundary will be clipped.

- *Background*

  This will set the background color of the whole figure when printing. The default is white.

- *Grid* NEW

  You may print a light-gray grid on the page by choosing a minor (thin) and/or a major (thick) grid spacing from pulldown menus or entering the spacing directly in the entry. The units of grid spacing are inches or mm, depending on the units of the figure.
  The grid options is only available for PostScript, Encapsulated PostScript, PSLatex, PDF and all the bitmap export formats.

- *Printer*

  Specify the printer name output should be directed to. If this field is empty, output will be directed to the default printer.
  The default value is specified by the resource `Fig*printer*string` or the environment variable `$PRINTER`.

  If your system uses **/etc/printcap** to define printers, **xfig** will make a pulldown menu of printers next to the entry.

- *Print Job Params*

  The string specified here will be passed as command-line options when executing `lpr` (`lp` on System V system). If `%f` is included in the string, (it may appear more than once) it will be replaced by the name of the figure.

  The default is empty, but it may be specified by `Fig*job_params*string` resource.

- *Figures in batch*

  This indicator shows how many figures have been put in the *batch file* for printing. Figures may be printed into the *batch file* by *Print FIGURE to Batch*, and those figures may be sent to the printer as one print job by clicking on *Print BATCH to Printer*.

- *Dismiss*

  Clicking this button will close the Print panel. The accelerator `Meta-C` will also perform this function.

- *Print all layers/Print only active*

  You may print the whole figure (all layers) or just the layers that are active according to the layer manager.

- *Print FIGURE/BATCH to Printer*

  Clicking this button will spool the figures in the *batch file* if any, or the current figure if none, to the printer. The accelerator `Meta-P` will also perform this function.

  The label of this button will be *Print BATCH to Printer* if there are any figures in the batch file, or *Print FIGURE to Printer* if there are none.

  When printing to the printer, **xfig** will first convert the figure to PostScript with **fig2dev** program, and pass the result to `lpr` (`lp` on System V system). When executing `lpr` (or `lp`), the printer name specified by *PostScript Printer* field and options specified by *Print Job Params* will be passed as command-line options.

- *Print FIGURE to Batch*

  Clicking this button will append the current figure to the *batch file*. The accelerator `Meta-B` will also perform this function.

  The figures stored in the *batch file* will be printed to the printer when *Print BATCH to Printer* is clicked later. You can use this facility when you want to send some figures to the printer at one time.

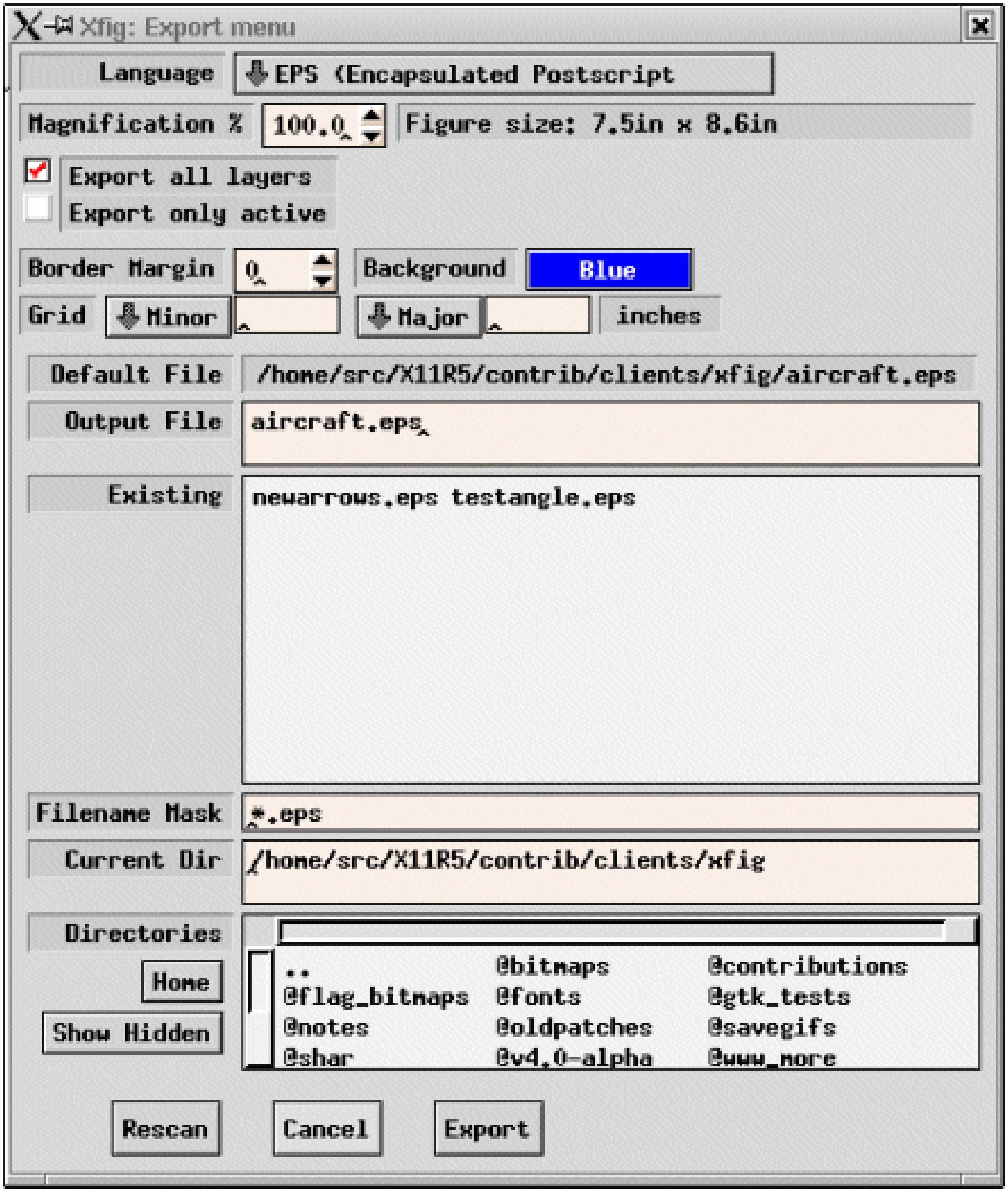
- *Clear Batch*

  Clicking this button will erase the accumulated figures in the *batch file*. The accelerator `Meta-X` will also perform this function. The figures are automatically deleted from the batch file when *Print BATCH to Printer* is clicked.

Exporting

The *Export...* entry in the *File* menu (or accelerator `Meta-X`) provides the facility to output the figure in various format such as PostScript, GIF, JPEG, HP-GL, etc. to a file. This is useful when you want to read figures by other applications (LaTeX or FrameMaker, for example). See LaTeX and Xfig for hints about using **xfig** with LaTeX.
The accelerator `Shift-Meta-X` will export using the previously set paramters (i.e. the export panel isn't popped up, but the figure is exported).

Use File/Print if you want to print the figure to a PostScript printer.



Xfig: Export menu

Language: EPS (Encapsulated Postscript

Magnification %: 100.0 Figure size: 7.5in x 8.6in

☑ Export all layers
□ Export only active

Border Margin: 0 Background: Blue

Grid: Minor Major inches

Default File: /home/src/X11R5/contrib/clients/xfig/aircraft.eps

Output File: aircraft.eps

Existing: newarrows.eps testangle.eps

Filename Mask: *.eps

Current Dir: /home/src/X11R5/contrib/clients/xfig

Directories:

```
..               @bitmaps         @contributions
@flag_bitmaps    @fonts           @gtk_tests
@notes           @oldpatches      @savegifs
@shar            @v4.0-alpha      @www_more
```

- *Language*

  Specify the format (language) to be generated as output. The default is *Encapsulated PostScript*, but may be changed with the resource `Fig.exportLanguage`.

  The following formats are available:

  Vector formats:

  - PostScript
  - Encapsulated PostScript (EPSF)

- EPS With ASCII (EPSI) preview NEW
- EPS With monochrome TIFF preview NEW
- EPS With color TIFF preview NEW
- PDF (Portable Document Format)
- Combined PostScript/LaTeX
- Combined PDF/LaTeX NEW
- LaTeX picture environment
- PicTeX macros
- IBMGL (HP-GL)
- Textyl \special commands
- TPIC
- PIC
- HTML Image map
- MF (MetaFont)
- MP (MetaPost) NEW
- MMP (Multi-MetaPost) NEW
- CGM (Computer Graphics Metafile - useful to import into Microsoft WORD, etc.)
- Binary CGM NEW
- EMF (Enhanced Metafile) NEW
- SVG (Scalable Vector Graphics - beta driver) NEW
- Tk (Tck/Tk toolkit canvas)

Bitmap formats:

- GIF Graphic Interchange Format
- JPEG
- PCX Paintbrush format
- PNG Portable Network Graphics
- PPM Portable Pixmap
- SLD (AutoDesk slide format)
- TIFF Tag Image File Format
- XBM X11 Bitmap
- XPM X11 Pixmap

As a variation of the LaTeX format, *epic*, *eepic* and *eepicemu* macros are also available. It is also possible to output the text part of the figure in LaTeX and the graphics part in PostScript using *Combined PostScript/LaTeX*. This is especially useful when complex numerical formulas are included in the figure (see also TEXT FLAGS).

Not all of the features in **xfig** are supported by all export languages. For example, imported pictures are not supported for IBMGL export. The PostScript export language supports all features of **xfig** and a fairly high quality output will be generated.

The **fig2dev** program, part of the **TransFig** package available with **xfig** does the actual conversion from Fig to the output language.

To export the figure in a bitmap format such as GIF or JPEG, you must have the GhostScript and netpbm packages on your system.

- *Magnification*

  Specify the magnification when exporting figure in percent of full size (%). The default is 100% and may be set by the `Fig.magnification` resource.

- *Export all layers/Export only active*

  You may export the whole figure (all layers) or just the layers that are active according to the layer manager.

- *Border Margin*

  When exporting to PostScript, Encapsulated PostScript, HTML MAP, or any of the bitmap formats (e.g. GIF, JPEG, etc.), you may add a margin space around the figure. The size of the margin is in pixels or 1/80th inch.

- *Background*

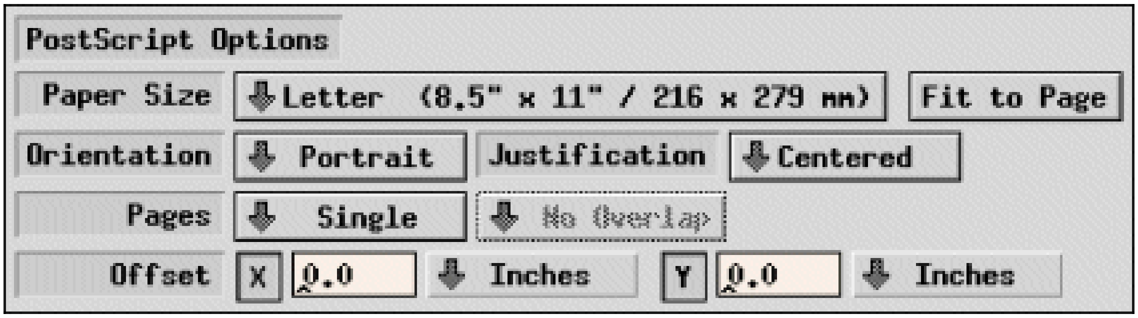
  This will set the background color of the whole figure when printing. The default is white.

- *Grid* NEW

  You may add a light-gray grid to the page by choosing a minor (thin) and/or a major (thick) grid spacing from pulldown menus or entering the spacing directly in the entry. The units of grid spacing are inches or mm, depending on the units of the figure.
  The grid option is only available for PostScript, Encapsulated PostScript, PSLatex, PDF and all the bitmap export formats.

  The default is *None*.

- PostScript Options - These options will appear below the Grid options when PostScript is selected as the export language.



PostScript Options

Paper Size | Letter (8.5" x 11" / 216 x 279 mm) | Fit to Page

Orientation | Portrait | Justification | Centered

Pages | Single | No Overlap

Offset | X | 0.0 | Inches | Y | 0.0 | Inches

- *Paper Size*

  Specify the size of the paper. See description in Print Panel about available paper sizes. This is effective only if *PostScript* is selected at *Language*.

- *Fit to Page*

  Clicking this button will set the *Magnification* automatically so that the figure size will just fill current *Paper Size* with at least 1/2 inch margin all around. This is effective only when *PostScript* is selected as *Language*.

- *Orientation*

  Specify the orientation of the output as *Landscape* (horizontal) or *Portrait* (vertical). The default orientation is same as the orientation of the canvas which may be changed by *Portrait/Landscape* in the *View* menu.

- *Justification*

  Specify if the figure should be *Flush Left* or *Centered* on the paper of the size selected by *Paper Size*. This is effective only if *PostScript* is selected at *Language*.

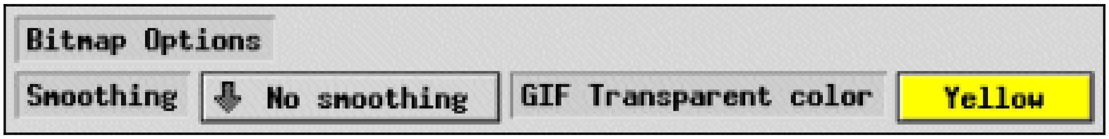
- *Pages*

  If *Multiple* is selected, the figure will be split into multiple pages if the figure is larger than *Paper Size*. If *Single* is selected, this facility will not used. This is effective only if *PostScript* is selected at *Language*.

- *Offset*

  When exporting figure, the figure will be shifted to the right or down by the amount specified here. Use negative numbers to shift it left and/or up. The unit of the amounts may be selected from *Inches*, *Centimeters*, and *Fig Units*(1/1200 inch in version 3.x).

- Bitmap Options - These options will appear below the Grid options when any bitmap format is selected as the export language.



Bitmap Options

Smoothing: No smoothing

GIF Transparent color: Yellow

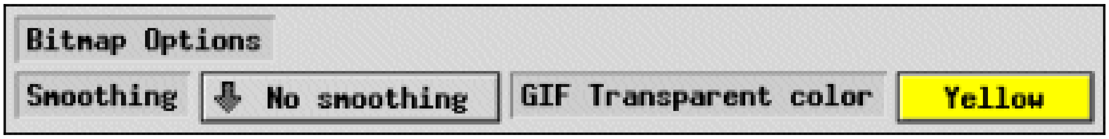
- *Smoothing*

  This will smooth the image by calling **fig2dev** with either the `-S 2' (*Some smoothing*) or -S 4 (*More smoothing*) option to tell GhostScript to render at 2x magnification which improves font rendering, then passes through pnmscale to reduce to original size, which also smooths the image by averaging colors of adjacent pixels.

  The default is *No smoothing*.

- *Transparent Color*

  For GIF export, it is possible to specify one of the colors as ``transparent''. When displaying the figure with GIF viewers that support Transparent GIF (such as Netscape Navigator, for example), the color will not appear but the background of the viewer will show through in place of the color. The default is *None*.



Bitmap Options

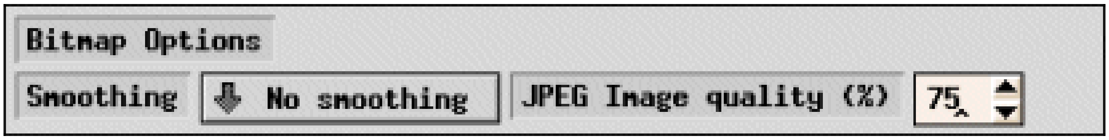
Smoothing: No smoothing

GIF Transparent color: Yellow

- *JPEG Image quality*

  If the export language is **JPEG**, an entry to select the "quality factor" appears.

  The default is *75*.



Bitmap Options

Smoothing: No smoothing

JPEG Image quality (%): 75

- *Default File*

  Output will be written to this file if *Output File* is empty.

  This file name is the figure name plus an extension that reflects the output format at the default, and it will be changed to the specified file name if export has been performed by specifying a file name in *Output File*.

- *Output File*

  Specify the file name the output should be written to. If this field is empty, the file name in the *Default File* field will be used.

  The file name in the *Output File* field may be changed by selecting a file name in the *Fig Files* list, or typing the file name from keyboard directly. If *Return* is typed after file name is entered, export to the file will be performed as if the *Export* button was clicked.

- *Alternatives*

  The list of files in the current directory (only files matching the pattern specified by *Filename Mask*) are displayed, and users may select a file for output from the list.

  Clicking a file name in this list with mouse button 1 will copy the file name to the *Output File* field. Double-clicking a file name in this list with mouse button 1 will cause exporting to the file as if *Export* button was clicked. Note that exporting to the existing file will over-write the old contents of the file.

- *Filename Mask*

  Only the files matching this pattern will be put in the *File Alternatives* list. The pattern is similar to the one used by the UNIX shell, and it is possible to use meta-characters like ``*" or ``?".

  Typing *Return* in this field will cause rescan of the current directory as if *Rescan* button was clicked. This string will be changed according to the language selected with the *Language* menu.

- *Current Dir*

  This shows the current directory, and files in the directory will be displayed in the *Alternatives* list.

  The directory name in the *Current Dir* field may be changed by clicking a directory name in *Directories* list, or by typing the directory name from keyboard directly. If *Return* is typed after directory name is entered, the directory will scanned as if *Rescan* button was clicked and the contents of *Alternatives* list will be updated.

- *Directories*

  The list of directories in the current directory is displayed here, and clicking any item in this list with mouse button 1 will cause a move to the directory. Normally, *hidden directories* are not displayed here, but this may be toggled by *Show Hidden* button.

  ``..'' indicates the parent directory. Moving to the parent directory may also be performed by clicking mouse button 3 on the *Alternatives* list or the *Directories* list.

- *Home*

  Clicking this button will move to the home directory of the user.

- *Show Hidden*

  This button controls if hidden directories (directories whose names start with ``.'') should be displayed or not. Clicking this button will toggle the state. Normally, hidden directories are not displayed.

- *Rescan*

  Clicking this button will scan files in the current directory and update the *Alternatives* list. The accelerator `Meta-R` will also perform this function.

- *Cancel*

  Clicking this button will close the Export panel. The accelerator `Meta-C` will also perform this function.

- *Export*

  Clicking this button will export to the file specified by *Output File* field if any, or the file in *Default File*. The accelerator `Meta-X` will also perform this function.

  When trying to export to an existing file other than *Default File*, popup panel will appear and the user will asked to confirm the export operation. If the figure is exported to a file other than *Default File*, then *Default File* will be set to the actual export file name.

Generating an HTML Image Map

It is possible to generate image map (clickable map) of HTML 3.2 by selecting *HTML Image Map* as *Language* on the Export panel.

To use this facility, using *Comments* on the Edit panel, comment like:

```
HREF="url" ALT="string"
```

must be set for objects you want to make it clickable. Here, `url` is URL of the target of the link, `string` is alternative string for browsers which will not display images (`ALT` attribute is required in HTML 3.2). `string` will be used as label of alternative text links **xfig** will generate with the image map, too.

TEXT objects can't be used for links. CIRCLE, ELLIPSE, SPLINE and ARC will be approximated with polygons. Open objects such as POLYLINE or OPEN SPLINE will be treated as if it is closed. ARC-BOX will treated as if is is a BOX.

[Contents | Introduction | Credits]

Xfig User Manual

Version 3.2.4 December 2002

- New Features in *xfig 3.2.4*
- New Libraries and Examples

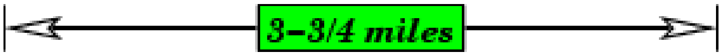
New Features in *xfig 3.2.4*

- The xfig documentation is nearly entirely GIF-free (TM). Only the frames images use GIF because they use a transparent background, which many browsers don't honor. PNG is used for all other images.

- Xfig reference manual now available in PDF besides html. The English version is in Doc/xfig_en.pdf and the Japanese in Doc/xfig_jp.pdf. These are also available in the Help menu.

- Binary and source RPM files available for systems with rpm (RedHat Package Manager).

- Dimension lines - Hold the shift key down while drawing a polyline, and after specifying the two endpoints a "dimension line" is created, which is actually a compound object containing the line drawn (with arrowheads if auto arrow mode is on) in the current color, "tick" lines at each endpoint, text showing the length of the line in the current units, and a box around the text filled in the current fill color. If this compound object is rescaled or one or more endpoints are moved, the actual length is recalculated and the text is updated.



- New Xfig logo courtesy of Marc-Aurele Darche.

- Can now paste text in text mode using Shift+Middle-mouse-button in addition to F20 and Sun's Paste key (F18).

- If xfig is compiled with #define WHEELMOUSE uncommented in the Imakefile, the wheel may be used for several things (see Miscellaneous/Wheel Mouse).

- New grid/positioning option for inches mode: -grid_unit. If the value given is "1/10", "ten", "tenth", "1/10" or "10", (e.g. -grid_unit ten) then the grid and point positioning is in decimal inches. Otherwise they are in 1/16 inch units (as before). When in this mode, the grid step sizes are 0.1, 0.2, 0.5 and 1 inch and the positioning steps are 0.05, 0.10, 0.2, 0.5, and 1.0 inch.

- Background (-g to fig2dev) option for tk export honored now.

- New command-line argument -correct_font_size (X resource Fig.correct_font_size) to make xfig use real points for font sizes instead of units of 1/80 inch. xfig has always claimed that the font size units are points (1/72inch), but in fact they were 1/80 inch. This new option is not the default because it will break some figures by making the text too large compared to other Fig objects, but it will be the default in a future version. The corresponding argument to fig2dev is -F (xfig automatically sets this when calling fig2dev).

- Added "Stop" button to stop loading library.

- Much more efficient when copying large imported pictures - existing pixmap is used instead of creating new one. This also precludes the need to remap colors when running out of colors when the colors are identical.

- When changing an attribute of an imported picture that doesn't change the picture itself (e.g. depth), the pixmap is not regenerated, therefore saving time. The same is true when copying a picture.

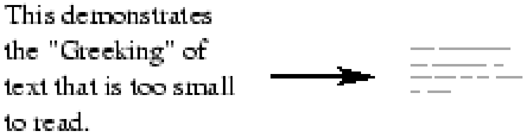
- New X resource Fig.ghostscript and command-line argument -ghostscript by which you can tell xfig what your ghostscript is called (e.g. "gs" or "gswin32").

- Can now add ASCII (EPSI) preview to EPS export. See export panel.

- Can now add monochrome or color TIFF preview to EPS export. This is needed by Microsoft Word, Powerpoint, etc. to display the EPS on the screen. Select from the pulldown menu of export languages. You must have fig2dev 3.2.4 for these features. See export panel.

- Translations of LATEX.AND.XFIG document into two popular Chinese encodings (GB2312 and Big5). These are called LATEX.AND.XFIG.zh_CN and LATEX.AND.XFIG.zh_TW respectively. From Mendel Chan

- When text becomes too small to see (e.g. small zoom), xfig will "Greek" it, meaning it draws a gray line segment for each word in the string.


This demonstrates the "Greeking" of text that is too small to read.

- When resizing canvas to a smaller height, the number of buttons per row of the mode (side) panel increases so that it doesn't protrude into the indicator (bottom) panel

- Smarter redraw when turning on depths from the depth manager. If depth N is turned on and there are no active depths < N, only depth N is drawn - otherwise, the canvas is erased and the whole figure is redrawn.

- If the user's pointer (mouse) buttons are switched (usually 3 2 1) xfig will automatically flip the mouse hints messages. This overrides the -flipvisualhints command-line argument.

- If the user clicks on or modifies an existing text, and presses return, the new text on the next line will use the attributes of the first text instead of the attributes from the indicator panel.

- Note in README and Imakefile about not casually setting the DESTDIR make variable.

- Picture objects are drawn as filled gray boxes when their layer is inactive in gray mode.

- xfig no longer requires fonts that are named "-adobe-..." (e.g. for RedHat systems with URW fonts).

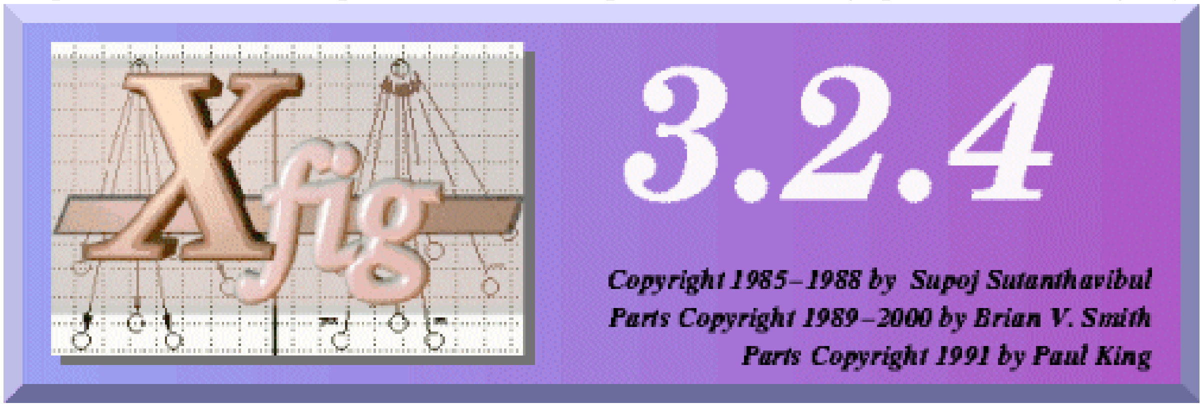
- Updating the attribute panel from a compound object will retrieve (only) the smallest depth from the compound.

- When scaling an object, the line thickness is no longer scaled with it

- New measuring modes: angle, length and area. See Measure Angle/length/area. From Alex Durner.

- New feature to add tangents and normal lines to curves. See Add Tangent. From Alex Durner.

- Splash screen on startup. Disable with -nosplash (resource Fig.splash). Interior logo by Marc-Aurèle Darche.


Xfig
3.2.4
Copyright 1985–1988 by Supoj Sutanthavibul
Parts Copyright 1989–2000 by Brian V. Smith
Parts Copyright 1991 by Paul King

- New command-line option -grid_color (resource grid_color) to set the color of the canvas grid (default: red).

- New option to print grid when exporting or printing (see printing/ exporting section in html docs).

- When drawing an arc by specifying the center point, a circle is drawn showing the radius of the arc as the user drags the mouse, then when the first end point is clicked an arc is drawn as the mouse is dragged. The user then clicks in the direction the arc should go, followed by the final angle to complete it.

- If the Ruler unit is Imperial (in), you may have xfig display measurements using fractions where possible. If a measurement can be shown as an integer multiple of 64ths of an inch (or 32nds, 16ths, etc.) it will show it as such (e.g. 3-5/64 in), otherwise a decimal value will be shown (e.g. 3.085 in).

- When opening a compound object, if the middle button us used to open the compound, all other objects on the canvas are kept visible. From Alex Durner.

- When opening a compound object, the "point positioning" indicator is enabled, allowing you to set the bounds of the compound according to the indicator when you close the compound.

- Zero-crossing lines on canvas have own command-line parameter (-zero_lines) and resource (Fig.zero_lines) for setting color separate from page border color.

- Smooth factor when exporting to bitmap format is now a pull-down menu with: No smoothing, Some smoothing, More smoothing.

- When flipping compound objects with text, a vertical flip will now shift the text to maintain its relative position to the other objects in the compound and its angle is rotated by 180 degrees minus the original angle. This behavior will for example, keep the orientation of a text inside a polygon consistent with the polygon. A horizontal flip of a compound containing text will change the angle in the same way.

- Three-D borders for buttons, etc reduced from 2 to 1 pixel for "lean" look (resource Fig*shadowWidth in Fig-color.ad).

- Additional JPEG file format (EXIF) added for import. These usually come from digital cameras.

- Validation for integer spinners - doesn't allow non-numeric input when typing in values directly.

- When scaling an object, the scale factor is shown in the message. window in addition to the width/height. From Alex Durner.

- New url for pstoedit (mentioned in installation.html).

- When picking points of an object at high zoom, the tolerance has been reduced to make it easier to pick the correct point.

- When "graying" inactive depths, the inactive objects are drawn first followed by active objects ensuring visibility of active objects.

- xfig now draws "grayed out" picture objects (inactive depth) as a gray box with the filename only.

- Line style in edit panel now shows graphic of style instead of textual description.

- Cap style in edit panel was inadvertantly left off open-splines.

- Popup dialog to edit behavior of spline point now pops up near the point being edited.

- When cancelling drawing a picture object from the popup editor, the empty object is removed rather than showing a <empty> in the box.

- Pressing Return or Escape in popup message window will dismiss it.

- In picture browser popup, double clicking or clicking and pressing Return will apply the selected picture AND close the browser.

- New popup query if updating the depth in a compound object would result in any depth exceeding the maximum. The popup gives the user the option to cancel the depth update part. From Marcus Eger (eger.m@gmx.de)

- More consistent placement of Cancel/Ok/Set etc. buttons in dialogs. Generally, the Cancel button is first followed by the others, and where possible, they are at the bottom of the dialog.

- More meaningful message "Cut buffer is empty" if there is no .xfig file in the user's home directory.

- New MetaPost driver for TransFig 3.2.4 supported.

- New option for binary CGM export (must have TransFig 3.2.4 too). The ralcgm program is required for this option. See ftp://ftp.cc.rl.ac.uk/pub/graphics/ralcgm/

- In popup edit panel for compound objects, one may easily shift a compound object by changing either corner's x, y values and pressing return or scale it by changing the width and/or height and pressing return.

- "condensed" alias for "narrow" fonts (urw changed their helvetica- narrow to helvetica-condensed).

- Pressing escape in a popup dialog will dismiss it

- Zoom/Unzoom (Shift-Z/z) changes zoom by a factor of 1.5 instead of adding or subtracting 1 from the zoom value. Also, the zoom/unzoom is centered on the canvas.

- Exporting to JPEG doesn't require compiling with USE_JPEG defined in Imakefile.

- Minimum and maximum depths of objects in a compound are displayed in popup edit panel.

- Rulers show units now every 10 units (e.g. 0cm, 10cm, ...)

- "Front" and "Back" labels in depth manager to show relative positioning of depths on drawing.

- If you wish to place a library object at the position it was originally created, e.g. for laser printer labels, hold down the Shift key and press mouse button 3 (`place at orign posn') to place it.

- New URL for chemtool, a chemical structure editor: http://ruby.chemie.uni-freiburg.de/~martin/chemtool/chemtool.html

- Named styles where user can save attributes that are often used for quick access. For example, the user may want dashed, thick, red lines often so he can save those attributes and call it (e.g.) "thickred", and easily call it up later. See the View/Manage Styles...) menu entry.

- The minimum and maximum depth of objects are shown in the edit panel for compound objects

- SVG (Scalable Vector Graphics) beta driver (for TransFig/fig2dev) from Anthony Starks with major improvements by Martin Kroeker (martin@ruby.chemie.uni-freiburg.de). Some caveats when using Mozilla-SVG to view these:

    1. the file must be exported with magnification = 6.6% (0.066 in fig2dev) to scale to screen coords
    2. the file suffix must be ".xml"

- EMF (Enhanced Metafile) export driver (TransFig/fig2dev) based on CGM driver. Converted by M. Schrick

New Libraries and Examples

- Examples moved to Libraries/Examples so they are installed in a convenient place for viewing (via the library feature)

- Poster advertizing German version of "Little Shop of Horrors" in Libraries/Examples/poster.fig. From Kai-Martin Knaak

- Three new Examples from Carlo Kopp (in Libraries/Examples):
  Python-4.fig - RAFAEL Python 4 all aspect agile heatseeking missile
  YF-23A.fig - Development example of the USAF's Northrop-MDC YF-23A
  tsr2-side.fig - Notional camouflages for BAC TSR.2 production aircraft.

- Over 160 new country flags in the Libraries/Flags directory. Now organized by region of the world (e.g. Africa, Oceania).

- New flags of Bosnia and Slovenia with vector coat of arms in Libraries/Flags directory from Roman Maurer

- New maps of Yugoslavia, Macedonia, Croatia, Bosnia-Herzegovina and Slovenia

- New Avery label templates in Libraries/Labels directory, including a CD label, Avery_5931, done by William Chimiak.

- New Avery and PerfectData Corp CD label templates in Libraries/Labels directory, done by Len Hickey.

- New computers and printers in Libraries/Computers library from Ernst Pisch

- New network library devices Extreme Networks Inc. Black Diamond and Summit 5i network switches, Net Gear Hub, and new CODECs in Audiovisual library from Bill Chimiak

- New DSP component library from Jakub Stastny (stastnj1@volny.cz)

- Standard welding bevels library from Erik Engh (Libraries/Welding)

- Canadian map from Seymour Green.

- A Uno hand (card game) drawn by Roland Rosendfeld (Libraries/Examples/uno_hand.fig)

[Contents | Introduction | Credits]

Xfig User Manual

Version 3.2.4

December 2002

Bugs Fixed in *xfig* 3.2.4

- Undo of join/split now works correctly. From Pieter-Tjerk de Boer (ptdeboer@cs.utwente.nl)

- Pulldown menu for object library didn't always go away while loading library

- If -library_dir was relative path, any imported pictures in library files wouldn't be found.

- If in text draw mode and user uses Alt- accelerator, the compose LED would stay on

- Middle mouse emulation for 2-button mice wasn't accepting Alt-Right- click, only Meta-Right-click.

- If one created a new figure and didn't save it before exporting, the export command was malformed resulting in an error message.

- Numeric locale setting moved to routines that read and write Fig files so that the user's numeric convention is used in everything else.

- Point positioning was truncated in the negative X and Y regions of the canvas instead of being rounded.

- -b (border margin) wasn't being passed to fig2dev for PDF and PSTEX output.

- xfig only recalculate the bounds of compound objects being read if there are no bounds following the "6" or they are all 0. This fixes the problem introduced in 3.2.3d where objects that were bound at a certain point positioning were rebound at the current positioning.

- When xfig reads in a .fig file (merge_file or load_lib_obj) it will not put the file into another compound if it contains only one compound. This will prevent the double compounding of library objects or merged files.

- If a compound with more than 64 strings was edited, xfig segfaulted because the limit wasn't checked. This is fixed, and the limit has been increased to 200.

- Typo in figx.h. "#ifdef XAW3d" instead of "#ifdef XAW3D" (capital "D")

- If the XFIGDOCDIR directory didn't already exist, install.doc failed.

- The "make install.all" directive did not install the executable, as the README file said is would. This has been fixed.

- Deleting a region (middle mouse button delete) of a figure resulted in all other objects dissappearing after a screen refresh.

- If the user mistakenly clicked the *left* mouse button on a point where he wanted to *split* a line, the ability to split was turned off.

- Toplevel library directory wasn't included in library menu when it contained .fig files.

- Subdirectories without any .fig files appeared in the library menus, contrary to the documentation.

- If xfig was renamed to anything else, the command-line arguments and X resources weren't parsed by the Intrinsics. This is fixed.

- Some versions of "make" don't like the blank line produced by the "#ifdef I18N" in the install.html section of the Imakefile, so it has been reworked to get around that bug.

- Array placement with horizontally/vertically constrained copy works now

- Html "options" page showed "-pa" as abbreviation for -paper_size, when in fact it is now "-pap" because of the new "-pageborder" option.

- If display zoom was large (> 20) and grid was turned on, very large pixmap was being created for the grid.

- Inconsistancy with "char" vs "unsigned char" in get_canvas_clipboard().

- Text input and editing more efficient when redrawing large fonts.

- Box curve indicator now appears when in "convert box<->arcbox" mode.

- If user called xfig with empty (but existing) file, the filename was not retained for subsequent save.

- Cleaned up redrawing of corner markers when editing objects from popup edit panel.

- Scaling arcbox didn't report scaling messages.

- When scaling rotated ellipses, scaling messages were overwritten.

- Improved redraw when moving/scaling/copying objects and screen is redrawn.

- Drawing units indicator was not updated when reading Fig file.

- Cleaned up "move point" code. When moving a corner point of a box or compound or the radius or diameter of a circle or ellipse, it would first make the object dissappear until the mouse was moved. This doesn't happen anymore.

- Command-line -userunit (X resource Fig.userunit) were overwritten when reading Fig file. Now it is only changed if units are inconsistent with value in file. E.g. if file has "Inches" and xfig has "cm", xfig changes to "in" when reading file.

- A few files were left open that should be closed.

- If the print and/or export panels are up and the figure units are changed from/to Metric/Imperial, the figure size wasn't updated in the print/export panel(s).

- Missing link from editing.html file to "Paste Objects" in main_menus.html.

- Bug in array place item - didn't work when number of X copies and Y copies = 1.

- Bug in **undo** of array place - didn't account for adding back object depths

- (dx, dy) values weren't displayed when moving the midpoints of a line or arc.

- While drawing freehand (middle mouse button start), if the user wanted to delete points by holding down the shift key and pressing the left button, moving the mouse would still add points to the object

- PNGINC Imakefile variable added to specify where png.h is located (was already there in fig2dev)

- The depth counts were erased when deleting a region of a figure.

- If you choose one object from a library to place on the canvas and then popup the library panel and choose another object, the depths from the temporary object following the pointer weren't accounted for.

- Added "Fig*AllowShellResize: false" to the app-defaults file to fix a problem that some window managers have with xfig continually resizing itself

- When saving path of imported picture, xfig thought path was relative to current directory when it wasn't

- Added "|| defined(apollo)" in fig.h to include sys/types.h

- Doc/FORMAT3.2 file had imported picture description line reversed with points lines

- Doc/FORMAT3.2 file had incorrect values for object types in the first part of section 3.

- Segfault when merging a Fig file with only color objects (no drawing objects)

- No message when toggling on/off depth manager

- If xfig couldn't open ~/.xfigrc for writing (e.g. because of permissions) it would segfault

- Segfault when converting a 2-point line to a polygon. Bugfix from Tom Sato.

- Segfault when creating a closed spline with three points, where the last two points are coincident. Bugfix from Tom Sato.

- Typo in attributes.html in describing the selection of the previous or next attribute with mouse buttons 2 and 3 respectively. For mouse button 2 it should read "previous" and for button 3 "next".

- Typo in Doc/FORMAT3.2 and Doc/html/fig-format.html for backward_arrow description: said "(0: no forward arrow, 1: on)" instead of ... no backward ...

- Added $(DUSESMALLICONS) to dependencies for main.c in Imakefile

- Reduced height of object edit panels by only displaying 3 comment lines and 6 x,y pairs before scrolling.

- Special texts weren't filtered by the depth filter when exporting to combined ps/latex or pdf/latex

- Various typographical errors found by Daniel Frèrejacque

- Segfault if fast export (Alt-Shift-X) is done before either the file, export or browse panel was created

- When xfig was compiled with WHEELMOUSE support, zooming was always "centered" even not zooming with wheel

- Typo in Fig.ad resource file for pasting text using F8 key

- When doing a panic save (segfault, etc.), xfig now closes any open compound objects first

- Bug in importing GIF file. File descriptor changed between first open and final close.

- Bug in Imakefile with JPEGINCDIR

- Had to include SimpleMenu.c and header files because of changes to some Xaw SimpleMenu implementations that were incompatible with xfig's own SmeBSB object

- Bug in "Figure size" label in export panel caused segfault sometimes

- Macro call (up_part) in e_update.c caused problems on HP/UX ANSI compiler

- Same compiler takes issue with empty action records in w_style.c: library_actions[] ={}; Removed because it isn't used anymore.

- After pressing the "Reread" button in imported picture edit popup, the Shrink, Expand and Original size buttons were disabled

- Bug in SmeCascadeP.h when not using 3D widget set (Xaw3d)

- One remaining use of sys_errlist[] should have been changed to strerror() in f_util.c

- Pressing Escape while drawing an object caused segfault

- When line type is dotted and the dot gap is < 0.5, XSetDashes() produced "Bad Value" error. Found by Tom Sato

- init_break_tag() and init_break_only() each had an extraneous parameter (loc_tag) which wasn't used and wasn't passed by the caller anyway. This caused havoc with (at least) the gcc optimizer on an Alpha processor. Found by Jay Estabrook.

- Didn't allow smoothing when exporting to AutoCAD slide file.

- Changing browser or pdfviewer in dialog had no effect.

- Fixed problem with RedHat imake rules not installing Doc/xfig.man properly.

- When making a compound object including lines that have arrowheads and large coordinates (>= 100000), an incorrect bounding box was produced. This same bug was in fig2dev 3.2.3d and earlier.

- Drawing an arc by starting with the center and dragging it out now reports the angles of the endpoints after clicking on first endpoint instead of incorrectly reporting the radius.

- Checks that closed splines have at least 3 points when reading Fig file and removes any that have fewer.

- Exporting with color TIFF preview option missing parameter when calling fig2dev

- Export grid option disabled for languages other than PS/EPS/PDF/bitmaps

- Not really a bug, but the message that tells the user when xfig is converting a figure from an older file format to the current format has been removed.

- When changing the height of xfig, the number of buttons per row on the mode (left) panel increase OR decrease as needed.

- Chemical Process Flowsheet library Valve_horizontal and Valve_vertical valves reversed.

- Bug in forcing absolute path for picture objects in libraries

- Bug where xfig may change the Major/Minor printing/export grid tick values from "None" to 0.0

- When an imported picture is compressed but the .fig file has the uncompressed name, export failed.

- Xfig allowed "splitting" a picture box into a polyline. This has been fixed to disallow it.

- Queries user if hasn't saved named styles after adding or deleting same.

- Print-to-batch-file didn't call print_to_file with all parms

- xfig crashed if the window manager deleted the color popup panel in the popup editors

[Contents | Introduction | Credits]

Xfig User Manual

Version 3.2.4 *December 2002*

Getting and Installing Xfig and TransFig

You need the **fig2dev** program in the **TransFig** package in addition to **xfig** to be complete. The **fig2dev** program is the program which allows you to print figures and export them to various languages and formats such as PostScript, GIF, JPEG, etc.

Getting and Installing Xfig

This is the installation procedure of **xfig**. See the `README` file for more details. For information about how to use internationalization facility of **xfig**, see also Internationalization.

1. Get the source of **xfig** (`xfig.3.2.4.full.tar.gz`) and *gunzip* and *untar* the file:
   Either do:
   1. `gunzip -c xfig.3.2.4.full.tar.gz | tar xvf -`
      or
   2. `gunzip xfig.3.2.4.full.tar.gz`
      `tar xvf xfig.3.2.4.full.tar`

   You can get the source from ftp://epb.lbl.gov/xfig, http://www.xfig.org/xfigdist, ftp://ftp.x.org/contrib/applications/drawing_tools/xfig and its mirror sites or from any CTAN machine (ftp://ftp.tex.ac.uk/pub/archive/graphics, for example).

2. Edit the `Imakefile` if you need to customize it:

- If you want to install **xfig** in a directory other than the default X11 binary directory:
  - add **"BINDIR=<directory>"** at the top of the Imakefile, where **<directory>** is the full path of the directory in which you want **xfig** to be installed.
  - Also, you may have to redefine **MKDIRHIER** because make looks for it relative to the **BINDIR** variable.
    Set it to: **"MKDIRHIER = /bin/sh <path>/mkdirhier"** where <path> is the path to the **mkdirhier** script or program.
  - Finally, uncomment and change **XAPPLOADDIR** to the directory where you want the app-defaults resource files to go. You will have to use the environment variable **XAPPLRESDIR** in your shell to point to this directory.
    On Sun systems running Openwindows you probably need to set the environment variable **XUSERFILESEARCHPATH** to point to your app-defaults directory, and append a "%N" to the path, e.g. /users/me/xfig/app-defaults/%N
- To use JPEG, you must have JPEG library (Version 5b or later). If you already has the JPEG library installed in your system area then make sure the USEINSTALLEDJPEG variable is uncommented (the default). If you don't have JPEG installed, get the source from ftp://ftp.uu.net/graphics/jpeg or ftp://ftp.x.org/contrib/libraries and **comment out** the USEINSTALLEDJPEG variable, and set the JPEGLIBDIR variable to the directory where source of JPEG library is stored.
  ** Make sure you delete or rename any older version of the JPEG library you might have on your system. Some Linux systems come with an older version which is incompatible with xfig.

  If you don't want JPEG support, comment out `#define USEJPEG' using the XCOMM comment directive (e.g. XCOMM #define USEJPEG).

- You need the PNG library (-lpng) and the zlib (-lz) compression library. You can find the PNG sources at http://www.libpng.org/pub/png/libpng.html and the zlib sources at ftp://ftp.cdrom.com/pub/infozip/zlib.

- To enable XPM support, uncomment `#define USEXPM' and modify the definition of XPMLIBDIR if necessary. To use XPM, you must have XPM3 package (Version 3.4c or later).

- If you want to use Xaw3d (Three-D Athena Widget) library, uncomment `#define XAW3D', and change the `XAW3DINC variable to point to the directory where your 3D Athena widget headers are located. When **Xaw3d** is used instead of **Xaw**, the buttons will appear 3 dimensional. The **Xaw3d** library should be compiled without -DARROW_SCROLLBAR.

- Small Icons - If you have a small screen (e.g. 800x600 or so) you may want to use the small mode panel buttons for **xfig**. If so, uncomment the `#define USESMALLICONS' line.

- If your system doesn't have strcasecmp() and/or strncasecmp(), uncomment
  `HAVE_NO_STRCASECMP = -DHAVE_NO_STRCASECMP'
  and/or
  `HAVE_NO_STRNCASECMP = -DHAVE_NO_STRNCASECMP'.

- If your system doesn't have strstr(), uncomment `#define NOSTRSTR'.

- If your system doesn't have `strerror()` (but support `sys_errlist`), uncomment `` `NEED_STRERROR = -DNEED_STRERROR' ``.

- NEW If you have a wheel mouse, uncomment the `` `#define WHEELMOUSE' `` line. See wheel mouse for a description of the wheel function in xfig.

- If you want to enable internationalization facility of **xfig**, uncomment ``` ``#define I18N'' ``` (remove (`XCOMM`). See Internationalization for more information.

  Suggestion to Package Maintainers: If you are preparing a package to be distributed to the world in the binary form, it is recommended to enable internationalization facility so that the package can be used also in Japan or Korea.

Check the `README` file for more information about specific systems.

3. Type `xmkmf` to generate a `Makefile` for your system.

4. Type `make` to compile **xfig**.

5. After the compilation has finished, type `make install` to copy the files to the appropriate directories.

   If you don't use `make install` and copy files manually, note that:

   - the `xfig` executable should go in your command search path,
   - `Fig.ad` and `Fig-color.ad` should go in the `app-defaults` directory with the names `Fig` and `Fig-color` respectively,
   - The `CompKeyDB` file , the `Doc/html` directory, and the `xfig-howto.pdf` and `xfig.pdf` files should be copied into the directory specified by the `XFIGLIBDIR` variable in the `Imakefile`

6. Type `make install.man` to copy the manual pages to the appropriate directory.

Installing TransFig/fig2dev

Here is the installation procedure for **TransFig** (**fig2dev**). See `README` file in for details. For information about how to use internationalization facility of **fig2dev**, see also Internationalization.

1. Get the source of **TransFig** (`transfig.3.2.4.tar.gz`), and *untar* the file. It should be available at the same site as **xfig**.

2. You shouldn't need to change the toplevel `Imakefile` or the `Imakefile` in `fig2dev/dev`.
   Edit `fig2dev/Imakefile` if you need to customize it:

   - If you don't want to use NFSS2 font selection for LaTeX text, comment out `` `DDNFSS = -DNFSS' `` using the `XCOMM` directive. If you don't want to use `epsfig` macros as distributes with LaTeX2e, comment out `` `DDEPSF = -DEPSF' `` in the same way.

Hint: Users of LaTeX2e will probably want to include both of these options; users of LaTeX 2.09 will probably want to include neither.

● To use JPEG, you must have JPEG library (Version 5b or later). If you already if the JPEG library installed in your system area then make sure the `USEINSTALLEDJPEG` variable is uncommented (the default). If you don't have JPEG installed, get the source from ftp://ftp.uu.net/graphics/jpeg or ftp://ftp.x.org/contrib/libraries

If you don't want JPEG support, comment out `` `#define USEJPEG' `` using the `XCOMM` comment directive (e.g. `XCOMM #define USEJPEG`).

● You need the PNG library (-lpng) and the zlib (-lz) compression library. You can find the PNG sources at http://www.libpng.org/pub/png/libpng.html and the zlib sources at ftp://ftp.cdrom.com/pub/infozip/zlib.

● To enable XPM support, uncomment `` `#define USEXPM' `` and change `XPMLIBDIR` if necessary. To use this facility, you must have XPM3 package (Version 3.4c or later).

● If your system doesn't have `strcasecmp()` and/or `strncasecmp()`, uncomment

```
    `HAVE_NO_STRCASECMP = -DHAVE_NO_STRCASECMP'
```

and/or

```
    `HAVE_NO_STRNCASECMP = -DHAVE_NO_STRNCASECMP'.
```

● If your system doesn't have `strstr()`, uncomment `` `#define NOSTRSTR' ``.

● If your system doesn't have `strerror()` (but support `sys_errlist`), uncomment `` `NEED_STRERROR = -DNEED_STRERROR' ``.

● On Solaris 2, you must add `-lnsl` to `LOCAL_LIBRARIES`. On HP-UX, you may need to reduce optimization level to `+O2`.

● If you want to enable internationalization facility of **fig2dev**, uncomment ``#define I18N" (remove (`XCOMM`). See Internationalization for more information.

See `README` and `Imakefile` for more details.

3. Type `xmkmf` to generate the `Makefile`.
4. Type `make Makefiles` to generate the `Makefiles`. Note the ``s" in ``Makefiles".
5. Type `make` to compile **TransFig** and **fig2dev**.

6. Type `make install` to copy the executable and files to the appropriate directories.

7. Type `make install.man` to copy the manual pages to the appropriate directory.

Installing Other Software

In addition to **xfig** and **transfig** (**fig2dev**). you will also need to install **GhostScript** and **netpbm**,

- **xfig** invokes **GhostScript** (the `gs` command) when displaying PostScript images, and exporting in bitmap formats (e.g. JPEG, XPM, etc).

  GhostScript must be compiled with pbmraw, ppmraw, pcx256, jpeg, and tiff24nc driver.

  GhostScript can be found at ftp://prep.ai.mit.edu/pub/gnu and its mirror sites, or at ftp.cs.wisc.edu/ghost/aladdin.

- **fig2dev** invokes commands in the **netpbm** package when exporting in some formats.

  Commands that may actually be used are: pbmtoxbm, pgmtopbm, pnmdepth, pnmscale, pnmtopng, pnmtotiff, ppmquant, ppmtoacad, ppmtogif, ppmtopcx, ppmtopgm, ppmtojpeg and ppmtoxpm.

  netpbm can be found at http://download.sourceforge.net/netpbm/ or ftp://ftp.x.org/contrib/graphics or its mirror sites in `/contrib/graphics`.
  Be sure that it includes **ppmtojpeg**. This may be found only in the newer releases, such as 9.0 and 9.1

Related Software

Libraries

- **Xaw3d (Three-D Athena Widget) library**
  If **Xaw3d** is used instead of normal **Xaw**, widgets such as buttons will have 3-D appearance. It is available from ftp://ftp.x.org/contrib/widgets and its mirror sites in `/contrib/widgets`.
- **XPM3 package**
  This package provides facilities to handle images in XPM format. It is available from ftp://ftp.x.org/contrib/libraries and its mirror sites in `/contrib/libraries`.
- **JPEG library**
  This package provides facilities to handle images in JPEG format. The official distribution site is ftp://ftp.uu.net/graphics/jpeg. It is also available from ftp://ftp.x.org/contrib/libraries and its mirror sites in `/contrib/libraries`.

Related Applications

Here is a list of related applications. See also *Scientific Applications on Linux (SAL)* for more applications for Linux (many of them may also work on other UNIX-compatible platforms) and http://duke.usask.ca/~macphed/soft/fig/ which a site maintained by Ian MacPhedran that has some other links for FIG-related applications.

- **fig2dev (TransFig)**
  fig2dev provides the facility to convert figures created with **xfig** into various formats. You must have **fig2dev** because **xfig** invokes it when Exporting or Printing. **fig2dev** is part of the **TransFig** package.
- **GhostScript**
  xfig will invoke **GhostScript** (the `gs` command) when displaying an imported PostScript image. **GhostScript** is copyrighted by Aladdin Enterprises, but it is (at least older versions are) distributed under GPL, and it is available from ftp://prep.ai.mit.edu/pub/gnu and its mirror sites, or ftp.cs.wisc.edu/ghost/aladdin.
- **cdraw2fig**
  Converts `Cheap Draw' format to **Fig** format. Cheap Draw 2.0 was written by Bob Pratt and is available from the UNIXPC archives at osu-cis.
- **chemtool**
  A chemical structure editor for X11 available from http://www.uni-ulm.de/~s_tvolk/chemtool that writes Fig output (and relies on transfig for anything more fancy than xbm)
- **figbuild**
  figbuild reads xfig files (Version 3.x, tested with 3.1, 3.2) plus its own config file for heights, and generates a Radiance input file. Written by Peter Apian-Bennewitz at Fraunhofer ISE, Germany. See: http://www.ise.fhg.de/radiance/pabs-toolbox/figbuild/figbuild.html
- **GIMP**
  A comprehensive image manipulation program, GIMP (Gnu Image Manipulation Program) is freely available from http://www.gimp.org/.
- **gnuplot**
  gnuplot is a plotting package which can output in **Fig** and other formats. The official distribution site is ftp://ftp.dartmouth.edu/pub/gnuplot, but you can also get it from the `comp.sources.misc` archive newsgroup.
- **grass2fig**
  The GRASS GIS (Geographic Resources Analysis Support System) is an open source Geographical Information System (GIS) with raster, topological vector, image processing, and graphics production functionality that operates on various platforms through a graphical user interface and shell in the X Window System. It is released under GNU General Public License (GPL). See http://grass.itc.it/. The grass2fig program will convert a GRASS binary vector map to a Fig file.
- **gzip (gunzip)**
  xfig will invoke **gunzip** when reading **gzip**ped files. **gzip** (**gunzip**) is a product of the Free Software Foundation, and is available from ftp://prep.ai.mit.edu and its mirror sites.
- **hp2xx**
  hp2xx can convert HP-GL files into other formats such as PostScript, LaTeX or **Fig**. It is available from ftp://ftp.gnu.org/gnu/hp2xx.
- **jfig**
  jfig is a Java-based program that is based on **xfig**. See http://tech-www.informatik.uni-hamburg.de/applets/javafig/applet.html
- **mifXfig**

mifXfig will convert files between MIF format (of Framemaker(TM)) and **Fig** format. See http://www.ida.liu.se/~vaden/teaching/m2f/ for details.

- **Msphinx**

  A friendly package to handle satellite images.

  Main features:

  Msphinx includes two types of analysis and processing functions:
  - Functions to process the basic data geometry and projection planes.
  - Functions to manipulate and modify the graphic display and print.

  Communication capabilities:
  - Data writes from external user programs (C or Fortran) into the memory planes.
  - User-to-user data exchange (text or images).
  - User image data base visualisation.
  - Direct links to other packages:
    - Mgraph (2d and contouring plotting)
    - HDFLook (HDF read write files)
    - **Xfig (Facility for Interactive Generation of figures)**
    - mpeg_encode (mpeg format nimations)

  Find it at: http://loasys.univ-lille1.fr/Msphinx/Msphinx_gb.html

- **netpbm**

  fig2dev will invoke the appropriate command from the **netpbm** package when exporting figures to bitmap formats. **netpbm** is available from ftp://ftp.x.org/contrib/applications/drawing_tools/xfig and its mirror sites or from ftp://ftp.cdrom.com/pub/X11/contrib/utilities.

- **pic2fig**

  Convertes PIC format (used with troff, ditroff, etc) to **Fig** format. It is available from ftp://ftp.cs.orst.edu/localsrc/graphics or ftp://ftp.x.org/contrib/utilities and its mirror sites in `/contrib/utilities`.

- **plot2fig**

  Converts UNIX plot format to **Fig** format. Available from any GNU site.

- **plotutils (GNU)**

  includes an application for XY plotting (graph), a plot filter (plot), a Tektronix translator (tek2plot), a program for displaying font character maps (plotfont), and a library for drawing 2-D vector graphics (libplot), all of which produce output in xfig format. by Robert S. Maier <rsm@math.arizona.edu>. See: http://www.gnu.org/software/plotutils/plotutils.html

- **pstoedit**

  pstoedit can convert PostScript files into **Fig** format. It is available from http://www.pstoedit.net/pstoedit.

- **xfigbar** and **xfigpie**

  Two C programs to produce bar and pie charts as Fig files. Written by John Conover <john@johncon.johncon.com>, these programs are available at: http://www.johncon.com/john/archive/xfigutilities.shar.gz

- **xgraph**

  xgraph is a plotting package which runs in X11. The version on ftp://iworks.ecn.uiowa.edu in `comp.hp` can output **Fig** files.

- **xpaint**

  xpaint is a painting program which runs under X11. It is available from http://www.danbbs.dk/~torsten/xpaint and any SunSite machines such as ftp://sunsite.unc.edu/pub/Linux/apps/graphics/draw/.

- **xv**

xv can be used to display images in various formats and allows some editing of those images (cropping, color editing scaling/rotating etc). By default, **xfig** will invoke **xv** when the Edit Image button is pressed in the edit PICTURE object panel. **xv** is shareware and is available via anonymous ftp from ftp://ftp.cis.upenn.edu/pub/xv.

[Contents | Introduction | Credits]

Xfig User Manual

Version 3.2.4 December 2002

Internationalization (Using Japanese, Korean, etc.)

xfig 3.2.X and **fig2dev 3.2.X** include code for internationalization, and it is possible to put characters of Japanese and some another languages in texts. At now, it is known to worked under Japanese and Korean.

Additional informations about this may available at *http://member.nifty.ne.jp/tsato/xfig/*. Send any questions or comments about this internationalization facility to *VEF00200@nifty.ne.jp (T.Sato)*.

Environment

To use this internationalization facility, you must have following environment:

- Library and X Server of X11R5 or X11R6.
  (They must be appropriately internationalized and they must be possible to connect to the *input method*. Use of X11R6 is recommended.)

- Appropriate *conversion server* (**Canna** or **Wnn**, for example), and an *input method* (**kinput2** or **htt**, for example). These facilities may be implemented as a single program.

- Fonts for the language (`jiskan16` for Japanese, for example)

Installation

Install xfig

1. Get **xfig.3.2.4.tar.gz** and *gunzip* and *untar* it.

2. Uncomment ``` `` ```#define I18N" in `xfig.3.2.4/Imakefile` (remove the `XCOMM` comment).

3. If your C library supports the required locale, remove `-DSETLOCALE` from the definition of `I18N_DEFS` in the `Imakefile`. If your C library doesn't support the required locale, make sure that `-DSETLOCALE` is specified.

4. Compile and install **xfig** in the usual way.

Install fig2dev

1. Get **transfig.3.2.4.tar.gz** and *gunzip* and *untar* it.

2. Uncomment ``` `` ```#define I18N" in `transfig.3.2.4/fig2dev/Imakefile` (remove the `XCOMM` comment).

3. If you want to install files like `japanese.ps` to a directory other than `/usr/local/lib/fig2dev`, modify the definition of `FIG2DEV_LIBDIR` and `I18N_DEV_DEFS` in the `Imakefile`.

4. Compile and install **TransFig** (**fig2dev**) in the usual way.

Startup

1. Set the locale name for the language to be used (such as `ja_JP.eucJP` or `ko_KR.eucKR`, for example) to the environment variable `LANG`.

2. If it is necessary, set the environment variable `XMODIFIERS` to specify the *input method* to be used.

3. Make sure that the appropriate *conversion server* (**Canna** or **Wnn**, for example) and an *input method* (**kinput2** or **htt**, for example) is available.

4. Type ``` ``xfig -international'' ```.

   Without the `-international` option, **xfig** will work as normal (no internationalization). If you put ``` ``Fig.international: true'' ``` into your resource file, `-international` option may omitted.

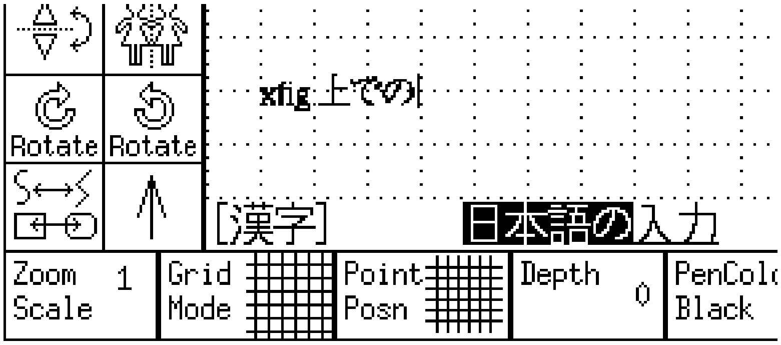
Entering Text

Using this internationalization facility, you may enter text in Japanese or some another languages (henceforce, call ``international text'') with the TEXT facility. When entering international text, ``*Times-Roman*'' or ``*Times-Bold*'' (may be displayed as ``*Times-Roman + Mincho*'' and ``*Times-Bold + Gothic*'' in Japanese environment) must be selected as TEXT FONT.

Input of international text will be started by typing a key to switch the input mode when it is ready to input text from the keyboard in TEXT mode. It depend on the environment as to which key will switch the input mode, but keys such as `Shift-SPACE`, `Control-SPACE`, `Control-O`, or `Control-\` may be used in many cases. Operations for conversion also depend on the environment, but will be the same as other applications which use the environment.

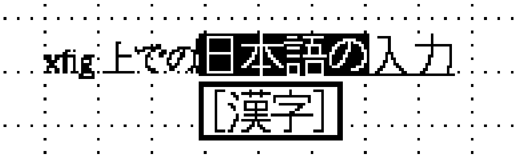
The *input style* may be selected from *Off the Spot*, *Over the Spot*, and *Root*. The *input style* to be used may be selected with the `inputStyle` resource or the `-inputStyle` option. For example, `xfig -international -inputStyle OverTheSpot` will select *Over the Spot* as the input style.

Off the Spot:
: The text under conversion will be displayed at the bottom of the canvas.



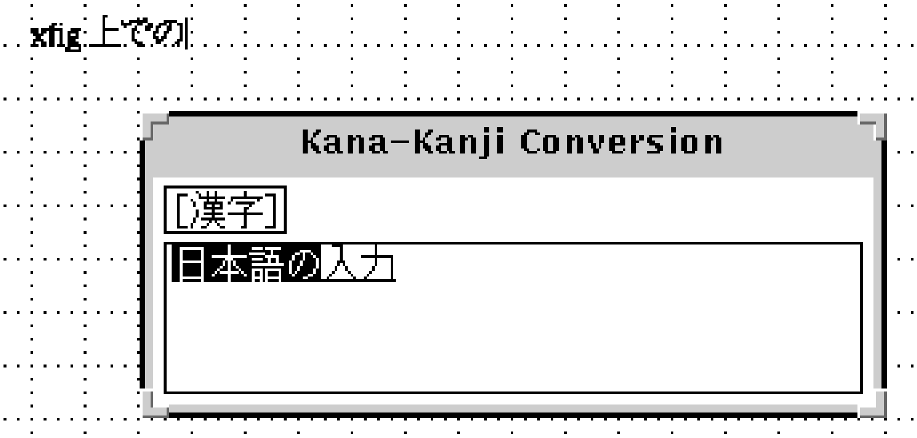
Over the Spot:

The text under conversion will be displayed at the position where it will end up. But the display may be somewhat strange because it will be displayed with a different font. Also, it may lead to somewhat unusual behavior, or the display on the canvas may get confused.



Root:

The text under conversion will displayed in a separate window.



Customization

Changing Display Fonts

By default, fonts to be used on the display (hardcoded in the program) is very loosely specified so that those fonts can found on any systems:

```
    Fig.normalFontSet: -*-times-medium-r-normal--16-*-*-*-*-*-*-*,\
        -*-*-medium-r-normal--16-*-*-*-*-*-*-*,\
        -*-*-*-r-*--16-*-*-*-*-*-*-*
    Fig.boldFontSet: -*-times-bold-r-normal--16-*-*-*-*-*-*-*,\
        -*-*-bold-r-normal--16-*-*-*-*-*-*-*,\
        -*-*-*-r-*--16-*-*-*-*-*-*-*
```

With this default specification, non-desirable fonts may loaded (bad appearance of texts or long delay when starting of **xfig** may caused as the result, for example) on some systems. In such case, you may need to specify those fonts more definitive in the resource file (`app-defaults/Fig`) and force the system to load the specified font:

```
    Fig*FontSet: -misc-fixed-medium-r-normal--14-*-*-*-*-*-*-*
    Fig.normalFontSet: -*-times-medium-r-normal--14-*-*-*-*-*-*-*,\
        -misc-fixed-medium-r-normal--14-*-*-*-*-*-*-*
    Fig.boldFontSet: -*-times-bold-r-normal--14-*-*-*-*-*-*-*,\
        -misc-fixed-medium-r-normal--14-*-*-*-*-*-*-*
```

If scalable fonts are available (when X server which support scalable fonts like X-TrueType Server is in use, for example) appearance of text may improved by specifying large fonts, as:

```
    Fig*FontSet: -*-times-medium-r-normal--16-*-*-*-*-*-*-*,\
        -foobar-mincho-medium-r-normal--16-*-*-*-*-*-*-*
    Fig.normalFontSet: -*-times-medium-r-normal--64-*-*-*-*-*-*-*,\
        -foobar-mincho-medium-r-normal--64-*-*-*-*-*-*-*
    Fig.boldFontSet: -*-times-bold-r-normal--64-*-*-*-*-*-*-*,\
        -foobar-gothic-medium-r-normal--64-*-*-*-*-*-*-*
```

Changing Fonts on PostScript Output

Fonts used when generating PostScript output are specified in the files like `japanese.ps` in **fig2dev** package, and it is possible to change them by modifying those files.

Japanese

By default, `Ryumin-Light` and `GothicBBB-Medium` will used if they are available, and `HeiseiMin-W3` and `HeiseiKakuGo-W5` otherwise.

Locale name can one of `japanese`, `ja`, `ja_JP`, `ja_JP.ujis`, `ja_JP.eucJP` and `ja_JP.EUC`.

Korean

By default, `Munhwa-Regular` and `MunhwaGothic-Bold` will used if they are available, and

HLaTeX-Myoungjo-Regular and HLaTeX-Gothic-Regular otherwise.

Locale name can one of korean, ko, ko_KR, ko_KR.ujis, ko_KR.eucKR and ko_KR.EUC.

Another Languages

Because configuration file for languages other than Japanese and Korean is not prepared, you must make the file for the language and available fonts.

The file must installed into the directory specified when **fig2dev** is installed. The filename must locale name followed by ``.ps''. For example, if locale name is zh_CN.eucCN, the filename must zh_CN.eucCN.ps.

Setting Resources

X Window System has mechanism to load locale-specific resource file to support internationalization (localization) of applications. With this mechanism, it is possible to make suitable settings for the language without specifying options when executing the application. To make **xfig** works properly for multiple languages, it may be necessary to make suitable settings using this mechanism.

In the default configuration of X11R6, if there is a resource file like /usr/X11R6/lib/X11/*locale*/app-defaults/Fig (here, *locale* is locale name or its ``language part''), it will be loaded instead of /usr/X11R6/lib/X11/app-defaults/Fig. Therefore, if you wrote setting for Japanese environment in /usr/X11R6/lib/X11/ja/app-defaults/Fig, the setting for Japanese environment will be used when environment variable LANG is set to ja_JP.eucJP or so, and default setting in /usr/X11R6/lib/X11/app-defaults/Fig will be used otherwise.

Notes

- When entering international text, you must select ``*Times-Roman*'' or ``*Times-Bold*'' (may be displayed as ``*Times-Roman + Mincho*'' and ``*Times-Bold + Gothic*'' in Japanese environment) as the TEXT FONT. If any other font is selected, Latin-1 characters will be available as in normal **xfig**.

- It is not possible to edit international text in the Edit Panel. But it is possible to edit text on the canvas in TEXT mode.

- When specifying international resource, you should specify as ``Fig.international: true'' but not ``Fig*international: true''.

- Use EUC for encoding of multi-byte text. You may need to set locale (environment variable `LANG`) appropriately for your system. On some systems, `japanese` may select non-EUC encoding.

- In Japanese environments, text may include only ASCII and JIS-X-0208 characters. If the environment supports it, it may be possible to enter characters of JIS-X-0201 *kana* characters or JIS-X-0212 (supplement *kanji*), but **fig2dev** will not accept those characters.

- Regrettably, making **xfig 3.2.X**'s Japanese entering feature available may difficult on many X11R5 systems. It is known to work on Japanese Solaris 2's OpenWindows and Solaris CDE, but unknown about another X11R5-based systems.

Tested Environments

At this time, **xfig**'s international facility has been successfully worked on the following environments.

Japanese

Operating System	**X**	**Input Method**
SunOS 4.1	X11R6	kinput2
Solaris 2.5	X11R6	kinput2
Solaris 2.5-2.6	OpenWindows, CDE	htt, ATOK
HP-UX 10.20	X11R6	kinput2
IRIX 6.3*	X11R6	kinput2
FreeBSD 2.2	XFree86	kinput2
Slackware Linux 3.1	XFree86	kinput2
RedHat Linux 4.2, 5.2	XFree86	kinput2
Debian GNU/Linux 2.x	XFree86	kinput2, skkinput

* On IRIX 6.3, you may need to compile **xfig** with IRIX's genuine `cc` (not with `gcc`), specifying compile option `-N32 -mips3`. Also, you may need to get source of JPEG library and compile it yourself, to avoid using JPEG library distributed with IRIX.

Korean

Operating System	**X**	**Input Method**

RedHat Linux 5.2, 6.0	XFree86	hanIM, ami

SEE ALSO:

Hangul and Internet in Korea FAQ
Hangul and Printing

[Contents | Introduction | Credits]

Xfig User Manual

Version 3.2.4 *December 2002*

Fig Format 3.2

This is a description of the *Fig Format 3.2*, which is a copy of `xfig.3.2.x/Doc/FORMAT3.2` in the **xfig 3.2.x** source distribution.

```
The new components in protocol 3.2 are the paper size, magnification,
single/multiple page indicator and transparent color for GIF export in the
header.

The other modification between version 3.1 and version 3.2 of the
protocol is the mathematical model used for splines. The new version
uses X-splines which allows the user to mix interpolation and approximation
points in a same curve. More precisely, it means that an X-spline curve
is neither an interpolated spline nor an approximated one, it is BOTH
(the behaviour of each point is controlled by one single parameter
called "shape factor"). For additional information about X-splines, see:

  "X-Splines: A Spline Model Designed for the End User"
  by C. Blanc and C. Schlick, Proceedings of SIGGRAPH'95

Caveat: Because spline models of previous versions (quadratic B-splines
and Bezier with hidden points) are no longer supported, curves that are
present in version 3.1 and older files are automatically converted to
X-splines. This translation is only an approximation process. It means
that the converted curves are not exactly the same as the original ones.
Though the translation usually provides almost identical curves, some
```

hand-fitting may be needed in some pathological cases.

Description of the Fig Format Follows

(1) The very first line is a comment line containing the name and version:

```
        #FIG 3.2
```

The character # at the first column of a line indicates that the line is a comment line which will be preserved when the Fig file is read in. The user may edit them with the popup editor.

The comment line(s) must immediately precede the object to which they are associated. In the case of the "whole figure comments" mentioned below, they immediately precede the (resolution,coord_system) line.

(2) The first non-comment line consists of the following:

```
        string  orientation             ("Landscape" or "Portrait")
        string  justification           ("Center" or "Flush Left")
        string  units                   ("Metric" or "Inches")
        string  papersize               ("Letter", "Legal", "Ledger", "Tabloid",
                                         "A", "B", "C", "D", "E",
                                         "A4",   "A3", "A2", "A1", "A0" and "B5")
        float   magnification           (export and print magnification, %)
        string  multiple-page           ("Single" or "Multiple" pages)
        int     transparent color       (color number for transparent color for GIF
                                         export. -3=background, -2=None, -1=Default,
                                         0-31 for standard colors or 32- for user
colors)
        # optional comment              (An optional set of comments may be here,
                                         which are associated with the whole figure)
        int     resolution coord_system (Fig units/inch and coordinate system:
                                           1: origin at lower left corner (NOT USED)
                                           2: upper left)
```

Fig_resolution is the resolution of the figure in the file. Xfig will always write the file with a resolution of 1200ppi so it will scale the figure upon reading it in if its resolution is different from 1200ppi. Pixels are assumed to be square.

Xfig will read the orientation string and change the canvas to match either the Landscape or Portrait mode of the figure file.

The units specification is self-explanatory.

The coordinate_system variable is ignored - the origin is ALWAYS the upper-left corner.

```
** Coordinates are given in "fig_resolution" units.
** Line thicknesses are given in 1/80 inch (0.3175mm) or 1 screen pixel.
   When exporting to EPS, PostScript or any bitmap format (e.g. GIF),  the
   line thickness is reduced to 1/160 inch (0.159mm) to "lighten" the look.
** dash-lengths/dot-gaps are given in 80-ths of an inch.
```

---

(3) The rest of the file contains various objects. An object can be one of six classes (or types).

```
     0)       Color pseudo-object.
     1)       Ellipse which is a generalization of circle.
     2)       Polyline which includes polygon and box.
     3)       Spline which includes
              closed/open approximated/interpolated/x-spline spline.
     4)       Text.
     5)       Arc.
     6)       Compound object which is composed of one or more objects.
```

In the following elaboration on object formats, every value of fig output are separated by blank characters or new line ('\n'). The value of the unused parameters will be -1.

Some fields are described as "enumeration type" or "bit vector"; the values which these fields can take are defined in the header file object.h. The pen_style field is unused.
These values may be defined in some future version of Fig.

The two color fields (pen and fill; pen only, for texts) are defined as follows:

```
        -1 = Default
         0 = Black
         1 = Blue
         2 = Green
         3 = Cyan
         4 = Red
         5 = Magenta
         6 = Yellow
         7 = White
      8-11 = four shades of blue (dark to lighter)
     12-14 = three shades of green (dark to lighter)
     15-17 = three shades of cyan (dark to lighter)
     18-20 = three shades of red (dark to lighter)
     21-23 = three shades of magenta (dark to lighter)
     24-26 = three shades of brown (dark to lighter)
     27-30 = four shades of pink (dark to lighter)
	    31 = Gold

	 values from 32 to 543 (512 total) are user colors and
	 are defined in color pseudo-objects (type 0)

	 Your X server may limit the number of colors to something less
	 than this, especially on a 8-bit PseudoColor visual, where
	 the number of usable colors will be 256 minus the number of colors
	 xfig preallocates for itself and the 32 standard colors (about 48).

    For WHITE color, the area fill field is defined as follows:

	-1 = not filled
	 0 = black
	...  values from 1 to 19 are shades of grey, from darker to lighter
	20 = white
	21-40 not used
	41-56 see patterns for colors, below

    For BLACK or DEFAULT color, the area fill field is defined as follows:

	-1 = not filled
	 0 = white
	...  values from 1 to 19 are shades of grey, from lighter to darker
	20 = black
	21-40 not used
	41-56 see patterns for colors, below

    For all other colors, the area fill field is defined as follows:

	-1 = not filled
	 0 = black
	...  values from 1 to 19 are "shades" of the color, from darker to lighter.
		A shade is defined as the color mixed with black
	20 = full saturation of the color
	...  values from 21 to 39 are "tints" of the color from the color to white.
		A tint is defined as the color mixed with white
	40 = white
	41 = 30 degree left diagonal pattern
	42 = 30 degree right diagonal pattern
	43 = 30 degree crosshatch
	44 = 45 degree left diagonal pattern
	45 = 45 degree right diagonal pattern
	46 = 45 degree crosshatch
	47 = horizontal bricks
	48 = vertical bricks
	49 = horizontal lines
	50 = vertical lines
	51 = crosshatch
	52 = horizontal "shingles" skewed to the right
	53 = horizontal "shingles" skewed to the left
	54 = vertical "shingles" skewed one way
	55 = vertical "shingles"skewed the other way
```

```
    56 = fish scales
    57 = small fish scales
    58 = circles
    59 = hexagons
    60 = octagons
    61 = horizontal "tire treads"
    62 = vertical "tire treads"

The depth field is defined as follows:

     0 ... 999 where larger value means object is deeper than (under)
               objects with smaller depth

The line_style field is defined as follows:

    -1 = Default
     0 = Solid
     1 = Dashed
     2 = Dotted
     3 = Dash-dotted
     4 = Dash-double-dotted
     5 = Dash-triple-dotted

The style_val field is defined as the length, in 1/80 inches, of the on/off
dashes for dashed lines, and the distance between the dots, in 1/80 inches,
for dotted lines.

The join_style field is defined FOR LINES only as follows:

     0 = Miter (the default in xfig 2.1 and earlier)
     1 = Round
     2 = Bevel

The cap_style field is defined FOR LINES, OPEN SPLINES and ARCS only as follows:

     0 = Butt (the default in xfig 2.1 and earlier)
     1 = Round
     2 = Projecting

The arrow_type field is defined for LINES, ARCS and OPEN SPLINES
only as follows:

     0 = Stick-type (the default in xfig 2.1 and earlier)

             \
                \
    _______________\
                   /
                /
             /

     1 = Closed triangle:

            |\
                |   \
        ________|      \
                |      /
                |   /
                |/

         2 = Closed with "indented" butt:

                |\
                \   \
                 \     \
        __________\       \
                  /       /
                 /     /
                /   /
                |/

         3 = Closed with "pointed" butt:

                   /\
                  /    \
                 /        \
        ________/            \
                \            /
                 \        /
                  \    /
                   \/

    The arrow_style field is defined for LINES, ARCS and OPEN SPLINES
    only as follows:

         0 = Hollow (actually filled with white)
         1 = Filled with pen_color
```

(3.0) Object Definitions

```
    (3.1) Color Pseudo-objects (user-defined colors)
          This is used to define arbitrary colors beyond the 32 standard colors.
          The color objects must be defined before any other Fig objects.

    First line:
        type    name                    (brief description)
        ----    ----                    -------------------
        int     object_code             (always 0)
        int     color_number            (color number, from 32-543 (512 total))
     hex string rgb values              (hexadecimal string describing red,
                                             green and blue values (e.g. #330099) )
```

(3.2) ARC

```
First line:
    type    name                        (brief description)
    ----    ----                        -------------------
    int     object_code                 (always 5)
    int     sub_type                    (1: open ended arc
                                         2: pie-wedge (closed) )
    int     line_style                  (enumeration type)
    int     line_thickness              (1/80 inch)
    int     pen_color                   (enumeration type, pen color)
    int     fill_color                  (enumeration type, fill color)
    int     depth                       (enumeration type)
    int     pen_style                   (pen style, not used)
    int     area_fill                   (enumeration type, -1 = no fill)
    float   style_val                   (1/80 inch)
    int     cap_style                   (enumeration type)
    int     direction                   (0: clockwise, 1: counterclockwise)
    int     forward_arrow               (0: no forward arrow, 1: on)
    int     backward_arrow              (0: no backward arrow, 1: on)
    float   center_x, center_y          (center of the arc)
    int     x1, y1                      (Fig units, the 1st point the user entered)
    int     x2, y2                      (Fig units, the 2nd point)
    int     x3, y3                      (Fig units, the last point)

Forward arrow line (Optional; absent if forward_arrow is 0):
    type    name                        (brief description)
    ----    ----                        -------------------
    int     arrow_type                  (enumeration type)
    int     arrow_style                 (enumeration type)
    float   arrow_thickness             (1/80 inch)
    float   arrow_width                 (Fig units)
    float   arrow_height                (Fig units)

Backward arrow line (Optional; absent if backward_arrow is 0):
    type    name                        (brief description)
    ----    ----                        -------------------
    int     arrow_type                  (enumeration type)
    int     arrow_style                 (enumeration type)
    float   arrow_thickness             (1/80 inch)
    float   arrow_width                 (Fig units)
    float   arrow_height                (Fig units)
```

(3.3) COMPOUND

```
A line with object code 6 signifies the start of a compound.
There are four more numbers on this line which indicate the
```

upper left corner and the lower right corner of the bounding box of this compound. A line with object code -6 signifies the end of the compound. Compound may be nested.

First line:

type	name	(brief description)
int	object_code	(always 6)
int	upperleft_corner_x	(Fig units)
int	upperleft_corner_y	(Fig units)
int	lowerright_corner_x	(Fig units)
int	lowerright_corner_y	(Fig units)

Subsequent lines:

```
    objects
    .
    .
```

Last line:

```
    -6
```

(3.4) ELLIPSE

First line:

type	name	(brief description)
int	object_code	(always 1)
int	sub_type	(1: ellipse defined by radii 2: ellipse defined by diameters 3: circle defined by radius 4: circle defined by diameter)
int	line_style	(enumeration type)
int	thickness	(1/80 inch)
int	pen_color	(enumeration type, pen color)
int	fill_color	(enumeration type, fill color)
int	depth	(enumeration type)
int	pen_style	(pen style, not used)
int	area_fill	(enumeration type, -1 = no fill)
float	style_val	(1/80 inch)
int	direction	(always 1)
float	angle	(radians, the angle of the x-axis)
int	center_x, center_y	(Fig units)
int	radius_x, radius_y	(Fig units)
int	start_x, start_y	(Fig units; the 1st point entered)
int	end_x, end_y	(Fig units; the last point entered)

(3.5) POLYLINE

First line:

```
    type    name                        (brief description)
    ----    ----                        -------------------
    int     object_code                 (always 2)
    int     sub_type                    (1: polyline
                                         2: box
                                         3: polygon
                                         4: arc-box)
                                         5: imported-picture bounding-box)
    int     line_style                  (enumeration type)
    int     thickness                   (1/80 inch)
    int     pen_color                   (enumeration type, pen color)
    int     fill_color                  (enumeration type, fill color)
    int     depth                       (enumeration type)
    int     pen_style                   (pen style, not used)
    int     area_fill                   (enumeration type, -1 = no fill)
    float   style_val                   (1/80 inch)
    int     join_style                  (enumeration type)
    int     cap_style                   (enumeration type, only used for POLYLINE)
    int     radius                      (1/80 inch, radius of arc-boxes)
    int     forward_arrow               (0: off, 1: on)
    int     backward_arrow              (0: off, 1: on)
    int     npoints                     (number of points in line)

Forward arrow line: same as ARC object

Backward arrow line: same as ARC object

For picture (type 5) the following line follows:
    type    name                        (brief description)
    ----    ----                        -------------------
    boolean flipped                     orientation = normal (0) or flipped (1)
    char    file[]                      name of picture file to import

Points line(s).  The x,y coordinates follow, any number to a line, with
as many lines as are necessary:
    type    name                        (brief description)
    ----    ----                        -------------------
    int     x1, y1                      (Fig units)
    int     x2, y2                      (Fig units)
      .
      .
    int     xnpoints ynpoints           (this will be the same as the 1st
                                        point for polygon and box)
```

```
 (3.6) SPLINE

First line:
    type    name                        (brief description)
    ----    ----                        -------------------
    int     object_code                 (always 3)
    int     sub_type                    (0: open approximated spline
                                              1: closed approximated spline
                                              2: open   interpolated spline
                                              3: closed interpolated spline
                                              4: open   x-spline
                                              5: closed x-spline)
        int     line_style              (See the end of this section)
        int     thickness               (1/80 inch)
        int     pen_color               (enumeration type, pen color)
        int     fill_color              (enumeration type, fill color)
        int     depth                   (enumeration type)
        int     pen_style               (pen style, not used)
        int     area_fill               (enumeration type, -1 = no fill)
        float   style_val               (1/80 inch)
        int     cap_style               (enumeration type, only used for open
splines)
        int     forward_arrow           (0: off, 1: on)
        int     backward_arrow          (0: off, 1: on)
        int     npoints                 (number of control points in spline)

    Forward arrow line: same as ARC object

    Backward arrow line: same as ARC object

    Points line: same as POLYLINE object

    Control points line :

    There is one shape factor for each point. The value of this factor
    must be between -1 (which means that the spline is interpolated at
    this point) and 1 (which means that the spline is approximated at
    this point). The spline is always smooth in the neighbourhood of a
    control point, except when the value of     the factor is 0 for which
    there is a first-order discontinuity (i.e. angular point).
```

```
    (3.7) TEXT
        type    name                    (brief description)
        ----    ----                    -------------------
        int     object                  (always 4)
        int     sub_type                (0: Left justified
                                             1: Center justified
                                             2: Right justified)
        int     color                   (enumeration type)
        int     depth                   (enumeration type)
        int     pen_style               (enumeration , not used)
        int     font                    (enumeration type)
        float   font_size               (font size in points)
        float   angle                   (radians, the angle of the text)
        int     font_flags              (bit vector)
        float   height                  (Fig units)
        float   length                  (Fig units)
        int     x, y                    (Fig units, coordinate of the origin
```

		of the string. If sub_type = 0, it is the lower left corner of the string. If sub_type = 1, it is the lower center. Otherwise it is the lower right corner of the string.)
char	string[]	(ASCII characters; starts after a blank character following the last number and ends before the sequence '\001'. This sequence is not part of the string. Characters above octal 177 are represented by \xxx where xxx is the octal value. This permits fig files to be edited with 7-bit editors and sent by e-mail without data loss. Note that the string may contain '\n'.)

The font_flags field is defined as follows:

Bit	Description
0	Rigid text (text doesn't scale when scaling compound objects)
1	Special text (for LaTeX)
2	PostScript font (otherwise LaTeX font is used)
3	Hidden text

The font field is defined as follows:

For font_flags bit 2 = 0 (LaTeX fonts):

0	Default font
1	Roman
2	Bold
3	Italic
4	Sans Serif
5	Typewriter

For font_flags bit 2 = 1 (PostScript fonts):

-1	Default font
0	Times Roman
1	Times Italic
2	Times Bold
3	Times Bold Italic
4	AvantGarde Book
5	AvantGarde Book Oblique
6	AvantGarde Demi
7	AvantGarde Demi Oblique
8	Bookman Light
9	Bookman Light Italic
10	Bookman Demi
11	Bookman Demi Italic
12	Courier
13	Courier Oblique

14	Courier Bold
15	Courier Bold Oblique
16	Helvetica
17	Helvetica Oblique
18	Helvetica Bold
19	Helvetica Bold Oblique
20	Helvetica Narrow
21	Helvetica Narrow Oblique
22	Helvetica Narrow Bold
23	Helvetica Narrow Bold Oblique
24	New Century Schoolbook Roman
25	New Century Schoolbook Italic
26	New Century Schoolbook Bold
27	New Century Schoolbook Bold Italic
28	Palatino Roman
29	Palatino Italic
30	Palatino Bold
31	Palatino Bold Italic
32	Symbol
33	Zapf Chancery Medium Italic
34	Zapf Dingbats

[Contents | Introduction | Credits]

Xfig User Manual

Version 3.2.4 December 2002

FAQ

General Questions

1. Where can I get **xfig/TransFig**?
2. Do I need to convert old **Fig** files to the new version?
   (Also, can **xfig** and **fig2dev** read older **Fig** files?)
3. Can I edit PostScript files with **xfig**?
4. I can't seem to control the drawing order of objects
5. My splines are being changed somehow from version 3.1 of **xfig**
6. How can I install **xfig** in a personal directory?
7. Is there a mailing list or newsgroup specifically for **xfig**?

Problems when running xfig or TransFig (fig2dev)

1. I don't get any color in **xfig**'s buttons
2. Icons in the side and bottom panels appear then disappear
3. A warning about some action not found occurs, like ``Warning: Actions not found: ModeOpenCompound"
4. An X error occurs with `X_SetClipRectangles` as the Request code, or thick line shows through arrowhead in certain orientations
5. When using LaTeX fonts in **xfig** I always end up with PostScript fonts
6. Arc-boxes don't appear correctly when viewing an exported PostScript file with GhostScript
7. OpenWindows sometimes loses track of **xfig**'s icon
8. On HP machines, the capitalization of some letters in the edit text window are wrong

9. I get a ridiculous %%BoundingBox in my exported PostScript file

10. Pattern fills are coming out all black when viewing the exported PostScript file with ghostscript 5.50

11. I am exporting to the MetaFont format and the generated file refers to the ``grafbase.mf'' file which I don't have

12. My pulldown menus don't work under CDE

13. I am getting get "mn" in my LaTeX text

14. Pattern fills are going beyond the bounds of the polygons when I export to EPS, GIF, etc., although they look fine on the screen

15. xfig says: "Warning: Representation size 4 must match superclass's to override underline"

16. fig2dev says: "can't open file: /usr/X11R6/lib/X11/fig2dev/en_US.ps" (or some other xx_xx.ps file)

17. xfig won't find the app-defaults file in my personal directory even though I have set the **XAPPLRESDIR** environment variable

18. I can't enter special characters such as a-umlaute in the popup editor for text objects.

Problems when compiling xfig

1. On a DEC Alpha the compiler may complain about something like ``.mask must have pcreg from .frame set if any bits are set in .mask or .fmask'' in the file `u_bound.c`

2. `XPointer` is undefined or unknown when I compile **xfig**

3. On Sun machines the linker may give erroneous error messages about `_get_applicationShellWidgetClass` and `_get_wmShellWidgetClass` are undefined

4. On some HP computers, the compiler dies with bus error in the file `u_undo.c`

5. On some HP computers using native `cc` compiler there is a problem with +O3 optimization

6. I'm getting errors such as:

   Unresolved:
   jpeg_std_error
   jpeg_create_decompress
   jpeg_destroy_decompress
   jpeg_stdio_src
   and others.

7. I am getting these errors when I am compiling **xfig**:

```
w_menuentry.c: In function `Redisplay':
w_menuentry.c:129: structure has no member named `fontset'
w_menuentry.c:143: structure has no member named `international'
```

General Questions

1. Where can I get xfig/TransFig?

xfig and **TransFig** are freeware and you can get them from several sites. See Installing **xfig** about this.

2. Do I need to convert old Fig files to the new version?

No. Every version of **xfig** and **TransFig (fig2dev)** can read **Fig** files of any older version. When you load one into **xfig** it converts it to the latest version. When you save it again it is saved in that version.

3. Can I edit PostScript files with xfig?

No, but you can ``paint'' over them with **Fig** objects to add annotation or other objects. Or you can convert the PostScript file into the **Fig** format using **pstoedit** and then edit the **Fig** file using **xfig** in the usual way.

4. I can't seem to control the drawing order of objects

When you have two or more objects that overlap and the drawing order is important, you must set the *depth* of each object to guarantee the drawing order.

5. My splines are being changed somehow from version 3.1 of xfig

xfig 3.2 uses a new type of spline called an X-Spline. X-splines allow the user to mix interpolation and approximation points in the same curve. More precisely, it means that an X-spline curve is neither an interpolated spline nor an approximated one, it is BOTH (the behaviour of each point is controlled by one single parameter called "shape factor").

Caveat: Because spline models of previous versions (quadratic B-splines and Bezier with hidden points) are no longer supported, curves that are present in version 3.1 and older files are automatically converted to X-splines. This translation is only an approximation process. It means that the converted curves are not exactly the same as the original ones. Though the translation usually provides almost identical curves, some hand-fitting may be needed in some pathological cases.

6. How can I install xfig in a personal directory?

There are several files that you need:

`xfig`, `fig2dev`
: These files should be put in your command search path.

`CompKeyDB`
: Set the `XFIGLIBDIR` variable in the `Imakefile` to the directory you want this file to go into, or specify the file with resource ``` ``Fig*keyFile: ```*`your_lib_area`*`/CompKeyDB`".

`Fig` and `Fig-color` (from the `Fig.ad` and `Fig-color.ad` files)
: Set the shell environment variable `XAPPLRESDIR` to the directory where you want these files to go.

`Doc/html`
: You must copy this whole directory (and its sub-directory) to the directory where `XFIGLIBDIR` in the Imakefile points (use `cp -r`).

`Doc/xfig-howto.pdf` and `Doc/xfig.pdf`
: You must copy these two files to to the directory where `XFIGLIBDIR` in the Imakefile points.

7. Is there a mailing list or newsgroup specifically for xfig?

There are no mailing lists for **xfig**, but you may send email to xfig-bugs@epb.lbl.gov or post questions about it to comp.windows.x, comp.windows.x.apps, comp.os.linux.x, comp.os.linux, or comp.os.linux.misc. Questions about **TransFig** or **fig2dev** may also be posted to comp.text.tex.

Problems when running xfig or TransFig (fig2dev)

1. I don't get any color in xfig's buttons

POSSIBLE CAUSE:
: You didn't install the app-defaults files correctly and/or you don't have the customization resource set.

SOLUTION:
: You should install the app-defaults files `Fig` and `Fig-color` (`Fig.ad` and `Fig-color.ad` in **xfig** source distribution) into the appropriate directory. You should also put the string ``` ``*customization: -color``` " in a global resource file that gets loaded into the X server resources (`.Xdefaults`), and then restart the X server.

Explanation: Since X11R5, there has been a general resource `*customization` which can be set to the string `-color` which tells the system to look for an app-default file with the suffix `-color` (e.g. `Fig-color`) and read that instead of the usual (`Fig`). Inside `Fig-color` is an `#include` statement to read the normal `Fig` app-default file to get everything else not pertaining to color.

2. Icons in side and bottom panels appear then disappear

POSSIBLE CAUSE:

On Linux (2.0.29 at least), the libNextaw version of the Athena Widget set (libXaw) seem to cause this problem.

SOLUTION:
Relink with vanilla libXaw or libXaw3d.

---

3. A warning about some action not found occurs, like ``Warning: Actions not found: ModeOpenCompound"

POSSIBLE CAUSE:
You didn't install the current app-defaults files.

SOLUTION:
Install the app-default files (`Fig.ad` and `Fig-color.ad`) which comes with the **xfig**.
This is most easily done with ``` ``make install ```".

---

4. X error occurs with `X_SetClipRectangles` as the Request code, or thick line shows through arrowhead in certain orientations

POSSIBLE CAUSE:
In X11R5 Xlib, there was a bug in the clipping algorithm.

SOLUTION:
Switch to X11R6 or apply the following patch to `mit/lib/X/XPolyReg.c` in the X11R5 library:

```
*** XPolyReg.c.orig     Tue Dec 15 12:01:22 1992
--- XPolyReg.c  Wed Nov 15 09:41:13 1995
***************
*** 402,408 ****
            if (numRects && pts->x == rects->x1 && pts->y == rects->y2 &&
                pts[1].x == rects->x2 &&
                (numRects == 1 || rects[-1].y1 != rects->y1) &&
!               (!i || pts[2].y > pts[1].y)) {
                rects->y2 = pts[1].y + 1;
                continue;
            }
--- 402,409 ----
            if (numRects && pts->x == rects->x1 && pts->y == rects->y2 &&
                pts[1].x == rects->x2 &&
                (numRects == 1 || rects[-1].y1 != rects->y1) &&
!               /* Fixed version from X11 R6 */
!               (i && pts[2].y > pts[1].y)) {
                rects->y2 = pts[1].y + 1;
                continue;
            }
```

---

5. **When using LaTeX fonts in xfig I always end up with PostScript fonts**

   POSSIBLE CAUSE AND SOLUTION:

   To put fonts under the control of LaTeX when generating PostScript output, you must tag the text object as ``special'' and then use Combined PostScript/LaTeX (pstex) output mode. This mode forces special text objects to be passed through to LaTeX, allowing the use of Math mode and Computer Modern fonts among other things.

6. **Arc-boxes don't appear correctly when viewing an exported PostScript file with GhostScript**

   POSSIBLE CAUSE:

   Ghostscript version 4.01 has a bug with the `arcto` operators.

   SOLUTION:

   Get Ghostscript 4.03 or newer.

7. **OpenWindows sometimes loses track of xfig's icon**

   POSSIBLE CAUSE:

   Cause unknown.

   SOLUTION:

   Here is a temporary kludge. Place the line:

   ```
     Fig*iconPixmap:  your_lib_area/fig.icon.X
   ```

   in your `.Xdefaults` file and copy the file `fig.icon.X` into directory *`your_lib_area`*.

8. **On HP machines, the capitalization of some letters in the edit text window are wrong**

   POSSIBLE CAUSE AND SOLUTION:

   Make sure that the `CompKeyDB` file has the lowercase letter definitions *before* the uppercase definitions. This should be the case for the `CompKeyDB` file shipped with **xfig** 2.1.5 and later.

9. **I get a ridiculous %%BoundingBox in my exported PostScript file**

   POSSIBLE CAUSE AND SOLUTION:

   The Solaris cc compiler version 4.2 has a bug in the optimizer. Recompile **fig2dev** with only **-O2** or **-O1** optimization.

10. ## Patterns are coming out all black when viewing the exported PostScript file with ghostscript 5.50

POSSIBLE CAUSE AND SOLUTION:

Ghostscript versions 5.10-5.50 have a bug which causes this to happen.
Get a newer version of Ghostscript.

---

11. ## I am exporting to the MetaFont format and the generated file refers to the ``grafbase.mf" file which I don't have

POSSIBLE CAUSE AND SOLUTION:

The grafbase.mf file should be on your system if you have MetaFont, but if you don't have it you can get it from ftp://ctan.tug.org/tex-archive/graphics/fig2mf/

---

12. ## My pulldown menus don't work under CDE

POSSIBLE CAUSE AND SOLUTION:

Under CDE you must turn off the **Num Lock** and **Caps Lock** keys for the pulldown menus to work.

---

13. ## I am getting get "mn" in my LaTeX text

POSSIBLE CAUSE AND SOLUTION:

The problem is in the \smash part of a command that was generated in the latex part of the export. If the `\mddefault` and `\updefault` are not set up properly on your system you need to define them to do nothing.

---

14. ## Pattern fills are going beyond the bounds of the polygons when I export to EPS, GIF, etc., although they look fine on the screen

POSSIBLE CAUSE AND SOLUTION:

Ghostscript versions 7.00 has a bug (#418928) which causes this to happen.
Get a newer version of Ghostscript.

---

15. ## xfig says: "Warning: Representation size 4 must match superclass's to override underline"

POSSIBLE CAUSE AND SOLUTION:

The 3D menu widget (smeBSBObject) was used instead of the 2D version or vice versa (check -DXAW3D in Imakefile/Makefile)
Don't mix the 2D and 3D widget libraries (Xaw and Xaw3d) and header files.

16. ## fig2dev says: "can't open file: /usr/X11R6/lib/X11/fig2dev/en_US.ps" (or some

other xx_xx.ps file)

POSSIBLE CAUSE AND SOLUTION:

This seems to happen only on RedHat systems. I believe that they have enabled internationalization even when it shouldn't be, and it is looking for a non-existent font encoding file.
Download newer versions of xfig and transfig RPMs from RedHat.

17. xfig won't find the app-defaults file in my personal directory even though I have set the XAPPLRESDIR environment variable

POSSIBLE CAUSE AND SOLUTION:

If you are using Sun's Openwindows X server you need to set a different environment variable, **XUSERFILESEARCHPATH** to point to your personal app-defaults directory, and append a "%N" to the path, e.g.

```
setenv XUSERFILESEARCHPATH $HOME/app-defaults/%N
```

18. I can't enter special characters such as a-umlaute in the popup editor for text objects.

POSSIBLE CAUSES AND SOLUTIONS:

1. The **CompKeyDB** file may not be installed. This is the file that tells xfig how to handle the multi-key sequences to produce the Latin-1 characters such as ä (a-umlaute), ç (c-cedilla), etc. You should do a `make install` to fully install xfig and its support files.
2. Your keyboard doesn't have a **"Multi_key"** key. On some systems (e.g. Sun Sparcstations), there is a compose key or "Multi_key" which handles these sequences. If your keyboard doesn't have such a key you can remap another key (e.g. the "Windows" keys, known to the X server as **Super_L** and **Super_R** to the Multi_key using the **xmodmap** program:

```
xmodmap -e "keysym Super_L = Multi_key"
xmodmap -e "keysym Super_R = Multi_key"
```

Problems when compiling xfig

1. On a DEC Alpha the compiler may complain about something about something like ``.mask must have pcreg from .frame set if any bits are set in .mask or .fmask" in the file `u_bound.c`

POSSIBLE CAUSE AND SOLUTION:

Set the optimization level to `-O1` at least for that file.

2. `XPointer` is undefined or unknown when I compile xfig

POSSIBLE CAUSE:

You are probably trying to compile **xfig** on an old OpenWindows system which doesn't have the `XPointer` typedef.

SOLUTION:

You will have to either upgrade to a newer version of OpenWindows or switch to the public X server from the Open Group.

3. On Sun machines the linker may give erroneous error messages about `_get_applicationShellWidgetClass` and `_get_wmShellWidgetClass` are undefined

POSSIBLE CAUSE:

It is a problem with the Sun shared libraries and the way X11 builds the shared Xmu library.

SOLUTION:

It doesn't affect the operation of **xfig** and you can ignore it.

Or you may compile with ``` ``-Bstatic -lXmu -Bdynamic ```".

To really solve the problem if you are using OpenWindows 3.0 (X11R4-based Xt), please contact your local Sun office and request the following patches:

```
Patch i.d.      Description
100512-02       4.1.x OpenWindows 3.0 libXt Jumbo patch
100573-03       4.1.x OpenWindows 3.0 undefined symbols when using
                        shared libXmu
```

A source patch for use with the MIT X11R4 libraries was developed by Conrad Kimball (cek@sdc.boeing.com); it retrofits into R4 some fixes made in R5 to get around this problem. The patch is on export in [1/93] `contrib/X11R4_sunos4.1.2_patch_version3.Z`.

The other option is to use X11R5 or X11R6.

4. On some HP computers, the compiler dies with bus error in the file `u_undo.c`

POSSIBLE CAUSE:

There is a bug in the HP compiler. This problem may occur on some HP computers, including HP 750 running HP-UX 8.05.

SOLUTION:

Workaround the problem by adding an extra statement between two statements in `u_undo.c`:

Original:

```
undo_glue()
{
    list_delete_compound(&objects.compounds, saved_objects.compounds);
```

```
        tail(&objects, &object_tails);
```

New:

```
    undo_glue()
    {
        list_delete_compound(&objects.compounds, saved_objects.compounds);
    >>>         /* HP-UX 8.05 compiler bug fix -- don't ask */
    >>>         last_action = last_action;
        tail(&objects, &object_tails);
```

---

5. On some HP computers using native `cc` compiler there is a problem with +O3 optimization

SOLUTION:

Use `+O2` instead of `+O3` or get patch from HP for the compiler; connect to http://support.mayfield.hp.com/ and choose **Patch Browsing and Downloading**.

---

6. I'm getting errors such as

Unresolved:
jpeg_std_error
jpeg_create_decompress
jpeg_destroy_decompress
jpeg_stdio_src
and others.

POSSIBLE CAUSE:

On most Linux machines, there is an old version of the JPEG library already installed, and is incompatible with **xfig**. **xfig** can't use a JPEG library older than Version 5b.

SOLUTION:

Delete its library files (usually `/usr/local/lib/libjpeg.so` and `/usr/local/lib/libjpeg.a`) and the header file (usually `/usr/local/include/jpegdata.h`).

---

7. I am getting these errors when I am compiling xfig:

w_menuentry.c: In function `Redisplay':
w_menuentry.c:129: structure has no member named `fontset'
w_menuentry.c:143: structure has no member named `international'

POSSIBLE CAUSE:

You probably have an old version of the 3D Athena Widget set (Xaw3d) which doesn't have the `fontset` and `international` components, but your Intrinsics library **does** have those components.

SOLUTION:

Either get a newer version of the Xaw3d widgets from ftp://ftp.x.org/contrib/widgets/Xaw3d or use your system's 2D Athena widgets (Xaw). The latter is acheived by:
1. commenting out the "XAWLIB = -lXaw3d" line in the Imakefile,
2. doing a "rm w_menuentry.o",
3. "make Makefile"
4. "make"

[Contents | Introduction | Credits]

Xfig User Manual

Version 3.2.4

December 2002

Credits

Xfig was originally written by Supoj Sutanthavibul at the University of Texas at Austin in 1985 for SunView. Later, Ken Yap, at Rochester, New York, did the first port of **xfig** to X11. In 1989, Brian V. Smith picked it up and added features such as multiple fonts, line thickness, area fill etc. Around 1991, Paul King at the University of Queensland, Austrailia added many features and overhauled the look and feel of **xfig** for version 2.0, to produce essentially what you see today. In 1992, Brian Boyter added the ability to import EPS files, and later it was expanded to import several other bitmap formats as well. Mr. Tom Sato from Japan added the Japanese text support and the spell checker and search/replace feature in 1997. He has added several other features mentioned in the New Features section and fixed several Bugs.

Brian V. Smith is the main contact for **xfig** and its incorporation of new features written either by himself or others.

There have been dozens of people who have contributed to the success (and code) of **xfig**, and they are too numerous to mention here. The man pages list some of those people. See below for the major copyrights.

Copyright/Permission Notices

This Documentation

This documentation was first written in Japanese by Tom Sato, based on the manual pages distributed with **xfig**. It was then translated into English with the help of some kind people. Later, it was updated to conform to the latest version by Brian Smith.

want to distribute modified version of this documentation.

You may also put these documentation files on your `public' WWW server. In such case, please send its URL to Brian V. Smith or Tom Sato so that we can maintain a list of such WWW servers.

Xfig

- Original Copyright (c) 1985 by Supoj Sutanthavibul
  Parts Copyright (c) 1989-2002 by Brian V. Smith
  Parts Copyright (c) 1991 by Paul King
  Other Copyrights may be found in various files

  Main Xfig copyright notice:

  Any party obtaining a copy of these files is granted, free of charge, a full and unrestricted irrevocable, world-wide, paid up, royalty-free, nonexclusive right and license to deal in this software and documentation files (the "Software"), including without limitation the rights to use, copy, modify, merge, publish and/or distribute copies of the Software, and to permit persons who receive copies from any such party to do so, with the only requirement being that this copyright notice remain intact.

  No representations are made about the suitability of this software for any purpose. It is provided ``as is'' without express or implied warranty.

- Parts Copyright (C) 1993 by Alan Richardson (mppa3@uk.ac.sussex.syma)
  The text rotation code in w_rottext.c was written by Alan Richardson. The above copyright notice holds for this work as well.

- Parts Copyright (c) 1994 by Anthony Dekker
  The Kohonen neural network code for color optimization was written by Anthony Dekker.

- Parts Copyright (c) 1995 by C. Blanc and C. Schlick
  The X-Spline code was written Carole Blanc (blanc@labri.u-bordeaux.fr) and Christophe Schlick (schlick@labri.u-bordeaux.fr) starting from an initial implementation done by C. Feuille, S. Grosbois, L. Maziere and L. Minihot as a student practice (Universite Bordeaux, France). For additional information about X-splines, see:
  ``X-Splines: A Spline Model Designed for the End User'' by C. Blanc and C. Schlick, Proceedings of SIGGRAPH'95 http://dept-info.labri.u-bordeaux.fr/~schlick/DOC/sig1.html

About GIF Support

Because Unisys has stated that they WILL charge royalties for the use of the LZW compression algorithm even in FREE programs, I have removed all traces of the GIF LZW compression/decompression code from ***xfig****.* ***Xfig*** *now calls giftopnm and ppmtopcx to import GIF files. Screen capture writes a PNG file. Exporting is handled by calling the ppmtogif program from fig2dev.*

Contact

Please send any questions, bug fixes, contributions and any comments to following destinations. When reporting a bug, please first check if the problem is mentioned in the FAQ section of the HTML files (xfig Help menu).

If it is not mentioned there, be sure to report the operating system you are using (e.g. SunOS 4.1.3), type of X server and version (OpenWindows 3, X11R6.4, XFree86 3.2.2, etc) and color depth (e.g. 8bpp, 16bpp), and most importantly the version of **xfig** or **fig2dev** that is having the problem.

It is also VERY useful to me if you can provide a stack trace from a debugger such as gdb, dbx, ups, etc.

- About this manual or **xfig** and **TransFig** themselves (*except Japanese support facility*)
  --- *xfig-bugs@epb.lbl.gov* (Brian V. Smith)

- About Japanese support facility
  --- *VEF00200@nifty.ne.jp* (Tom Sato)

Japanese users can also send questions about **xfig** or **TransFig** to *VEF00200@nifty.ne.jp*.

[Contents | Introduction | Credits]