Xfig User Manual
Version 3.2.8b-dev   **Aug 2021**

Table of Contents

[**English Version** | Japanese Version]

written by us

Xfig User Manual
Version 3.2.8b–dev   Aug 2021

[**English Version** | Japanese Version]

Table of Contents

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, and e.g. under Darwin on the MacIntosh and any X server under Microsoft Windows. It is freeware, and can be downloaded freely. See the installation section for details.

In **xfig**, figures may be drawn using objects such as circles, boxes, lines, spline curves, text, 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". It is also possible to embed images in formats such as GIF, JPEG, EPS (PostScript), etc.

Here is a screen image of **xfig**. Click on the image below for larger version (349k).
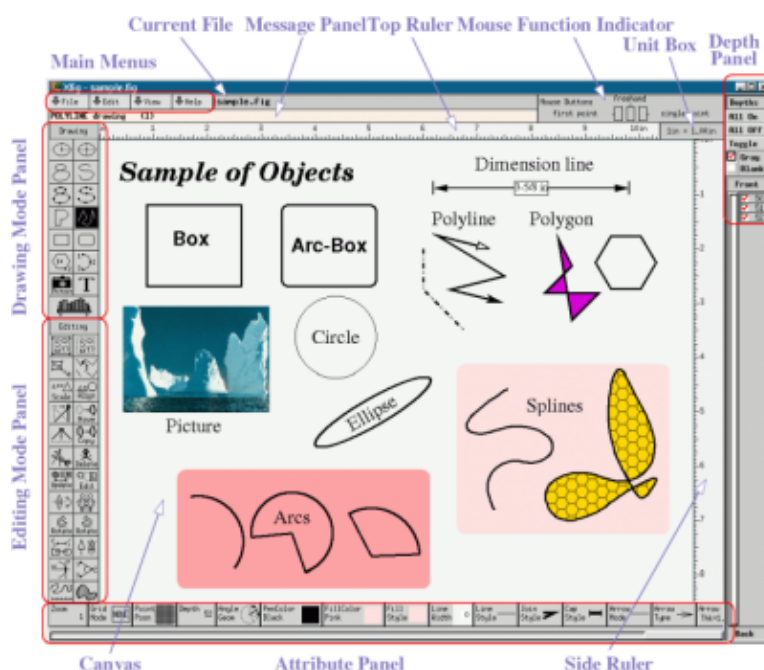
And here are some example figures extracted from the **xfig** distribution. Click on them to see a larger version.
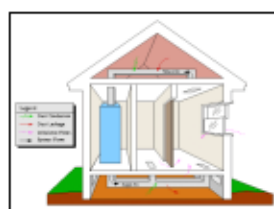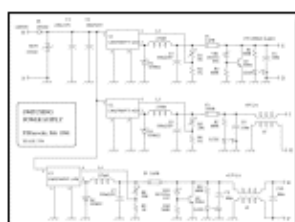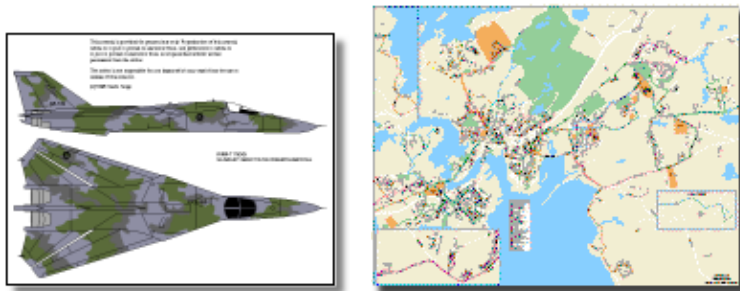
Xfig saves figures in its native *Fig format*, but they may be converted into various formats such as PostScript, GIF, JPEG, HP-GL, etc. **xfig** has a facility to print figures to a PostScript printer, too.

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 embed 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.

🔴 <u>Mouse Function Indicator</u>:

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.

🔴 <u>Rulers</u>:

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.

🔴 <u>Depth Panel</u>:

This panel shows the <u>depths</u> of all objects on the canvas. The user may hide or show any depth by clicking on the checkboxes next to the depth number.

🔴 <u>Units</u>:

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 <u>top and side rulers</u> or the arrow keys on the keyboard.

[<u>Contents</u> | Introduction | <u>Credits</u>]

Xfig User Manual
Version 3.2.8b–dev   **Aug 2021**

Main Menus

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

- <u>File</u>: File menu
- <u>Edit</u>: Edit menu
- <u>View</u>: View menu
- <u>Snap</u>: Snap menu
- <u>Help</u>: Help menu

```
New          (Meta-N)
Open...      (Meta-O)
Merge...     (Meta-M)
Save         (Meta-S)
Save As...   (Meta-A)
Export...    (Meta-X) (Quick = Shift-Meta-X)
Print...     (Meta-P) (Quick = Shift-Meta-P)
Exit         (Meta-Q)
1 /home/joe/sample1.fig
2 /home/joe/plan.fig
3 /home/joe/figs/testfig.fig
4 /home/joe/figs/floorplan.fig
```

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 <u>Undo</u>.
- **Open...**

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

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

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

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

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

    Popup panel to print current figure to PostScript printer. See <u>Printing</u> 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.

```
Undo              (Meta-U)
Paste Objects     (Meta-T)
Paste Text        (F18/F20)
Search/Replace... (Meta-I)
Spell Check...    (Meta-K)
Delete All        (Meta-D)
Global settings... (Meta-G)
Set units...
```

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.

```
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
```

View Menu

- **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 <u>Manage Styles...</u> 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 <u>Exporting</u> or <u>Printing</u> a figure.

    The default orientation may be specified by <u>command line options</u>. The orientation is automatically changed when <u>loading</u> 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 <u>Zooming</u>.

- **Zoom out**

    Zoom in on the canvas by a factor of two. A lowercase `z` will also perform this function. See also <u>Zooming</u>.

- **Zoom to Fit canvas**

    Zoom the figure so that it just fits in the canvas. `Ctrl-Z` will also perform this function. See also <u>Zooming</u>.

- **Pan to origin**

    Pan the canvas to the origin (0,0). See also <u>Panning the Canvas</u>.

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 <u>Exporting</u> or <u>Printing</u> for page sizes supported).

- **Show depth manager**

    Show or hide the <u>depth manager panel</u> to the right of the side ruler. See <u>Layers (Depths)</u> for details.

- **Show info balloons**

    Turn on or off displaying of <u>information balloons</u>. 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 <u>Global Settings</u> 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.

```
Hold
Release
Endpoint
Midpoint
Focus
Normal
Tangent
Intersection
```

Snap Menu

The Snap operations are used to snap the next point clicked by the user to some location on a selected object. For example, after invoking the Endpoint snap mode, the next point clicked by the user will be used to identify the nearest object and then, if that object has some feature that can be reasonably interpreted as an "endpoint" the input point will be translated to that endpoint. If the selected object has more than one endpoint, the one closest the initial point will be used.

Snaps can be used either in drawing mode, as described above, or in edit mode. You could, for example, move an ellipse such that one of its foci coincides wiith the midpoint of a polyline segment:

1. Select "Move"
2. Select "Focus" from the Snap pulldown.
3. Select the ellipse at some point on it near the focus.
4. Select "Midpoint" from the Snap pulldown.
5. Select the polyline segment.

The following is a list of the available snap modes.

● **Hold**

Usually, the selected snap operation is automatically deselected immediiately after the the next point is clicked. This button suppresses that behaviour, keeping the selected snap operation in effect until it is manually released with the Release button, q.v. The Hold capability allows, e.g., the connection of a series of polyline vertices by another polyline.

● **Release**

Releases a locked snap mode being retained by the Hold button, or cancels a snap operation after it's been selected.

● **Endpoint**

Endpoint snap mode snaps the input point to some feature of the selected object that may be interpreted as an endpoint. The feature selected depends on the type of object selected:

● Polylines (including boxes and polygons):
Snaps to the nearest vertex.
● Ellipses (not including circles):
Snaps to the nearest endpoint.of any of the semi-axes. (This is not relevant to circles.)
● Text
Snaps to the nearest vertex of the bounding box.
● Arcs
Snaps to the nearest of the three arc definition points.

- **Midpoint**

    Midpoint snap mode snaps the input point to some feature of the selected object that may be interpreted as a midpoint. The feature selected depends on the type of object selected:

    - Polylines (including boxes and polygons):
        Snaps to the midpoint of the nearest polyline segment.
    - Ellipses (including circles):
        Snaps to the centerpoint.
    - Text
        Snaps to the midpoint of the nearest side of the bounding box.
    - Arcs
        Snaps to the arc centerpoint.

- **Nearest**

    While a polyline is being drawn, this operation simply snaps to the nearest point on the nearest object. This works for target polyline, circle/ellipse, text (the bounding box), and arc objects.

- **Focus**

    Focus snap mode snaps the input point to some feature of the selected object that may be interpreted as a focus. The feature selected depends on the type of object selected:

    - Polylines (including boxes and polygons):
        Snaps to the unweighted centroid of the vertices.
    - Ellipses (including circles):
        Snaps to the nearest ellipse focus, or to the centerpoint of circles.
    - Text
        Snaps to the centroid of the bounding box.
    - Arcs
        Snaps to the arc centerpoint.

- **Diameter**

    While a polyline is being drawn, this operation snaps to a point such that the midpoint of the segment specified by that point and the "current" point is at the centerpoint of the selected object. This works for all objects except splines.

- **Normal**

    While a polyline is being drawn, this operation snaps to a point on the selected object such that the segment from the "current" point to the object is perpendicular to the object. If there are multiple possible normal points (like to the near and far sides of a circle), the snap is to the one closest to where the user clicked. This works for target polyline, circle/ellipse, text (the bounding box), and arc objects.

- **Tangent**

    While a polyline is being drawn, this operation snaps to a point on the selected object such that the segment from the "current" point to the object is tangent to the object. If there are multiple possible tangent points (like to the sides of a circle), the snap is to the one closest to where the user clicked. This works for target circle/ellipse and arc objects.

- **Intersection**

    Lets the user select a second object and snaps to the nearest intersection of the two objects. Works between all combinations of circles, ellipses, polylines, and arcs except (yet) the case of two ellipses at different angles

```
Xfig Reference (HTML)...
Xfig Reference (PDF)...
Xfig Man Pages (HTML)...
How-To Guide (PDF)...
About Xfig...
```

Help Menu

- **Xfig Reference (HTML)...**
  This starts up your web browser (resource `Fig.browser`, default *xdg-open*) to view the main **xfig** reference manual.
- **Xfig Reference (PDF)...**
  These start up your PDF viewer (resource `Fig.pdfviewer`, default *xdg-open*) to view the PDF version of the reference manual.
- **How-To Guide (PDF)...**
  This starts up your PDF viewer (resource `Fig.pdfviewer`, default *xdg-open*) to view the **xfig** tutorial, written by Peter Hiscocks.
- **Man pages (HTML)...**
  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**.
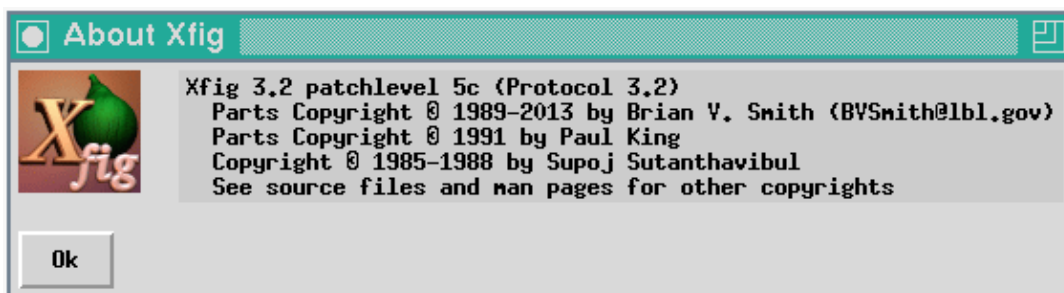
File Panels

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

- Choosing *Open...* from the <u>*File* menu</u> will pop up up the **Open Figure** panel. The accelerator `Meta-O` will also perform this function.
- Choosing *Merge...* from the <u>*File* menu</u> will pop up up the **Merge Figure** panel. The accelerator `Meta-M` will also perform this function.
- Choosing *Save As...* from the <u>*File* menu</u> 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":
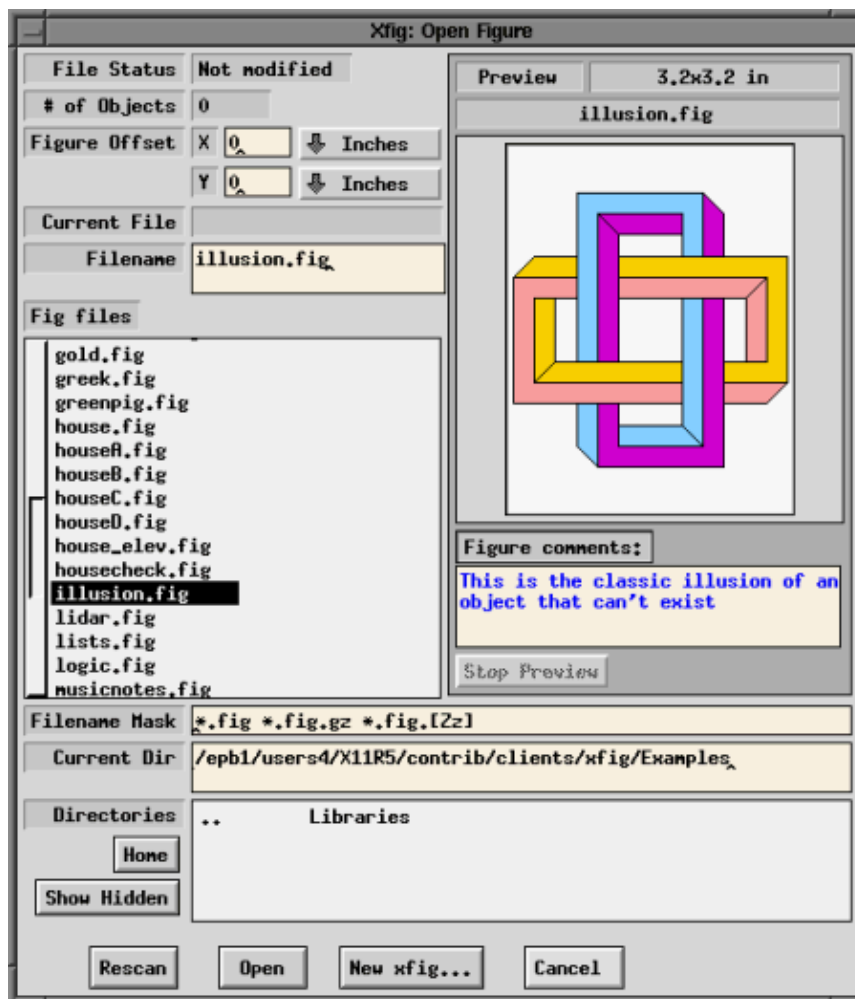
• *File Status*

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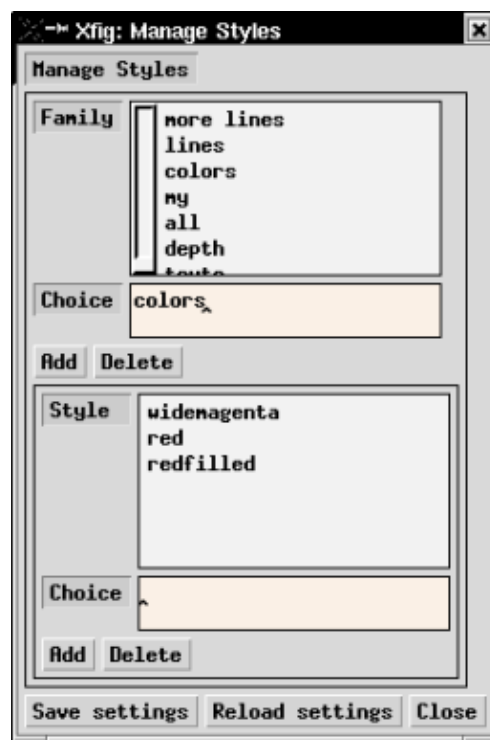
The style manager lets you save and load groups of attributes (e.g. line color, thickness, font size etc.) by name. For example you may want a thick, dashed, red line style often, so using the style manager you could choose those attributes and save and recall them in a style called "thickred".

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 3.2.8b-dev   Aug 2021

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

Polygons

Regular Polygons

Rectangular Boxes

Arc Boxes (Rectangular Boxes with rounded corners)

Arcs

Picture Objects (Imported image files)

Texts

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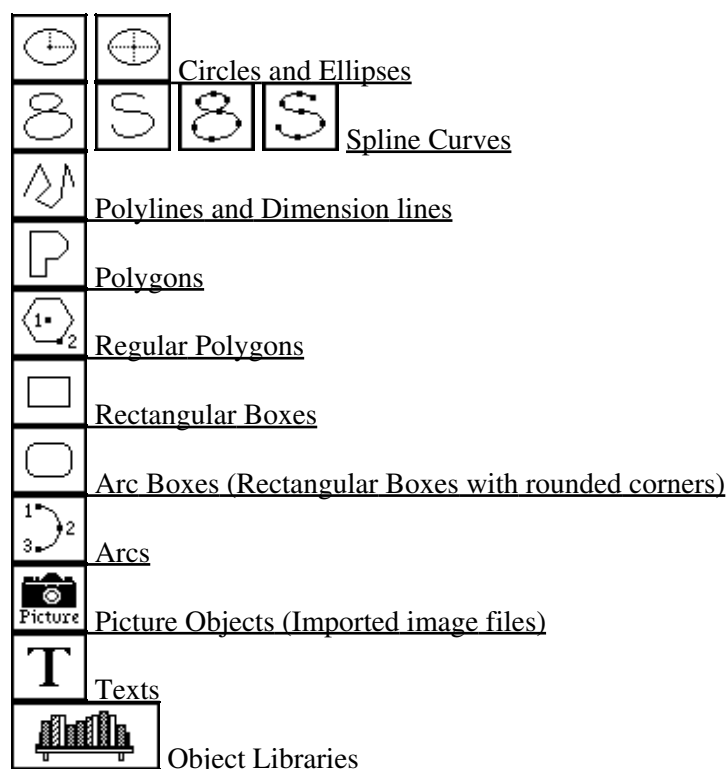
The drawing mode is selected by clicking on the corresponding button. The image on the button will turn negative, indicating that it is selected. Settings in the Attribute Panel will be applied to newly created objects, so users should set those attributes before creating new objects.

For overlapping objects, the depth attribute is important for **xfig** to know the drawing order. Please see Layers (Depths) for details.

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.
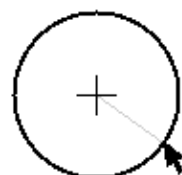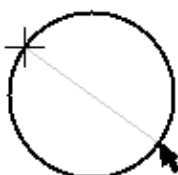
Circle by radius   Circle by diameter

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.
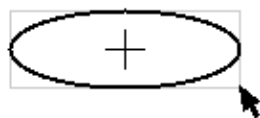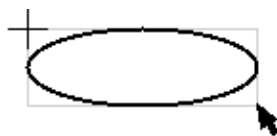
Ellipse by radius   Ellipse by diameter

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)

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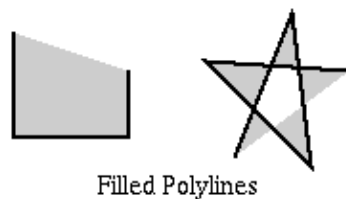
For all objects, clicking mouse button 3 (`cancel') will cancel all input. For objects that may have more than two points, clicking mouse button 1 while pressing the SHIFT key (`del point') will cancel entered points one by one.

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

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
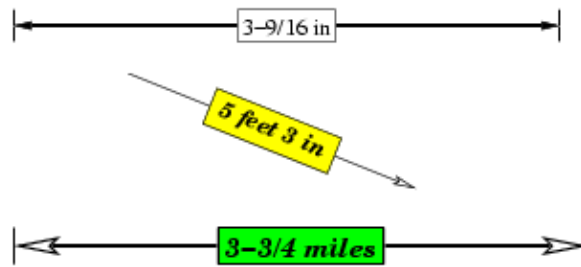
dimension lines:

Related Attributes:
> dimension line settings

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

> **Related Attributes:**
> > <u>depth</u>, <u>pen color</u>, <u>fill color</u>, <u>fill style</u>, <u>line width</u>, <u>line style</u>, <u>join style</u>

 (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 <u>NUM SIDES</u> in advance.

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

> **Related Attributes:**
> > <u>depth</u>, <u>pen color</u>, <u>fill color</u>, <u>fill style</u>, <u>line width</u>, <u>line style</u>, <u>join style</u>, <u>num sides</u>

 (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 <u>FILL STYLE</u>, 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 <u>POLYGON</u> (it may be easier to create non-rotated rectangle first and then rotate it by <u>ROTATE</u>). You may also use <u>REGULAR POLYGON</u> to create a rotated square.

> **Related Attributes:**
> > <u>depth</u>, <u>pen color</u>, <u>fill color</u>, <u>fill style</u>, <u>line width</u>, <u>line style</u>, <u>join style</u>

 (ARC-BOX)

Create rectangular boxes with rounded corners. This is identical to <u>BOX</u> except that the corners of the box will be arcs with the radius specified by <u>BOX CURVE</u>.

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

Related Attributes:
> <u>depth</u>, <u>pen color</u>, <u>fill color</u>, <u>fill style</u>, <u>line width</u>, <u>line style</u>, <u>join style</u>, <u>box curve</u>

(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 <u>ARC TYPE</u>.
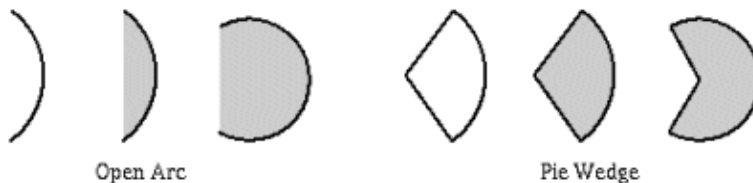
Open Arc   Pie Wedge

Related Attributes:
> <u>depth</u>, <u>pen color</u>, <u>fill color</u>, <u>fill style</u>, <u>line width</u>, <u>line style</u>, <u>cap style</u>*, <u>arrow mode</u>*, <u>arrow type</u>*, <u>arc type</u>
> (* These attributes are only effective for OPEN ARC)

(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 <u>Edit Panel</u> 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 *<u>Screen Capture</u> button*.

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

> ● <u>EPS (Encapsulated PostScript)</u>

- 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 xdg-open 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 EPS 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, 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 `ç' (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 <u>international-**xfig**</u>, it is possible to <u>enter local text such as Japanese</u> 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:

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

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

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

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 <u>compound</u> 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:

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

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

Related Attributes:
   <u>depth</u>, <u>pen color</u>, <u>text font</u>, <u>text size</u>, <u>text step</u>, <u>text flags</u>, <u>text/ellipse angle</u>, <u>text justification</u>

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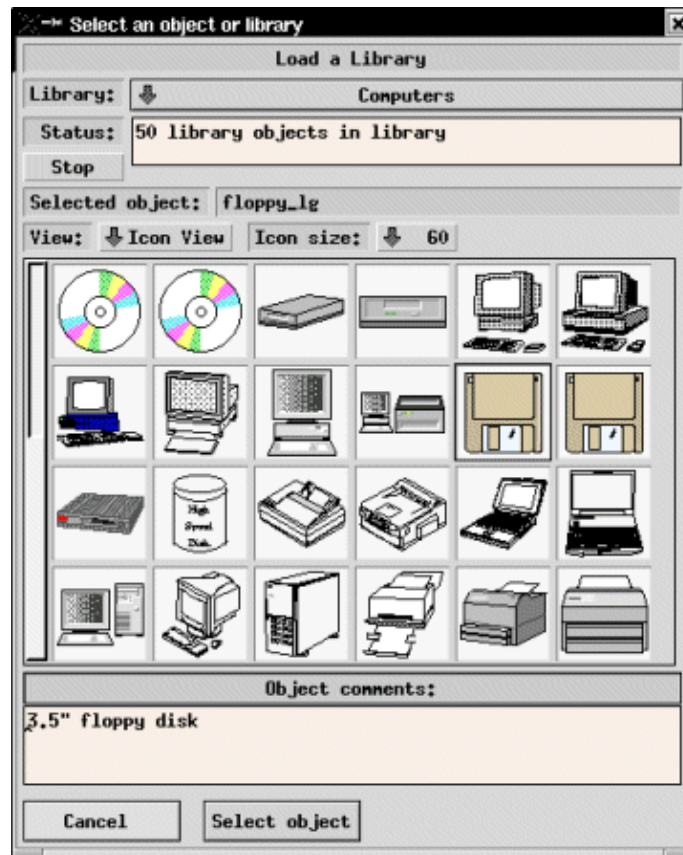
In the panel there is a pull-down menu of the underlined available libraries. Select one and the library will automatically be loaded. The status window will show how many objects it contains after it loads the library, and the icons of the library objects will appear in a view box. Double clicking on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

A diamond on the left side of the menu entry indicates that objects are at that level in addition to
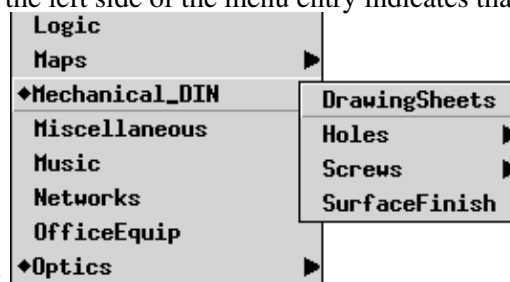
subdirectories:
The size of the icons may be chosen from a pulldown menu. The options are 40, 60, 80, 100 and 120 pixels.

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.

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.

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 thomas.loimer@tuwien.ac.at. 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 <u>installing</u> **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 <u>Drawing Mode Panel</u> of **xfig**. It is also possible to change the type of spline curves using <u>SPLINE <-> LINE</u>, or adjust the shape of the curve by adjusting the "<u>shape factor</u>" after a curve has been created.

🔵 CLOSED APPROXIMATED SPLINE

Smooth closed curve which approximates specified points.

🔵 OPEN APPROXIMATED SPLINE

Smooth curve which approximates specified points.

🔵 CLOSED INTERPOLATED SPLINE

Smooth closed curve which passes through specified points.

🔵 OPEN INTERPOLATED SPLINE

Smooth curve which passes through specified points.

Closed Approx. Spline   Open Approx. Spline   Close Interp. Spline   Open Interp. Spline
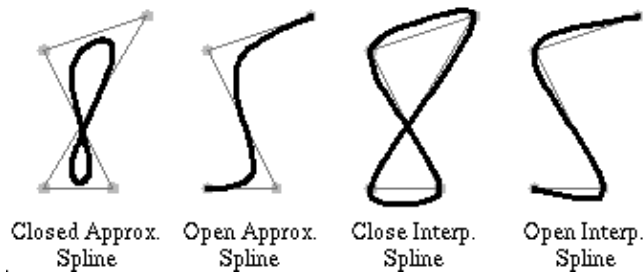
Using splines, curves such as the following may be easily drawn.
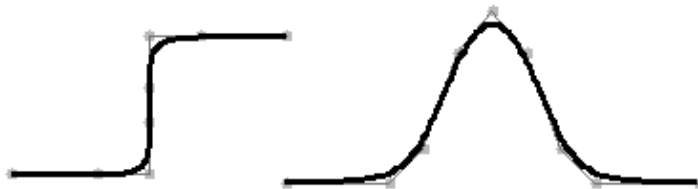
🟢About X-Splines

xfig's new X-Spline is bit more flexible. Each point (*control point*) of the spline has a `*shape factor*'. The value of the *shape factor* controls the behavior of the curve near the point. The value of the *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.
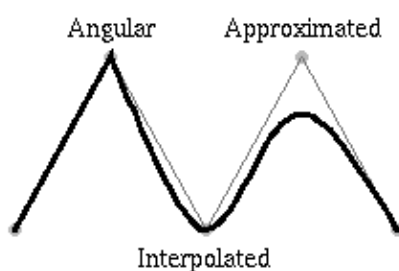
Caveat: Because previous spline models (quadratic B-splines and Bezier with hidden points) are no longer supported per se, curves that are present in **xfig** version 3.1 and older files are automatically converted to X-splines. This translation is only an approximation process. It means that the converted 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 <u>TEXT</u> 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	14	…	D−	∠	O"	√
??	⌟	12	⏐			O/	¬
C/	′	34	—	E`	∪		
L−	≤	**	.	E'	⊃	o`	∫
OX	⁄	−:	|	E^	⊇	o'	
Y−	∞	/u	∝	E"	⊄	o^	
||	ƒ	P!	∂	e`	⌊	o~	
SO	♣	A`	ℵ	e'	⌈	o"	
""	♦	A'	ℑ	e^	|	o/	
CO	♥	A^	ℜ	e"	⌊		
RO	→	A~	℘	I`	⊂	P|	⇒
_a	♠	A"	⊗	I'	⊆	p|	
_o	≡	A*	⊕	I^	∈	ss	⇓
<<	↔	AE	∅	I"	∉	U`	∧
>>	≈	a`	◊	i`	⌈	U'	∨
−|	←	a'	⟨	i'	{	U^	⇔
+−	±	a^	®	i^	|	U"	⇐
^−	↓	a~	©	i"	|	u`	
^*	°	a"	™	N~	∇	u'	
^.	•	a*	∑	n~	〉	u^	
^1	≠	ae	⌈	O`	®	u"	
^2	″	C,	∩	O'	©	Y'	⇑
^3	≥	c,	|	O^	™	y'	
\\\\	×	,,	÷	O~	∏		

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

!	!	8	8	O	Ο	g	γ
"	∀	9	9	P	Π	h	η
#	#	:	:	Q	Θ	i	ι
$	∃	;	;	R	Ρ	j	φ
%	%	<	<	S	Σ	k	κ
&	&	=	=	T	Τ	l	λ
'	∋	>	>	U	Υ	m	μ
(	(	?	?	V	ς	n	ν
)	)	@	≅	W	Ω	o	o
*	*	A	Α	X	Ξ	p	π
+	+	B	Β	Y	Ψ	q	θ
,	,	C	Χ	Z	Ζ	r	ρ
-	−	D	Δ	[	[	s	σ
.	.	E	Ε	\	∴	t	τ
/	/	F	Φ	]	]	u	υ
0	0	G	Γ	^	⊥	v	ϖ
1	1	H	Η	_	—	w	ω
2	2	I	Ι	`		x	ξ
3	3	J	ϑ	a	α	y	ψ
4	4	K	Κ	b	β	z	ζ
5	5	L	Λ	c	χ	{	{
6	6	M	Μ	d	δ	|	|
7	7	N	Ν	e	ε	}	}
				f	φ	~	~

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 [◆].

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 3.2.8b–dev   **Aug 2021**

Editing Objects

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

<u>Glue COMPOUND</u>

<u>Break COMPOUND</u>

<u>Open COMPOUND temporarily</u>

<u>Join/Split Lines/Splines/etc</u>

<u>Chop Object</u>

<u>Scale Object</u>

<u>Align Objects</u>

<u>Move Point</u>

<u>Add Point</u>

<u>Delete Point</u>

<u>Move Object</u>

<u>Copy Object</u>

<u>Delete Object</u>

<u>Update Attributes of Object</u>

<u>Edit Attributes of Object</u>

<u>Flip Object Vertically or Horizontally</u>

<u>Rotate Object</u>

<u>Convert SPLINE to/from LINE or BOX to/from ARC-BOX</u>

<u>Add/Delete Arrows</u>

<u>Add Tangent or Normal to Curve</u>

<u>Measure Angle</u>

<u>Measure Length of Lines and Arcs, and Perimeters of Circles and Ellipses</u>

Measure Area of Polygons and Ellipses

Glue selected objects and make them a COMPOUND object.
This is useful to <u>move</u> or <u>copy</u> some objects together at the same time. It is also possible to scale part of a figure by <u>scaling</u> 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 <u>BREAK COMPOUND</u>.

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 <u>GLUE COMPOUND</u> again.

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

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.

If mouse button 2 (`open, keep visible') is clicked on the COMPOUND, the other objects not in the COMPOUND object will become shades of gray, making it easy to align objects inside the compound with those outside.

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 <u>BREAK COMPOUND</u>.

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, <u>delete</u> an object in that COMPOUND object, close the COMPOUND object and <u>Undo</u> 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.

Chopping objects.

The Chop operation allows you to chop polylines, arcs, and circles into individual objects that, together, approximate the original object. (Ellipses will be supported in a future release.)

Left-clicking on one or more objects selects them as "axes". When another object is subsequently selected with a middle click, all possible intersections between that object and all "axe" objects are computed. These intersections are then used to create a collection of new objects that approximate the original. Additional "axe" objects can be added at anytime, and any number of objects may be chopped. Right-clicking clears the axe list.

Circles will be partitioned into pie-wedge arcs and arcs will be partitioned into either open or pie-wedge arcs, depending on the type of the original arc. (In both cases, the type can be changed by editing the newly created objects.) Open polylines and unconstrained polygons (i.e., closed polylines other than boxes, arcboxes, and picture boxes) will be chopped into one or more objects of the same type. Constrained polygons cannot be chopped.

How polygons are chopped is controlled by the "Arc Type" mode in the indicator panel. If "Open" mode is selected, the polygons that result from the cut operation will defined only by the vertices of the original polygon and the intersection points; "Closed" mode will result in polygons that include, in addition to the original vertices and intersects, the geometric centerpoint of the original polygon.

Chop operations cannot (yet) be undone.

Scale object

Clicking any **corner** of <u>BOX</u>, <u>ARC-BOX</u>, or <u>COMPOUND</u> 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 <u>TEXT</u> objects directly with this function. To scale TEXT objects, use <u>EDIT OBJECT</u> or <u>UPDATE</u> and change the size of the text (see <u>TEXT SIZE</u>). However, if a <u>COMPOUND</u> 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 <u>MOVE POINT</u> 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 <u>VERTICAL ALIGN</u> and <u>HORIZONTAL ALIGN</u>.

Clicking on a <u>COMPOUND object</u> 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 <u>TEXT</u> objects. For <u>BOX</u>, <u>ARC-BOX</u>, and <u>COMPOUND</u> objects, not only the selected point but the edges connected to the point will be moved. The object created by <u>REGULAR POLYGON</u> is an ordinary <u>POLYGON</u> object, and only the selected point will be moved.

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

Add a new point to a <u>POLYLINE</u>, <u>POLYGON</u>, or <u>SPLINE</u> 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 <u>BOX</u> or <u>ARC-BOX</u> objects. The object created by <u>REGULAR POLYGON</u> is an ordinary <u>POLYGON</u> object, and it is possible to add new points anywhere in it.

Points may also be <u>moved</u> or <u>deleted</u>.

Delete point from <u>POLYLINE</u>, <u>POLYGON</u>, or <u>SPLINE</u> 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 <u>BOX</u> or <u>ARC-BOX</u> object. The object created by <u>REGULAR POLYGON</u> is ordinary <u>POLYGON</u> object, and it is possible to delete points from it.

Points may also be <u>moved</u> or <u>added</u>.

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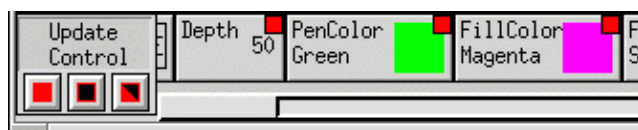
The attributes of objects may also be modified using EDIT OBJECT. Some attributes can't be modified by UPDATE but may be modified by EDIT OBJECT.

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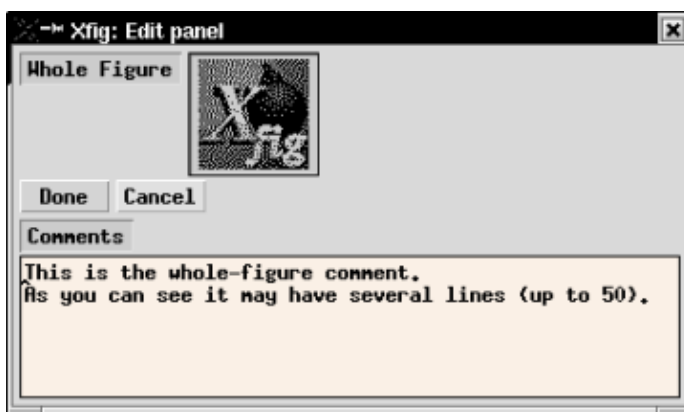
If the **Shift** key is held down while mouse button 2 (`show comments') is clicked on an object, any comments in the object are displayed in a popup until the mouse button is released:

  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 <u>ROTATE</u>.

  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 <u>ROTATION ANGLE</u> 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 <u>ROTATION ANGLE</u>. The number of copies to be created may be set by <u>NUMBER OF COPIES</u>.

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

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

  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

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

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

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

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

Xfig: Edit panel

POLYLINE: Polyline

Done | Apply | Cancel

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 | Thick 1.0

Width 4.0 | Width 4.0

Height 8.0 | Height 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).

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

TEXT object
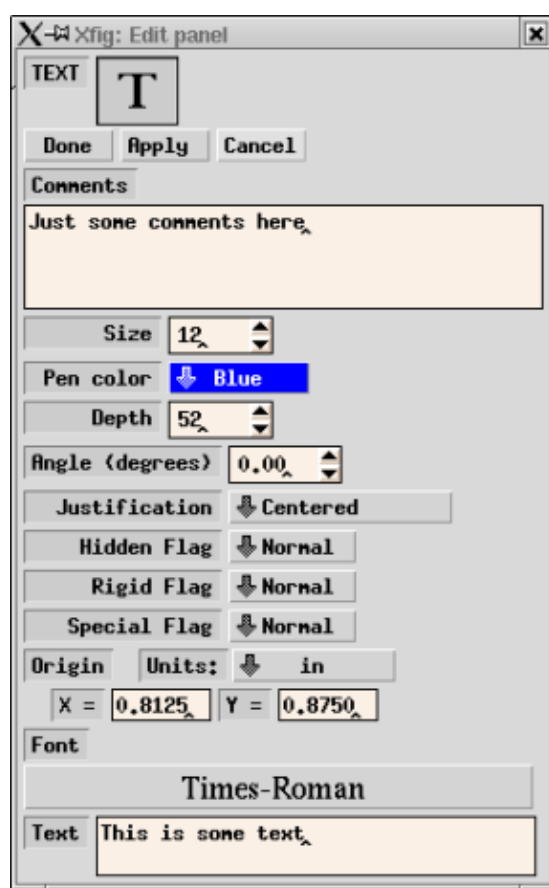
- *Size*
  The size of the text, in points. (see TEXT SIZE)
- *Pen Color*
  The color of the text. (see PEN COLOR)
- *Depth*
  The `depth' of the object. (see DEPTH)
- *Angle*
  The angle of the text, in degrees. Angles > 0 are counterclockwise. (see TEXT/ELLIPSE ANGLE)
- *Justification*

Indicates which part of the text is placed at the specified point. (see <u>TEXT JUSTIFICATION</u>)

● *Hidden Flag, Rigid Flag, and TeX Flag*

The state of those flags. (see <u>TEXT FLAGS</u>)

● *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 <u>TEXT FONT</u>)

● *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 <u>TEXT</u> mode, simply click any point of the text to be edit.

PICTURE object
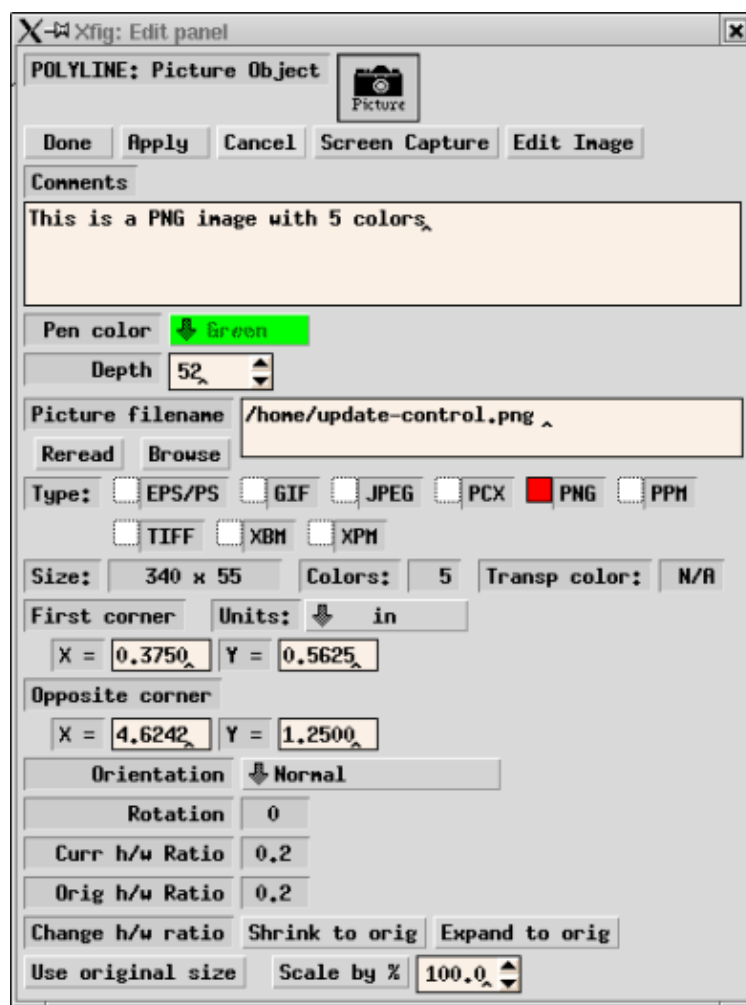
● *Screen Capture*

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.
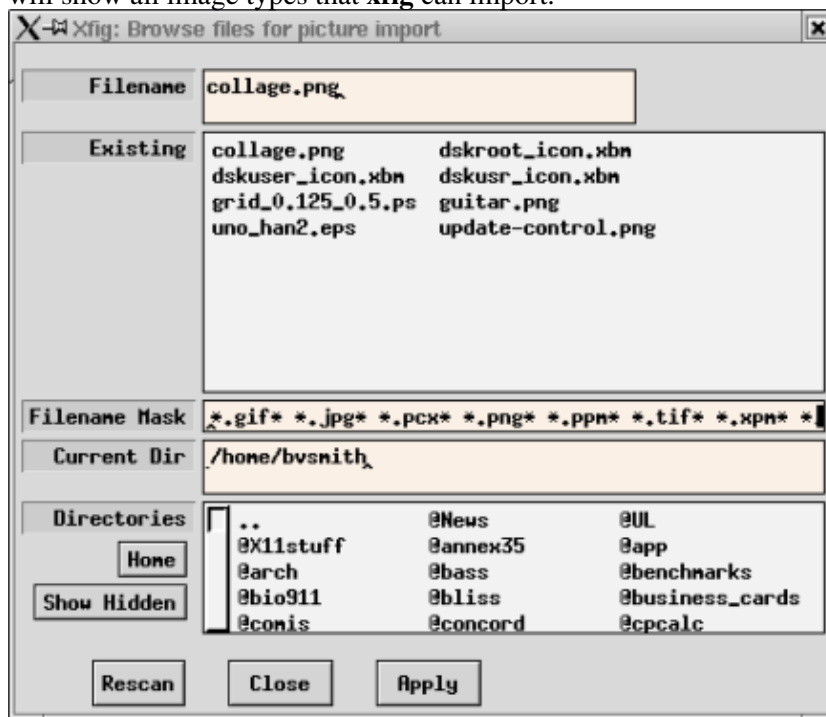
● *First Corner, and Opposite Corner*

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.
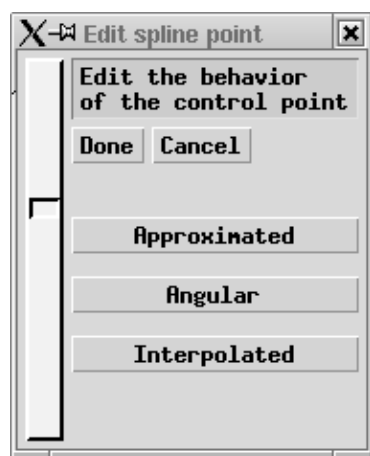
Clicking *Approximated*, *Angular*, or *Interpolated* button will set the shape factor to a preset value. The shape factor may also be set to arbitrary values by moving the slider on the left side of the panel. The modified spline curve will be displayed interactively in either case.
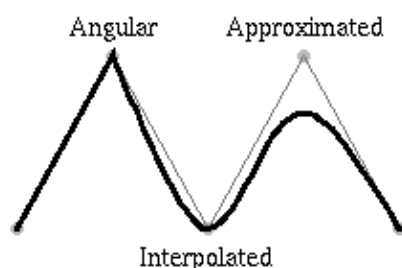
The shape factor of a control point may also be modified without popping up the Edit Point panel by clicking the control point while pressing both of SHIFT key and CONTROL key. Clicking a control point with mouse button 1 (`More approx') will decrease the shape factor, and clicking a control point with 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]

EDIT POINT

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 (magnify) figure in canvas

Grids
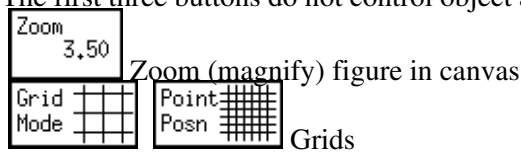
These are the attribute buttons:

Object depths

Colors and patterns

Width and shape of lines

Font and other attributes of TEXT objects

Arrows

Corner radius of ARC-BOX objects

Shape of ARC objects

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
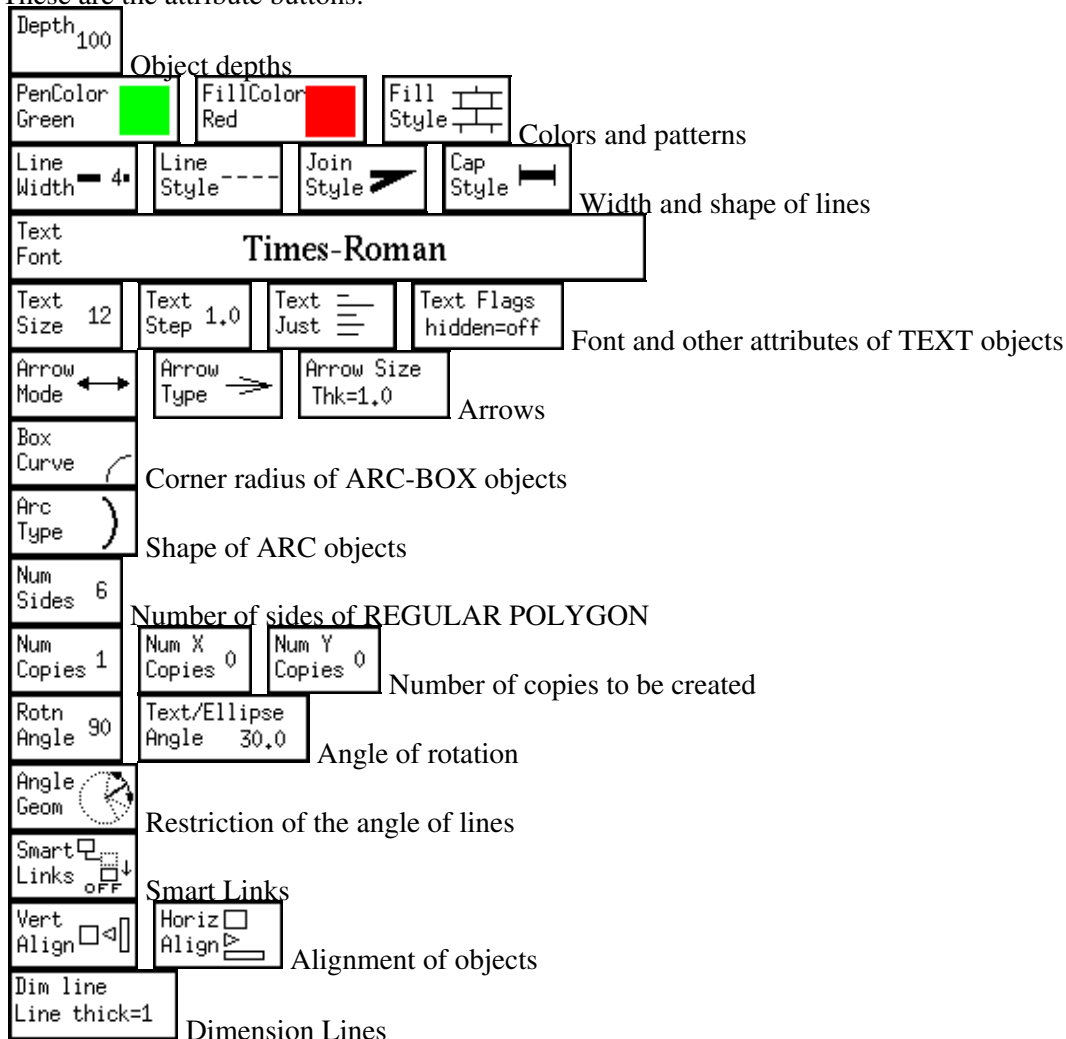
Normally, only the attribute buttons which are significant to the selected mode are placed on the attribute

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.
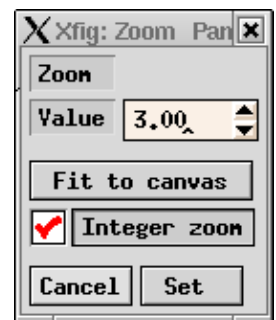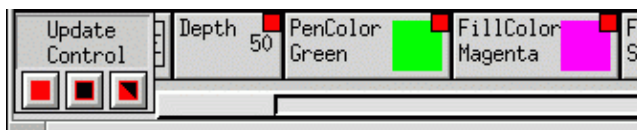
 (ZOOM SCALE)

> 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 you use a wheel mouse, 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.
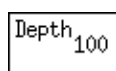
(DEPTH)
Set the `*depth*' of objects.

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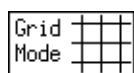
(GRID MODE)
Set the size of the grid displayed on the canvas. If this is set to *None*, no grid will be displayed.

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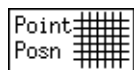
(POINT POSITION)
Set the size of virtual grid (it is not visible) to control the restriction of positioning of objects. If this is not set to *Any*, objects may only be placed on the virtual grid of the selected size. If this is set to *Any*, objects can be placed anywhere.

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.
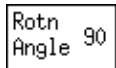
`Rotn Angle 90` (ROTATION ANGLE)

Set the angle for rotating objects with <u>ROTATE</u>. 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 <u>TEXT</u> or <u>ELLIPSE</u> objects may also be set by <u>TEXT / ELLIPSE ANGLE</u>.

Note that some objects may only be rotated by 90 degrees (e.g. <u>BOX</u>)
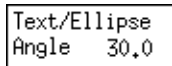
`Text/Ellipse Angle 30.0` (TEXT / ELLIPSE ANGLE)

Set the angle of <u>TEXT</u> or <u>ELLIPSE</u> 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 <u>ROTATE</u>. See also <u>ROTATION ANGLE</u>.
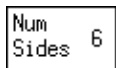
`Num Sides 6` (NUMBER OF POLYGON SIDES)

Set the number of sides used in creating a regular polygon with <u>REGULAR POLYGON</u>.
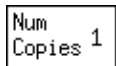
`Num Copies 1` (NUMBER OF COPIES)

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

The number of copies to be made with *ARRAY PLACEMENT* should be set with <u>NUMBER OF X COPIES</u> and <u>NUMBER OF Y COPIES</u>.
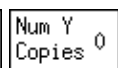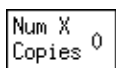
`Num X Copies 0` `Num Y Copies 0` (NUMBER OF X COPIES / NUMBER OF Y COPIES)

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

The number of copies to be made with *COPY & ROTATE* should be set with <u>NUMBER OF COPIES</u>.
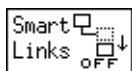
`Smart Links OFF` (SMART-LINKS MODE)

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 <u>COMPOUND object</u> (henceforth called *links*) will be treated specially when those objects are <u>moved</u> or <u>copied</u>.

`OFF` OFF

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

`MOVE` 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` 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.
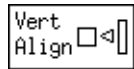
Vert Align (VERTICAL ALIGN)

Set the vertical alignment mode for ALIGN.

NONE 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.
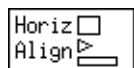
Horiz Align (HORIZONTAL ALIGN)

Set the horizontal alignment mode for ALIGN

NONE 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.
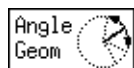
Angle Geom (ANGLE GEOMETRY)

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 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 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.

(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.
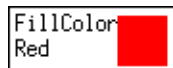
(FILL COLOR)

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*.
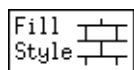
(FILL STYLE)

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:
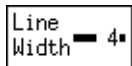
Line
Width ▬ 4▪ (LINE WIDTH)

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.
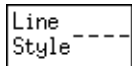
Line
Style ____ (LINE STYLE)

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:
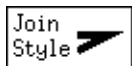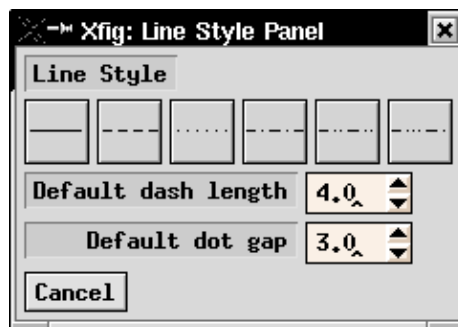
Join
Style ➤ (JOIN STYLE)

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.
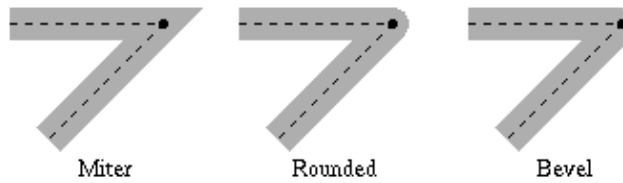
Miter   Rounded   Bevel

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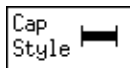
(CAP STYLE)

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.
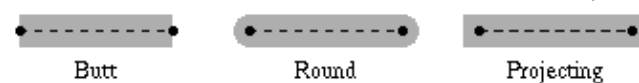
Butt   Round   Projecting

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.
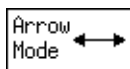
(ARROW MODE)

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.
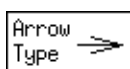
(ARROW TYPE)

Sets the shape of arrow heads.
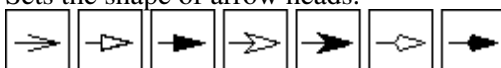
The size of arrow head may be set using ARROW SIZE.
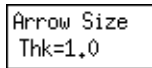
Arrow Size
Thk=1.0 (ARROW SIZE)
Set the size of arrow head.

Clicking this button with mouse button 1 will popup this panel:
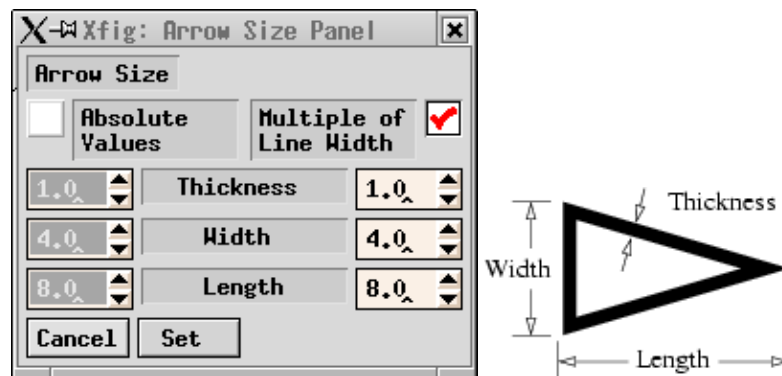
The *Thickness* controls the thickness of the line forming the arrow head.
The *Width* controls the width at the back of the arrow head.
The *Length* controls the length of the arrow head.

All these units are the same as for <u>LINE WIDTH</u>. This is a change from previous versions of **xfig**, where the width and length 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 <u>line width</u> setting to arrive at the sizes.
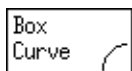
Box Curve (BOX CURVE)
Set the corner radius of <u>ARC-BOX</u> objects. The units are the same as for <u>LINE WIDTH</u>.
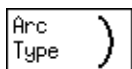
Arc Type (ARC TYPE)
Set the style of <u>ARC</u> object.

OPEN ARC
Draw only the arc.

PIE-WEDGE
Adding to the arc, draw lines which connect each end of the arc to the center.
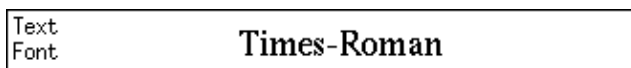
Text Font Times-Roman (TEXT FONT)
Set the font used to draw <u>TEXT</u> objects. The size of the font may be set by <u>TEXT SIZE</u>.

Clicking this button with mouse button 1 will popup the <u>Font Panel</u> to select the font.
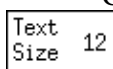
Text Size 12 (TEXT SIZE)
Set the size (point size) of <u>TEXT</u> object. The default is 12 points. The font itself is set by <u>TEXT FONT</u>.
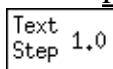
Text Step 1.0 (TEXT STEP)

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.
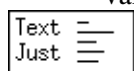
(TEXT JUSTIFICATION)

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.
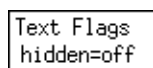
(TEXT FLAGS)

Set the *Hidden*, *Rigid* and *TeX* 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:

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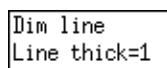
Dimension Lines

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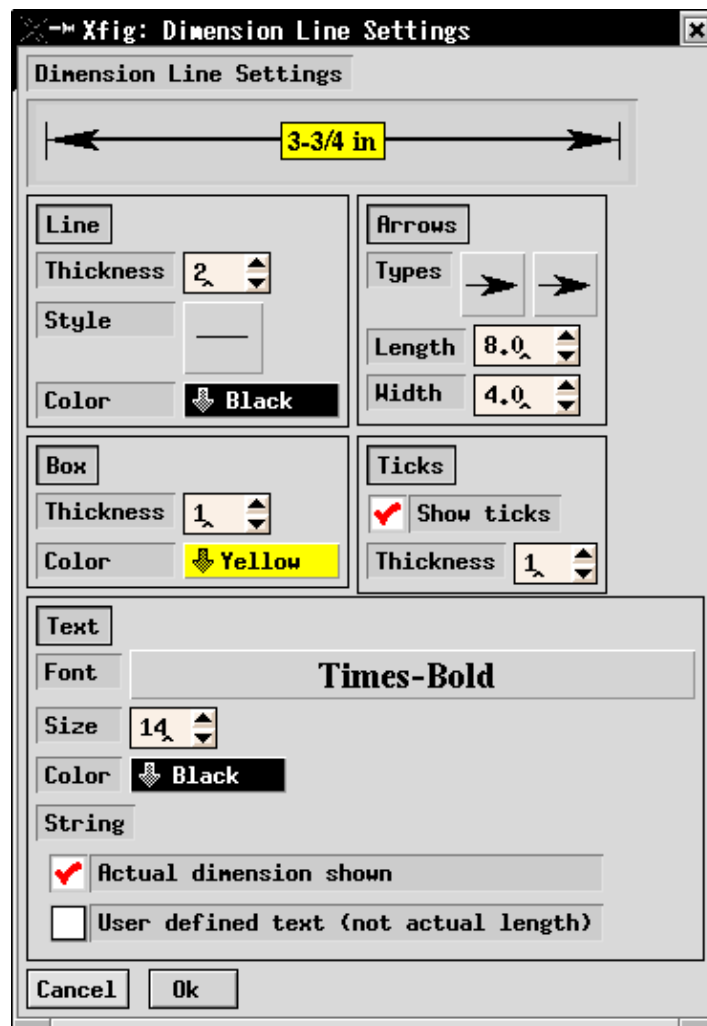
As you make changes to the settings, the image at the top of the panel shows what the dimension line would look like.

- ● *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.
- *Ticks*
- *Show ticks*
    Checkbox to show or hide end ticks on main line.
- *Thickness*
    The thickness of the ticks
- *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 <u>PEN COLOR</u> button or <u>FILL COLOR</u> button is clicked.
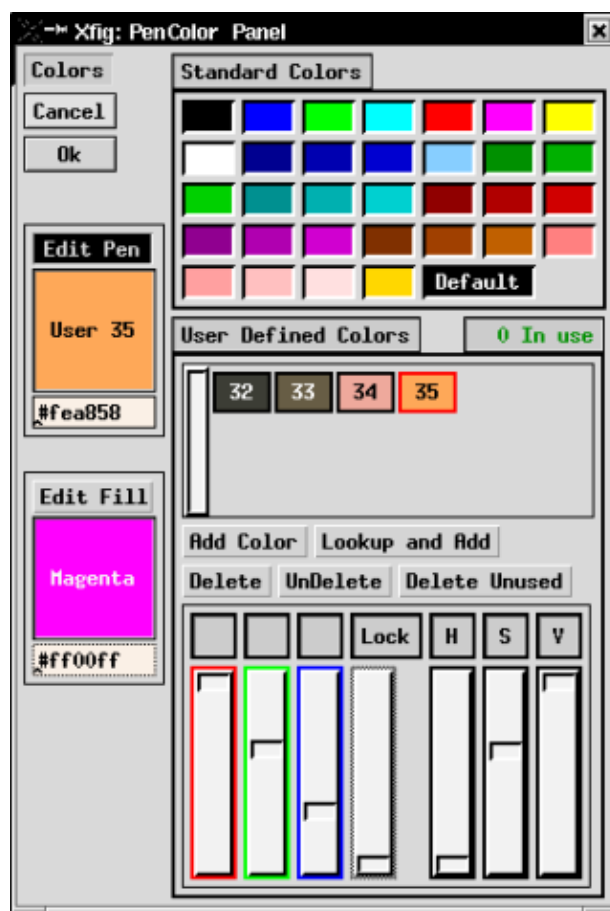
- *Edit Pen*
    If this button is selected, the PEN COLOR (color to draw outline of object) may be selected. If this panel is popped up by clicking <u>PEN COLOR</u> button, this button is selected initially.
- *Edit Fill*

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 <u>FILL COLOR</u> 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 <u>TEXT FONT</u> button is clicked. Only the font name may be selected here; the size of the font must be set using <u>TEXT SIZE</u>.

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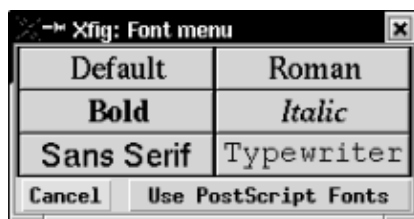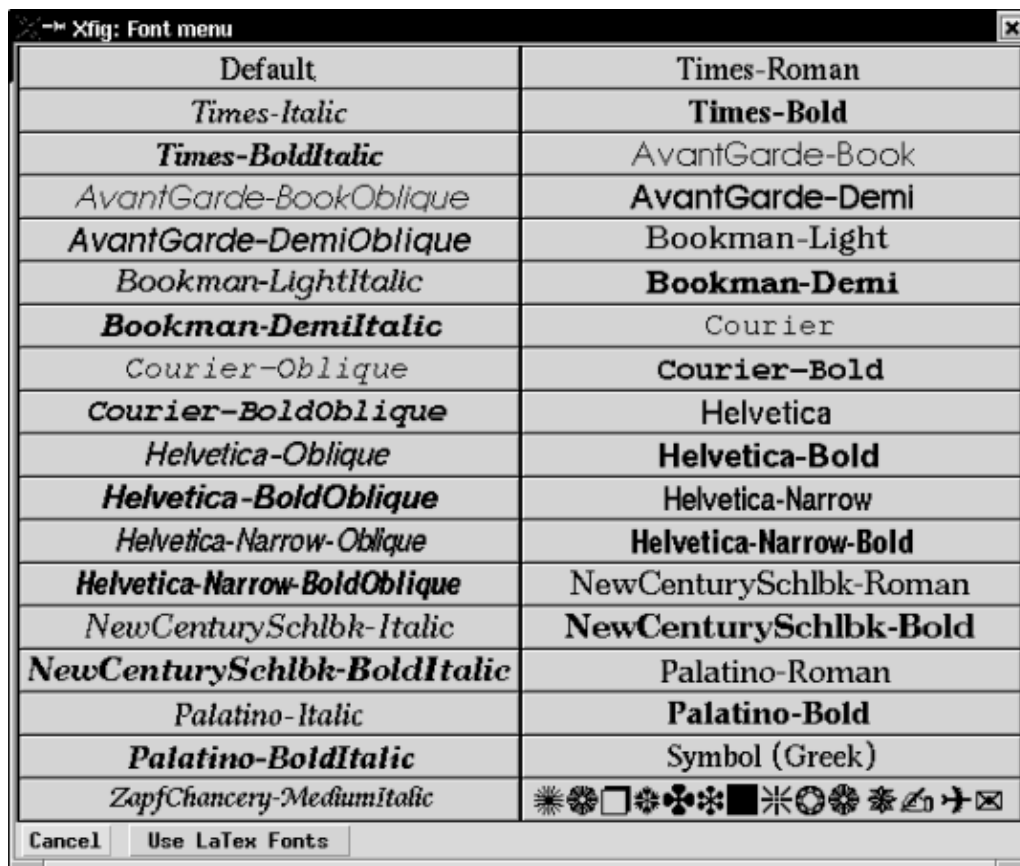
Clicking a font button (the labels of the buttons are drawn using the font) will select that font. Clicking the *Use LaTeX Fonts* button will switch to the panel for LaTeX fonts, and clicking *Use PostScript Fonts* button will switch to the panel for PostScript fonts.

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.

In international-**xfig**, "*Times-Roman*" and "*Times-Bold*" will be replaced with

in the Japanese environment, and

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]

Panning, Zoom, Ruler and Units

Panning the Canvas

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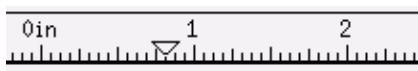
The rulers are placed at the top and right (left, if the drawing mode panel is *right*) of the canvas, and the scale selected by Units is displayed on them (i.e. inches or cm). Normally, triangular markers which show the mouse cursor position are also displayed on the rulers. The rulers may also be used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

If you use 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.
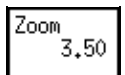
Zoom to Fit Canvas

- 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.

  If you use 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`.

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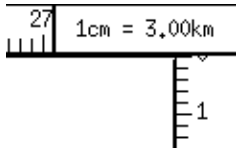
The Units box is placed where the top and side rulers meet. The units of the ruler (*in* or *cm*) and the scale of the drawing (e.g. 1cm=3km) will be displayed here.

Set Units

Clicking on the Units box with mouse button 3 or pressing `Shift-U` (`Set Units/Scale') will pop up a panel to set the units. This panel is also available from the <u>Edit Menu</u>
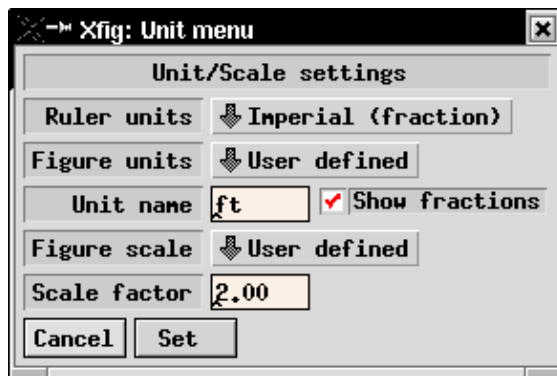
- *Ruler units*
  Select <u>ruler</u> unit from *Imperial (fraction)*, *Imperial (decimal)* or *Metric (cm)*. When *Imperial (fraction)* is chosen, then fractions of the units will be shown in measurement messages, e.g. **3-5/8 in**, or **3 ft 7-1/2 in** For the other choices, decimal values are shown, 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*
  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.

[Contents | Introduction | Credits]

Xfig User Manual
Version 3.2.8b−dev   **Aug 2021**

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*.

When exporting or printing your figure, you can tell xfig to export or print the whole figure or just the active layers.

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

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

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

[Contents | Introduction | Credits]
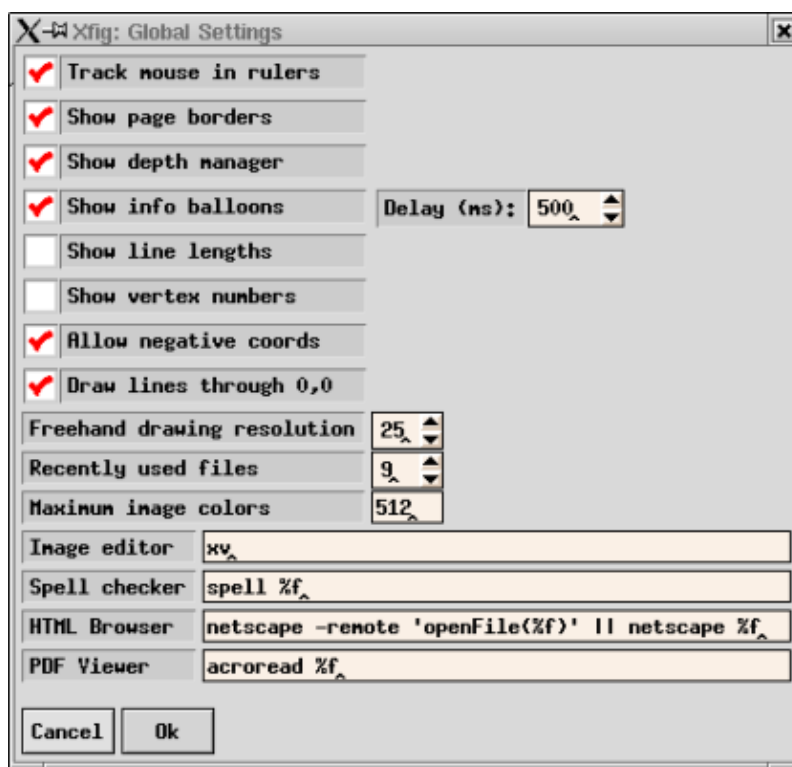
Global Settings

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 <u>-balloons delay</u> option or <u>balloon delay</u> resource.

- Show lengths on lines

    If this is checked, **xfig** will show the lengths of lines as they are being drawn, thus:
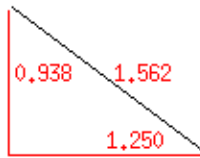
    This option may also be turned on or off with the <u>-showlength/-dontshowlength</u> command-line option or <u>showlength</u> resource.

- Show vertex numbers

    If this is checked, **xfig** will show the number of each vertex of objects on the canvas thus:
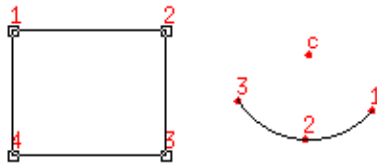
    For arcs, it also shows the center point. This is probably only useful for debugging of Fig files. This option may also be turned on or off with the <u>-shownums/-dontshownums</u> command-line option or <u>shownums</u> resource.

- 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 <u>-allownegcoords / -dontallownegcoords</u> option or <u>allownegcoords</u> resouce.

- Draw axis lines

    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 <u>-showaxislines / -dontshowaxislines</u> option or <u>showaxislines</u> 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 <u>imported images</u>.
    The default is **64** and may also be set with the <u>-max image colors</u> command-line option or <u>max image colors</u> resource.

- Image editor

    This is the external program used when the *<u>Edit Image</u> button* is pressed in the import picture editor. The default is "xdg-open", which opens the preferred image viewer/editor of your desktop. 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 <u>-image editor</u> option or <u>image editor</u> 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 <u>Edit menu</u>, or Meta-K accelerator).
    The default is "spell %f", but it can be changed with <u>-spellcheckcommand</u> option or <u>spellcheckcommand</u> 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 `"xdg-open %f"`, it says to use the default html-browser of your desktop
>   environment. The **%f** is necessary to tell **xfig** to insert the filename into the browser command. This
>   can be set by <u>browser</u> resource.

● PDF Viewer

>   This is the external PDF (Portable Document Format) viewer that **xfig** will The default is
>   `"xdg-open %f"`, which will invoke the default pdf-viewer of your desktop environment. The **%f** is
>   necessary to tell **xfig** to insert the filename into the command. This can be set by <u>pdfviewer</u> resource.

[<u>Contents</u> | <u>Introduction</u> | <u>Credits</u>]

Xfig User Manual
Version 3.2.8b–dev   Aug 2021

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*
  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.

However, be warned that some window managers may use the Meta-Right button combination to do window operations (e.g. resize). If this is the case you will have to change the setting in the window manager.

Wheel Mouse

If you use 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:

Keyboard Coordinate Entry

In any drawing or edit mode, the location of the next point may be specified from the keyboard rather from the pointer. Pressing Ctrl-k while the pointer is in the canvas will pop up a dialog that allows a coordinate to be entered.

The coordinates may be specified in either rectangular or polar form, either absolutely or realative to the current point, and in either decimal or fractional notation. Rectangular coordinates are specified as

x-coord, *y-coord*

or

r *x-coord*, *y-coord*

where the comma is optional and a leading 'r' (or 'R') indicates that the point specified is relative to the current point. Coordinates may be entered either in decimal form or fraction form like, for example "2-7/8" and are in the currently selected units.

Polar coordinates are entered as:

distance < *angle*

where the *distance* is always relative to the current point and, like rectangular coordinates, may be expresed as a decimal or a fraction.

The *angle* may be suffixed with any of the three letters "d", "r", or "p", in either upper or lower case. A "d" suffix indicates that the angle is in degrees, an "r" suffix indicates radians, and a "p" suffix indicates pi-radians. (Pi-radians are a multipler of ; ".5p" would represent the angle /2.) The default if no suffix is specified is degrees.

Keyboard entry is completed by pressing the Enter key. An unshifted Enter is equivalent to a left-button pointer click at the entered location; Shift-Enter is the equivalent of a middle-button (Button2) click and Ctrl-Enter is the equivalent of a right-button (Button3) click.

Keyboard entry may be cancelled by pressing Escape.

The keyboard entry mechanism maintains a stack of recently entered coordinates. While the text cursor is in the keyboard entry text field, the Ctrl-P, Ctrl-N, Up-arrow, and Down-arrow keys navigate up and down in the stack.

[Contents | Introduction | Credits]

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.

```
/*
 * FIG : Facility for Interactive Generation of figures
 * Copyright (c) 1985 by Supoj Sutanthavibul
 * Parts Copyright (c) 1989-2002 by Brian V. Smith
 * Parts Copyright (c) 1991 by Paul King
 * Parts Copyright (c) 1995 by C. Blanc and C. Schlick
 *
 * 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.
 */

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.

Note about metric units: To preserve a regular grid on the canvas the
centimeter is defined to be 450 Fig units and not 472.4 (1200/2.54).
For drawings done in metric units, fig2dev magnifies the output when
exporting or printing to compensate for the difference (472.4/450).
Also, if you make a drawing in one unit scale and switch to the other
units in xfig, the drawing will be rescaled on the screen by xfig to
remain consistent.

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, solid, dash, dotted, etc.)
        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, specification for dash/dotted lines)
        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, solid, dash, dotted, etc.)
    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, specification for dash/dotted lines)
    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, solid, dash, dotted, etc.)
    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, specification for dash/dotted lines)
    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               (enumeration type, solid, dash, dotted, etc.)
          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, specification for dash/dotted lines)
          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. For positive values of this
factor, the spline is approximated at this point, for negative values the
spline is interpolated 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., an angular point).
Recommended values for this factor are 1.0 for an approximated spline, 0.0
for an angular point and -0.5 for interpolated splines.
The shape factor used here corresponds to the parameter s_k defined in
section 4.1 of Blanc & Schlick (1995) for values greater than 0, for values
smaller than 0 it corresponds to the negative of the parameter q defined in
section 5.1 of Blanc & Schlick (1995).


```
==============================================================================
(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]

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.3.2.5d.tar.gz** and *gunzip* and *untar* it.
2. Uncomment "`#define I18N`" in `xfig.3.2.5d/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.3.2.5e.tar.gz** and *gunzip* and *untar* it.
2. Uncomment "`#define I18N`" in `transfig.3.2.5e/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.

   N.B. If you run **fig2dev** standalone you must use the `-j` option to enable internationalization.

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 <u>TEXT FONT</u>. If any other font is selected, <u>Latin-1 characters</u> will be available as in normal **xfig**.
- It is not possible to edit international text in the <u>Edit Panel</u>. But it is possible to edit text on the canvas in <u>TEXT</u> 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

Xfig User Manual
Version 3.2.8b−dev   *Aug 2021*

LaTeX and Xfig

- How to Import Xfig Figures in Your LaTeX Files
- Changing the Size of Pictures
- Xfig and Metapost by Josselin Mouette (jmouette@ens-lyon.fr)

How to Import Xfig Figures in Your LaTeX Files

Xfig can generate output to many different formats which TeX or LaTeX can read. The final document is produced by one of the following methods, depending on the export language.

- **LaTeX picture, LaTeX with pict2e macros, TeX/LaTeX with tikz macros, LaTeX with pstricks macros, LaTeX with epic/eepic/eepicemu macros:**

    The file generated from xfig is directly included, e.g., `\input{fig1.tikz}`, or the entire file contents is copied into your TeX/LaTeX file. Therefore, the TeX/LaTeX document can contain all text and graphics. Usually, it will be necessary to load a macro package to enable TeX or LaTeX to process the file. Text that should be interpreted by TeX/LaTeX must have the *TeX Flag* set to *ON*, see TEXT FLAGS. Otherwise, characters special to TeX will be quoted and, e.g., `$x$` will print as x, not *x*.

    `latex`

    > Include the generated file with, e.g., `\input{fig1.latex}` into your document, or simply embed it. No packages are required, but the capabilities of the `latex` output language are quite limited.

    `tikz`

    > `\input{fig1.tikz}` or embed the contents of the generated tikz file into the TeX-document. In the preamble of a LaTeX document, use
    > ```
    > \usepackage{tikz, graphics}  % graphics, if the figure
    > contains embedded images
    > \usetikzlibrary{arrows.meta, % if the figure contains
    > arrow-tips
    >                 bending,      % arrow tips on arcs are "bent,"
    > i.e., deformed a bit
    >                 patterns      % if the figure contains pattern
    > fills
    >                 }
    > ```
    > The export to TeX/LaTeX with tikz macros supports all capabilites of xfig. LaTeX and Postscript fonts can be freely used and even mixed in the same figure, and the typesetting capabilities of TeX can be used.

    `pict2e`

    > Embed or `\input` the generated file into your LaTeX document. In the preamble include
    > ```
    > \usepackage{pict2e,
    >             color,    % if the figure contains color
    >             graphics  % if the figure contains embedded images
    >             }
    > ```
    > Export to LaTeX with pict2e macros does not support pattern fills. All other features of a xfig drawing are reproduced. LaTeX and Postscript fonts can be used for text in the xfig drawing.

```
pstricks
        \usepackage{pstricks,
                    pstricks-add, % for complex line styles and hollow
        PSTricks arrows (-R 1 option)
                    graphicx      % if the figure contains embedded
        images
                    }
```
The pstricks output matches the quality of the Postscript driver. Text is rendered black, but LaTeX font color-changing code can be embedded in the text.

```
epic
        \usepackage{epic}
eepic
        \usepackage{epic,eepic} % order matters!
```

- **EPS, PDF, EPS and PDF:**

    Export the figure from xfig to eps or pdf (or both) and import the file with `\includegraphics` into your document. Write

    `\usepackage{graphics}`

    into the preamble and include the drawing with, e.g., `\includegraphics{fig1.eps}` or `\includegraphics{fig1}`. This method supports all capabilities of xfig, but only Postscript fonts are used to render text. To make use of the fonts and of the typesetting possibilities of LaTeX, one could use the `psfrag` package to replace text strings in an eps file with code rendered by LaTeX.

- **Combined PS/LaTeX, Combined PDF/LaTeX, Combinded PS/PDF/LaTeX:**

    Export the figure to two files or even three files, partially to eps or pdf or to both, and partially to a LaTeX file, and include the LaTeX file in your document. The LaTeX file contains the text and includes either the eps or pdf file which contains the graphics. Text marked with the *TeX Flag* set to *ON* is rendered by LaTeX and superimposed over the included graphics. In the preamble, put

    `\usepackage{graphics,color} % color, if the text is colored`

    Write `\input{fig1.pstex_t}` for latex + eps, `\input{fig1.pdf_t}` for latex + pdf, or `\input{fig1.pspdftex}` for latex + eps or pdf at the location where the drawing should appear. The export to Combined PS/PDF/LaTeX, latex + eps or pdf, can be processed by any LaTeX engine.

A variant of the second method above is to

- **create a stand-alone tex file and import the generated EPS or PDF:**

    The tex-file cannot be exported directly from xfig, but you must invoke `fig2dev` from the command line with the `-P` option of the tikz or pict2e driver. Save your drawing, and invoke `fig2dev` on the saved fig-file, e.g., for the `tikz` driver,

    ```
    $ fig2dev -L tikz fig1.fig fig1.tex
    $ pdflatex fig1.tex
    ```

    Include the final `fig1.pdf` with `\includegraphics{fig1}` into your LaTeX document. This tex-file is processed by LaTeX to create a pdf or eps graphics that can be included into your LaTeX document. The advantage of this method above the direct export to pdf or eps is that text can be rendered with LaTeX- and Postscript fonts, and TeX/LaTeX expressions are interpreted in text on which the *TeX flag* is set to *On*.

To set the TEXT FLAG *TeX Flag* to *ON* and use LaTeX-fonts for all text, invoke xfig with the commandline

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

To make the above permanently and start all drawings with LaTeX fonts and the *TeX flag* set to *ON*, put the following in your X-resources file. This is probably the file `.Xresources` in your home directory.

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

Changing the Size of Pictures

If possible, try to not scale drawings at all. Different line thicknesses should be a avoided, similarly to avoiding differently sized letters in text. The rulers at the top and at the side, and the magnification setting in the lower left corner of the xfig window may aid at producing the graphics at the right size.

If scaling cannot be avoided, the commands `\scalebox` and `\resizebox` from the graphics package can be used,

```
\scalebox{1.5}{\input{fig1.pstex_t}}    % \scalebox{factor}{object}
\resizebox{10cm}{!}{\input{fig1.latex}} % \resizebox{width}{height}{object}
```

An exclamation mark in place of one of the dimensions retains the aspect ratio of the object.

To set the width of tikz-drawings produced by xfig, you can use

```
\newlength\XFigwidth\XFigwidth84mm
\newlength\XFigheight\XFigheight56mm
```

before input'ing your figure. This will scale the coordinates of the drawing, but not line widths or text. If only one of the two dimensions is given, the figure retains its aspect ratio. To render subsequent figures in their original size, both dimensions must be undefined or set to zero,

```
\XFigwidth0pt \XFigheight0pt
```

Xfig and Metapost

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

1. 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}{1}{1}
   ```

   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).
2. 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?

[<u>Contents</u> | <u>Introduction</u> | <u>Credits</u>]

Xfig User Manual
Version 3.2.8b–dev   **Aug 2021**

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*
    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.

However, be warned that some window managers may use the Meta-Right button combination to do window operations (e.g. resize). If this is the case you will have to change the setting in the window manager.

Wheel Mouse

If you use 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:

Keyboard Coordinate Entry

In any drawing or edit mode, the location of the next point may be specified from the keyboard rather from the pointer. Pressing Ctrl-k while the pointer is in the canvas will pop up a dialog that allows a coordinate to be entered.

The coordinates may be specified in either rectangular or polar form, either absolutely or realative to the current point, and in either decimal or fractional notation. Rectangular coordinates are specified as

x-coord, *y-coord*

or

r *x-coord*, *y-coord*

where the comma is optional and a leading 'r' (or 'R') indicates that the point specified is relative to the current point. Coordinates may be entered either in decimal form or fraction form like, for example "2-7/8" and are in the currently selected units.

Polar coordinates are entered as:

distance < *angle*

where the *distance* is always relative to the current point and, like rectangular coordinates, may be expresed as a decimal or a fraction.

The *angle* may be suffixed with any of the three letters "d", "r", or "p", in either upper or lower case. A "d" suffix indicates that the angle is in degrees, an "r" suffix indicates radians, and a "p" suffix indicates pi-radians. (Pi-radians are a multipler of ; ".5p" would represent the angle /2.) The default if no suffix is specified is degrees.

Keyboard entry is completed by pressing the Enter key. An unshifted Enter is equivalent to a left-button pointer click at the entered location; Shift-Enter is the equivalent of a middle-button (Button2) click and Ctrl-Enter is the equivalent of a right-button (Button3) click.

Keyboard entry may be cancelled by pressing Escape.

The keyboard entry mechanism maintains a stack of recently entered coordinates. While the text cursor is in the keyboard entry text field, the Ctrl-P, Ctrl-N, Up-arrow, and Down-arrow keys navigate up and down in the stack.

[Contents | Introduction | Credits]

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)

```
Undo             (Meta-U)
Paste Objects    (Meta-T)
Paste Text       (F18/F20)
Search/Replace... (Meta-I)
Spell Check...   (Meta-K)
Delete All       (Meta-D)
Global settings... (Meta-G)
Set units...
```

View Menu Accelerators:

View **(Meta-V)**

```
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
```

Help Menu Accelerators:

Help **(Meta-H)**

```
Xfig Reference (HTML)...
Xfig Reference (PDF)...
Xfig Man Pages (HTML)...
How-To Guide (PDF)...
About Xfig...
```

Drawing and Editing Mode Accelerators:

[Contents | Introduction | Credits]

Xfig User Manual
Version 3.2.8b–dev   **Aug 2021**

Command Line Options and Resources

Command-line Options

- Mode Settings
- PRINT and EXPORT
- Fonts or Colors, etc.
- Size of Windows, etc.
- Other Settings

X Resources

- Mode Resources
- PRINT and EXPORT
- Fonts or Colors, etc.
- Size of Windows, etc.
- Other Resources
- Resources for Internationalization i18n

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.

–autorefresh
> Make xfig look at the timestamp on the .fig file and automatically load it and display it every time it changes.

–dontshowaxislines (**–dontshowa**)
> Don't draw the axis 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 <u>TEXT FLAGS</u> 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 <u>Set Units</u>.

–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 <u>View/Portrait/Landscape</u> 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 <u>Set Units</u>.

–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 <u>TEXT FLAGS</u> 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 <u>-update</u> option to do a batch scaling of figures.

–showaxislines (**–showa**)
> Draw the axis lines crossing through 0,0 on the canvas. See also <u>axislines</u> option for the color of the zero lines.

–smallicons (**–sma**)
> Use small button icons on mode panel.

–specialtext (**–spec**)
> Set the *TeX* flag of <u>TEXT FLAGS</u> to *ON*. The default is *off*.

–startfontsize *pointsize* (**–startfo**)
> Set the initial value of <u>TEXT SIZE</u>. The default is 12pt.

–startpsFont *font* (**–startps**)
> Select the initial PostScript font as <u>TEXT FONT</u>. The default is *Times-Roman*.

–startlatexFont *font* (**–startla**)
> Select the initial LaTeX font as <u>TEXT FONT</u>. This is effective only if −latexfonts option is given. The default is *Default*.

–starttextstep *stepsize* (**–startt**)
> Set the initial value of <u>TEXT STEP</u>. The default is 1.2.

–startfillstyle *stylenumber* (**–startfi**)
> Set the initial setting (-1 to 21) of <u>FILL STYLE</u>. The default is -1 (*No fill*).

–startgridmode *modenumber* (**–startg**)
> Set the initial <u>GRID MODE</u>. 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 <u>POINT POSITION</u>. 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 <u>LINE WIDTH</u>. 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 <u>Set Units</u>.

–userunit *units* (**–useru**)

> Set the unit name used when displaying dimensions. See also <u>Set Units</u>.

–axislines *color* (**–axisl**)

> Draw the axis lines crossing through 0,0 on the canvas in this color. The **–showaxislines** option must be on. 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 <u>ZOOM SCALE</u>. The default is 1.0.

PRINT and EXPORT Settings

–center (**–cente**)

> Select *Center* as *Justification* when <u>Exporting</u> or <u>Printing</u>. This the the default.

–exportLanguage *language* (**–exportL**)

> Set the initial export language when <u>Exporting</u>. *Language* may be one of the following:

```
Vector formats:
    box        LaTeX box (figure boundary)
    shape      ShapePar definition
    latex      LaTeX picture
    epic       LaTeX picture + epic macros
    eepic      LaTeX picture + eepic macros
    eepicemu   LaTeX picture + eepicemu macros
    tikz       LaTeX picture + tikz macros
    pict2e     LaTeX picture + pict2e macros
    pstricks   LaTeX + pstricks macros
    pictex     PiCTeX macros
    ibmgl      IBMGL (or HPGL)
    eps        Encapsulated PostScript
    ps         PostScript
    pspdf      EPS and PDF (two files)
    pdf        Portable Document Format
    pstex      Combined PS/LaTeX (both PS and LaTeX parts)
    pdftex     Combined PDF/LaTeX (both PDF and LaTeX parts)
    pspdftex   Combined PS/PDF/LaTeX (three files)
    textyl     Textyl \special commands
    tpic       TPIC
    pic        PIC
    html       HTML Image map
    mf         MF (MetaFont)
    mp         MP (MetaPost)
    cgm        CGM (Computer Graphics Metafile)
    emf        EMF (Enhanced Metafile Format)
```

```
         dxf         DXF (Drawing Interchange Format)
         gbx         GBX (Gerber, RS-247-X)
         tk          Tk (aka Tcl/Tk)
         ptk         Perl/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 <u>Exporting</u> or <u>Printing</u>. The default is *Centered*.

–jpeg_quality *quality* (**–j**)

Set the quality used when <u>Exporting</u> to the JPEG (JFIF) image format.

–magnification *mag* (**–mag**)

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

–multiple (**–mu**)

Set *Pages* to *Multiple* when <u>Exporting</u> or <u>Printing</u>. The default is *Single*.

–paper_size *size* (**–pap**)

Set the paper size when <u>Exporting</u> or <u>Printing</u>. *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

Paper	Size	Option name
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.

–correct_font_size

> Normally, *fig2dev* uses 1/80 inch for font size increments (for historical reasons), instead of the more proper ``points'' (1/72 inch) that *xfig* uses. This option makes xfig call fig2dev with the `-F` option to make it use points (1/72 inch).

–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 [*WxH*]**+***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 light red).

–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 <u>Edit Panel</u> for PICTURE object is clicked.

–inputStyle *style* i18n

Set the input style when <u>entering international text</u> 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 <u>internationalization facility</u>. 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 <u>available libraries</u>.

–library_icon_size *size* (**–library_i**)

Set the size of the library icons to *size* pixels square. See also <u>-icon_view</u>.

–list_view (**–list_**)

Show the list view of the <u>library objects</u>. The default is to show the icon view. See also <u>-icon_view</u>.

–nosplash

Don't show the startup screen on startup. The default is to show the splash screen

–nowrite_bak

–write_bak

When saving a drawing into an existing .fig file **xfig** will first rename that file by appending ".bak" to the name. These options turn on and off that feature.

–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 <u>depth manager</u> This will allow more room for the canvas. The default is to *show* the depth manager.
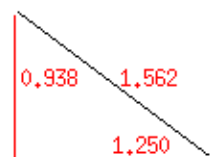
–showlengths (**–showl**)

–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.
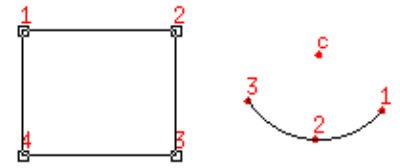
–shownums (**–shown**)
–dontshownums (**–dontshown**)
>   Makes **xfig** show the number of each vertex of objects on the canvas. For arcs, it also shows the center point. This is probably only useful for debugging of Fig files.

–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.
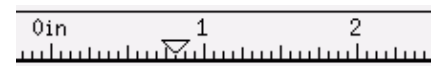
–track (**–trac**)
–notrack (**–not**)
>   Turn on or off cursor (mouse) tracking arrows. The default is to track the cursor position by displaying arrows on the top and side ruler.

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.

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

autorefresh

Make xfig look at the timestamp on the .fig file and automatically load it and display it every time it changes.

Option: −autorefresh

Default: off

hiddentext (**HiddenText**)

Set initial value of the *Hidden* flag of the <u>TEXT FLAGS</u>.

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 <u>TEXT FLAGS</u>.

Option: −rigidtext

Default: off

specialtext (**SpecialText**)

Set initial value of the *TeX* flag of the <u>TEXT FLAGS</u>.

Option: −specialtext

Default: off

startpsFont (**StartpsFont**)

Select the specified PostScript font as the initial <u>TEXT FONT</u>.

Option: −startpsFont

Default: Times−Roman

startlatexFont (**StartlatexFont**)

Select the specified LaTeX font as the initial <u>TEXT FONT</u>.

Option: −startlatexFont

Default: Default

startfontsize (**StartFontSize**)

Set initial value of <u>TEXT SIZE</u>.

Option: −startfontsize

Default: 12

starttextstep (**StartTextStep**)

Set initial value of <u>TEXT STEP</u>.

Option: −starttextstep

Default: 1.2

startfillstyle (**StartFillStyle**)

Set the initial value (-1 to 21) of <u>FILL STYLE</u>.

Option: −startfillstyle

Default: -1 (None)

startgridmode (**StartGridMode**)

Set the initial value (0 to 4) of <u>GRID MODE</u>.

Option: −startgridmode

Default: 0 (None)

startposnmode (**StartPosnMode**)

      Set the initial value (0 to 5) of <u>POINT POSITION</u>.

      Option: −startposnmode

      Default: 1 (1/16inch, or 1mm)

startlinewidth (**StartLineWidth**)

      Set the initial value of <u>LINE WIDTH</u>.

      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 <u>ZOOM SCALE</u>.

      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 <u>Exporting</u>.

      Option: −exportLanguage

      Default: eps (Encapsulated PostScript)

flushleft (**FlushLeft**)

      Set the initial *Justification* when <u>Exporting</u> or <u>Printing</u>.

      Option: −flushleft

      Default: off (Centered)

jpeg_quality *quality* (**Quality**)

      Set the quality used when <u>Exporting</u> to the JPEG (JFIF) image format.

      Option: jpeg_quality

      Default: 75

magnification (**Magnification**)

      Set the initial *Magnification* when <u>Exporting</u> or <u>Printing</u>.

      Option: −magnification

      Default: 100

multiple (**Multiple**)

      Set the initial selection of *Pages* when <u>Exporting</u> or <u>Printing</u>.

      Option: −multiple

      Default: off

paper_size (**Papersize**)

      Set the initial *Paper Size* when <u>Exporting</u> or <u>Printing</u>. See <u>paper sizes</u>.

      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

smallicons (**Icons**)
>   Show the small icons in the mode panel.
>   **Option:** -smallicons
>   **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 light red).
> **Option:** `-grid_color`
> **Default:** #FFCCCC

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:** xdg-open

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:** xdg-open %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**)

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 <u>Browse Panel</u>. 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 <u>File Panel</u>. 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 <u>Print Panel</u>. 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 <u>Print Panel</u>. 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 <u>internationalization facility</u> (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 <u>TEXT FONT</u>. `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 <u>entering international text</u> 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 <u>TEXT FONT</u>. 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 <u>Print</u>. 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 <u>international-**fig2dev**</u> (`japanese.ps`, etc.) are placed. If this is not set, the setting when compiling **fig2dev** will be used.

[<u>Contents</u> | <u>Introduction</u> | <u>Credits</u>]

Xfig User Manual
Version 3.2.8b−dev   **Aug 2021**

Printing and Exporting

- Printing
- Exporting
- 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, PDF, EPS, LaTeX, MetaFont, and bitmap formats such as GIF, JPEG, PPM and several others. See the *Language* description of the Exporting section and the hints on LaTeX and Xfig.
Hence, to export or print, you must have **fig2dev** installed on your system along with **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).

🔴 *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*

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.

🔴 *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 <u>*File*</u> 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 <u>LaTeX and Xfig</u> 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 <u>File/Print</u> 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

[Home]
[Show Hidden]

```
..                @bitmaps        @contributions
@flag_bitmaps     @fonts          @gtk_tests
@notes            @oldpatches     @savegifs
@shar             @v4.0-alpha     @www_more
```

[Rescan]   [Cancel]   [Export]

🔴 *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 (EPS)
🔵 PDF (Portable Document Format)
🔵 EPS and PDF (two files)
🔵 LaTeX box (figure boundary)
🔵 LaTeX picture environment
🔵 TeX/LaTeX with pict2e macros
🔵 TeX/LaTeX with tikz macros
🔵 LaTeX with pstricks macros
🔵 LaTeX with epic / eepic / eepicemu macros
🔵 Combined PostScript/LaTeX
🔵 Combined PDF/LaTeX
🔵 Combined PostScript/PDF/LaTeX (three files)
🔵 PicTeX macros
🔵 IBMGL (HP-GL)
🔵 DXF (Drawing Interchange Format)
🔵 GBX (Gerber, RS-247-X)

- Textyl \special commands
- TPIC
- PIC
- HTML Image map
- MF (MetaFont)
- MP (MetaPost)
- CGM (Computer Graphics Metafile - useful to import into Microsoft WORD, etc.)
- EMF (Enhanced Metafile)
- SVG (Scalable Vector Graphics)
- Tk (Tck/Tk toolkit canvas)
- Perl/Tk
- Shape (LaTeX \shapepar definition)
- SVG (Scalable Vector Graphics)

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

For combination with LaTeX, it is possible to either describe the entire figure with TeX/LaTeX macros, or to output the text part to LaTeX and the graphics part to PostScript and/or PDF using *Combined PostScript/LaTeX*, *Combined PDF/LaTeX* or *Combined PostScript/PDF/LaTeX*. See also TEX FLAGS and LaTeX and Xfig.

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, PDF, Tex/LaTeX with tikz macros and SVG export languages support all features of **xfig** and a fairly high quality output will be generated.

The **fig2dev** program, part of the **Fig2dev** 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 ghostscript installed 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*
  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 available for PostScript, Encapsulated PostScript, PDF, PSLatex, PDFLatex, pstricks, tikz 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 <u>Print Panel</u> about available paper sizes. This is effective only if *PostScript* is selected at <u>*Language*</u>.

🔴 *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 <u>*Language*</u>.

🔴 *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 <u>*Portrait/Landscape*</u> in the <u>*View* menu</u>.

🔴 *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 <u>*Language*</u>.

🔴 *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 <u>*Language*</u>.

🔴 *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 <u>Netscape Navigator</u>, 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 <u>language</u> 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 <u>*Language* menu</u>.

🔴 *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 3.2.8b–dev   Aug 2021

LaTeX and Xfig

- How to Import Xfig Figures in Your LaTeX Files
- Changing the Size of Pictures
- Xfig and Metapost by Josselin Mouette (jmouette@ens-lyon.fr)

How to Import Xfig Figures in Your LaTeX Files

Xfig can generate output to many different formats which TeX or LaTeX can read. The final document is produced by one of the following methods, depending on the export language.

- **LaTeX picture, LaTeX with pict2e macros, TeX/LaTeX with tikz macros, LaTeX with pstricks macros, LaTeX with epic/eepic/eepicemu macros:**
    The file generated from xfig is directly included, e.g., `\input{fig1.tikz}`, or the entire file contents is copied into your TeX/LaTeX file. Therefore, the TeX/LaTeX document can contain all text and graphics. Usually, it will be necessary to load a macro package to enable TeX or LaTeX to process the file. Text that should be interpreted by TeX/LaTeX must have the *TeX Flag* set to *ON*, see TEXT FLAGS. Otherwise, characters special to TeX will be quoted and, e.g., `$x$` will print as x, not *x*.
    `latex`
        Include the generated file with, e.g., `\input{fig1.latex}` into your document, or simply embed it. No packages are required, but the capabilities of the `latex` output language are quite limited.
    `tikz`
        `\input{fig1.tikz}` or embed the contents of the generated tikz file into the TeX-document. In the preamble of a LaTeX document, use
        `\usepackage{tikz, graphics}  % graphics, if the figure`
        `contains embedded images`
        `\usetikzlibrary{arrows.meta, % if the figure contains`
        `arrow-tips`
        `                  bending,    % arrow tips on arcs are "bent,"`
        `i.e., deformed a bit`
        `                  patterns    % if the figure contains pattern`
        `fills`
        `                  }`
        The export to TeX/LaTeX with tikz macros supports all capabilites of xfig. LaTeX and Postscript fonts can be freely used and even mixed in the same figure, and the typesetting capabilities of TeX can be used.
    `pict2e`
        Embed or `\input` the generated file into your LaTeX document. In the preamble include
        `\usepackage{pict2e,`
        `             color,     % if the figure contains color`
        `             graphics   % if the figure contains embedded images`
        `             }`
        Export to LaTeX with pict2e macros does not support pattern fills. All other features of a xfig drawing are reproduced. LaTeX and Postscript fonts can be used for text in the xfig drawing.

```
pstricks
        \usepackage{pstricks,
                    pstricks-add, % for complex line styles and hollow
        PSTricks arrows (-R 1 option)
                    graphicx      % if the figure contains embedded
        images
                    }
```
The pstricks output matches the quality of the Postscript driver. Text is rendered black, but LaTeX font color-changing code can be embedded in the text.

```
epic
        \usepackage{epic}
eepic
        \usepackage{epic,eepic} % order matters!
```

- **EPS, PDF, EPS and PDF:**

    Export the figure from xfig to eps or pdf (or both) and import the file with `\includegraphics` into your document. Write

    ```
    \usepackage{graphics}
    ```

    into the preamble and include the drawing with, e.g., `\includegraphics{fig1.eps}` or `\includegraphics{fig1}`. This method supports all capabilities of xfig, but only Postscript fonts are used to render text. To make use of the fonts and of the typesetting possibilities of LaTeX, one could use the `psfrag` package to replace text strings in an eps file with code rendered by LaTeX.

- **Combined PS/LaTeX, Combined PDF/LaTeX, Combinded PS/PDF/LaTeX:**

    Export the figure to two files or even three files, partially to eps or pdf or to both, and partially to a LaTeX file, and include the LaTeX file in your document. The LaTeX file contains the text and includes either the eps or pdf file which contains the graphics. Text marked with the *TeX Flag* set to *ON* is rendered by LaTeX and superimposed over the included graphics. In the preamble, put

    ```
    \usepackage{graphics,color} % color, if the text is colored
    ```

    Write `\input{fig1.pstex_t}` for latex + eps, `\input{fig1.pdf_t}` for latex + pdf, or `\input{fig1.pspdftex}` for latex + eps or pdf at the location where the drawing should appear. The export to Combined PS/PDF/LaTeX, latex + eps or pdf, can be processed by any LaTeX engine.

A variant of the second method above is to

- **create a stand-alone tex file and import the generated EPS or PDF:**

    The tex-file cannot be exported directly from xfig, but you must invoke `fig2dev` from the command line with the `-P` option of the tikz or pict2e driver. Save your drawing, and invoke `fig2dev` on the saved fig-file, e.g., for the `tikz` driver,

    ```
    $ fig2dev -L tikz fig1.fig fig1.tex
    $ pdflatex fig1.tex
    ```

    Include the final `fig1.pdf` with `\includegraphics{fig1}` into your LaTeX document. This tex-file is processed by LaTeX to create a pdf or eps graphics that can be included into your LaTeX document. The advantage of this method above the direct export to pdf or eps is that text can be rendered with LaTeX- and Postscript fonts, and TeX/LaTeX expressions are interpreted in text on which the *TeX flag* is set to *On*.

To set the <u>TEXT FLAG</u> *TeX Flag* to *ON* and use LaTeX-fonts for all text, invoke xfig with the commandline
```
    $ xfig -specialtext -latexfonts -startlatexFont default
```
To make the above permanently and start all drawings with LaTeX fonts and the *TeX flag* set to *ON*, put the following in your X-resources file. This is probably the file `.Xresources` in your home directory.
```
    Fig.latexfonts: true
    Fig.specialtext: true
```

Changing the Size of Pictures

If possible, try to not scale drawings at all. Different line thicknesses should be a avoided, similarly to avoiding differently sized letters in text. The rulers at the top and at the side, and the magnification setting in the lower left corner of the xfig window may aid at producing the graphics at the right size.

If scaling cannot be avoided, the commands `\scalebox` and `\resizebox` from the graphics package can be used,

```
\scalebox{1.5}{\input{fig1.pstex_t}}    % \scalebox{factor}{object}
\resizebox{10cm}{!}{\input{fig1.latex}} % \resizebox{width}{height}{object}
```

An exclamation mark in place of one of the dimensions retains the aspect ratio of the object.

To set the width of tikz-drawings produced by xfig, you can use

```
\newlength\XFigwidth\XFigwidth84mm
\newlength\XFigheight\XFigheight56mm
```

before input'ing your figure. This will scale the coordinates of the drawing, but not line widths or text. If only one of the two dimensions is given, the figure retains its aspect ratio. To render subsequent figures in their original size, both dimensions must be undefined or set to zero,

```
\XFigwidth0pt \XFigheight0pt
```

Xfig and Metapost

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

1. 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}{1}{1}
   ```

   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).
2. 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?

[Contents | Introduction | Credits]

Xfig User Manual
Version 3.2.8b−dev   **Aug 2021**

Printing and Exporting

- Printing
- Exporting
- 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, PDF, EPS, LaTeX, MetaFont, and bitmap formats such as GIF, JPEG, PPM and several others. See the *Language* description of the Exporting section and the hints on LaTeX and Xfig.
Hence, to export or print, you must have **fig2dev** installed on your system along with **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).

- *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*

    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.

- *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 <u>*File* menu</u> (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 <u>LaTeX and Xfig</u> 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 <u>File/Print</u> if you want to print the figure to a PostScript printer.

● *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 (EPS)
- ● PDF (Portable Document Format)
- ● EPS and PDF (two files)
- ● LaTeX box (figure boundary)
- ● LaTeX picture environment
- ● TeX/LaTeX with pict2e macros
- ● TeX/LaTeX with tikz macros
- ● LaTeX with pstricks macros
- ● LaTeX with epic / eepic / eepicemu macros
- ● Combined PostScript/LaTeX
- ● Combined PDF/LaTeX
- ● Combined PostScript/PDF/LaTeX (three files)
- ● PicTeX macros
- ● IBMGL (HP-GL)
- ● DXF (Drawing Interchange Format)
- ● GBX (Gerber, RS-247-X)

- Textyl \special commands
- TPIC
- PIC
- HTML Image map
- MF (MetaFont)
- MP (MetaPost)
- CGM (Computer Graphics Metafile - useful to import into Microsoft WORD, etc.)
- EMF (Enhanced Metafile)
- SVG (Scalable Vector Graphics)
- Tk (Tck/Tk toolkit canvas)
- Perl/Tk
- Shape (LaTeX \shapepar definition)
- SVG (Scalable Vector Graphics)

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

For combination with LaTeX, it is possible to either describe the entire figure with TeX/LaTeX macros, or to output the text part to LaTeX and the graphics part to PostScript and/or PDF using *Combined PostScript/LaTeX*, *Combined PDF/LaTeX* or *Combined PostScript/PDF/LaTeX*. See also TEXT FLAGS and LaTeX and Xfig.

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, PDF, Tex/LaTeX with tikz macros and SVG export languages support all features of **xfig** and a fairly high quality output will be generated.

The **fig2dev** program, part of the **Fig2dev** 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 ghostscript installed 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*

  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 available for PostScript, Encapsulated PostScript, PDF, PSLatex, PDFLatex, pstricks, tikz 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 <u>Print Panel</u> about available paper sizes. This is effective only if *PostScript* is selected at <u>*Language*</u>.

🔴 *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 <u>*Language*</u>.

🔴 *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 <u>*Portrait/Landscape*</u> in the <u>*View* menu</u>.

🔴 *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 <u>*Language*</u>.

🔴 *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 <u>*Language*</u>.

🔴 *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 <u>Netscape Navigator</u>, 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 <u>language</u> 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 <u>*Language* menu</u>.

🔴 *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 3.2.8b–dev   **Aug 2021**

New Features

Current release (xfig.3.2.8b): No new features, only bug fixes.

Older releases of xfig:

xfig.3.2.8

User interface changes:

● Panning sideways is enabled with Shift + Wheel up or down. Shift + Wheel Up pans left, Shift + Wheel Down pans right. Alt or Meta + Wheel Up pops up the edit panel, Alt or Meta + Wheel Down pops up the drawing panel. Previously, Shift + Wheel Up or Down popped up the edit or drawing panel, respectively, and panning sideways could only be achieved by moving the wheel sideways.

New features:

● To display gif image files, either giftopnm and ppmtopcx from the netpbm package, or imagemagick, or graphicsmagick must be installed.
● Obey the resolution given in embedded png, tiff and jpeg image files.
● Embedded image files are displayed in full color, not reduced to a maximum of 256 colors.
● Display embedded pdf files.

xfig.3.2.7b

● New drawings in Electronic/Schematic library. Thanks to Niko Sauer.

xfig.3.2.7

● Several new export languages (gbx, dxf, ptk). The backend-driver fig2dev had the capability before, but the language chooser in the export panel did not show these languages.
● Export to full-page PDF and multi-page PDF (similar to PostScript) now possible.

xfig.3.2.6a

● Lines on the screen are now drawn with the same width as in PostScript and other outputs. Previously, lines on screen were thicker.

xfig.3.2.6

● Update of documentation.

xfig.3.2.6-rc

These changes were made because the original view was of the drawing on the page instead of the modern view of the observer moving left/right/up/down:

● Left arrow moves observer left (used to scroll canvas left)
● Right arrow moves observer right
● Up arrow moves observer up
● Down arrow moves observer down
● Ctrl+Wheel up zooms in, Ctrl+Wheel down zooms out (was the other way around)

xfig.3.2.6-beta

● New export options: pict2e and tikz.
● Build using autoconf (./configure: make; make install), replacing xmkmf.

xfig.3.2.5c

● New library objects from Markus Laner: Networks: NodeB.fig, cloud.fig, router.fig, wlan.fig Computers: datacard.fig, screen.fig, usbpen.fig, xbox.fig
● New command-line option -nowrite_bak to turn off automatic renaming of .fig to .fig.bak when saving .fig file. X resource is "write_bak". Also command-line option -write_bak to force renaming to true.
● Can now specify small buttons with command-line argument -smallicons From Ed Rosten

xfig 3.2.5b: No new features, only bug fixes.

xfig 3.2.5a

●"Chop" tool which allows you to chop polylines, arcs and circles into individual objects that together, approximate the original object. From Chris Moller
● Fig-color merged into Fig app-defaults file because almost no one knows about the "*customization: -color" option to make X programs read the color def file.
● 22 new arrowhead types
● changed default browser to firefox and default pdf viewer to xpdf
● HTML MAP export (fig2dev) produces reference to .png file instead of .gif now
● A note added to i18n.html that says if you run fig2dev standalone, you must pass the -j option to it
● No need to have the C99 complex functions in the math library.
● New computers from Andre Esser (Libraries/Computers)
● Some network icons colored and/or combined by Roland Rosenfeld.
● When exporting only active layers have choice of bounding area of whole figure or only the active layers
● Two new library figures - piano_keyboard.fig, top view of a piano keyboard by Reinaert Albrecht, and piano_keyboard_perspect.fig, a perspective view of a piano keyboard by Brian Smith, both in the Libraries/Music directory.
● New library figure - scissors.fig in Libraries/OfficeEquip from Kasie Breezer Talbot
● Search tolerance increased from 4 pixels to 10 for zoom &lt 20
● Isometric grids to ease making isometric drawings From Jasper Wesselingh

• New 37-pin D connector and 50-pin ribbon connector in Libraries/Electronic/Physical

xfig 3.2.5

• Shift-U accelerator added to popup units panel
• The library menu button sizes with the library panel, to show longer path names if the user wants.
• The cursor for drawing lines, splines, etc. is a crosshair now instead of an arrow.
• Snap features to allow snapping points to endpoints, midpoints and intersections of other objects. From Chris Moller.
• Keyboard input of object coordinates for precise positioning. From Chris Moller.
• Can grab arc anywhere along the curve instead of just at grab points From Chris Moller.
• Added -noflipvisualhints for cases where, due to "different" pointer devices, they aren't really flipped, but the mapping indicates they are.
• Export to all of PS, PDF and LaTeX in one operation From Alistair Ramsey
• Reorganized Networks library and new devices: ASX-200, 1000, and 4000 by Bill Chimiak
• New figures "betty.fig" (Betty Boop) and "nikke.fig" (German detective Nick Knatter) in Libraries/Examples drawn by Markku Reunanen
• Can finish text input by pressing Escape. This removes the cancellation of a compose sequence (e.g. a-umlaut)
• Actually a bug fix, import of PDF files is now possible. The code was there since version 3.2.3, but the user interface was not.
• Nearly-full ANSI protoization of code by Harald Koenig
• Installation of Libraries and doc files faster with tar instead of shell loop
• New Gregorian chant music symbols from Bill Chimiak. Also, Music library divided into "chant" and "modern"
• Popup edit panel slightly more compact vertically
• New library "Fasteners" containing various bolts and screws From Jim Yuzwalk
• Increased size of numeric entry fields in indicator panels
• N-channel IGBT in Electronic/Schematics Library from Art Blair
• starting values for arrow type, width, length, thickness may be specified on command-line or in X resources: -startarrowtype, -startarrowwidth, -startarrowlength, -startarrowthick

xfig 3.2.5-alphas

• Added note to README and FAQ html file: If the Xaw or Xaw3d Athena widget sets are compiled with the ARROW_SCROLLBAR style of scrollbars, there is no StartScroll action and you won't be able to scroll using the wheel on the mouse. If you want to be able to scroll using the wheel, you must recompile the Xaw library from sources, disabling the ARROW_SCROLLBAR option.
• Astrological symbols for planets in Miscellaneous/Astrology by Andrew Collier
• Added key bindings to text dialogs to be more like modern systems (the Athena Widget Set that xfig uses is very old): Home: beginning-of-line End: end-of-line Del: delete-character-right
• Added *.jpeg* to picture browse options
• Snap mode to snap the point to selected features of selected objects.
• Keyboard entry capability to allow users to specify drawing or editing points from the keyboard.
• Chop editing operation to slice objects into pieces.

xfig 3.2.5-alpha4

Note: There is a minor change to the copyright/permission notice for xfig. Basically, I have restored the part that allows one to sell xfig regardless of whether it is bundled as part of a package or not. This is identical to the original copyright/permission notice for xfig, which was based on the MIT (then later, the X Consortium) copyright notice. Here is the new notice (different files have different author copyrights at the top):

```
> FIG : Facility for Interactive Generation of figures
> Copyright (c) 1985-1988 by Supoj Sutanthavibul
> Parts Copyright (c) 1989-2012 by Brian V. Smith
> Parts Copyright (c) 1991 by Paul King

> 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 distribute, sublicense and/or sell copies of
> the Software, and to permit persons who receive copies from any such
> party to do so, with the only requirement being that the above copyright
> and this permission notice remain intact.
```

- Introduction.html and installation.html updated to include Macintosh port of xfig
- Button to collapse depths of a compound object (make all same depth) in popup object editor
- Grid changed from dotted line to light red solid line
- Candle in Libraries/Miscellaneous by Dr. Lyman Hazelton
- Right-click on depth checkbox sets current depth in indicator panel to that depth
- Library of symbols used when diagramming folding instructions to make origami models and example contributed by Marc Vigo
- Can adjust width and height directly in popup editor for picture objects
- Support for David Hawkey's Xaw3D version 1.5E (ftp://ftp.visi.com/users/hawkeyd/X/Xaw3d-1.5E.tar.gz)

xfig 3.2.5-alpha3

- introduction.html and installation.html updated to include Macintosh port of xfig
- Button to collapse depths of a compound object (make all same depth) in popup object editor
- Grid changed from dotted line to light red solid line
- Candle in Libraries/Miscellaneous by Dr. Lyman Hazelton
- Right-click on depth checkbox sets current depth in indicator panel to that depth
- `-autorefresh` command-line option (resource: `Fig.autorefresh`) which will make xfig look at the timestamp on the .fig file and automatically load it and display it every time it changes.
- Removed requirement to compile with WHEELMOUSE when using wheelmouse
- When placing library objects, the name and comments are displayed in message window
- Mouse wheel can be used to scroll through filename lists in File and Export panels and icons or object lists in Library panel
- Thickness of ticks in dimension lines are user-adjustable now
- For attribute popup dialogs with only one text entry, keyboard now focuses on the entry as long as the pointer is anywhere in the dialog
- When using the "Open compound, keep rest visible", the other objects are drawn in shades of gray similar to the inactive layers feature.
- Zooming in or out with the Z or z key respectively will keep the canvas centered on the mouse pointer
- Full version and patchlevel is included in Fig file header for diagnostics
- Can explicitly set the rotation of imported pictures in edit panel after importing
- Export option to produce both EPS and PDF (in two files) in one step. Useful for those who both use LaTeX and PDFLaTeX
- "Epoch" added to rpm spec
- New HP/GL2 (fig2dev) driver from Glenn Burkhardt with paper size selection, offset, centering and orientation options

[Contents | Introduction | Credits]

Xfig User Manual
Version 3.2.8b–dev   **Aug 2021**

Bugs Fixed

Current release:

Bugs Fixed in *xfig 3.2.8b*

- Adding points to splines works if boxes are present in the figure.
- Stay in the working directory when browsing picture files.

Older releases of xfig:

Bugs Fixed in *xfig 3.2.8*

- Read version 1.3 fig files.
- The default resolution of embedded raster images is changed from 80 dpi to 72 dpi.
- Read compressed files, but do not silently uncompress them.
- Get correct bounding box (/MediaBox) of embedded pdf files.

Bugs Fixed in *xfig 3.2.7b*

Ticket numbers refer to https://sourceforge.net/p/mcj/tickets/#.
- Fix core dump when clicking File -> Export. Ticket #45.
- When exiting, cleanly pull down all windows. Ticket #54.
- Adding and deleting arrows to arcs works, ticket #46.
- Allow more than 100 comment lines, ticket #47.
- Make `xfig -update` work again, would segfault.
- Correctly draw interpolated splines, ticket #35.
- Position points on grid, if appres.userspacing unequal 1. Ticket #31.

Bugs Fixed in *xfig 3.2.7a*

Debian bug numbers refer to https://bugs.debian.org/#.
- Really fix a bug that a figure was saved in a second directory upon using the "Open.." dialog and pressing "Cancel". Ticket #9, debian bug #270113.

Bugs Fixed in *xfig 3.2.7*

- Interpolated splines are now created with the recommended value of the shape factor of -0.5, not with -1.0. A description of the shape factor as used in xfig has been added in the file doc/FORMAT3.2
- Apply xfig-3.2.6a-fallback-for-nonscalable-fonts.patch and xfig-3.2.6a-scalable-fonts.patch from Hans de Goede.

- Apply debian patches 08_every_time.patch, 10_RELEASE_DATE.patch, 11_declare_swap_colors.patch, from Roland Rosenfeld.
- Apply debian patch 07_colorsliderarrows.patch. Debian bug #870365.
- Change map of India to disputed boundaries, from https://en.wikipedia.org/wiki/File:India_disputed_areas_map.svg, retrieved 2018-02-25. Debian bug #270799, 09_india_disputed.patch
- Prevent integer overflow in {floor,ceil}_coords_{x,y}(). Fixes debian bug #300436. Also prevent integer overflow in round_coords().
- Completely remove alloca.h and alloca() from the code.
- When entering the "Save as..." dialog, moving to a new directory and pressing "Cancel", a subsequent "Save" wrote to the new directory. This happened analogously for the "Export" dialog. Fixed, ticket #9.

Bugs Fixed in *xfig 3.2.6a*

- Add compile-switch --enable-tablet for using an input tablet.
- Eventually, sorted out the xpm switches. Only use --enable-xpm-splash.
- Improve ./configure and add diagnostic output.

Bugs Fixed in *xfig 3.2.6*

- Update documentation to LaTeX and Xfig. From Roland Rosenfeld :
- Update LaTeX documentation, use \documentclass instead of \documentstyle, \usepackage instead of \input.
- Builds on hurd, had to #include in a few places.
- Use application/x-xfig, not image/x-xfig in xfig.desktop file. See https://bugs.launchpad.net/ubuntu/+source/xfig/+bug/690067 and https://bugs.launchpad.net/ubuntu/+source/xfig/+bug/1045848 .

Bugs Fixed in *xfig 3.2.6-rc*

- Should compile on Mac (Darwin) after installing dependencies using Macports, Fink or Homebrew.
- To compile on ancient systems, try make -DANCIENT.
- Report correct paths to installed files in the man page.
- Add pathmax.h and dirstruct.h, to define DIRSTRUCT and PATH_MAX.

From Vladislav Zavjalov; Commit numbers refer to http://git.altlinux.org/people/slazav/packages/?p=xfig.git;a=commit;h=#
- Preserve comments when deleting objects; Commit #6c42e5c
- Restore colors and main comment after Delete All + Undo; #34f6105
- Draw horizontal scrollbar if property buttons are too wide to fit on the screen. Commit #e9c1394
- Add save8bit app-resource to allow saving 8-bit files. #5bd9fbb
- Move Xfuncs.h and Xosdefs.h out from fig.h. Commit #b433478
- Move -l flags from LDFLAGS to LDADD. Commit #2d57c85

From Roland Rosenfeld. Bug numbers refer to https://bugs.debian.org/#.
- Do not link with -lXmu, if linking with Xaw3d > 1.5e.
- Make japanese documentation available for LANG=ja_JP.ujis.
- Set the Creation date in xfig_man.html to the date when creating the source tarball.
- Really fix reading png files, https://bugzilla.redhat.com/show_bug.cgi?id=1150330
- Delete some superfluos files, fix typos (Caribbean), add semicolon in xfig.desktop (debian bug #812477).

From Brian V. Smith:

- old link for "Xfig On Microsoft Windows (Getting and Installing Xfig html page)
- dash_list wrong dimension causes segfault on some line styles
- Should handle systems where REG_NOERROR (regular expression) is undefined now
- Missing xfig_man.html
- When exporting, wasn't setting locale to "C" for all command-line parameters, so floating point numbers would contain "," instead of "."
- Didn't allow enough space for fonts (would segfault when popping up File panels)
- Moving arc point would segfault

Bugs Fixed in *xfig 3.2.6-beta*

From Roland Rosenfeld. Bug numbers refer to https://bugs.debian.org/#. In fact, most of the bugs below, also those from other sources, were fixed by applying the debian-patches.
- Position independent executables work, fixes #756791.
- Revert previous change of X-spline parameter.
- Use C locale for fig2dev, to correctly interpret decimal point. Fixes debian bug #782737.
- Correct grid mode distances in man page. Debian bug #314820.
- Extend xfig.desktop.
- Fix unsecure use of temporary file. From Thomas Hoger, gentoo. Debian bug #565341.
- Escape - in man-page.
- Securely print string.
- Restore old shadow behaviour.
- Fix reading "/MediaBox" when importing pdf. Closes debian bug #530898.
- With -papersize b1, use B1, not B10. Debian bug #535181.

From Vladislav Zavjalov. Information on bugs is found at https://bugzilla.altlinux.org/show_bug.cgi?id=#
- Do not use fontsets for symbol and dingbat fonts, #26579.
- Fix compiler warnigs in w_indpanel.c. Commit feabe27.
- Restore correct depths when loading a figure, then undo. Commit b5c79b3.
- Restore filename after delete region, undo. Commit 219d690.
- Restore user color when opening new fig-file. Commit fccd915.
- Restore correct depths when joining lines with different depths. Commit a77bcaf.
- Fix xfontlist->fset initialization. Commit a592399.
- Fix default origin (100000,100000) -> (0,0) for filled objects. Commit 4d87e60.
- Fix 100% cpu-load by save panel. Commit 81a4596.
- Do not use local SimpleMenu.c with XAW3D1_5E. Would segfault on some 64-bit systems when opening main menu.
- Missing argument may cause crash or undefined behaviour. Debian bug #795642. From Michael Tautschnig.

From Hans de Goede. Bug information at https://bugzilla.redhat.com/show_bug.cgi?id=#.
- Patch for importing png-files, bug #1150330. From David Kaufmann.
- Fix crash when changing arrow size. Red Hat bug #1046102.
- Fix crash when creating dash-dotted lines. Red Hat bug #1023744.
From Michael Srb, thanks to Maurizio Pollini and David Kaufmann.
- Fix crash on exit. From Peter Volkov at gentoo

Bugs Fixed in *xfig 3.2.5c*

- Increased default width of layer panel from 58 to 64 to accomodate scrollbar width
- missing comma in FIXED_JAPANESE_PDF #ifdef case in w_cmdpanel.c
- printer name enclosed in apostrophes when printing in case has spaces in name
- new link from Elizabeth Bailey for Fig applications referenced in installation.html

- old link to duke.uask.ca for Fig applications has new link in installation.html
- Typo in message "GIF read error on extention ..." should be "extension"
- Possibility of stack overflow with malformed Fig files.
- In version 1.4 of the PNG library dither was removed so xfig now uses quantize when importing PNG images with palettes
- Other updates for PNG library version 1.5 from Peter Volkov
- Some versions of Cygwin don't have either REG_NOERROR or REG_OKAY defined so REG_NOERROR defined to be 0
- Changed X-Spline parameter to match original intent of X-Spline authors: changed definition of Q(s) from -s to -0.5 * s in u_draw_spline.c
- Allowance for stricter ghostscript in -dSAFER mode. Was causing error on reading EPS images. From Hans de Goede
- Security vulnerability with importing images fixed (RedHat bug # 657981 - xfig buffer overflow when opening .fig file with malicius color definition)
- When exporting to combined PS/PDF/LATEX the -D option to exclude all but active layers was not passed to fig2dev
- Uses 24-bit color instead of 8-bit when importing eps files (pcx24b driver for ghostscript)
- Bug in freeing null fontset
- Semicolon (;) added at end of MimeType line in xfig.desktop as per http://standards.freedesktop.org/desktop-entry-spec/desktop-entry-spec-latest.html
- Reference to "Darwin Ports" changed to "MacPorts"
- Renamed O_TEXT to O_TXT to avoid conflicts with system define
- On lines with Round or Projecting cap style and arrowheads, the line endpoint stuck out beyond the arrowhead

From Vladsilav Zavjalov:

- Uses fontsets for all fonts in international mode
- SEGV when zooming while creating object (e.g. text)
- Add locale_encoding appres which acts similar to euc_encoding, but uses system locale and mbrlen function for multibyte character length calculation.
- Move "file not found" error message from check_docfile() to launch_viewer(), remove excess check_docfile() call. Error message is not shown when looking for a locale-dependent docs. It is shown only if no docs found at all.
- Do not do strcpy with equal arguments in w_cmdpanel.c/update_cur_filename()
- Print to file had extra argument in sprintf in international mode
- Message panel not tall enough with 3D Athena widgets
- Warnings about different size of int and void * on 64-bit system. To fix this problem I changed int to intptr_t in some places.

Bugs Fixed in *xfig 3.2.5b*

- Dimension line indicator incorrectly showed arrow length/width
- Several patches from Fedora xfig maintainer
- divide by 0 under certain circumstances in ruler code - From Libor Pechacek
- Segfault if PRINTER env variable was not defined

Bugs Fixed in *xfig 3.2.5a*

- dimension line arrowhead width and length used %d instead of %f in dialog, always displaying 0
- leftover debugging printf(...color =) in startup
- finally fixed size of mode panel so it doesn't cover indicator panel under certain circumstances
- removed "Alpha" from splash

- updated copyright date in Help/About
- problem with depth panel height due to snap mode indicator
- snap indicator indicated "Focus" when "Diameter" was selected
- w_keyboard.c had pointer assignment reversed
- extraneous "done" in Imakefile in "install.jhtml" section From Eric Scott
- many protoytpe cleanups from Eric Scott
- #ifndef __FreeBSD__ added around #include &lt alloca.h &gt from Eric Scott
- uses rint() instead of lrint() for those with non-C99 compilers
- Solaris doesn't have REG_NOERROR for regex, so have new #define for that
- added dependency on version.h and patchlevel.h for f_util.c and f_save.c to Imakefile
- In imperial fractional scale mode, if the user scale factor wasn't 1.0, xfig would switch to decimal mode. This has been fixed to remain in fractional mode.
- Added condition for GLIBC to not declare srandom(int)
- Better resizing of message popup panel
- Map of Spain included Portugal. From Eugen Dedu.
- Double-clicking on either style or family would crash xfig
- Export/Print grid units were not reset when loading Fig file of different units
- When converting empty export/print grid units to fractional inches would make 0/2
- Escaped hyphens in man page From Roland Rosendfeld
- When exporting to all three: PostScript, PDF and TeX, incorrectly generated PostScript and PDF that included LaTeX-only (special) text
- Bug when doing Save As to a shorter filename - would corrupt name sometimes
- Note about 450 pixels/cm didn't make it from the FORMAT3.1 file to FORMAT3.2
- Print command was using -P instead of -d for lp
- On Cygwin, needed to double-escape PRINTER environment when backslashes in name
- w_library.c had incompatible poiner type in call to ScanLibraryDirectory
- Not enough characters allocated for ruler inch/cm indicators. Would cause segfault on zooming out because of longer ruler text, e.g. -10210cm

Bugs Fixed in xfig 3.2.5

- xfig would crash on 64-bit processors because of a missing include, which causes new_string() to be declared implicitly as returning an integer. This may cause the top 32 bits to get lost and hence the crash.
- Blanks not preserved in imported picture filename when reading Fig file
- When in metric mode, decimal precision was ignored for dimension lines
- When showing vertex numbers on objects, first vertex is now 0 to match edit window vertex numbers
- Also, vertex numbers are not shown on inactive (grayed) layers
- Widget shadow resources moved from Fig-color to Fig because they don't really have anything to do with color. There was a problem when running KDE because it set a global resource *Scrollbar*height: which made xfig freeze when *customization: -color was NOT used.
- The point positioning indicator is turned on when editing a compound object to show the user that it is used
- Increased the maximum size of images that can be imported. There was a limit in the PostScript image encoder of 4096x4096 pixels. This was increased to 8192x8192.
- Centered text was changed to left-justified when flipped horizontally inside a compound object.
- With -v option, after reporting the version, xfig would say that -v was an unknown option
- Local locale was being used when writing the xfig cut buffer file instead of switching to "C" locale. This created commas instead of decimal (.) for numbers in those particular locales. From Dirk Osswald
- Local locale was used when forming command for calling fig2dev, resulting in commas instead of decimals for floating point numbers.
- count_user_colors buffer overflow fixed
- Under Cygwin, temporary file stayed around after unlink(), causing error when importing more than one ps/eps/pdf file
- Segfault when using -update because appres resources were NULL
- -update option failed when not first option passed to xfig

- Clicking window manager "close window" button in library titlebar didn't close the window
- Wasn't distributing objects in compounds when there were only 2 objects
- Some compilers complain about the order of declaration in u_fonts.h - fixed
- Northern part of map of India was incorrect (Libraries/Maps/Asia/india.fig)
- Map of Serbia was misnamed "yugoslavia.fig" (Libraries/Maps/Europe/yugoslavia.fig)
- Map of Central Europe had old Yugoslavia instead of Croatia, Serbia, Bosnia- Herzegovina, and Macedonia (also, Slovenia was mistakenly inside the border)
- Editing a compound object with more than 200 texts would crash xfig

Bugs Fixed in *xfig 3.2.5-alpha5*

- typo in latex_and_xfig.html and LATEX_AND_XFIG files. Text should be: \convertMPtoPDF{foo.0}{1}{1} It was missing parameters {1}{1}
- -correct_font_size missing from -help option list and man pages
- -help and -version didn't work unless they were first in the options
- -O option wasn't passed to fig2dev for overlapping pages in multiple page mode for PostScript export

Bugs Fixed in *xfig 3.2.5-alpha4*

- When loading a library object, if it contained only a compound and nothing else, when xfig promoted that compound to the toplevel the main comment was lost
- Incorrect header files used for SmeBSB resulted in either segfault or none of the command panel entries being underlined
- Drawing very large splines (e.g. at zoom = 0.01) caused integer roundoff errors, making xfig loop indefinitely
- Bug where a line that had a zero width or length arrowhead was not redrawn after being moved, copied, canvas redraw, etc.
- The page border and axis lines would obscure Fig objects when moving, copying etc. other objects on the canvas.
- In the popup picture editor, if the relative position of the corners of the picture were changed, the rotation field was not updated (this bug was in 3.2.5-alpha3 only)

Bugs Fixed in *xfig 3.2.5-alpha3*

- Missing #ifdef XAW3D in SimpleMenu.c
- Bug when breaking a compound object - depths were added twice to the counts
- When implicitly cancelling the placement of a library object by choosing another mode, xfig would tell user to cancel or finish the current operation, but there was no way to do that
- Picture Reread button was active before file was read the first time
- Rereading picture in edit popup produced bad colors
- When using multiple copies of an imported picture, h/w ratio was not computed for copies
- Importing PCX images were incorrect when bytes per line different from width*bpp
- When passed a filename containing a directory name for a Fig object library using -library_dir, and that directory only contains Fig files and no subdirectories with Fig files, xfig segfaulted
- Several checks for memory allocation failure added to the library loading procs
- Forgot to free allocated memory when return abnormally from loading libraries
- Path length check in loading libraries fixed
- Missing include for put_msg prototype in e_measeure.c
- Needed #ifdef for XtVersion in SmeBSB.c (X11R5 doesn't have international fontset)
- Added SIGPIPE signal to ignore in case an external program dies when we're using pipes
- Screen capture on an MSBFirst X server with 24/32 bits per pixel was incorrect.

- Importing any image file on such a server was incorrect (bytes/bits reversed).
- Reading GIF or PCX files on 24-bit server *and* on big-endian machine (e.g. Sparcstation) resulted in bytes being swapped and funny colors
- Could popup unit dialog when drawing/editing objects
- Embedded whitespace in filenames in recently loaded files weren't parsed properly (.xfigrc)
- When pasting an object on the canvas, point positioning grid wasn't used
- Bug when reading a compressed eps file (file handle was passed to open proc instead of name)
- Minor grid spec used twice instead of minor/major when passed to fig2dev
- Fixed conversions of export/print grid values when switching to/from metric, decimal or fraction
- When loading or merging a file, xfig appended ".fig" to the name if there wasn't ".fig" in the name. Now it only appends ".fig" if there is no suffix (no ".").
- When appending the ".fig" before the previous change, xfig would segfault
- Libraries/Electronic/Schematic/transformer and transformer_ironcore aligned to 1/16" grid
- Bugs in indicator panel display of text flags, dimension line params and arrow size params when cycling through settings with middle or right mouse button
- Bug in callbacks for dimension line checkboxes that select actual length or user text
- When exporting to Combined PDF/LaTeX it uses ".pdf" and ".pdf_t" suffixes because LaTeX doesn't recognize ".pdftex" as a PDF file
- Better clipping around arrowheads on thick lines (lines that are thicker than the arrowhead is wide)
- Checks for open splines of less than 2 points when reading figure file and removes them
- Clicking middle mouse button after creating first point of closed spline switched to freehand mode
- Export panel sections would get messed up when changing export languages
- Now checks whether scrollbars support StartScroll before trying to use it for the wheel scrolling. When the Xaw widgets are compiled with ARROW_SCROLLBAR, there is no such action.
- pstex_t export lacked border option (-b) to align LaTeX text when pstex figure specified border (also fixed in fig2dev)
- Create one picture object with no filename, then create another and xfig crashed
- Some bugs when freeing dimension line components
- Bad choices for grid dot spacing in metric mode in the 5mm grid, and decimal inch mode in the 0.5 and 1.0 inch grids
- -international flag missing from xfig.html and xfig.man docs
- Added call to XsetLocaleModifiers() when initializing input method (-international mode only)
- xfig was limiting arrowhead lengths to 50 pixels instead of 50 inches, and the width to 10 pixels instead of 10 inches.
- edit panel for circles shouldn't have "angle" entry
- edit panel wasn't allowing typing in of negative angles for text and ellipses
- Objects were sometimes drawn with a wild point when zooming
- Full path was being added to default export filename and wasn't changing when user changed directories
- When drawing a box or rounded box with the "show line lengths" on, the sizes were in Fig units (1200ppi) instead of user units.
- Spacing cedilla (ISO 0xB8 / octal 270) was missing from CompKeyDB file
- Changing the units in the popup edit panel for a text object caused a segfault.
- Rulers and grid didn't change scale when user scale was != 1.0. Even though the message window showed the correct user scale when drawing objects, the rulers and grid still showed the unscaled values.
- Axis lines through 0,0 now drawn after page border so it remains visible when there is a grid
- Segfault if current directory was deleted after starting xfig
- Positioning grid was set to "ANY" when editing a compound object, causing the original bounding box to be lost
- Bug in arc drawing caused arcs to be drawn as circles at high zoom
- Computing the area of a polygon larger than 38x38 inches overflowed calculation
- Bug in bounds calculation for ellipses and circles that increased bounding box even with line width = 1
- Limit on number of styles in a family wasn't checked
- If all depths were turned off and any edit operation was attempted on the canvas such as move object, delete object, xfig would hang, searching for objects indefinitely.

[Contents | Introduction | Credits]

Xfig User Manual

Version 3.2.8b–dev   Aug 2021

Getting and Installing Xfig and Fig2dev

You need the **fig2dev** program in addition to **xfig** to be complete. **Xfig** saves figures in its own, proprietary .fig format. The **fig2dev** program translates the .fig files to various languages and formats such as PostScript, png, jpeg, pstricks, tikz, etc. **Fig2dev** allows you to print and export figures.

Most probably, **xfig** and **fig2dev** are available as pre-built (binary) packages using the package or software manager of your operating system. You will have to install both the xfig as well as the fig2dev or transfig package. Until about 2016, the fig2dev package was called transfig.

Binary packages of xfig and fig2dev (or transfig) are available for a number of Linux operating systems (debian, ubuntu, fedora, centos, suse, arch, etc.), some BSD's (netbsd, openbsd, freebsd), and for Darwin (fink, macports, brew).

For Windows, xfig can run under cygwin. Xfig and fig2dev are available as cygwin packages. Fig2dev can probably be compiled with VisualStudio.

Installing Other Software

In addition to **xfig** and **fig2dev**, it is very useful to have

- **ghostscript** (the `gs` command) and one out of the
- **netpbm**, **imagemagick** or **graphicsmagick** packages installed.

Xfig will call **ghostscript** to display postscript or pdf files embedded as a picture.
Fig2dev will need **ghostscript** to be able to export to pdf and to the bitmap formats (gif, jpeg, tiff, png, etc.). For some bitmap formats, fig2dev will try to call programs from either the **netpbm**, **imagemagick** or **graphicsmagick** packages to produce smaller output files.

Getting and Installing from Sources

You can get the sources for **xfig** and **fig2dev** from https://sourceforge.net/projects/mcj. Files such as `xfig+fig2dev-3.2.Xx.tar.xz` contain the sources for both programs, while `xfig-3.2.Xx.tar.xz` and `fig2dev-3.2.Xx` contain the sources for xfig and fig2dev, respectively.

Uncompress and untar the archives (`tar xaf *.tar.xz` or `tar xJf *.tar.xz`), navigate into the base directory, e.g.,
```
cd fig2dev-3.2.??
```
configure, build and install the program:
```
./configure
make
make install
```

See the INSTALL files of xfig and fig2dev for further instructions. Especially xfig needs some further options to ./configure.

To build xfig, the minimum requirements are the runtime libraries and the header files for libx11, libxt, libxaw (or libxaw3d), libxi, and libxpm. For fig2dev, the zlib (zlib1g) runtime libraries and development (header) files are needed. Optionally, the png, jpeg and tiff libraries are recommended for reading and displaying image files in xfig. Fig2dev will need the png library to export .fig files containing a png image to postscript, pdf or any image format.

Installing Ghostscript's Type 1 fonts under X

You can use Ghostscript's high quality Type 1 fonts on your X server (and therefore in xfig). Copy the necessary files from Ghostscript's fonts directory to /usr/X11R6/lib/X11/fonts/Type1/ (or wherever your X server Type 1 fonts are).

```
$ cp /usr/share/ghostscript/fonts/*l.pfb /usr/X11R6/lib/X11/fonts/Type1
```

Append the file `/usr/share/ghostscript/fonts/fonts.scale` to `Type1/fonts.scale`. The first line in the latter file contains is the number of fonts listed in that file. Change that number to include the additional fonts.

Now run `mkfontdir(1)` to update the font database:

```
$ mkfontdir
```

If your operating system has the `chkfontpath` command, alternative is to add the Ghostscript font directory to the X server font path:

```
chkfontpath -a /usr/share/ghostscript/fonts/fonts/
```

In either case, if your X-server is running already you have to tell it to reread the font databases with `xset(1)`:

```
$ xset fp rehash
```

Related Software

⚜ **GhostScript**
  xfig will invoke **GhostScript** (the `gs` command) when displaying an imported <u>PostScript image</u> or when exporting to pdf or any bitmap format.
⚜ **AutoTrace**
  Converts bitmap to vector graphics - has a Fig file output driver. Available from <u>http://autotrace.sourceforge.net</u>. Written by Martin Weber.
⚜ **chemtool**
  A chemical structure editor for X11 available from <u>http://ruby.chemie.uni-freiburg.de/~martin/chemtool/chemtool.html</u> that writes Fig output (and relies on fig2dev 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, then at Fraunhofer ISE, Germany. See <u>http://www.pab-opto.de/progs/figbuild</u>
⚜ **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 http://www.gnuplot.info/.

🔴 **hp2xx**

hp2xx can convert HP-GL files into other formats such as PostScript, LaTeX or **Fig**. It is available from https://www.gnu.org/software/hp2xx.

🔴 **Msphinx**

A friendly package to handle satellite images. It can directly link to **Xfig**. Find it at: https://www-loa.univ-lille1.fr/informatique/logiciels/Msphinx.html.

🔴 **netpbm**

fig2dev will invoke the appropriate command from the **netpbm** package when exporting figures to bitmap formats. **netpbm** is available from https://sourceforge.net/projects/netpbm/, see also http://netpbm.sourceforge.net.

🔴 **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), and a translator to **Fig** format (plot2fig). By Robert S. Maier <rsm@math.arizona.edu>. See https://www.gnu.org/software/plotutils.

🔴 **pstoedit**

pstoedit can convert PostScript files into **Fig** format. It is available from https://pstoedit.sourceforge.io, see also http://pstoedit.net.

🔴 **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.tar.gz

🔴 **xpaint**

xpaint is a painting program which runs under X11. It is available from https://sourceforge.net/projects/sf-xpaint/files/sf-xpaint, see http://sf-xpaint.sourceforge.net.

🔴 **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 from http://www.trilon.com/xv.

[Contents | Introduction | Credits]

Xfig User Manual

Version 3.2.8b–dev   Aug 2021

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.3.2.5d.tar.gz** and *gunzip* and *untar* it.
2. Uncomment "`#define I18N`" in `xfig.3.2.5d/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.3.2.5e.tar.gz** and *gunzip* and *untar* it.
2. Uncomment "`#define I18N`" in `transfig.3.2.5e/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. <u>Compile and install</u> **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 <u>resource file</u>, `-international` option may omitted.

    N.B. If you run **fig2dev** standalone you must use the `-j` option to enable internationalization.

Entering Text

Using this internationalization facility, you may enter text in Japanese or some another languages (henceforce, call "international text") with the <u>TEXT</u> 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 <u>TEXT FONT</u>.

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 <u>TEXT</u> 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 <u>TEXT FONT</u>. If any other font is selected, <u>Latin-1 characters</u> will be available as in normal **xfig**.
- It is not possible to edit international text in the <u>Edit Panel</u>. But it is possible to edit text on the canvas in <u>TEXT</u> 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

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.

```
/*
 * FIG : Facility for Interactive Generation of figures
 * Copyright (c) 1985 by Supoj Sutanthavibul
 * Parts Copyright (c) 1989-2002 by Brian V. Smith
 * Parts Copyright (c) 1991 by Paul King
 * Parts Copyright (c) 1995 by C. Blanc and C. Schlick
 *
 * 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.
 */

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.

Note about metric units: To preserve a regular grid on the canvas the
centimeter is defined to be 450 Fig units and not 472.4 (1200/2.54).
For drawings done in metric units, fig2dev magnifies the output when
exporting or printing to compensate for the difference (472.4/450).
Also, if you make a drawing in one unit scale and switch to the other
units in xfig, the drawing will be rescaled on the screen by xfig to
remain consistent.

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, solid, dash, dotted, etc.)
        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, specification for dash/dotted lines)
        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, solid, dash, dotted, etc.)
    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, specification for dash/dotted lines)
    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, solid, dash, dotted, etc.)
    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, specification for dash/dotted lines)
    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            (enumeration type, solid, dash, dotted, etc.)
          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, specification for dash/dotted lines)
          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. For positive values of this
factor, the spline is approximated at this point, for negative values the
spline is interpolated 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., an angular point).
Recommended values for this factor are 1.0 for an approximated spline, 0.0
for an angular point and -0.5 for interpolated splines.
The shape factor used here corresponds to the parameter s_k defined in
section 4.1 of Blanc & Schlick (1995) for values greater than 0, for values
smaller than 0 it corresponds to the negative of the parameter q defined in
section 5.1 of Blanc & Schlick (1995).


```
==============================================================================
(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
```

[<u>Contents</u> | <u>Introduction</u> | <u>Credits</u>]

Xfig User Manual
Version 3.2.8b–dev   **Aug 2021**

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?

Problems when running xfig or TransFig (fig2dev)

1. When I pull up a menu, **xfig** crashes
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.
19. When I enter values in the "spinner" widgets, the digits are entered in reverse order.
20. LaTeX asks for missing macro \k *or* the ogonek diacritic mark isn't visible.
21. Text is positioned differently when exported than it is on the screen
22. I cannot type anything into the popup dialogs in xfig when I use the fvwm window manager
23. When I start xfig I get the message: "Warning: Actions not found: StartScroll"

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`".

`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.

Problems when running xfig or TransFig (fig2dev)

1. When I pull up a menu, xfig crashes

POSSIBLE CAUSE:
> Your version of the 3D Athena Widget Set may be incompatible

SOLUTION:
> If your vendor has installed the Xaw3D1.5E verison of the 3D Athena Widget set but has called it Xaw3D, then it is incompatible with **xfig**'s menu widget. Fix this by uncommenting the **#define XAW3D1_5E** line in the Imakefile (remove the "XCOMM") and recompiling **xfig**. Be sure to do:
> > xmkmf
> > make clean
> > make install

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 file.

SOLUTION:

Install the app-default file (`Fig.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:

A warning about some action not found occurs, like "Warning:Actions not found: ModeOpenCompound"

```
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"
```

19. **When I enter values in the "spinner" widgets, the digits are entered in reverse order.**

POSSIBLE CAUSE AND SOLUTION:

There is a bug in the RedHat Xaw Athena widget set. Any of the Xaw3d (3-D) Athena widget sets will work properly. Uncomment the "#define XAW3D" line in the Imakefile and recompile xfig according to the instructions in the README file.

Pattern fills are going beyond the bounds of the polygons when Iexport to EPS, GIF, etc., although they look

20. **LaTeX asks for missing macro \k *or* the ogonek diacritic mark isn't visible.**

    POSSIBLE CAUSE AND SOLUTION:
    > Define the macro `\def\k#1{\c{#1}}` before the `\begin{document}` directive in your LaTeX document.

21. **Text is positioned differently when exported than it is on the screen**

    POSSIBLE CAUSE AND SOLUTION:
    > Your X server (screen) fonts may not be the same as the PostScript fonts that fig2dev uses when exporting. If you use the ghostscript fonts in your X server things will improve. To do this, see Installing Ghostscript's Type1 fonts under X.

22. **I cannot type anything into the popup dialogs in xfig when I use the fvwm window manager**

    POSSIBLE CAUSE AND SOLUTION:
    > There have been reports of this problem when using the fvwm window manager, possibly under Cygwin. Try mwm or twm or KDE.

23. **When I start xfig I get the message: "Warning: Actions not found: StartScroll"**

    POSSIBLE CAUSE AND SOLUTION:
    > When the Xaw or Xaw3d Athena widget sets are compiled with the **ARROW_SCROLLBAR** style of scrollbars there is no StartScroll action and you won't be able to scroll using the wheel on the mouse. If you want to be able to scroll using the wheel, you must recompile the Xaw library from sources, disabling the **ARROW_SCROLLBAR** option.
    > From xfig version 3.2.5 on, it checks if this action exists and doesn't give the warning.

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]

On some HP computers using native cc compiler there is aproblem with +O3 optimization   174

Xfig User Manual
Version 3.2.8b–dev   **Aug 2021**

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, Australia 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.

Thomas Loimer 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

Copyright (c) 1998-2012 by Tom Sato and Brian V. Smith.

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.

Permission to use, copy and distribute this documentation for any purpose and without fee is hereby granted. Modification of this documentation is also granted as long as this copyright/permission notice remain intact and name of the person who made the modification is explicitly written in the documentation. However, contact with Tom Sato and/or Brian V. Smith is strongly recommended if you want to distribute modified version of this documentation.

Xfig

- Original Copyright (c) 1985 by Supoj Sutanthavibul
  Parts Copyright (c) 1989-2012 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*)
  --- *thomas.loimer@tuwien.ac.at* (Thomas Loimer)
- 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*.