

The KGraphViewer Handbook

Gaël de Chalendar
Federico Zenith



The KGraphViewer Handbook

Contents

1	Introduction	5
2	Using KGraphViewer	7
2.1	The Main Window	7
2.1.1	Moving the Graph inside the Window	8
2.1.2	Zooming	8
2.1.3	Graph Files Handling	9
2.1.3.1	Files Modified on Disk	9
2.1.3.2	Files and Windows	10
2.1.3.3	Session Handling	10
2.2	Graph Printing	11
2.3	Configuring KGraphViewer	14
2.4	Miscellaneous Functions	16
3	Command Reference	18
3.1	The KGraphViewer Menus	18
3.1.1	The File Menu	18
3.1.2	The View Menu	18
3.1.3	The Settings and Help Menu	19
3.2	The KGraphViewer Toolbars	19
3.2.1	The Main Toolbar	19
3.2.2	The View Toolbar	20
3.2.3	The Help Toolbar	20
4	Credits and License	21
4.1	Program	21
4.1.1	Main Authors	21
4.1.2	Contributors (to be completed)	21
4.2	User Guide	21
4.2.1	Contributors	21
4.3	Other Credits	21
4.4	Licenses	21

Abstract

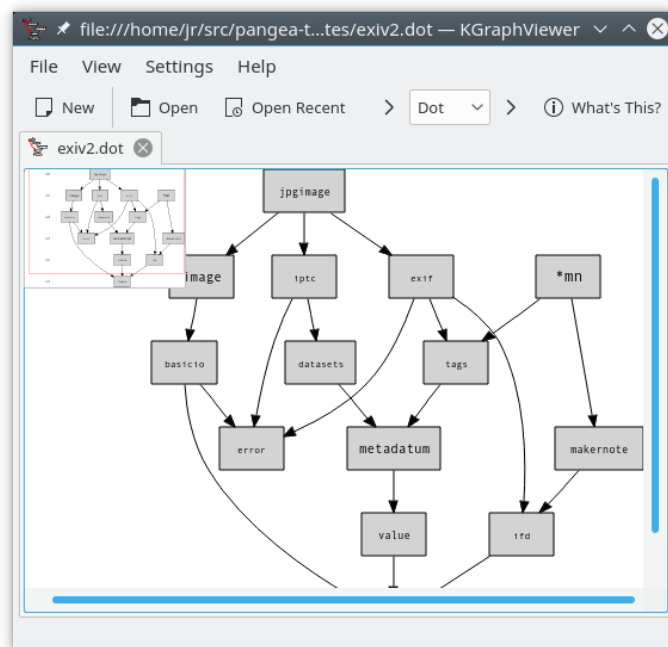
KGraphViewer is a [Graphviz](#) DOT graph viewer. It is developed in the KDE community. The [Graphviz](#) programs are free-software layout engines for graphs. Graphs are commonly used in scientific domains and particularly in computer science.

KGraphViewer displays the graphs in a modern, user-friendly GUI.

Chapter 1

Introduction

KGraphViewer is a [Graphviz](#) DOT graph viewer. The [Graphviz](#) programs are free-software layout engines for graphs. KGraphViewer displays the graphs in a modern, user-friendly GUI. It has the following features:



- Zooming;
- Loading of several graphs in tabs at the same time;
- Storage of a list of recent files;
- A bird's eye view of the graph;
- Graph panning by mouse dragging;
- Full-featured printing;
- Perfect drawing of all [Graphviz](#) example graphs;
- Node and edge colors and all node shapes are supported;

The KGraphViewer Handbook

- Automatic choice of DOT for directed graphs and NEATO for undirected ones;
- Manual or automatic reloading with user confirmation of (externally) modified files (configurable);
- Opening of new program instances as new tabs in an existing instance (configurable);
- Help system;
- Internationalization.

Planned features are:

- Eventual use of the Qanava library to support internal graphs representation;
- Save session data about graphs (layout program used, zoom factor, etc.)
- Integration in Kate and Konqueror (so that a change in Kate's view of the graph would re-layout and reload the file in KGraphViewer);
- Menu to open the file in Kate *or* to open a katepart inside the KGraphViewer window;
- Loading of large and huge graphs (currently, there are performance problems at around 1000 nodes, and a hack to ignore nodes above a specified limit);
- Progressive support of more and more DOT attributes and values;
- Layout in background (with threading in order not to block the interface, with a progress bar).

NOTE

Note that this program includes code from the GPL tool KCacheGrind by Josef Weidendorfer, with his authorization.

NOTE

Note also that the printing system is from the Calligra's Kexi database application.

Other required software: KGraphViewer is a program that uses the [Graphviz](#) tool DOT for graph layout. Its dependencies are:

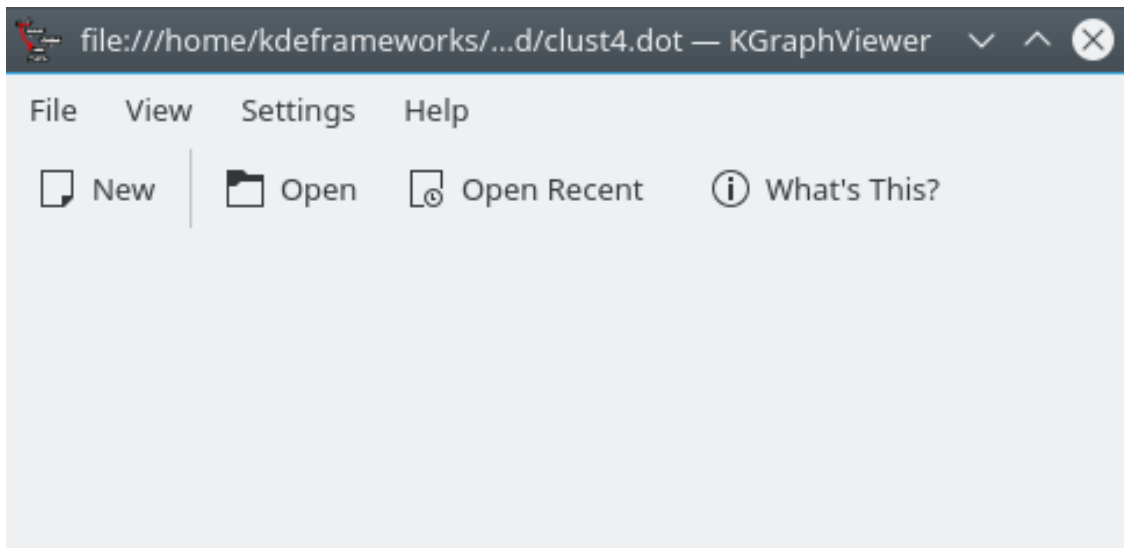
- The Qt™ and KF5 libraries version 5.x;
- The [Graphviz](#) tools as externals programs (tested with version 2.8);
- The [Boost](#) library version 1.36.

Chapter 2

Using KGraphViewer

2.1 The Main Window

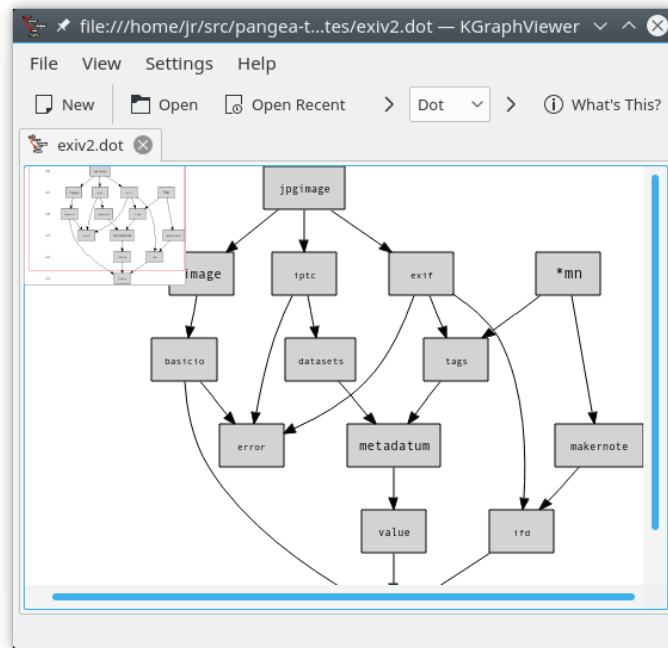
The main window displays the loaded graphs, each in its own tab. Initially, if you start KGraphViewer without arguments, its main window will be empty as in the screenshot below. In this case, the toolbar contains only the **Open File** and the (empty) **Recently Opened Files** items. Click on the first one to open the standard **Open File** dialog.



The initial main window when opening KGraphViewer for the first time

After selecting one or several DOT files, the main window looks like the next screenshot below:

The KGraphViewer Handbook



The main window of KGraphViewer with several files opened



As you can see, if the graph is larger than the available area, a bird's-eye view is shown in one corner. You can choose with the context menu whether the optimal position of this overview should be automatically computed for you, or place it where you want yourself.

2.1.1 Moving the Graph inside the Window

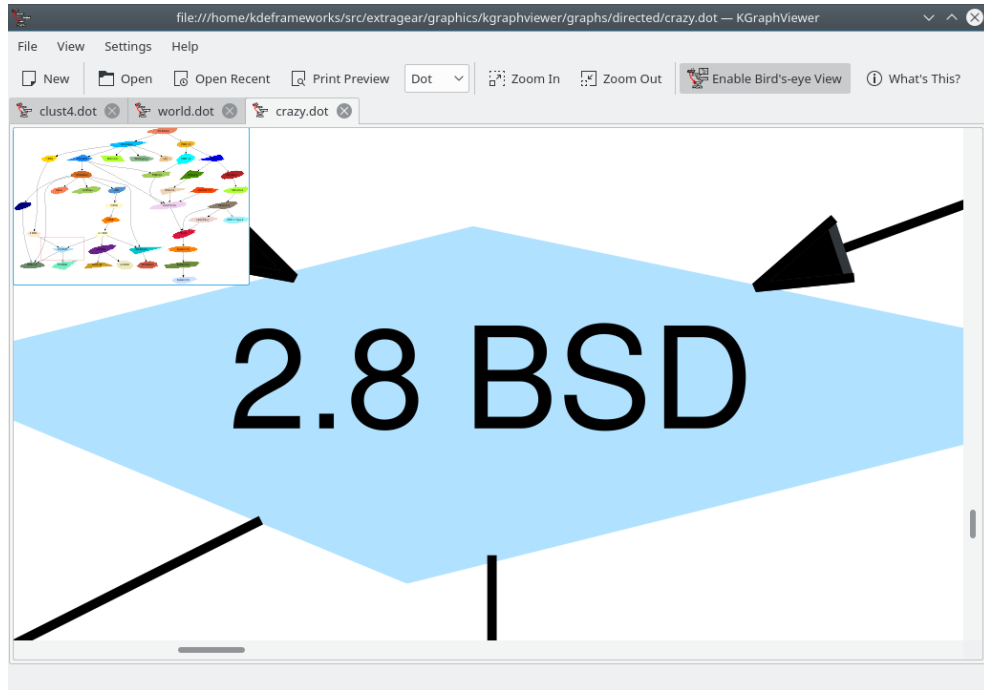
To move the graph, you can:

- Click and drag it;
- Use the scroll bars;
- Press the arrows keys;
- Click somewhere in the bird's-eye view;
- Use the mouse wheel (up and down with no modifier, left and right with the **Alt** key pressed);
- or click and drag in the bird's-eye view.

2.1.2 Zooming

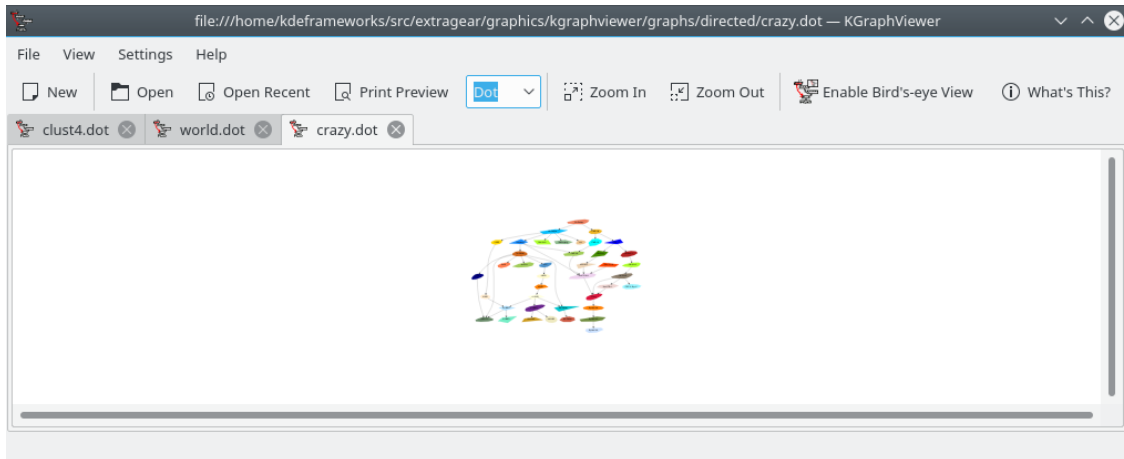
To zoom, you can either use the  and  toolbar buttons to zoom in and out, or hold down the **Shift** key while rolling your mouse wheel. While zooming in, the red rectangle in the bird's-eye view that indicates the displayed zone gets smaller and smaller. If it were to get smaller than 10 pixels in its shorter side, it will be replaced by a filled red rectangle in order to stay visible. You can see that on the screenshot below.

The KGraphViewer Handbook



A graph at the maximal zoom in level

Zooming is limited in both directions by a factor of 10. You can see below the maximum zoom-out factor on a graph.

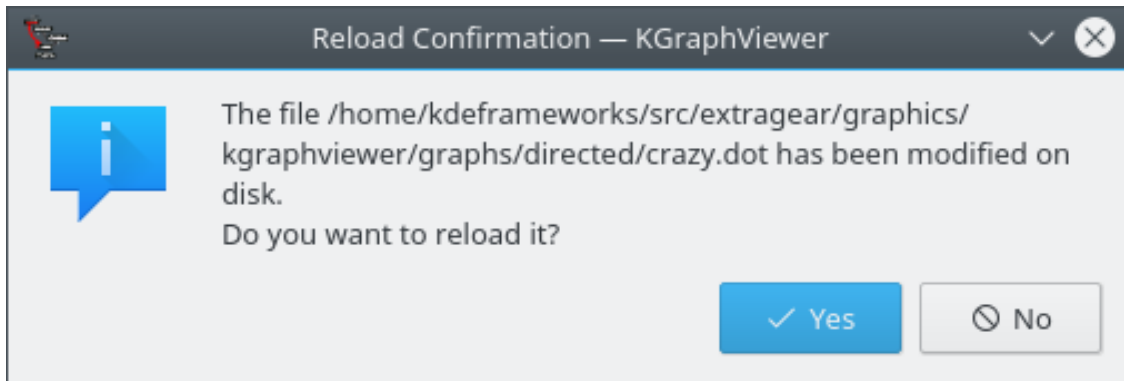


A graph at the maximum zoom-out level

2.1.3 Graph Files Handling

2.1.3.1 Files Modified on Disk

If a file currently loaded in KGraphViewer is modified by another program, the following dialog is shown:

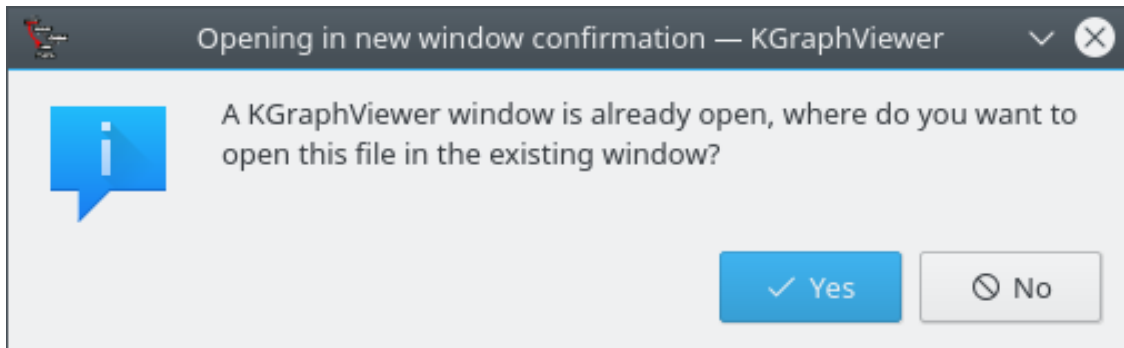


Dialog to reload a file modified on disk

You can choose to reload the graph or to ignore the change. You can also get KGraphViewer to remember your choice, so it will do the same thing the next time. If you later want to modify your choice, choose the **Settings** → **Configure KGraphViewer** menu item. See Section 2.3.

2.1.3.2 Files and Windows

When you open a graph file from outside a running KGraphViewer instance, e.g. from a file browser or the command line, you have the choice to open a new KGraphViewer window or to open the graph in a new tab in the existing window. This is done with the dialog below.

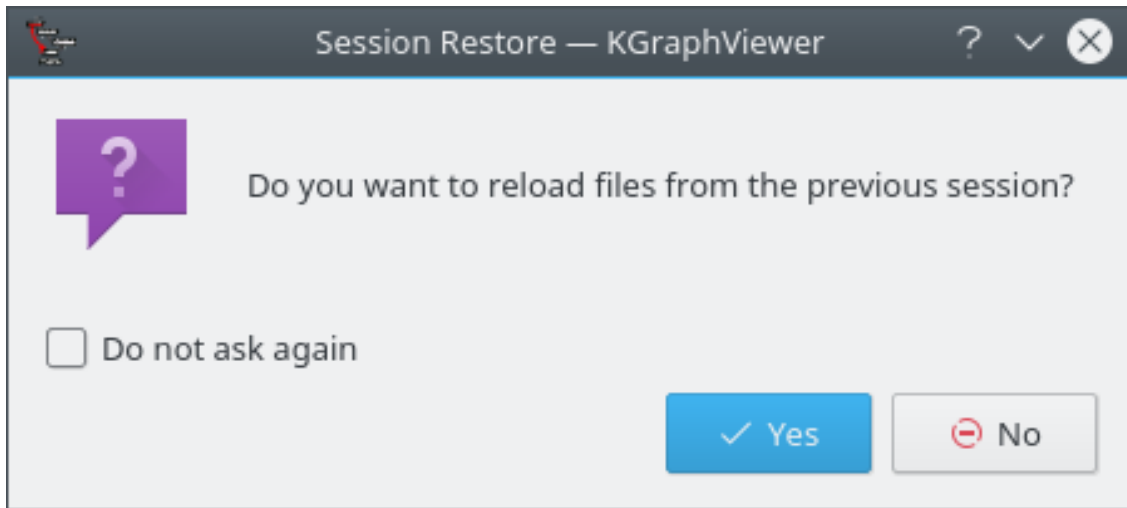


Dialog to open a new window

As for the previous option, you can save your default choice and modify it later in the configuration dialog.

2.1.3.3 Session Handling

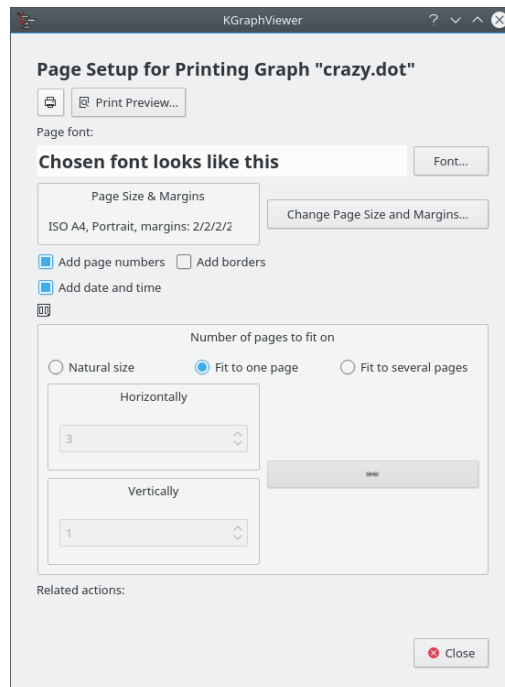
At start-up, you can choose to reopen the graphs that you had opened the previous time. Again, you can save your choice and change it later.



Dialog to reload the graph of the previous session at start-up

2.2 Graph Printing

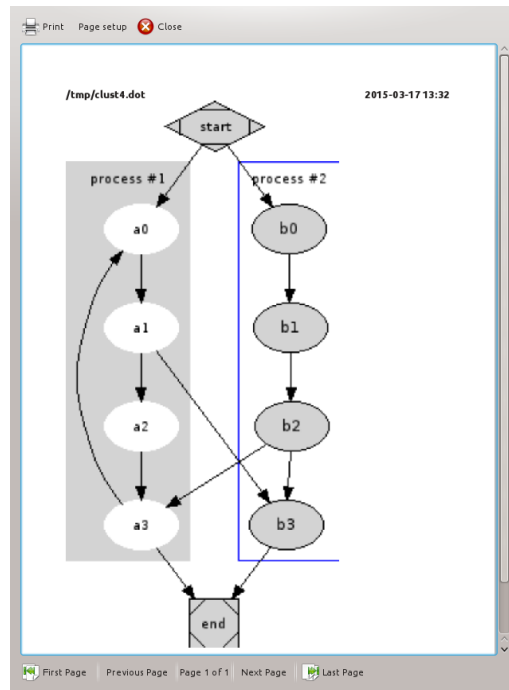
Try the **Print preview** or the **Page setup** buttons to explore the printing options.



Page setup dialog

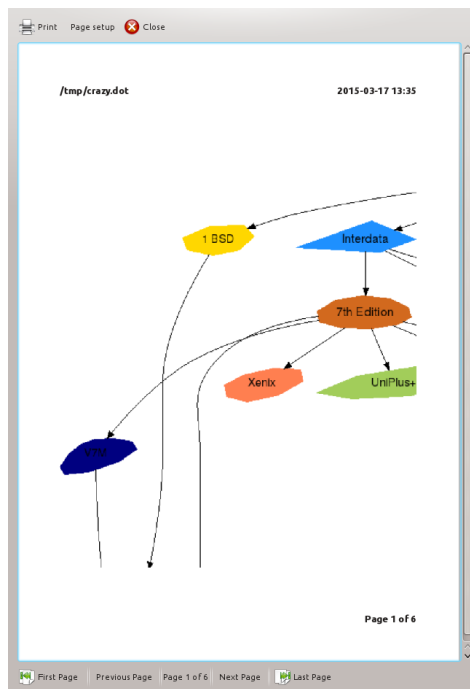
You can setup the printing options to fit the graph on exactly one page, as below:

The KGraphViewer Handbook



Page preview setup to fit on one page

You can also choose how to divide the graph on multiple pages. The number of horizontal and vertical pages can be chosen independently.

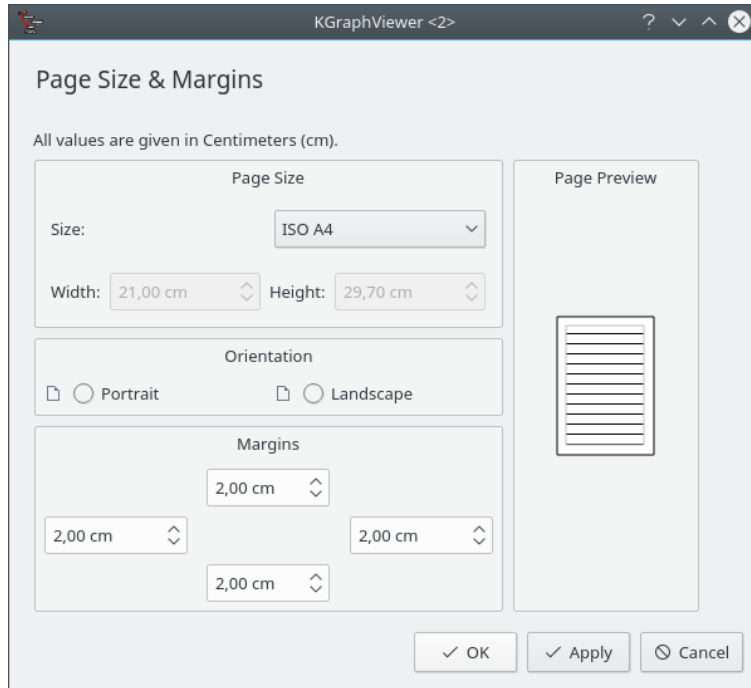


Page preview set up to fit on multiple pages and to fill them

Finally, when using multiple pages, you can choose to make the part of the graph on each page to fit its page like above, not keeping the aspect ratio, or to keep it. In the latter case, the pages will not be filled.

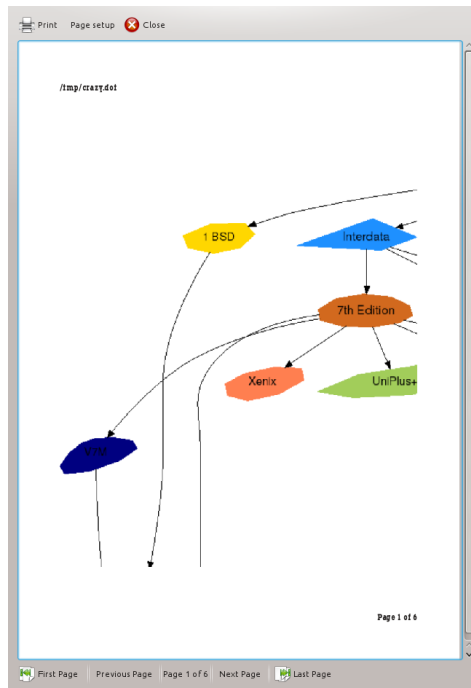
You can change the size of pages and the margins:

The KGraphViewer Handbook



Page size and margins dialog

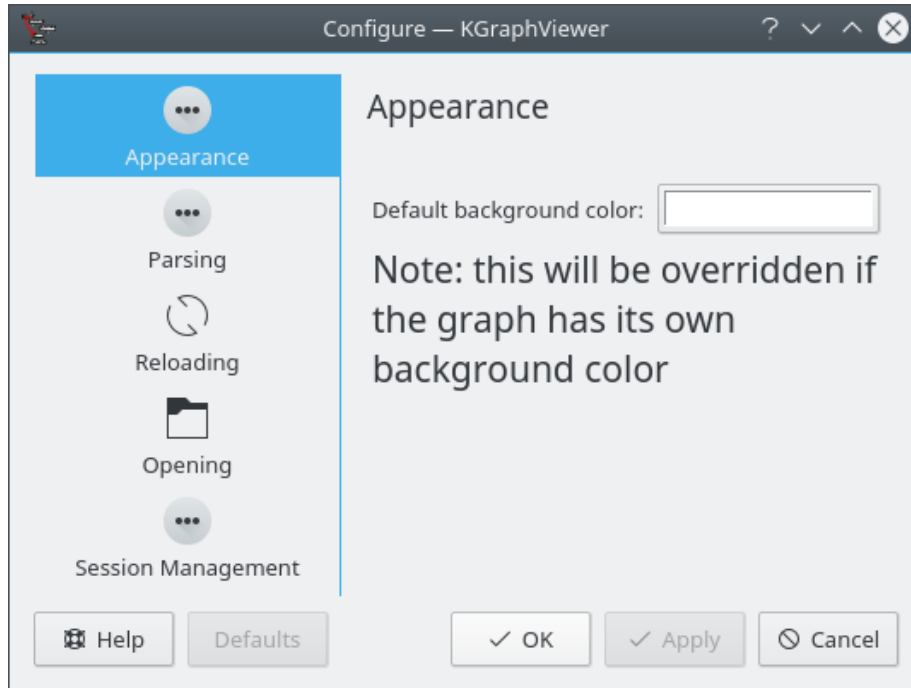
You can also change the font, headers and footers:



Page preview setup to fit on multiple pages without header and a different font

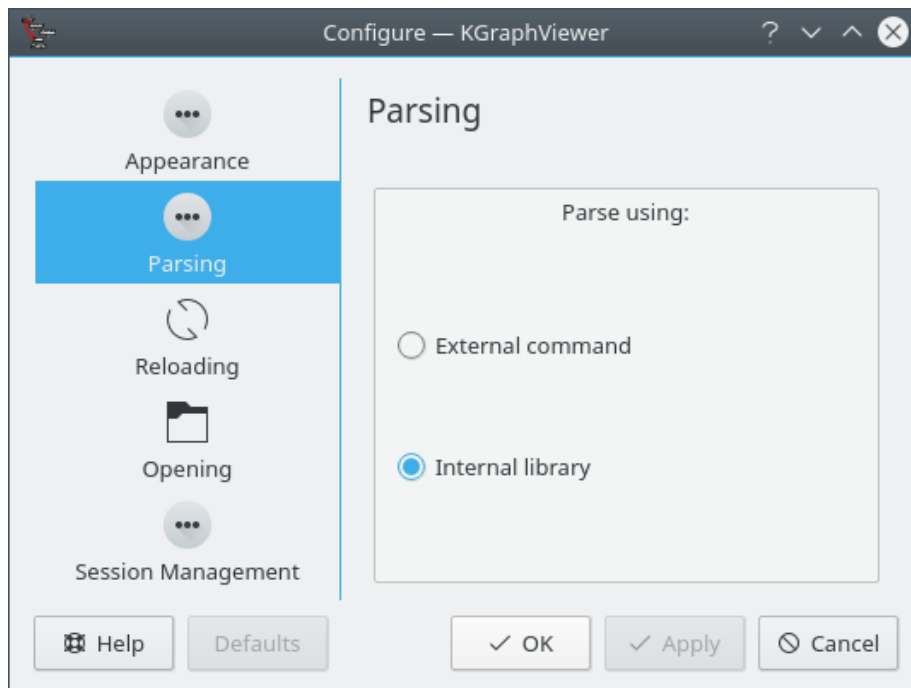
2.3 Configuring KGraphViewer

The **Appearance** page allows to change the default background color used in KGraphViewer if a graph does not have its own background color.



Page to change the appearance of a graph

The **Parsing** page allows to select an external command or the internal library for parsing Graphviz DOT files.



Page to reload files modified on disk in the configuration dialog

The KGraphViewer Handbook

The **Reload** page allows to change the behavior of KGraphViewer when a file is changed on disk by another application. You can choose a default action (reload automatically or ignore changes), or to be asked each time.



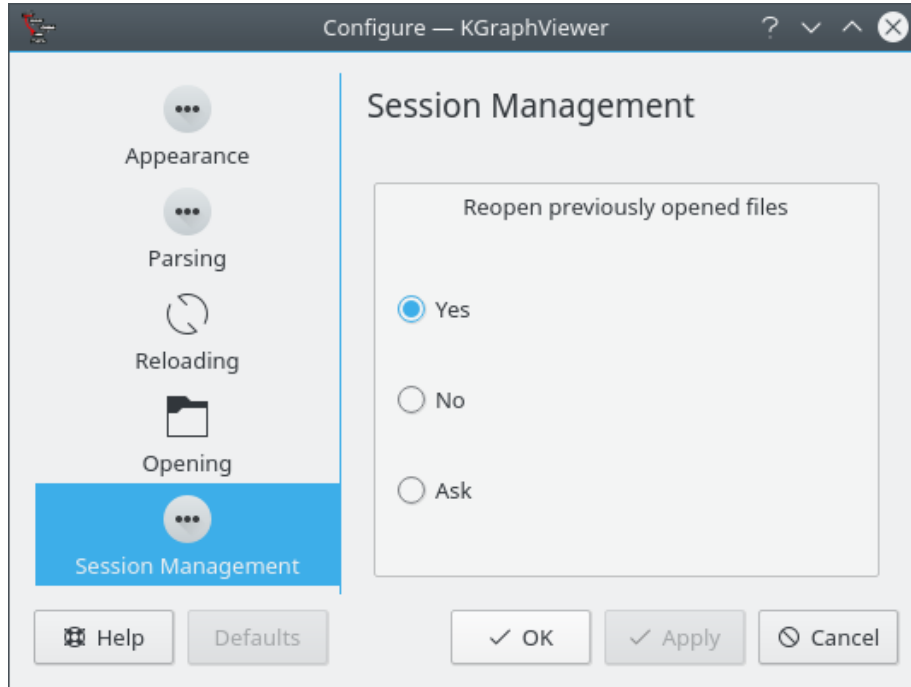
Page to reload files modified on disk in the configuration dialog

The **Opening** page allows to change the behavior of KGraphViewer when a graph file is opened from outside the current session. You can choose a default action (open in an existing window or in a new one), or to be asked each time.



Page to open new files in an existing window in the configuration dialog

The **Session Management** page allows to change the KGraphViewer's start-up behavior. You can choose a default action (reopen files of the previous session or not), or to be asked each time.



Page to reopen files of the previous session in the configuration dialog

2.4 Miscellaneous Functions

The context menu, usually obtained by clicking the right mouse button in the graph view, offers some other possibilities:

Layout

This allows to choose between various [Graphviz](#) layout algorithms or to choose your own one. The entries, as described on the [Graphviz](#) homepage, are the following:

DOT

DOT algorithm, default for directed graphs. It produces 'hierarchical' or layered drawings of directed graphs. The layout algorithm lays edges in the same direction (top to bottom, or left to right), and then attempts to avoid edge crossings and to reduce edge length.

NEATO

NEATO algorithm, default for undirected graphs. It produces 'spring model' layouts. It uses the Kamada-Kawai algorithm, which is equivalent to statistical multi-dimensional scaling.

TWOPI

TWOPI algorithm, for directed and undirected graphs. Radial layout, after Graham Wills 1997.

FDP

FDP algorithm, for undirected graphs. It produces 'spring model' layouts. It implements the Fruchterman-Reingold heuristic including a multigrid solver that handles larger graphs and clustered undirected graphs.

CIRCO

CIRCO algorithm. It produces a circular layout, after Six and Tollis 1999, Kauffman and Wiese 2002. Suitable for certain diagrams of multiple cyclic structures.

Specify layout command

Opens a dialog allowing you to specify a layout command executed by the shell. This command should accept one parameter, the graph file name, and write its layout on stdout in the xdot language. This way, you can give specific options to one of the previous commands, or apply a filter modifying your graph before running it through DOT.

Reset layout command to defaults

Determine the graph type (directed or not) and apply the default algorithm to it, currently DOT and NEATO respectively.

Export Graph

As Image...

Allows to export the full graph to all supported image formats.

Enable Bird's-eye View

When checked, the Bird's-eye view is displayed when necessary. Otherwise, it is hidden and its positioning menu is disabled.

Bird's-eye View

Configure where to place the bird's-eye view in the graph view. The possible choices are:

Top Left

Places the bird's-eye view at the top left corner;

Top Right

Places the bird's-eye view at the top right corner;

Bottom Left

Places the bird's-eye view at the bottom left corner;

Bottom Right

Places the bird's-eye view at the bottom right corner;

Automatic

Automatically choose the best position to hide as little of the graph as possible.

Chapter 3

Command Reference

3.1 The KGraphViewer Menus

3.1.1 The File Menu

File → New (Ctrl+N)

Opens a new KGraphViewer window.

File → Open (Ctrl+O)

Displays the **Open File** dialog.

File → Open Recent

Displays the list of the ten most recently opened graph files.

File → Export Graph → As Image...

Allows to export the full graph to all supported image formats.

File → Print... (Ctrl+P)

Prints the current graph using the currently set print parameters.

File → Print Preview... (Ctrl+Shift+P)

Opens the **Print Preview** window for the current graph.

File → Page Setup

Opens the **Page Setup** dialog for the current graph.

File → Quit (Ctrl+Q)

Quits KGraphViewer.

3.1.2 The View Menu

View → Zoom In (Ctrl++)

Zoom in the current graph by 10%.

View → Zoom Out (Ctrl+-)

Zoom out the current graph by 10%.

View → Redisplay

Reloads the current graph.

View → Select Layout Algo

Opens a submenu with various [Graphviz](#) layout algorithms for the current graph.

View → Enable Bird's-eye View (Ctrl+B)

When checked, the Bird's-eye view is displayed when necessary. Otherwise, it is hidden and its positioning menu is disabled.

View → Birds-eye View

Configure where to place the bird-eye view in the graph view. Choices are:

Top Left

Places the bird's-eye view at the top left corner;

Top Right

Places the bird's-eye view at the top right corner;

Bottom Left

Places the bird's-eye view at the bottom left corner;

Bottom Right

Places the bird's-eye view at the bottom right corner;

Automatic

Automatically choose the best position to hide as little of the graph as possible.

3.1.3 The Settings and Help Menu

KGraphViewer has the common KDE **Settings** and **Help** menu items, for more information read the sections about the [Settings Menu](#) and [Help Menu](#) of the KDE Fundamentals.

3.2 The KGraphViewer Toolbars

3.2.1 The Main Toolbar



Shows the **Open File** dialog.

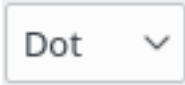


Shows the list of recently opened files.



Shows the **Print Preview** dialog.

3.2.2 The View Toolbar



Edits the layout command or shows the list of preset layout algorithms.



Zoom in the current graph by 10%.



Zoom out the current graph by 10%.

3.2.3 The Help Toolbar



Displays the **What's This** mouse pointer. Click with it on a GUI element to obtain help using it (if there is such help available).

Chapter 4

Credits and License

4.1 Program

4.1.1 Main Authors

Program copyright 2005-2006 Gaël de Chalendar kleag@free.fr

4.1.2 Contributors (to be completed)

- Reimar Döffinger Reimar.Doeffinger@stud.uni-karlsruhe.de

4.2 User Guide

Documentation copyright 2006 Gaël de Chalendar kleag@free.fr

4.2.1 Contributors

- Federico Zenith federico.zenith@member.fsf.org

4.3 Other Credits

This program contains reused code from the following Free Software projects:

KCacheGrind

Initial graph model and view classes;

Kexi

Print setup and preview classes.

4.4 Licenses

This documentation is licensed under the terms of the [GNU Free Documentation License](#).

This program is licensed under the terms of the [GNU General Public License](#).