



Contents

| | | |
|----------|--------------------------------------|----------|
| 1 | FSView - the File System View | 4 |
| 1.1 | Introduction | 4 |
| 1.2 | Visualization Features | 4 |
| 1.2.1 | Item Drawing Options | 4 |
| 1.2.2 | TreeMap Drawing Algorithms | 5 |
| 1.3 | User Interface | 5 |
| 1.4 | Credits and License | 5 |

1 FSView - the File System View

1.1 Introduction

The FSView Plugin for Konqueror is another view mode for objects of mime-type `inode/directory` for local files. It can be viewed as an alternative to the various iconviews and listviews for browsing the content of your local file system.

The unique property of FSView is its ability to show whole nested folder hierarchies using a so-called tree map for graphical visualization. Tree maps allow for displaying metrics of objects in nested structures: each object is represented by a rectangle whose area is proportional to its metric. The sum of the children's metrics must be equal to or smaller than the parent object's metric.

For FSView, sizes of files and folders are chosen as metric, where the size of a folder is defined to be the sum of the sizes of its subitems. This way, big files even deep down in the folder hierarchy can be spotted easily by looking for large rectangles. So FSView can be looked at as a graphical and interactive version of the UNIX® `du` command.

The integration of FSView as part of Konqueror allows it to use the standard features like context file menus, and mime type sensitive actions. But it was chosen not to implement automatical update of file system changes by watching: if you delete a file shown in FSView outside of Konqueror, you have to update manually by e.g. pressing `F5`. The reason for this decision is the potentially huge amount of files which would have to be watched for changes, and thus, could put an unreasonable pressure on system resources for a small feature.

1.2 Visualization Features

This section explains the graphical visualization of FSView in detail.

1.2.1 Item Drawing Options

In FSView, an item of the tree map visualization is a rectangle representing a file or folder of your file system. For easier navigation, rectangles have a meaningful coloring and labeling options.

The color of a rectangle, switchable via menu item `View → Color Mode` can be either `Depth` for easy detection of the nesting, or a color mapping of different file attributes like name, owner, group, or mime type.

A rectangle can be labeled with the various attributes of the corresponding file or folder. For an attribute, you can choose, whether it should be shown at all, only if space is available, or if space should be taken from children (thus introducing errors to pure tree map drawing constrains). Additionally, you can choose the relative location of the label in the rectangle.

1.2.2 TreeMap Drawing Algorithms

For drawing algorithms in tree maps, the rule that the area is proportional to the metric of an item should hold true. With the tree map in FSView, this goal isn't always met: we draw borders to show the nesting of items, and this border takes space for a item which is possible lost for the areas of child items. Note that less space is lost for the border if the rectangle is quadratic.

Similar, it is good for a better overview to let all rectangles at least have its name as label, taking space, too. It is an option to always force space for labels, or only to draw labels when empty space is available. In the latter case, the tool tips appearing when the mouse is kept over an item for some time, can help a lot. They show information for the item the mouse is currently over, together with its parent relations up to the root item of the tree map.

How the space of an item is split into subareas for children is left to the implementation. It is always better to try to split areas in such a way that rectangles are as quadratic as possible, both for labeling and less space lost by borders. The best methods in this regard are Rows, Columns, or Recursive Bisection.

You can choose the minimal area of items to be drawn via the menu item View → Stop at Area. Instead, a hash pattern will be drawn over the parents area to signal that this space is in fact occupied by a child.

1.3 User Interface

FSView supports multiple selection of items, similar to the iconview and listview. This allows for actions to be done simultaneously to a set of files. A simple mouse click always selects a single item below the mouse pointer. Use a mouse click in combination with pressing the **Shift** key for range selection or the **Ctrl** Key for selection toggle. Note that by selecting an item, subitems can no longer be selected. Selecting an item will also clear the selection of all its parent items before.

There is keyboard navigation available in tree maps: use **Left Arrow** and **Right Arrow** keys to move the current item between siblings, and the **Up Arrow** and **Down Arrow** keys to go up and down the nesting hierarchy. **Space** selects the item, in conjunction with **Shift** does range selection, and with the **Ctrl** key hold down simultaneously, toggles selection of the current item. Press **Return** to run the open action onto the current item.

1.4 Credits and License

Originally, FSView was meant as a small test application and usage tutorial for the TreeMap widget developed within KCachegrind.

Copyright by Josef Weidendorfer, licensed using GPL V2.

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

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