

Managing notes with Treeline

LEAF BY LEAF

Mindmaps are useful for organizing ideas and data. Treeline is a promising mindmap tool for KDE.

BY FRANK WIEDUWILT AND ANDREAS BOHLE

Treeline [1] is a mind mapping tool that helps users apply order to the chaotic task of data collection. The Python-based Treeline relies on PyQt [2].

Treeline organizes data in a tree structure. Each element (or *node*) in the tree can contain an arbitrary number of data fields of different types. There is no restriction to the number of nodes. Various sorting functions help users search the data collection. The program focuses on structured management but also provides print and export functions that let users convert their data into a variety of formats.

XML a Panacea?

Treeline uses an XML dialect for data storage. The information is stored in a file with a *.trl* extension, which is editable with any text editor. The structure for the XML data is defined in a Document Type Definition (DTD) or schema file.

Treeline uses **XSL** transformation for data conversion, creating

HTML files or bookmark files for various browsers. The Mozilla bookmark format, or the XBEL format [3] used by Galeon and Konqueror, are both available for browser-based processing.

Installation

Treeline requires the Qt library Version 3 or newer and Python [4] Version 2.3 or newer, as well as PyQt 2.4 or newer and the XML parser from the PyXML package [5]. If your system fulfills all these requirements, you can unpack the Treeline sources, which are available from the program homepage, then change to the *treeline* directory created when you unpack, become root, and give the *python install.py* command.

If you prefer to install the program in a directory other than */usr/local/treeline*, pass the required path to the installation script using the *-p* for *pathname* option.

By default, Treeline comes with an English user interface and online help. Users who need other languages should look for *treeline-i18n-0.14.1a.tar.gz* on the program homepage. Follow the steps for installing the program to set up the language packages.

After completing the install, you can launch Treeline in a terminal window by entering *treeline &*. The program comes up with a well organized interface.

The tree view, for your notes, is located on the left below the menu and toolbar. The lower right side of the window has three tabs: *Data output*, *Data editor*, and *Title list*.

If you select *Data output*, the program will display the formatted content of the

selected node at the top, with the titles of the subordinate nodes below. You can use the *Data editor* tab to edit the current node in the top half and the subordinate notes in the bottom half. The *Title list* tab restricts the view to only the node titles.

Within the tree view, you can click and drag the mouse to move individual nodes (apart from the main element) to any position in the tree. To insert a new element in front of the current element, press the keyboard shortcut [Ctrl] + [B]. [Ctrl] + [I] inserts a new element after the current element.

To start a new element below the current node, press [Ctrl] + [A]. Treeline automatically highlights the new node so that you can enter a name.

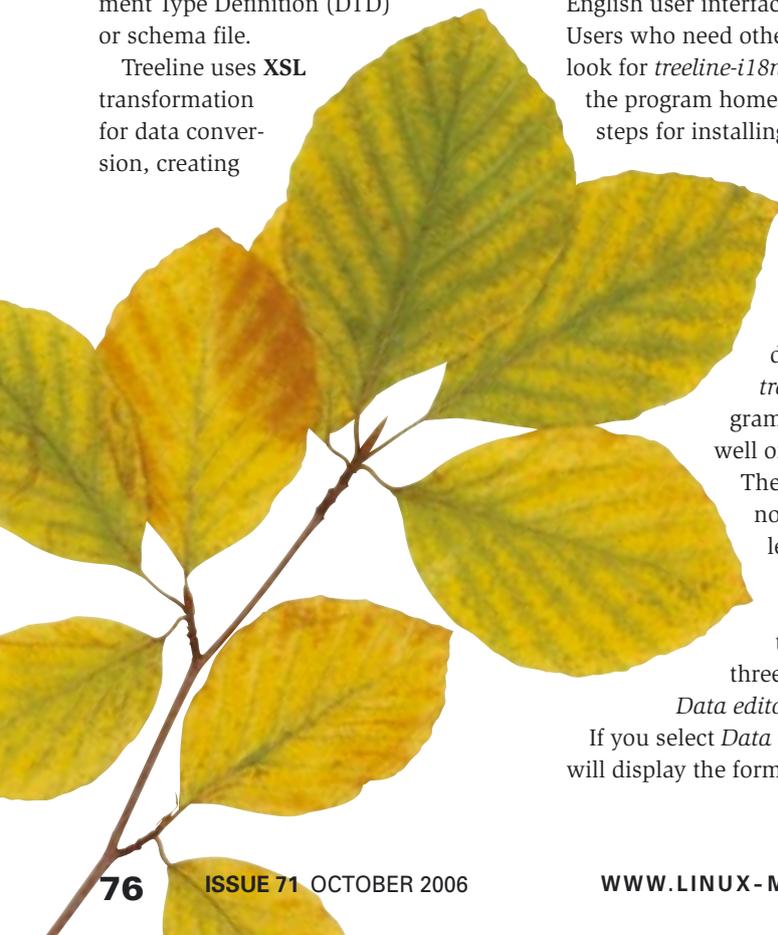
Data Input

When you launch the program for the first time, or when you create a new file, the collection will contain a single node or *Main element*. The node expects just one piece of information, a name. You can modify the name, just like the content of any element, by selecting the *Data Editor* tab on the right of the window. A text box appears; you can enter a name for the (main) element.

Treeline's main strength is its versatile storage ability. Nodes not only provide simple text elements, but data types for notes, literary references, or interesting websites. To insert an element of this kind, select *Data | Configure data types* in the menu.

Select the data type you want to edit in the *Data type* field, or click *Modify list* to create a new type. Selecting *Change icon* takes you to a dialog with a collection of icons for the new data type.

The *Title format* field contains the text that Treeline will use as a label for the node. This can be any data field or a fixed string. Use the *Output format* box to specify what Treeline will display in



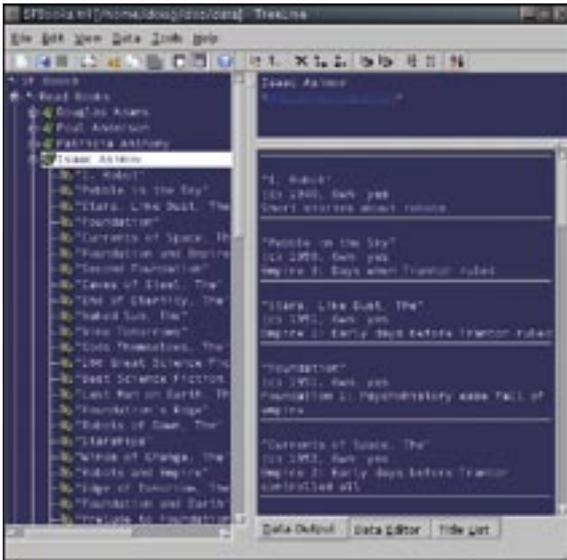


Figure 1: Treeline gives users a simple user interface.

the *Data view* tab in the main window. Again, you can use any data field; the order is arbitrary.

Use the *Fields* section to specify the data fields you require. Clicking on *New field...* opens a dialog where you can enter the field names. You can then select *Field type...*

Select the required type in the *Field type* list. Table 1 provides a list of field types supported by Treeline. In the *Extra text* box, you can specify if the text should appear before (*Prefix*) or after (*Suffix*) the field content.

Output format for date fields lets you specify the date format; for number fields, you can specify the number of decimal places and zero prefixes. *Text format* for the *Text* field type lets you specify if you prefer clear text or simplified HTML.

If you choose simplified HTML, you can use HTML tags to format the field content, and organize content with the help of HTML lists. Unfortunately, neither the interface nor the document tell you which tags are supported. Tags for bold type, underlining, line breaks, or lines are common.

Although you can enter block elements, such as `<p>` or `<div>`, and although Treeline will display the content, the program just ignores the underlying functions and does not warn you that it will do so.

Searching and Finding

To search your data, select *Tools | Search* in the menu. You can then input one or

multiple search keys in the *Search* dialog, and Treeline will search your data. The *Data | Filter data* menu item lets you define simple filtering rules for specific data types. This tells Treeline to display nodes whose content matches the filter rules, thus keeping large collections of data manageable.

Export and Print

The *File | Export* menu takes you to the export dialog, where you can select the required output format. As of this writing, the selection of import and ex-

port filters is fairly restricted.

You can use *Export type* to specify the format in which Treeline will export your data. The most interesting aspect of this option is the support for HTML book-mark formats, which gives users the ability to present their data on their own homepages. *Export preferences* lets you specify which elements Treeline will export; you can also specify whether you want to export the header and footer lines.

If you have a large collection of data, you might prefer to distribute the output over multiple HTML files. To do this, the program creates directories named after the first data field in the corresponding nodes. Thus, you should make sure that the field values at a specific level are unique and only contain characters that can be used as directory names.

To present information, Treeline supports printing of all, or selected, nodes. The menu item *File | Print options...* takes you to the print dialog. By default, Treeline will print all the elements below a selected node. The *Include root node* option specifies whether or not to print the selected element.

Conclusions

Treeline can help users organize information on a specific subject. The ability to define various data types ensures a basic structure for the data. Search and sort functions help users analyze the material they have collected. Treeline's online help is a useful aid if you get stuck.

Treeline does come with some problems. For example, you will not find a description of the data structure for the file format. The Treeline user interface seems cluttered and less than intuitive. For example, the program gives users the ability to define additional root nodes, but this does not mean that the nodes will be inserted at the root level, where only one node is supported. Some dialogs are overloaded with options and elements. The pre-defined field types are a mixture of genuine field types and controls, such as the choice field type. This shows that the software lacks a stringent design.

With respect to data type definitions, it looks like Treeline is headed in the right direction. This feature gives users the ability to create more complex elements, to verify input against conditions, and to specify defaults.

All in all, Treeline is a useful mind-mapping tool, and the implementation is fine for home use. Thanks to open source licensing, there is nothing to stop Python programmers from adding the features the current version of the program lacks. ■

Table 1: Pre-defined Field Types

Field Type	Description
Text	Text of any length
Number	Numbers of any size
Choice	Choice of a series of values
Combination	Multiple choice
Date	Date values in any format
Time	Time values in any format
Boolean Variable	Yes/No
URL	URL
Path	Path to a file
Internal link	Reference to another element
Execute link	Link to a program
Email address	Mail address
Picture	Path to an image file

INFO

- [1] Treeline: <http://bellz.org/treeline/>
- [2] PyQt: <http://www.riverbankcomputing.co.uk>
- [3] XBEL: <http://pyxml.sourceforge.net/topics/xbel/>
- [4] Python: <http://www.python.org>
- [5] PyXML: <http://pyxml.sourceforge.net>