Celsius Library System
Version 3.0.3

(w) by Christian Sämann, available under GPLv3

This is the manual to the Celsius[1] Library System, an open source collections manager. Celsius is highly adaptable to everyones needs: It can be used to classify, automatically sort and tag arbitrary electronic documents. In particular, it comes with all the features of a bibliography manager. Celsius can also manage your movie or music collections, automatically downloading relevant metadata from the internet. Even your favourite list of geocaches can be imported into Celsius, after which you can group, tag, sort and then re-export them.

[1]The library shown on the splash screen is called Celsus Library, but I prefer this slight change in name.

Contents

1. Introduction

Celsius originated from a collection of scripts which I started writing back in 2004 to manage my electronic copies of scientific papers downloaded from `www.arxiv.org`: They would look for the arXiv number in a postscript file, download metadata from the internet and rename the papers according to their title and authors. Since more and more people have shown interest in this software, I decided to extend these scripts into something more professional and to compile a proper distribution. This resulted in Celsius 1, which was simultaneously a document and bibliography manager. It has always been a speciality of Celsius to sort documents according to predefined rules into user-specified categories.

The main improvements in Celsius 2 were its extensibility via plugins and many additional features, turning it into a comfortable bibliography manager. Also the graphical user interface (GUI) was redesigned to bring as many features as possible within a few mouse clicks.

As I realized that it might be handy to have Celsius also manage my eBooks collections, my sheet music files, my movies, video and music files, I decided to extend the flexibility of Celsius. This led to the development of Celsius 3. A second concern was to make it look more professional and simpler to use by grouping similar features together and streamlining the GUI.

Celsius 3 is still strongest in handling scientific papers and has all the standard features of a bibliography manager. In particular, it comes with a number of plugins designed to help with papers in high-energy and mathematical physics, i.e. papers form the arXiv or those in the database of SLAC Spires.

Celsius 3 can also manage all your eBooks, and it comes with a plugin to import metadata from Amazon. In case you'd like to manage your physical books, entering their barcode number is sufficient and Celsius will download both metadata and a thumbnail.

A more exotic feature is Celsius' template and import function for Geocaches. It seems that GSAK, a very popular software amongst geocachers, is only available for Windows. Celsius tries to provide the basic features needed for a geocache manager: you can import geocaches, put them into various categories and sort them and re-combine them into gpx files.

Before going into further details, here are the main features of Celsius:

- ▷ Celsius manages databases ("libraries") containing arbitrary items, possibly linked to one or more electronic documents. When adding new items, Celsius tries to find as much metadata as possible using plugins, which can be easily written yourself.
- ▷ Celsius comes with templates for collections of scientific papers, eBooks, sheet music, movies and music files. Each template uses certain plugins to access metadata from the internet.
- ▷ There are many standard operations for various types of libraries (e.g., creating of a BibTeX-files or sending documents by email).
- ▷ Each library template comes with many search options.
- ▷ One can specify a hierarchy of categories/folders, into which Celsius will sort automatically all documents according to easily adjustable rules. The papers can also be sorted manually via drag and drop.

▷ For scientists working in high-energy physics, there is a large collection of helpful plugins available, which help with obtaining and updating metadata automatically.

Most recent versions, updates and help can be found on the project's homepage at

`http://celsiusls.sf.net` .

This site also hosts a forum, where you can submit suggesstions for improvements, feature requests, bug reports and questions about Celsius.

Personally, I use the system heavily in my everyday work, and I found it to be quite useful. (The scientific library on my laptop, e.g., consists currently already of about 10000 papers, preprints and lecture notes with 750,000 pages sorted in about 150 different categories.)

Note that Celsius v3.0 can read libraries from Celsius v2.1. Libraries from older versions of Celsius, however, should first be converted to Celsius v2.1 using the converter contained in that version.

Legal stuff

Acknowledgements

Many thanks to everyone who made suggestions for improvements, in particular to Kersti Anear, Dan McNamee, Sean Murray, Marco Krohn and Martin Wolf. Some of the icons used are taken from the silk and silkcompanion iconsets by famfamfam.

2. Installation

2.1. First steps

Celsius requires JAVA 1.6 to run, which is available for all modern operating systems. To test whether you have JAVA installed try typing the the command `java -version` in a terminal/command line editor. If the output does not say that you are running JAVA 1.6 (or greater), just download it from `http://java.com/download`.

You should unpack the zip-archive, preserving its folder structure; if necessary, you can adjust the folder structure later on. The standard folders have the following meaning: all the configuration files are located in the root folder, `plugins` contains all the plugins and the plugin configuration file, `Libraries` contains the sample library included in the distribution and `Icons` contains an extensible set of icons. The folder `toinclude` is empty at first; its purpose is that you put all the documents you'd like to add in this folder. You can, however, choose any other folder for this purpose.

Celsius can be started by either double clicking on `Celsius.jar` or by typing `java -jar Celsius.jar` in the command line after changing to the installation folder.

2.2. Celsius on various operating systems

Linux. Celsius should run on linux almost out of the box, if a recent java runtime environment is installed. To handle pdf, postscript and djvu-files, make sure the packages "poppler-tools" (for `pdftotext`), "ghostscript" (for `ps2ascii`) and "djvulibre" (for `djvused`) are installed. If you're not using KDE, you might want to adjust minor settings in `file support` and `email settings`.

MacOS. Celsius runs fine with a few adjustments. To get started, install the xpdf-tools from foolabs. A compiled version is available at `http://users.phg-online.de/tk/MOSXS/xpdf-tools-3.dmg`. To handle djvu files, install the djvulibre application available from sourceforge. Once this is done, add in the configuration dialogue under `file support`, `djvu` the string `/Applications/DjView.app/Contents/bin` in front of `djview` and `djvused`. Remember to click on `Apply Changes` before closing the dialogue window. For more details on configuring auxiliary programs, see the next subsection.

Windows. Celsius runs fine with a few adjustments. Install the Windows version of the xpdf-tools from foolabs and make sure the executables are in the path. A compiled version is available at `http://www.foolabs.com/xpdf/download.html`. To handle djvu files, install the djvulibre application available from sourceforge. For more details on configuring auxiliary programs, see the next subsection.

2.3. Setting up the auxiliary programs: details

If you would like to work with libraries handling files, you have to set up the proper file viewers and text extractors for the file types you want to work with. Most of the time, the viewers will work with the included configuration file as it uses the system settings. However, the text extractors have to be set up manually. On a standard Linux/Unix machine, the included configuration file should work fine. On Windows, you can easily adjust the settings using the configuration dialog.

To tell Celsius how to handle various file types, go to `File`→`Configuration` and choose the tab `File support`. For every file type, it is necessary to specify commands for viewing and extracting texts. Optionally, one can specify further commands like alternative viewers or starting up viewers with different options. Note that the command for the document viewers should end with something like `%from%`, where `%from%` will be substituted by the full file name of the document. Similarly, the command for the text extractor should contain both `%from%` and `%to%`. (The text extractor should place an end-of-page character at the end of each extracted page. Otherwise, Celsius cannot distinguish between the abstract and the body of a paper.) On Windows computers, it might help to place hyphens around `%from%` etc. On Linux machines, file names containing a space character can create some trouble. Note that Celsius allows you to redirect the output of a program by adding `>%to%`. On Apple computers, the `open` command can handle almost all the file types and under KDE 4, `okular` is capable of reading every document format one might encounter.

If you press the `Standard` button, then the standard viewer of the operating system will be used.

To save the configuration, click on `Apply Changes`, to test the commands, click on the corresponding buttons labelled `Test`. If everything works, you are ready to go. (You do not have to worry about gzipped files, Celsius will always unzip them first, before trying to extract text.)

For Windows, you can try the enclosed example file. Moreover, I recommend to use the following text extraction software: Postscript is most easily extracted with Ghostscript (see the example). For DjVu, I use the DjVuLibre-library and the program djvused. PDF files can be extracted, e.g., by the pdftotxt software by Glyph & Cog, LLC. (The available JAVA-classes did not perform equivalently, this is why they are not included.)

2.4. Log-file

To help with writing a bug report and to track the actions of the program, there is the logging file `output.txt` in the base directory, which should be deleted from time to time (`File`→`Delete logging file`) to keep it from growing annoyingly big.

The logging file contains quite a lot of information, so if anything should not work properly, it can tell you, what is actually going wrong. Always attach the log-file (or the portion since the last call of Celsius) when submitting reports of bugs with an unclear source. Also, it might be helpful to run Celsius from a command prompt, as every error will yield a trace of the stack printed to the standard output port.

2.5. Remarks on the way Celsius stores information

Celsius stores all the data using a rather simple but fast XML engine, which creates plain text files. These files in turn can be easily modified by hand (after which, however, the `Synchronize library` command of the menu `Tools` should be invoked, to verify the integrity of the modified information). To be more precise, there is a big file, `libraryindex.xml` in the base directory of every library, which contains the index for this library. For every document, there is (at least) a second file called `<id>.xml` in the subdirectory `information/`, which is also linked in the index under the tag `addinfo`. In the following, we will refer to this file as well as to the information it contains as `metadata`. While the index is completely loaded when the library is (and thus changes are only visible in the file if the library is saved), the metadata for a document is loaded whenever a document is selected and changes are saved immediately.

3. Getting started

After starting Celsius for the first time, you should create a library to which you can then add your items.

3.1. Libraries

Libraries are the databases Celsius stores items and files in. Usually, you will have one library containing your scientific papers and one containing your music files. For each library, you

can specify the way items appear in the table, what information on items will be displayed, which plugins will be used and many further functions.

You can change the properties of a library by selecting the menu `Libraries→Edit library properties`. You can give a library an arbitary `Name`. The fields listed in `Index` will be quickly accessible and always kept in memory. Celsius will automatically create the fields in the index and those listed in `Standardfields` for each item. In `Tablecolumns`, you specify the fields which you want to be displayed in the table, while `Columnheaders` gives the title of the corresponding columns. The `Columntypes` help Celsius in determining how to deal with the contents of the various columns. The `Columnsizes` determine the widths of the various columns in the table. Here, positive numbers will be taken as relative widths, while negative numbers are fixed widths. The column specified in `Autosortcolumn` is the one that will automatically be sorted when a table is displayed. If this column is set to 1000, then they are sorted according to the string `Item-Sort- Representation`. All fields containing lists of persons should be given in `People`, as Celsius will then allow for search this. The fields `Plugins-Manual, Plugins-Auto` and `Plugins-Export` contain lists of the corresponding plugins used in the library. The compatible file types are given in `Filetypes`. The fields which should be explicitly searchable are listed in `Searchtags`. The fields which have to be present in each item (besides the id) should be given in `Essential-Fields`, the ones that are checked for distinguishing items are given in `Differentiating-Fields`. Under `Item-Representation`, one can specify, how an item is displayed in dialogs and `Item-Sort-Representation` gives a rule for sorting items. In both these fields, only fields from the index should be used. The naming convention for files is specified in `Naming-Convention`. Fields which can only take a limit number of values can be listed in `Choice-Fields`, followed by a `:` and by a comma-separated list of possible values. The fields which are displayed as icons instead of texts are given in `Icon-Fields`, and `Icon-Dictionary` contains a translation from the values to the actual icon file names. The `Default-Add-Method` specifies if items are automatically moved to the libraries `Item-Folder`, or if they should be kept at their location.

3.2. Viewing documents

Just select either a category (e.g. `String theory`), seach for an author (e.g. `Saemann`) or just perform a general search with an empty search string. In the table in the middle, you should see the corresponding documents appear. A double click on any of the entries in the table will open the document if the corresponding viewers are set up properly; alternatively you can select `View selected document` from the `Documents` menu, press ALT-V or select the link in the HTML window below the table. Further additional information on a document is also provided in the lower information pane. The related features should be straightforward to use.

3.3. Searching for documents

The easiest way of accessing documents is to select the category in which they are registered. A list of all documents not registered in any category is produced when selecting `Library`, the root of the category tree.

Most of the time, however, searching a library will be the most effective method of accessing documents. First, one can search for documents by authors. When typing letters into the `Search authors` text field, Celsius shows a list of matching authors. If an author is selected in this way, the additional information on the author contained in `authorremarks.xml` is displayed.

Second, one can look for identifiers by typing letters into the `Search identifiers` text field. Celsius will look for the entered strings in the `identifier` tags of the documents.

Third, one can search for keywords (which Celsius will automatically extract, e.g., from pdf files).

Finally, one can perform a more detailed search using the `General Search` text field. Most of the functions are obvious: selecting `in Keywords` tells Celsius to look also in the `keywords` tag of the document, etc. The search is performed as you type in letters, however, if the search is very time consuming (as is the case if `in Plain Text` is selected), you have to start the search manually. The `Date added` option refers to the timestamp of the document's metadata-file, and thus also shows documents modified within this period. The `Date-tag` textfield allows you to search for documents by date (documents which have no `date`-tag are given a random one dating back to 1977). Please note that the format for entering a date is YYYY-MM-DD. If you selected `doc ref` from the filetypes combo box, then only document references, i.e. entries without an associated file, will be shown.

If you leave the search string in `General Search` blank, all documents matching the other search criteria will be shown.

You can similarly search for categories in the category tree using the `Search in Categories` text field.

3.4. Adding new documents

There are four ways, one can add new documents to the library. Either one adds a single document, all the documents in a certain folder, a document reference or a BibTeX record. All these options are found in the menu `Libraries`. The first two options add actual electronic documents to the library, while the second two just add the bibliographic information as a record. For the first two options, it is therefore important that the text extraction software has been set up properly.

In the first two cases, the following happens: After selecting a certain file or directory, Celsius will first adjust the filetypes and gzip any postscript file which is not already gzipped. Celsius will then continue with extracting plain text, and obtaining metadata by calling the installed automatic plugins in the order they are listed in their configuration dialog. A copy of the plain text is created in the same directory as `[filename].txt.gz`.

All this is done in parallel, using threads; the maximum number[2] of threads is, however, limited. A text field indicating the number of running threads is found in the upper right corner of the dialog window.

You can also add additional tags to a document and apply plugins manually.

The buttons all work in an obvious way; note that the button `Add all recognized files` adds automatically all files which have the tag `recognition` associated with the value

[2]This number can be specified in the file `configuration.xml`.

100.

If the checkbox `Leave file at current location` is not checked (the default setting), the file will be moved to the `documents/` folder of the current library's base folder. Although this option should work fine, it is highly recommended to let Celsius move the documents to the document folder of the corresponding library. Documents are not lost and can easily be exported to other folders or send via email to other people.

Even if a document is not available as an electronic copy, it can be helpful to have its data cataloged, e.g. for creating a BibTeX-file and accessing its SLAC-page easily or just for keeping an overview over one's private library. For this purpose, one can add a document reference or a bibtex record using the corresponding menu in `Libraries`. A dialog will open and ask for the essential information, and the document reference will be added. Providing a barcode is actually enough, as the rest of the information will be completed automatically.

3.5. Adding document types

Document types are simply added or removed by adding or removing png files of size 16x16 from the doctypes subfolder.

3.6. Sorting documents

The documents are automatically sorted according to the current rules when being added to a library. Via `Registration`→`Start autoregistration`, one can have Celsius sort all the documents in the library later on, e.g. after adjusting the registration rules. A third possibility is to manually drag and drop documents from the table in the middle into the categories on the left. Most of the related functions in the menu `Registration` are self-explicatory, it is just important to understand that there are two ways in which a document can belong to a category: by "autoregistration" or by (manual) registration. One can move "autoregistrations" to "registrations", e.g., by `Registration`→`Fix registration of currently shown documents`.

3.7. Defining the category tree

To insert and move a category in the tree on the left, one uses the functions in the menu `Categories` in a straightforward manner. Contrary to Celsius v1.0, the name of a category is also its ID, so you should not use the same name twice, unless you want to have some overlap. So instead of naming a category "Books", name it "Books (QFT)".

Also, renaming a category just changes its name. You will be asked whether the items currently listed in this category should be adjusted accordingly.

4. The main window and the menus

4.1. The tabs of the information panel

Note that some of the tabs might be hidden, depending on the selected library template.

Info tab: This tab displayes information concerning the current item, the currently selected category, author, or keyword, the current table of items or the last search results. The information displayed can be adjusted by editing the HTML templates. If you select an author in this tab and choose `Go to selected author` from its popup menu, the selected author and his documents will be displayed. If you drag and drop an image onto the info tab, a copy of it will be associated with the current item and registered under the tag `thumbnail`.
Bibliography tab: In this tab, one can display and edit bibliographic information. The information itself is stored in the BibTeX-format, the output is returned by the currently selected export filter. The buttons for editing the BibTeX record are self-explaining.
Links tab: Here, items to which the current items has a link are displayed, sorted by type of link. For example, when using a library created from the template scientific papers, the links correspond to papers citing the current item and those appearing in the bibliography of the current item. In a library created from the template GeoCaches, the links contain additional waypoints.
Remarks tab: This tab allows you to add remarks to items which will be written to the tag `remarks`.
Search tab: Here, plain text passages of the current document at which the search terms appear are displayed. This is helpful for examining the context in which a term searched for appears without actually opening the document. Also, the text of the first page of the current document can be displayed.
Thumbnail tab: This tab allows you to manage a thumbnail associated with the current document.
Data tab: This tab gives direct access to the way information is stored in Celsius. Note that tags which refer to data which contains line breaks (as the abstract or the BibTeX record) are not shown. The left hand side shows the information contained in the library's main index, while the right hand side shows all data contained in the metadata-file. One can use this tab to modify the records manually and even add new tags to the records. Note, however, that line breaks are not allowed and that information contained in both the library's index and the metadata should be edited in the right panel and such information will always be synchronized automatically.

4.2. Using HTML templates for the info tab

The display in the information tab is fully adjustable to ones needs for each library by specifying templates. In such a template, you can use plain HTML code (within the boundaries of a JAVA 1.6 JEditorPane). All tags found between two sharps are automatically replaced by the corresponding entries of the main library index or the additional information associated with the item. For example, in

```
<p>#authors#<br>#title#</p>
```

the authors and the title of the document will be inserted at the designated positions. To format tags which contain names, try the following extensions:

```
#authors&1#<br>#authors&2#<br>#authors&3#<br>#authors&4#<br>
```

Fields that contain lists separated by a pipe can be displayed as lists as follows:

```
#|references#
```

If an item contains longituted/latitude information, one can format this as

```
#lon&#<br>#lat&#
```

The distance and the heading compared to the current location are displayed via

```
#distance&#<br>#heading&#
```

If you would like to display the value of an icon-field as an icon, use, e.g. for the field `type`:

```
<img src='file:#type&id#'>
```

Moreover, fields contained in the BibTeX record can be accessed as

```
Publication year: #bibtex.year#
```

There is a further possibility of controlling the output: If a line starts with `#if#tag#`, it will only be included, if the value of the tag is not empty. For example,

```
#if#remarks# <p>#remarks#</p>
```

will only show the tag "remarks", if there is any information there to be displayed. Similarly, adding a ! in front of the tag will display the line only if the tag is *not* set, e.g.:

```
#if#!remarks# <p>No remark available.</p>
```

One can test if a journal link is available as follows:

```
#if#journallink# <p>Journal link available.</p>
```

Arbitrary external links can be included and a click will open the viewer for HTML, as e.g.

```
#if#bibtex.doi#<a href='http://dx.doi.org/#bibtex.doi#'>doi-link</a>
```

Celsius knows a number of special commands:

```
<a href='http://$$view'>#filetype#</a>
```

displays an HTML link, which, when clicked, opens the main file associated with the current item. If there is no file associated, Celsius first tries to open a journal link, then it will try to open a webpage specified under the tag `url`. If the item has the tag `combine` specified, Celsius will open a new tab with the matching items.

```
<a href='http://$$view-alt-1'>alternate version</a>
```

displays an HTML link which opens the alternative version number 1.

```
<a href='http://cid-000000'>Item with Celsius ID 000000</a>
```

displays an HTML link which opens item with Celsius ID 000000.

`#listsimilar#`

inserts a list of similar items according to the combine rule of the current item.

`#$$links#`

inserts a link that will open all the links in a new tag.

`#$$links-references#`

inserts a link that will open all links of type references in a new tag.

`#$$linksList#`

inserts a list of all the links.

`#$$linksList-references#`

inserts a list of all links of type references.

`#filesize#`

gives the size of the file associated with the item

`#altversions#`

list all the secondary files associated with the current item.

When displaying the contents of certain categories or search results, the insertions `#currentdocuments#`, `#currentpages#` and `#currentduration#` will display the number of items, pages or the total duration of the currently displayed in the library.

Detailed example files demonstrating many of these features are included in the library templates.

4.3. Defining menu shortcuts

The shortcut keys for Celsius are defined in the file `celsius.shortcuts` in the base folder of the program. This file can be direclty accessed unter `File`→`Configuration`→`General Configuration`→`Edit ShortCuts`. A defining line contains the character sequence `::`, separating a String giving the caption of a menu from the hotkey definition. The hotkeys are defined using the standard format for keystrokes in JAVA. More precisely, the string following `::` is parsed by the `java.awt.AWTKeyStroke.getAWTKeyStroke(String s)` command. Defining menu shortcuts should be obvious from the definitions included in the distribution.

4.4. Defining journal commands

To perform certain commands for papers in certain journals (like e.g. to open the corresponding webpage), you can specify commands in Celsius' configuration Dialog. You can then use the `#journallink#` command in the HTML template to create a link to this command, see the example HTML template.

4.5. Tools

In the `Tools` menu, there are the following commands:
Synchronize library. If Celsius should ever crash or you forgot to save the library before leaving Celsius, then the documents you added are not gone. To check for documents, which are included in the library but do not appear on the index and inversely, to check that all the files mentioned in the index and in the documents' metadata are indeed present, you can use this tool.
Setup USB device/Synchronize with USB device. This allows you to synchronize a folder on an external USB device with the files associated to the documents registered in certain categories. This is helpful in copying files, e.g. on an eBook reader or an mp3-player.
Edit my location/Set my location... These menues are used to specify your location for libraries containing information on geolocation, as e.g. libraries created from the template GeoCaching.
Edit library templates. This function allows you to directly add library templates and to create a template from the currently selected library.

4.6. Bibliography

The first three menues copy various bibliography related information to the clipboard, which can then be used in text editors or word processing software.
Show all papers cited in a tex file. Goes through a specified tex file and displays all the papers in the current library which are cited in this paper in the order they appear in. This is helpful for creating a bibliography without using BibTeX: For this, one selects all the displayed papers and chooses `Bibliography→Bibliographic record for selected documents to clipboard`. The contents of the clipboard can then be inserted into any LaTeX file as the bibliography.
Create BibTeX of current library. Creates a single file with all the BibTeX records available in the current library.
Check BibTeX integrity. Sometimes, BibTeX information obtained from the internet is corrupt. To easily correct this, Celsius allows you to search for those files with a corrupt BibTeX record.
Create literature list. If a category is selected, Celsius will automatically create a literature list of all the contained items as well as the subcategories and the items those contain using the currently selected bibliography plugin.

4.7. Hiding unwanted menus and tabs

To hide menus and tabs which do not apply for a certain library, one can use the `hide` field in the library properties. Currently, Celsius supports the following entries:

```
Menu:Bibliograpy
MenuItem:Set my location to current item's location
Tab:Bibliography|Tab:Search|Tab:Thumbnail
```

This will hide the Bibliography tab, the Bibliography menu, the Search tab (which is only used for searching the full text of an item) and the thumbnail tab.

5. Registration rules

One of the strengths of Celsius is that you can define certain rules, which will automatically sort newly added items into categories. You might, for example, want to collect all the scientific paper, which refer to a certain publication, in one category to have an overview over a certain subfield of your research.

Note that the syntax for registration rules changed from Celsius 2.1 to Celsius 2.2 to allow for more flexibility.

In one of the tabs under the category tree, the rules having an effect on the currently selected category will be displayed. If `Library` is selected, all the rules will be shown. Rules are edited by opening the Registrations rule editor. Here, one directly edits the `rules.xml` file belongin to the library.

There are two types of registration rules: testing fields and more complex testing, as explained below. Note that the order of the fields specified in the XML-entries are always irrelevant. Note furthermore, that if the field "parse" of the item is set to "none", the registration rules won't be applied.

5.1. Testing single fields

The following line checks if the field author has the value "Witten, Edward". If so, it registers it in the categories "interesting" and "cool":

```
<equals tag="title" value="Witten,Edward" target="interesting|cool"/>
```

To test, whether a tag contains a certain term, you can use the following rule:

```
<contains tag="title" target="Relativity" value="relativ"/>
```

This rule registers the item in the category "Relativity", if its title contains the term "relativ". The following line does the same, but the term "relativ" has to make up 2/3rds of the title:

```
<contains-mainly tag="title" target="Relativity" value="relativ"/>
```

To register items, in whose plain text appears the string "hep-th/0611108", use

```
<contains tag="plaintext" target="BLG theory" value="hep-th/0611108"/>
```

This command will collect all papers which cite the paper with this arXiv identified. Note that scanning of the plain text can be forbidden by putting the field "parse" of the item to the value "header" or "none".

5.2. More complex testing

To construct more sophisticated tests, you can use the `test` rule. Consider the following example:

```
<test>
   <contains tag="title" action="a+=2" value="flag"/>
   <contains tag="title" action="a+=2" value="quant"/>
   <contains tag="title" action="a+=2" value="fuzzy"/>
   <eachtime tag="plaintext" value="Fuzzy Flag" actions="a+=1"/>
   <if conditions="a>3" target="Fuzzy Flag Varieties"/>
</test>
```

The rules are essentially the same as when testing a single field, however, instead of specifying a target, an action is given. An action consists out of a variable name (one character only, here 'a'), an operation ('=' puts the variable to a value, while '+=' and '-=' increase and decrease its value by the given quantity) and the actual value. The if-condition then can test a specified condition, and, if fulfilled, registers the item in the categories specified by target. A condition consists again out of a variable name, a comparator ('=', '¿' or '¡'), and a value. Variables, which were not modified, are automatically assumed to have value 0. The only new rule here is `eachtime`, which executes the specified action each time a match is found in the given field.

6. Special functions

6.1. Combiners

to be completed...

6.2. Java look and feel

The java look and feel can be specified in the configuration dialog. The standard cross-platform look and feel is set to "Nimbus". Note that Celsius needs to be restarted for changes in the look and feel to take full effect.

7. Plugins

Plugins for Celsius are tiny java programs which come in two flavors: information plugins, which are used to retrieve additional information from the internet or the plain text of a document and bibliographic plugins, which translate the bibliographic information to a desired format. You can easily write and adapt the plugins yourself if you are familiar with a little JAVA or C++. Most of the time, just studying the examples included in the distribution should provide you with enough understanding for writing your own ones. If you should have useful and working plugins others might be interested in, you can publish them on the Celsius homepage.

7.1. Installation and administration of plugins

Administration of plugins is done directly in Celsius by clicking e.g. on the `Manage` button in the main screen.

The plugins included in the distribution are all installed. To install a new plugin, place the class file in the `plugins/` subfolder. Then click on the `Add` button for the relevant category of plugin. Note that the information plugins are split into two categories: those, which are applied automatically when adding new documents, and those, which are applied manually. If necessary, parameters can be associated with every plugin in every category separately.

Note that the automatic plugins are called in the order they appear in the configuration file. This is important if one plugin depends on the result of another one (as, e.g., in the case of `PluginSpires`, which depends on the result of `PluginArXiv`).

Clicking with the right mouse button on the plugin list in the main window displays a popup-menu, which allows you to reload the plugins or to have Celsius display a short message with information on the selected plugin.

7.2. Interface of a plugin

A very basic plugin looks as follows[3]:

```
import java.util.ArrayList;
import java.util.HashMap;

public class PluginTester extends Thread {

    public static final HashMap<String,String> metaData=new HashMap<String,String>() {
        {
            put("title"             ,"Test-Plugin");
            put("author"            ,"Christian Saemann");
            put("version"           ,"1.0");
            put("help"              ,"This plugin is for testing purposes only and"+
                                     "printlns the records' tags with its values.");
            put("needsFirstPage"    ,"no");
            put("longRunTime"       ,"no");
            put("requiredFields"    ,"fullpath");
            put("type"              ,"auto|manual");
            put("defaultParameters" ,"");
            put("parameter-help"    ,"none.");
        }
    };

    public celsius.MProperties Information;
    public ArrayList<String> Msgs;
```

[3]Note that the `HashMap` which has been used in Celsiusv2.0 to store all information has been replaced by a new class, `celsius.MProperties`.

```
    public void Initialize(celsius.MProperties i, ArrayList<String> m){
        Information=i; Msgs=m;
    }

    public void run() {
        for (String key : Information.keySet()) {
            System.out.println(key+" :: "+Information.get(key));
        }
    }

}
```

The set of fields defined here is minimal and should be self-explanatory. The fields `title`, `author` and `help` are static information provided by the plugin for the user interface. The field `needsFirstPage` specifies, whether the information variable provided by Celsius should contain the key `firstpage` which is mapped to a String containing the plain text of the first page. Putting this to false increases the speed, if the plugin is, e.g., applied to all documents in a library.

The information variable provided by Celsius when initializing a plugin contains a number of standard keys mapping to information on the paper. These included `title`, `authors`, `pages` and `identifier`. The key `fullpath` is mapped to the full path of the actual document file. The key `$$params` contains parameters in an arbitrary format.

You can add arbitrary user-defined keys to the information variable; you should, however, use lowercase letters only for defining a key. If you want to refer to files, put the marker /$ before the filename add another marker /$ followed by the filetype:

```
Information.put("thumbnail","/$"+thumbfile+"/$.jpg");
```

This will enable Celsius to move the file to the information directory of the current library. The plugin `PluginAmazon`, e.g., uses this technique.

The ArrayList Msgs is a protocol array, i.e. you can add strings to this list, which will then be written to the standard loggin file `output.txt`.

For reading webpages, you may use the routines provided in `TextFile.java`, the routines of `Parser.java` might help with parsing the results.

Note that bibliographic plugins should return an additional field with the key name `output`, in which the properly formatted bibliographic record should be contained.

7.3. Special functionality

tags with $$keep- should be carried over to the next thread. metadata.get("finalize") yes: call plugin one more time with all $$keep data, but nothign else plugin gets field "$$finalize" put to yes

metadata.get("questions") line:Name for GPX data:$$keep-name—multiline:...:... won't work, if field is already set

7.4. Technical remarks on writing plugins

Obviously, you will have to install the source developement kit of JAVA, as you will need a compiler for your plugins. Note furthermore, that the plugins which are automatically started when adding a document are run in parallel, so that user interaction and output is not possible without risking deadlocks etc. The plugins which can be manually started, however, are executed sequentially, so that user interaction is permitted, see e.g. the plugin `PluginAmazon`.

7.5. The plugins included in the distribution

Adjust Audio Titles. This plugin tries to adjust the title/maintitle/fulltitle fields.
Adjust Identifier. This plugin adjust the identifiers according to bibtex entries.
Adjust Type. This plugin tries to automatically adjust the document type.
BibTeX. This plugin dumps the BibTeX field of a document
Convert ps.gz to pdf. This plugin converts a gzipped postscript file into pdf format.
Create List of Publications. This plugin creates a list of publications, using the ListOfPublications.conf configuration file.
Create List of Publications/Web. This plugin creates a list of publications, using the ListOfPublicationsWeb.conf configuration file for use online.
Create Playlist. This plugin creates a playlist for the linked files.
Create Thumbnails (epub,djvu,pdf). This plugin creates a thumbnail from the first page of the pdf file. The following programs are needed: for epub: nothing for pdf: convert (from ImageMagick) for ps.gz: convert (from ImageMagick) for djvu: ddjvu.
CropPDF/eReader. This plugin crops the pdf files located on an ebook reader. The parameter is the folder for the ebook reader.
Enumerate items. This plugin sets the number field of the selected items to consecutive numbers.
Export GPX. This plugin creates a GPX file out of the selected Caches
Extended LaTeX. This plugin creates extended LaTeX output
Get Audio Data. This plugin reads out Audio Metadata. It requires audioreader/jaudiotagger
Get from DOI. This plugin tries to identify the DOI and looks up publication information.
Get from Filename. This plugin names a file according to its filename, if this is of the form title (author).
Get from Inspire. This plugin gets information for arXiv preprints from Inspire.
Get from International Music Score Library Project. This plugin looks for an entry at IMSLP and downloads information.
Get from JSTOR. This plugin looks for an entry at JSTOR and downloads information.
Get from Mutopia Project. This plugin looks for an entry at Mutopia.org and downloads information.
Get from PDF Header. This plugin reads out the header information of a pdf-file.
Get from arXiv. This plugin looks for an arXiv identifier in the text and downloads the information belonging to this file.
Get from filename (audio). This plugin interprets a file's position in the folder hierarchy.
Get from header. This plugin looks for metadata in the plain text information of a file.

Get from header (mp3). This plugin looks for the metadata of an mp3-file.
Get journal data from Spires. This plugin looks for an entry at SLAC/Spires for records with a certain journal reference.
GetDOI. This plugin updates records from SLAC/Spires.
Look at ADSabs. This plugin looks for a record corresponding to title (and author, if specified) at adsabs.
Look at Amazon. This plugin tries to find a record at Amazon.com and downloads the thumbnail. It is also used as the barcode plugin.
Look at Amazon/Music. This plugin tries to find an album at Amazon.com and downloads the thumbnail.
Look at IMDB. This plugin looks for a record corresponding to the title at the Internet Movie Database.
Look at Inspire. This plugin looks for a record at Inspire. The search string is the string entered as "title".
Look at PhilSci Archive. This plugin looks for an entry at the PhilSci Archive and downloads information.
Look up DOI. This plugin looks up publication information from the DOI.
Normalize Journals. This plugin substitutes long journal names in BibTeX by their abbreviation. A list of journal names is found in the configuration file "journal substitutions.txt".
Repair PDF. This plugin repairs pdf files. It needs pdftk installed.
Simple reference. This plugin creates a simple literature reference
Stamp Audio Header. This plugin writes metadata into an audio file. It requires audiowriter/jaudiotagger.
Stamp PDF Header. This plugin writes title/authors information to a pdf-file. It needs pdftk installed.
Standard LaTeX. This plugin creates standard LaTeX bibliography output.
Test-Plugin. This plugin is for testing purposes only and printlns the records' tags with its values.
Update Citation Count. This plugin updates citation counts from Inspire.
Update Links. This plugin updates all links from Inspire.
Update Publication Record. This plugin updates records using Inspire.
Update Recent Publication Record. This plugin updates records of papers from the last two years using Inspire.net.
Update to Inspire. This plugin updates all records with a Spires-ID to Inspire.The parameter is the Inspire base URL, e.g. http://inspirebeta.net/.
Watch Citation Count. This plugin updates and compares citation counts from Inspire.