

Take Notes

Notes handling module

Abstract

Module for processing notes. Notes are classified according to categorie and contain information about subject, date of intake, etc. The presentation of notes can be filtered according to several criteria.

Introduction

Active email lists, as for example the ConT_EXt-list, produce many emails discussing some problem or other topics. Often these discussions form threads chaining successive postings; where each posting incorporates much content of the previous one. Collecting all the emails for later reference produces a lot of redundancy. Moreover, the continuous repetition of previous content tends to render reading a tedious business.

Clearly, better accessibility can be attained when the threads worth retaining are condensed to a single or a few successive notes describing the gist of the discussion. This idea motivated the development of this module. It serves as a means to collect, store, select and reproduce the stored notes. Usage of this module is not restricted to storing email discussions, of course. It can be applied to everything one deems worth remembering.

Overall structure

This module is dependent on three other modules: `hvdn-lua`, `hvdn-ctx` and `hvdn-xml`; these are loaded from within the main module file `hvdn-note`.

Notes can be stored in files and have the following general structure:

```
<note attributes>
  <subject> ... </subject>
  <author> ... </author>
  <short> ... </short>
  <source> ... </source>
  <text> ... </text>
  <remark> ... </remark>
</note>
```

In order to have them processed properly they should be enclosed inside an arbitrary node; which functions as for that particular tree. For example choosing `<notes>` as root:

```
<!-- Collected ConTEXt discussion notes -->
<notes>
  <note> ... </note>
  <note> ... </note>
  ...
</notes>
```

Notes are typeset with the following setup:

```
\usemodule[hvdn-note]
<takenotes attributes>
  <!-- Included from file -->
  <include file="notes-1.xml"/>
  <include file="notes-2.xml"/>
  ...
  <!-- Local -->
  <notes>
    <note> ... </note>
    ...
  </notes>
</takenotes>
```

Note structure

The data pertaining to a note are split between the attributes on the `<note>` and its child nodes. Attributes are used to classify the notes and will serve as keys in the filtering of notes during the processing stage. These attributes are

- `category` – Freely chosen designation for the topic under which the subsequent notes are categorized by the user. **Must be present on each note or a T_EX-error will result.**
- `subcategory` – A category can be refined into subcategories, making finer meshed searches possible.
- `date` – The date pertaining to the note; presumably the date on which the content was produced, or the date it was made. However, in case of a series of notes on historical events one may use the date attribute for the date of the event. Usually a date will have eight digits `yyyymmdd`. Thus `20160427` will be interpreted as `27-04-2016`, but `yymmdd` is acceptable too and designates a year in the 21th century. When date has four digits only, it will be no surprise that it will be interpreted as a year.

A missing or empty date will never match true on a filter operation. The same holds for dates given as 'May 12, 1981'; such dates are printed as encountered, but not taken into account in a search.

- `key` – A number of comma separated keywords that can be used to filter out notes of a particular interest. A missing key will always match false on a filter operation.
- `language` – Signifies the language in which the content of the note is written, the `\language`-setting adapts accordingly. Legends like 'remark' etc. however, are typeset in the language of the current document. To change this behavior see the example at the end of this article. For languages other than English, Dutch, German and French you will have to add the translations yourself; it is easy enough, see the list at the end of the source code.
- `obsolete` – Signals the obsolescence of the note, and is used to suppress output without the need to remove it from the dataset. If present, any value other than `no` makes the note obsolete.

The subnodes carry the information proper. These are:

- `<subject>` – Short description of the subject of the note.
- `<author>` – Author of the information or the writer of the note. More than one author node can be given.
- `<source>` – The source of the information. For example: email, internet, personal communication, etc.
- `<remark>` – Special remark to be made.
- `<text>` – The content proper of the note.

In the output the order in which the subnodes appear is fixed. Empty subnodes are suppressed, which makes it easier to work with a template.

Template for a note:

```
<note category=""
  subcategory=""
  date=""
  key=""
  language=""
  obsolete="">
<subject></subject>
<author></author>
<source></source>
<remark></remark>
<text></text>
</note>
```

Takenotes structure

The `<takenotes>` governs the extraction and presentation from the collection of notes. Attributes steer the selection of notes according to criteria related to the

attributes on the note. All these selection criteria may be specified independently from each other.

- `category` – When present, the value of this attribute is compared to the `category` attribute on each note. If they are equal, then that note is selected. The comparison of the attributes is done regardless of case.
- `date` – An attribute `from` carries the lowest date on the note that will be selected. Conversely `until` selects the last date included in the output. If both are present they constitute an interval. Dates must be formatted as four, six or eight digit values as was explained above.
- `key` – The content of this attribute will be matched to the comma separated list on the note. If there is a match, then this note will be selected.
- `language` – Filter notes on having explicitly a language in the given comma separated list of two-letter language codes, for example `language="en,nl"`.
- `obsolete` – Set to `yes` in order to include obsolete notes in the output.
- `list` – This attribute is kind of special. When not empty it generates a list of the filtered notes according to the selector given in the attribute. Values for the selector are: `category`, `language`, `subject`, `author`, `source`.

Other attributes on `<takenotes>` regulate the output of the subnodes. These are the same as mentioned above: `subject`, `author`, `source`, `text`, `remark`. For example, `author="off"` will suppress the output of the `<author>` subnode. All are on by default.

There are flags to influence the output. These too are off by default.

- `framed` – Put each note into a frame, thus preventing it being broken between pages. The background can be colored by setting `backgroundcolor`. Implies `nobreak`.
- `nobreak` – Prevents breakup of a note over a page boundary by placing it in a `vbox`. The `framed` option takes precedence over `nobreak`.
- `newpage` – Start a newpage at each new note.
- `numbered` – Separate notes by numbered rules.
- `verbose` – Extra information goes into TeX's log.

Finally some embellishments. The content of the notes can be given different font and style from the rest of the document. An example of the use of a different font for the text would be to typeset programcode in a monospaced font.

The following values are available for the style options: `bf`, `it`, `bi`, `bs`, `sl`, `tt`, `sc`, `rm`, `ss`, `tf`, `tfa`, `tfb`, `tfc`, `tfd`, `tfx`, `tfxx`, `ttx`, `txx`, `os`, `hw`, `cg`, `bold`, `italic`, `bolditalic`, `italicbold`, `slant`, `slanted`, `boldslanted`, `slantedbold`,

smallcaps, oldstyle, mediaeval, normal, serif, regular, roman, sans, sansserif, mono, type, teletype, handwritten, calligraphic, big, verybig, heavy, veryheavy, small, tiny, smallmono, tinymono. The values given here translate to appropriate font commands. Several font commands can be combined as for example `legendstyle="bf,os"`, but not all combinations are effective.

- `legendfont` – Font used for the items.
- `legendstyle` – Style used for the items.
- `textfont` – Font used for the text.
- `textstyle` – Style used for the text.
- `textoffset` – Space between the text node and the legends above..
- `textcolor` – Color for the note; default is black.
- `obsoletecolor` – Color for an obsoleted note; default is darkgray.
- `background` – Effective only when `framed` or `nobreak` is chosen; default is white. Used to make the notes stand out against their environment.
- `indent` – The legends are not indented but for the text it can be set; default is none.
- The legends have zero whitespace but for the text it can be set; default is none.

Example

Example of a note in the Dutch language as designated by the attribute `nl`. The attribute `use="de"` on the `<vocabulary>`, however, forces the legends in German.

<p><i>Thema: Cryptografie cursus</i> <i>Kategorie: Cryptografie/cursus</i> <i>Datum: 1-1-2018</i> <i>Schlüssel: cryptografie,cryptoanalyse</i> <i>Autor: Hans van der Meer</i> <i>Bemerkung: Diese Kurs ist in Niederländisch.</i></p> <p>Algemene inleiding tot de cryptografie en cryptoanalyse met oefeningen voor studenten.</p>	<i>Verfallen</i>
---	------------------

```
<takenotes obsolete="yes" verbose="on"
  category="cryptografie"
  from="2000-01-01" until="2099-12-31"
  legendfont="small" legendstyle="it,os"
  framed="on" textoffset="1mm">
<vocabulary use="de"/>
<notes>
<note category="Cryptografie" subcategory="cursus"
  key="cryptografie,cryptoanalyse"
  date="20180101" language="nl" obsolete="yes">
  <subject>Cryptografie cursus</subject>
  <author>Hans van der Meer</author>
  <text>
    Algemene inleiding tot de cryptografie en
    cryptoanalyse met oefeningen voor studenten.
  </text>
  <remark>
    Deze Kurs ist in Niederländisch.
  </remark>
</note>
</notes>
</takenotes>
```

Example of the output when extracting the subjects from a series of notes. Selectionfilters may be put in action while processing the data thus reducing the output to the areas of interest. Note that output related options like `framed`, style options, etc. are ineffective while producing a list.

1. Example of `xmlstrip` usage
2. Information about `xmlstrip`
3. Initialize `startuseMPgraphic` variables
4. Unresolved `xmlstrip` issue
5. Variables on `startuseMPgraphic`

```
<takenotes list="subject">
<notes>
  <note>...</note> ...
</notes>
</takenotes>
```

Hans van der Meer
H.vanderMeer@uva.nl

