mikojdoe
dcm.sty: An Infrastructure for marking up Dublin Core Metadata in \LaTeX\ documents*

Michael Kohlhase  John Doe

March 20, 2019

Contents

1 Introduction 3

2 The User Interface 3
  2.1 Package Options ................................................. 3
  2.2 The DC Metadata Block ........................................... 3
  2.3 DCM Metadata Block Styles ....................................... 4
  2.4 Configuration .................................................... 4

3 Limitations 5

4 The Implementation 5
  4.1 Package Options .................................................. 5
  4.2 The DC Metadata Block ........................................... 5
  4.3 DCM Block Styles .................................................. 7
  4.4 Dealing with ISO Dates ............................................ 8
  4.5 Configuration ..................................................... 8

*Version v0.3 (last revised 2019/03/20)
1 Introduction

The dcm package allows mark up Dublin Core Metadata [DCM03] in TeX documents so that it can be harvested by automated tools or exported to PDF¹. This package allows to attribute authorship to arbitrary text fragments.²

2 The User Interface

2.1 Package Options

`showmeta` The dcm package takes a single option: `showmeta`. If this is set, then the metadata keys are shown (see [Koh16] for details and customization options).

2.2 The DC Metadata Block

`DCmetadata` The dcm provides the environment `DCmetadata` for Dublin Core Metadata Blocks. `DCmetadata` defines local macros for the specifying the relevant Dublin Core metadata fields and takes an optional argument that specifies the presentation of the metadata block, see Figure 1 for an example which would generate the title block for the dcm package. Let us now come to the macros themselves.

`\DCMcreators` The `\DCMcreators` and `\DCMcontributors` macros are used to specify the authors and contributors to a text fragments. These macros take one argument, the authorship of a document specified in terms of `id`s of persons specified via `\WAperson` before. They can occur multiply in a metadata block.

`\DCMtitle` The `\DCMtitle` macro takes one argument, the

`\DCMshorttitle` The `\DCMshorttitle` macro takes one argument, the

`\DCMsubject` The `\DCMsubject` macro takes one argument, the

`\DCMdescription` The `\DCMdescription` macro takes one argument, the

`\DCMpublisher` The `\DCMpublisher` macro takes one argument, the

`\DCMdate` The `\DCMdate` macro takes one argument, the

`\DCMtype` The `\DCMtype` macro takes one argument, the

`\DCMidentifier` The `\DCMidentifier` macro takes two arguments, the first one is the identification system, and the second one the identifier string itself.

`\DCMsourse` The `\DCMsourse` macro takes one argument, the

`\DCMlanguage` The `\DCMlanguage` macro takes one argument, the

`\DCMrelation` The `\DCMrelation` macro takes one argument, the

`\DCMrights` The `\DCMrights` macro takes one argument, the

`\DCMlicense` The `\DCMlicense` macro takes one argument, the

`\DCMabstract` The `\DCMabstract` macro takes one argument, the

`\DCMlicensenotice` The `\DCMlicensenotice` macro takes one argument, the

`\DCMcopyrightnotice` The `\DCMcopyrightnotice` macro takes one argument, the

¹EdNote: This still needs to be implemented, see http://www.wlug.org.nz/PdfLatexNotes for details
²EdNote: continue
Example 1: The DC Metadata block for the dcm package documentation

2.3 DCM Metadata Block Styles

The DCMetadata environment takes an optional argument that specifies the style the metadata block is rendered in. The dcm package supplies two styles: maketitle and titlepage. The former uses the `\maketitle` macro from the calling class to assemble a title, whereas the latter builds a title page from scratch. The title block of this documentation has been created by the `maketitle` style.

To add a further metadata block style `(sty)`, we simply have to supply a `\dcm@{sty}@block` macro that expands to the intended presentation. This macro does not take any arguments, but can use the internal token registers defined by the DCMetadata environment. Generally, for any of the metadata commands `\DCM{md}` defined in `?user.dcm.mdblock?` there is a token register `\dcm@{md}` that contains the value specified in the key.

2.4 Configuration

The dcm package provides a set of macros that customize (e.g. for multiple languages) the generated content.
3 Limitations

In this section we document known limitations. If you want to help alleviate them, please feel free to contact the package author. Some of them are currently discussed in the \LaTeX{} GitHub repository \LaTeX{}.

1. none reported yet

4 The Implementation

4.1 Package Options

The first step is to declare (a few) package options that handle whether certain information is printed or not. They all come with their own conditionals that are set by the options.

\begin{verbatim}
\DeclareOption*{\PassOptionsToPackage{\CurrentOption}{rdfmeta}}
\PassOptionsToPackage{\CurrentOption}{workaddress}
\ProcessOptions

The first measure is to ensure that the KeyVal package is loaded (in the right version). For \LaTeXXML{} we also initialize the package inclusions.

\begin{verbatim}
\RequirePackage{workaddress}
\RequirePackage[sectioning]{rdfmeta}
\end{verbatim}

4.2 The DC Metadata Block

Then we make an environment for defining the metadata. Note that since we have defined the omdoc:metadata element to auto-open and auto-close, we do not have to (and should not for that matter) supply it in the DCmetadata element.

\begin{verbatim}
\newenvironment{DCmetadata}[1][{}]{% to set the way things are presented.
\ifundeфинìed{dcm\@style \@block}{%
\message{style \@style not defined}}%
\csname dcm\@style \@block\endcsname}
\end{verbatim}
Here come the constructors, most of them are relatively straightforward

\DCMcreators the \DCMcreators macro checks whether all ids are defined.
12 \def\DCMcreators#1{\@for\@I:=#1\do{\wa@ref{person}\@I{id}}}
13 \gdef\dcm@creators{#1}}

\DCMcontributors the \DCMcontributors macro also checks whether all ids are defined.
14 \def\DCMcontributors#1{\@for\@I:=#1\do{\wa@ref{person}\@I{id}}%}
15 \def\dcm@contributors{#1}}

\DCMtitle
16 \def\DCMtitle#1{\def\dcm@title{#1}\providecommand{\dcm@shorttitle}{#1}}

\DCMsubtitle
17 \def\dcm@subtitle{}
18 \def\DCMsubtitle#1{\def\dcm@subtitle{#1}}

\DCMshorttitle
19 \def\dcm@shorttitle{}
20 \def\DCMshorttitle#1{\def\dcm@shorttitle{#1}}

\DCMsubject
21 \def\DCMsubject#1{\def\dcm@subject{#1}}

\DCMdescription
22 \long\def\DCMdescription#1{\long\def\dcm@description{#1}}

\DCMpublisher
23 \def\DCMpublisher#1{\def\dcm@publisher{#1}}

\DCMdate the \DCMdate uses \today as a default.\footnote{EdN:3}
24 \def\dcm@date{\today}
25 \def\DCMdate#1{\def\dcm@date{#1}}

\DCMtype
26 \def\DCMtype#1{\def\dcm@type{#1}}

\DCMidentifier
27 \def\DCMidentifier#1#2{\def\dcm@scheme{#1}\def\dcm@identifier{#2}}

\DCMsource
28 \def\DCMsource#1{\def\dcm@source{#1}}

\DCMlanguage
29 \def\DCMlanguage#1{\def\dcm@language{#1}}

\footnote{EdNote: @DEYAN: do that in latexml}
4.3 DCM Block Styles

We now define various commonly used styles.

This style builds up a title page from scratch

```latex
\dcm@titlepage@block
\begin{titlepage}
\null\vfil\vskip 60\p@
\begin{center}
\ifx\dcm@title\@empty
\PackageWarning{dcm}{No title specified}{\LARGE Add title here\par}
\else\LARGE \dcm@title \par\fi
\ifx\dcm@subtitle\@empty\Large \dcm@subtitle \par\Large
\else\large\lineskip .75em\WAauthorblock\dcm@creators\Large
\fi
\ifx\dcm@date\@empty
\PackageWarning{dcm}{No date specified}{\large \today\par}
\else\large \dcm@date \par\fi
\end{center}
\vskip 2em
\ifx\dcm@abstract\@empty
\PackageWarning{dcm}{No Abstract specified}{\large \dcm@abstract\par}
\else\begin{quote}\textbf\dcm@abstract@heading:\dcm@abstract\end{quote}\fi
\vskip 2em
\noindent\dcm@rights\dcm@license
\end{titlepage}
```

This style makes use of the title facility of the document class.

\begin{verbatim}
def\dcm@maketitle@block{
def\title\dcm@title\ifx\dcm@subtitle\empty\else\newline\dcm@subtitle\fi
\def\@author{\WAauthorblock\dcm@creators}\%
def\@date{\dcm@date}\maketitle}
\end{verbatim}

4.4 Dealing with ISO Dates

The first step is to build a macro for making ISO dates.

\begin{verbatim}
def\ISOtimestamp{\count1=\time\divide\count1 by 60 % hours
\count2=\count1\multiply\count2 by 60% minutes in \count1 hours
\count3=\time\advance\count3 by -\count2 % minutes\the\year -\ifnum\month>9\else0\fi\the\month-\ifnum\day>9\else0\fi\the\day
T\ifnum\count1>9\else0\fi\the\count1:\ifnum\count3>9\else0\fi\the\count3:00Z}
\end{verbatim}

4.5 Configuration

\begin{verbatim}
def\dcm@abstract@heading{Abstract}
def\dcm@creators@heading{Author(s)}
def\dcm@contributors@connector{with contributions from}
def\dcm@chapter@heading{Chapter}
def\dcm@section@heading{Section}
def\dcm@subsection@heading{Subsection}
def\dcm@subsubsection@heading{Subsubsection}
def\dcm@paragraph@heading{Paragraph}
\end{verbatim}

References


[sTeX] KWARC/$s$TeX. URL: \url{https://github.com/KWARC/sTeX} (visited on 05/15/2015).