

# `workaddress.sty`: An Infrastructure for managing Addresses and Affiliations in L<sup>A</sup>T<sub>E</sub>X\*

Michael Kohlhase                      John Doe  
Jacobs University                      DFKI  
<http://kwarc.info/kohlhase>   <http://dfki.de/jdoe>

April 14, 2015

## Abstract

The `workaddress` package allows manage addresses and Affiliations in a  
bibT<sub>E</sub>X-like manner.<sup>1</sup>

## Contents

---

\*Version v0.4 (last revised 2012/09/23)

<sup>1</sup>EdNOTE: `continue`

EdN:2

# 1 Introduction

The `workaddress` package allows manage Addresses and affiliations of persons in a `bibTeX`-like manner.<sup>2</sup>

## 2 The User Interface

### 2.1 Package Options

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

### 2.2 Database Entries for Persons

The `workaddress` package recognizes that from a metadata perspective, persons are complex entities. In particular, specifying metadata is a tedious and repetitive task that leads to embarrassing errors. Therefore the `workaddress` package takes a hint from `bibTeX` and allows to specify personal metadata in a database and use it by a database key. The `\WAperson` macro allows to specify personal metadata<sup>3</sup>

```
\WAperson[id=jdoe,affiliation=dfki,department=skss,
          url=http://dfki.de/jdoe]
{John Doe}
\WAperson[id=miko,affiliation=jacu,department=case,
          url=http://kwarc.info/kohlhase]
{Michael Kohlhase}
```

**Example 1:** A small database of Persons

EdN:3

`\WAperson`

with the following keys:

---

<sup>2</sup>EdNOTE: continue

<sup>3</sup>EdNOTE: This should be synchronized with the FOAF specification [`FOAF:spec`]

key	value	comment
id	string	identifier of this person
birthdate	date	birthdate
email		the primary e-mail address
url	URI	primary home page
affiliation	Inst. identifier	the primary professional affiliation
personaltitle	string	the personal title e.g. <b>King</b>
academictitle	string	the academic title e.g. <b>Prof. Dr.</b>
department	Inst. identifier	the department specified in the work address
workaddress	long string	the work address
privaddress	long string	the private address
worktel	string	work telephone number
privtel	string	private telephone number
workfax	string	work fax number
privfax	string	private fax number
worktelfax	string	if the phone and fax share a prefix, give this as well
privtelfax	string	dito

In Figure ?? we have specified (minimal) metadata for the authors of the **workaddress** package. The metadata can be accessed by specifying the identifiers (given by the **id** key) in the **workaddress** macros defined below, see for instance the **\WAcreeators** macro in Figure ??, which leads to the title block of this note.

Like in bibTeX [**Patashnik:b88**], it is a good idea to collect the metadata in a separate file that is input in the document. In practice it may be possible to generate these files from conventional address databases.

## 2.3 Institutions

Institutions are treated analogously to persons. The **\Wainstitution** macro al-

```

\Wainstitution[id=case,partof=jacu,acronym=CASE,
               url=http://jacobs-university.de/ses/case]
  {Center for Advanced Systems Engineering}
\Wainstitution[id=jacu,url=http://jacobs-university.de]
  {Jacobs University Bremen}
\Wainstitution[id=skss,partof=dfki,url=http://dfki.de/sks,acronym=SKS]
  {Safe and Secure Cognitive Systems}
\Wainstitution[id=dfki,url=http://dfki.de,shortname=DFKI,acronym=DFKI]
  {German Research Center for Artificial Intelligence}

```

**Example 2:** A small Database of Institutions and their Parts

**\Wainstitution**

lows to specify personal metadata<sup>4</sup> with the following keys:

<sup>4</sup>EDNOTE: This should be synchronized with the FOAF specification [**FOAF:spec**]

key	value	comment
id	string	identifier of this person
url	URI	primary home page
partof	Inst. identifier	parent institution

## 2.4 Applications

`\WAauthorblock` The data from the address database can be used in various ways. For instance, the `\WAauthorblock` macro creates a block of users and their affiliations. In the context of the database from Figures ?? and ??, `\WAauthorblock{miko,jdoe}` creates

Michael Kohlhasse   John Doe

`\wa@institution@logo` `\wa@institution@logo` creates the logo of an institution from the database, and (if that is not there create a box and a message instead.)

## 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 `sTeX` TRAC [`sTeX:online`].

1. none reported yet

## 4 The Implementation

The `workaddress` package generates two files: the `LATEX` package (all the code between `<*package>` and `</package>`) and the `LATEXML` bindings (between `<*ltxml>` and `</ltxml>`). We keep the corresponding code fragments together, since the documentation applies to both of them and to prevent them from getting out of sync.

### 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.

```

1 <*package>
2 \DeclareOption{showmeta}{\PassOptionsToPackage{\CurrentOption}{metakeys}}
3 \ProcessOptions
4 </package>
```

The first measure is to ensure that the `KeyVal` package is loaded (in the right version). For `LATEXML` we also initialize the package inclusions.

```

5 <*package>
6 \RequirePackage{sref}
7 \RequirePackage{pgf}
```

```

8 </package>
9 <*!xml>
10 # -*- CPERL -*-
11 package LaTeXXML::Package::Pool;
12 use strict;
13 use LaTeXXML::Global;
14 use LaTeXXML::Package;
15 DeclareOption('showmeta',sub {PassOptions('metakeys','sty',ToString(Digest(T_CS('\CurrentOption
16 ProcessOptions();
17 RequirePackage('sref');
18 RequirePackage('graphicx');
19 </!xml>

```

## 4.2 Persons

To implement the `\WAperson` macro, we need to implement its keywords.

```

20 <*package>
21 \addmetakey*{wa@person}{id}
22 \addmetakey*{wa@person}{birthdate}
23 \addmetakey*{wa@person}{email}
24 \addmetakey*{wa@person}{url}
25 \addmetakey*{wa@person}{affiliation}
26 \addmetakey*{wa@person}{personaltitle}
27 \addmetakey*{wa@person}{academictitle}
28 \addmetakey*{wa@person}{department}
29 \addmetakey*{wa@person}{workaddress}
30 \addmetakey*{wa@person}{privaddress}
31 \addmetakey*{wa@person}{worktel}
32 \addmetakey*{wa@person}{privtel}
33 \addmetakey*{wa@person}{workfax}
34 \addmetakey*{wa@person}{privfax}
35 \addmetakey*{wa@person}{worktelfax}
36 \addmetakey*{wa@person}{privtelfax}

```

`\wa@def` The next macro is an auxiliary one that puts the value into an appropriate token register.

```

37 \def\wa@def#1#2#3#4{\expandafter\xdef\csname wa@#1@#2@#3\endcsname{#4}}
38 </package>

```

At the L<sup>A</sup>T<sub>E</sub>XML side we have a function `ExportMetadata` that does a similar job, fishing out the metadata keys from the keyval arguments and storing them in a safe place so they can be accessed later.

```

39 <*!xml>
40 sub getKeyValue_noDelim {
41   my ($keyval,$key)=@_;
42   my $valuelist = $keyval && ToString($keyval->getValue($key));
43   $valuelist =~ s/^\{(.*)\}$/1/g if $valuelist;
44   return $valuelist;
45 }

```

```

46 sub ExportMetadata {
47   my $keys = shift;
48   my($id, $email,$affill,$address,$url,$name)=$keys
49   && map(getKeyValue_noDelim($keys,$_),qw(id email affiliation address url name));
50   if ($id) {
51     AssignValue('WA_'. $id.'_email',$email,'global') if $email;
52     AssignValue('WA_'. $id.'_affiliation',$affill,'global') if $affill;
53     AssignValue('WA_'. $id.'_address',$email,'global') if $email;
54     AssignValue('WA_'. $id.'_url',$url,'global') if $url;
55     AssignValue('WA_'. $id.'_name',$name,'global') if $name;
56   } else {print STDERR "Warning: key 'id' undefined in \\WAperson\n";
57     return;}#$
58 }/ltxml)

```

`\wa@ref@test` This macro tests whether the information specified is defined, and gives an error message else.

```

59 (*package)
60 \def\wa@ref@test#1#2#3{%
61 \@ifundefined{wa@#1#2#3}%
62 {\PackageError{workaddress}{reference to undefined #3 of #1 #2}%
63 {you must define a #1 with #2=#3\MessageBreak%
64 via the macro \protect\WA#1, before you can use it!}}
65 {}

```

With this, referencing is simple

`\wa@ref`

```

66 \def\wa@ref#1#2#3{\wa@ref@test{#1}{#2}{#3}\csname wa@#1#2#3\endcsname}

```

With this we can define the `\WAperson` macro, it just clears the keys, sets them again, and stores them in token registers. If course only if a `id` attribute is given, else we raise an error.

`WAperson`

```

67 \let\wa@persons=\relax
68 \newcommand\WAperson[2][\metasetkeys{wa@person}{#1}
69 \ifx\wa@person@id\empty\@latex@warning{key 'id' undefined in WAperson}\else
70 \wa@def{person}\wa@person@id{id}{\wa@person@id}% redundant, but useful for checking
71 \wa@def{person}\wa@person@id{name}{#2}
72 \wa@def{person}\wa@person@id{email}{\wa@person@email}
73 \wa@def{person}\wa@person@id{birthdate}{\wa@person@birthdate}
74 \wa@def{person}\wa@person@id{url}{\wa@person@url}
75 \wa@def{person}\wa@person@id{affiliation}{\wa@person@affiliation}
76 \wa@def{person}\wa@person@id{workaddress}{\wa@person@workaddress}
77 \wa@def{person}\wa@person@id{privaddress}{\wa@person@privaddress}
78 \wa@def{person}\wa@person@id{personalttitle}{\wa@person@personalttitle}
79 \wa@def{person}\wa@person@id{academictitle}{\wa@person@academictitle}
80 \wa@def{person}\wa@person@id{department}{\wa@person@department}
81 \wa@def{person}\wa@person@id{workaddress}{\wa@person@workaddress}

```

```

82 \wa@def{person}\wa@person@id{privaddress}{\wa@person@privaddress}
83 \wa@def{person}\wa@person@id{worktel}{\wa@person@worktel}
84 \wa@def{person}\wa@person@id{privtel}{\wa@person@privtel}
85 \wa@def{person}\wa@person@id{workfax}{\wa@person@workfax}
86 \wa@def{person}\wa@person@id{privfax}{\wa@person@privfax}
87 \wa@def{person}\wa@person@id{worktelfax}{\wa@person@worktelfax}
88 \wa@def{person}\wa@person@id{privtelfax}{\wa@person@privtelfax}
89 \@ifundefined{wa@persons}
90 {\xdef\wa@persons{\wa@person@id}}
91 {\xdef\wa@persons{\wa@persons,\wa@person@id}}
92 \fi}
93 \newcommand\DCMperson[2][\WAperson[#1]{#2}%
94 \PackageWarning{workaddress}{\protect\DCMperson\space is deprecated, use \protect\WAperson\space}
95 \</package>
96 \<*txml>
97 DefKeyVal('wa@person','id','Semiverbatim');
98 DefKeyVal('wa@person','birthdate','Semiverbatim');
99 DefKeyVal('wa@person','email','Semiverbatim');
100 DefKeyVal('wa@person','url','Semiverbatim');
101 DefKeyVal('wa@person','affiliation','Semiverbatim');
102 DefKeyVal('wa@person','personaltitle','Semiverbatim');
103 DefKeyVal('wa@person','academictitle','Semiverbatim');
104 DefKeyVal('wa@person','department','Semiverbatim');
105 DefKeyVal('wa@person','workaddress','Semiverbatim');
106 DefKeyVal('wa@person','privaddress','Semiverbatim');
107 DefKeyVal('wa@person','worktel','Semiverbatim');
108 DefKeyVal('wa@person','privtel','Semiverbatim');
109 DefKeyVal('wa@person','workfax','Semiverbatim');
110 DefKeyVal('wa@person','privfax','Semiverbatim');
111 DefKeyVal('wa@person','worktelfax','Semiverbatim');
112 DefKeyVal('wa@person','privtelfax','Semiverbatim');
113
114 DefConstructor('\WAperson OptionalKeyVals:wa@person {}','',
115 afterDigest=>sub {
116   my ($stomach,$whatsit)=@_;
117   my $keys=$whatsit->getArg(1);
118   my $name=ToString($whatsit->getArg(2));
119   $keys->setValue('name',$name);
120   ExportMetadata($keys);
121   return;
122 });#<
123 \</txml>

```

## 4.3 Institutions

To implement the \WAinstitution macro, we need to implement its keywords first.

```

124 <*package>
125 \addmetakey*{wa@institution}{id}

```

```

126 \addmetakey*{wa@institution}{shortname}
127 \addmetakey*{wa@institution}{acronym}
128 \addmetakey*{wa@institution}{url}
129 \addmetakey*{wa@institution}{partof}
130 \addmetakey*{wa@institution}{countryshort}
131 \addmetakey*{wa@institution}{logo}
132 \addmetakey*{wa@institution}{streetaddress}
133 \addmetakey*{wa@institution}{townzip}
134 \addmetakey*{wa@institution}{type}
135 \addmetakey*{wa@institution}{country}

```

and we proceed as for \WAperson,

```

136 \let\wa@institutions=\relax

```

#### WAinstitution

```

137 \newcommand\WAinstitution[2][\metasetkeys{wa@institution}{#1}
138 \ifx\wa@institution@id\@empty\@latex@warning{key 'id' undefined in WAinstitution}\else
139 \wa@def{institution}\wa@institution@id{id}{\wa@institution@id}% redundant, but useful for check
140 \wa@def{institution}\wa@institution@id{name}{#2}
141 \wa@def{institution}\wa@institution@id{shortname}{\wa@institution@shortname}
142 \wa@def{institution}\wa@institution@id{acronym}{\wa@institution@acronym}
143 \wa@def{institution}\wa@institution@id{url}{\wa@institution@url}
144 \wa@def{institution}\wa@institution@id{partof}{\wa@institution@partof}
145 \wa@def{institution}\wa@institution@id{countryshort}{\wa@institution@countryshort}
146 \wa@def{institution}\wa@institution@id{logo}{\wa@institution@logo}
147 \wa@def{institution}\wa@institution@id{townzip}{\wa@institution@townzip}
148 \wa@def{institution}\wa@institution@id{streetaddress}{\wa@institution@streetaddress}
149 \wa@def{institution}\wa@institution@id{country}{\wa@institution@country}
150 \wa@def{institution}\wa@institution@id{type}{\wa@institution@type}
151 \@ifundefined{wa@institutions}
152 {\xdef\wa@institutions{\wa@institution@id}}
153 {\xdef\wa@institutions{\wa@institutions,\wa@institution@id}}
154 \fi}
155 \newcommand\DCMinstitution[2][\WAinstitution[#1]{#2}%
156 \PackageWarning{workaddress}{\protect\DCMinstitution\space is deprecated, use \protect\WAinsti
157 \>/package}
158 \<*\txml>
159 DefKeyVal('wa@institution','id','Semiverbatim');
160 DefKeyVal('wa@institution','url','Semiverbatim');
161 DefKeyVal('wa@institution','partof','Semiverbatim');
162 DefConstructor('\WAinstitution OptionalKeyVals:wa@institution {}','',
163 afterDigest=>sub {
164 my ($stomach,$whatsit)=@_;
165 my $keys=$whatsit->getArg(1);
166 my $name=ToString($whatsit->getArg(2));
167 $keys->setValue('name',$name);
168 ExportMetadata($keys);
169 return;
170 });#<
171 \</txml>

```



## 4.4 Applications

`\WAauthorblock` This internal macro builds an author block from a list of `\WAperson` labels in `\wa@creators`.

```

172 <*package>
173 \addmetakey[false]{WAauthorblock}{dept}[true]
174 \addmetakey[false]{WAauthorblock}{aff}[true]
175 \addmetakey[false]{WAauthorblock}{url}[true]
176 \def\@true{true}
177 \newcounter{authors}
178 \newcommand\WAauthorblock[2][{}]{%
179 \metasetkeys{WAauthorblock}{#1}
180 {\let\tabularnewline\relax
181 \@for\@I:=#2\do{\stepcounter{authors}}
182 \def\@authors{}\def\@affs{}\def\@depts{}\def\@urls{}
183 \@for\@I:=#2\do
184   {\xdef\@authors{\@authors&\wa@ref{person}\@I{name}}
185    \xdef\@dept{\wa@ref{person}\@I{department}}
186    \xdef\@shortname{\csname wa@institution@\@dept @shortname\endcsname}
187    \xdef\@dept{\ifx\@shortname\@empty\wa@ref{institution}\@dept{name}\else\@shortname\fi}
188    \xdef\@depts{\@depts&\@dept}
189    \xdef\@aff{\wa@ref{person}\@I{affiliation}}
190    \xdef\@shortname{\csname wa@institution@\@aff @shortname\endcsname}
191    \xdef\@aff{\ifx\@shortname\@empty\wa@ref{institution}\@aff{name}\else\@shortname\fi}
192    \xdef\@affs{\@affs&\@aff}
193    \xdef\@urls{\@urls&\wa@ref{person}\@I{url}}}
194 \message{\theauthors authors: \@authors}}
195 \begin{tabular}[t]{l*{\theauthors}{c}}
196   \@authors\\
197   \ifx\WAauthorblock@dept\@true\@depts\\\fi
198   \ifx\WAauthorblock@aff\@true\@affs\\\fi
199   \ifx\WAauthorblock@url\@true\@urls\\\fi
200 \end{tabular}}
201 </package>
202 % \begin{macrocode}
203 % \end{macro}
204 %
205 % \begin{macro}{\wapname}
206 % \begin{macrocode}
207 <*package>
208 \newcommand\wapname[1]{\wa@ref{person}{#1}{name}}
209 </package>

```

`\waptname`

```

210 <*package>
211 \newcommand\waptname[1]{\wa@ref{person}{#1}{personaltitle} \wa@ref{person}{#1}{name}}
212 </package>

```

`\wa@institution@logo`

```

213 <*package>
214 \newcommand\wa@institution@logo[2][\wa@ref{institution}{#2}{logo}}
215 \pgfdeclareimage[#1]{logo}{\wa@ref{institution}{#2}{logo}}
216 \IfFileExists{\wa@ref{institution}{#2}{logo}}%
217 {\pgfuseimage{logo}}
218 {\fbox{#2 logo}\message{still need logo for #2}}
219 </package>

```

## 4.5 Finale

Finally, we need to terminate the file with a success mark for perl.

```

220 <ltxml>1;

```