pathsuris.sty: Paths and URIs for STEX*

Jinbo Zhang, Michael Kohlhase, Dennis Müller FAU Erlangen-Nürnberg

October 21, 2020

Abstract

This package provides macros to deal with paths and base URIs for STEX. In particular, it offers a path canonicalizer, which is used in package modules, in order to support modules specified with relative path.

Contents

| 1 | | r Interface | | | | | | | | | | | | | |
|----------|-----|-----------------------|--|--|--|--|--|--|--|--|--|--|--|--|--|
| | 1.1 | Base URIs | | | | | | | | | | | | | |
| | 1.2 | Using Absolute Paths | | | | | | | | | | | | | |
| | 1.3 | Path Canonicalization | | | | | | | | | | | | | |
| | 1.4 | URIs | | | | | | | | | | | | | |
| 2 | | Implementation | | | | | | | | | | | | | |
| | 2.1 | Base URIs | | | | | | | | | | | | | |
| | 2.2 | Using Absolute Paths | | | | | | | | | | | | | |
| | 2.3 | Path Canonicalization | | | | | | | | | | | | | |
| | 2.4 | URIs | | | | | | | | | | | | | |

^{*}Version v2.1 (last revised 2020/09/30)

User Interface 1

Base URIs 1.1

\baseURI1 \baseURI

1.2 Using Absolute Paths

\defpath

EdN:1

Finally, the separation of documents into multiple modules often profits from a symbolic management of file paths. To simplify this, the modules package supplies the \defpath macro: $\defpath[\langle baseURI \rangle] \{\langle cname \rangle\} \{\langle path \rangle\}\$ defines a command, so that $\langle csname \rangle \{\langle name \rangle\}$ expands to $\langle path \rangle / \langle name \rangle$. So we could have used

```
\defpath{OPaths}{../other}
\importmodule[load=\OPahts{bar}]{bar}
```

instead of the second line in Example ??. The variant \OPaths has the big advantage that we can get around the fact that TEX/IATEX does not set the current directory in \input, so that we can use systematically deployed \defpath-defined path macros to make modules relocatable by defining the path macros locally. The optional parameter $\langle baseURI \rangle$ is for the LATEXML transformation, which (if $\langle baseURI \rangle$ is specified) resolves $\langle path \rangle$ to an absolute URI according to [BFM05, section 5.2].

1.3 Path Canonicalization

By calling $\operatorname{Qcpath}\{\langle path \rangle\}$, the canonicalized path will be stored in $\operatorname{QCanPath}$. To print a canonicalized path, simply use $\mathbf{cpath}\{\langle path \rangle\}$. Here is a set of examples with their canonizalized paths for testing.

| path | canonicalized path | expected |
|---------------|--------------------|-------------|
| paun | canonicanzed path | cxpccica |
| aaa | aaa | aaa |
| //aaa | //aaa | //aaa |
| aaa/bbb | aaa/bbb | aaa/bbb |
| aaa/ | | |
| //aaa/bbb | //aaa/bbb | //aaa/bbb |
| /aaa//bbb | /bbb | /bbb |
| /aaa/bbb | /aaa/bbb | /aaa/bbb |
| aaa/bbb//ddd | aaa/ddd | aaa/ddd |
| aaa/bbb/./ddd | aaa/bbb/ddd | aaa/bbb/ddd |
| ./ | | |
| aaa/bbb// | | |

 $^{^{1}\}mathrm{EdNote}$: document it

1.4 URIs

By calling \seturi[\meta{macroname}] { $\langle path \rangle$ }, the URI will be split into its components \macronamescheme, \macronameauthority, \macronamepath, \macronamequery and \macronamefragment, and the resolved URI itself is stored in \macronameuri, as in the following example. If the optional macroname is not provided, the default name is pathsuris@curruri@.

In order to differentiate between empty and missing components, a missing component will be equal to \makeuri@empty, whose expansion is \relax.

\seturi[myuri]{http://this.isatest/foo/bar/?query#fragment}

| | macro | value |
|---------|-----------------|--|
| | \myuriuri | http://this.isatest/foo/bar?query#fragment |
| | \myurischeme | http |
| yields: | \myuriauthority | this.isatest |
| | \myuripath | foo/bar |
| | \myuriquery | query |
| | \myurifragment | fragment |

\makeuri{\meta{scheme}}{\meta{authority}}{\meta{query}}{\meta{fragment}}\ constructs a URI from its individual components. The (expanded and resolved) individual components will be stored in \makeuri@scheme, \makeuri@authority, etc.; the resolved URI will be stored in \makeuri@uri.

\asuri{\meta{macroname}}{\meta{uri}}, similarly to setpath, defines a new macro \macroname[\meta{newmacroname}]{\meta{command}} that allows manipulating uri in various ways. \asuri calls \seturi[\meta{macroname}]{\meta{uri}}, so the individual components and full uri (as string) are subsequently stored in \macronamescheme, \macronameauthority, etc.

If an optional new macro name is given in \macroname, then the result of the modification is stored in that new macro, as if defined via \asuri; otherwise, the macro is modified "in place".

- \macroname{drop query} drops the query component.
- \macroname{drop fragment} drops the fragment component.
- \macroname{/other/path} drops query and fragment, appends other/path to the path and resolves the URI.
- \macroname{?newquery} drops the fragment and either declares newquery as a new query component, or appends ?newquery to the existing query component, if it is not \makeuri@empty. Note, that this behaviour diverges from the official URI specification, but it conforms to MMT URI's, which use ? as separator between DPaths, modules names and declaration names.
- \macroname{#newfragment} analogously to {?newquery}.

2 The Implementation

```
1 \*package\
2 \RequirePackage{stex-base}
3 \RequirePackage{xstring}
4 \RequirePackage{etoolbox}
```

2.1 Base URIs

\baseURI On the LATEX side we do nothing (for the moment).

5 \newcommand\baseURI[2][]{}

2.2 Using Absolute Paths

\defpath \defpath[optional argument] {macro name} {base path} defines a new macro which can take another path to formal one integrated path. For example, \MathHub in every localpaths.tex is defined as:

\defpath{MathHub}{/path/to/localmh/MathHub}

then we can use \MathHub to form other paths, for example,

\MathHub{source/smglom/sets}

will generate /path/to/localmh/MathHub/source/smglom/sets.

- 6 \newrobustcmd\defpath[3][]{%
- 7 \expandafter\newcommand\csname #2\endcsname[1]{#3/##1}% 8 }%

2.3 Path Canonicalization

We define two macros for changing the category codes of common characters in URIs, in particular #.

```
9 \def\pathsuris@setcatcodes{%
10
      \edef\pathsuris@oldcatcode@hash{\the\catcode'\#}%
      \catcode'\#=12\relax%
11
      \edef\pathsuris@oldcatcode@slash{\the\catcode'\/}%
12
13
      \catcode'\/=12\relax%
      \edef\pathsuris@oldcatcode@colon{\the\catcode'\:}%
14
      \catcode'\:=12\relax%
15
      \edef\pathsuris@oldcatcode@qm{\the\catcode'\?}%
16
      \catcode'\?=12\relax%
17
18 }
19 \def\pathsuris@resetcatcodes{%
      \catcode'\#\pathsuris@oldcatcode@hash\relax%
20
21
      \catcode'\/\pathsuris@oldcatcode@slash\relax%
      \catcode'\:\pathsuris@oldcatcode@colon\relax%
      \catcode'\?\pathsuris@oldcatcode@qm\relax%
23
24 }
```

```
We define some macros for later comparison.
         25 \def\@ToTop{..}
         26 \left( \frac{0}{2} \right)
         27 \def\@Colon{:}
         28 \def\@QuestionMark{?}
         29 \left(0\right)
         31 \pathsuris@setcatcodes
         32 \def\@Fragment{#}
         33 \pathsuris@resetcatcodes
         Implement \@cpath.
\@cpath
         34 \left(\frac{9}{2}\right)
                \edef\pathsuris@cpath@temp{#1}%
         35
                \def\@CanPath{}%
         36
         37
                \IfBeginWith\pathsuris@cpath@temp\@Slash{%
                  \@cpath@loop%
         38
                  \edef\@CanPath{\@Slash\@CanPath}%
         39
                }{%
         40
                    \IfBeginWith\pathsuris@cpath@temp{\@Dot\@Slash}{%
         41
                         \StrGobbleLeft\pathsuris@cpath@temp2[\pathsuris@cpath@temp]%
         42
                         \@cpath@loop%
         43
                    }{%
         44
         45
                         \ifx\pathsuris@cpath@temp\@Dot\else%
         46
                         \@cpath@loop\fi%
                    }%
         47
                }%
         48
                \IfEndWith\@CanPath\@Slash{%
         49
         50
                  \ifx\@CanPath\@Slash\else%
                    \StrGobbleRight\@CanPath1[\@CanPath]%
         51
                  \fi%
         52
                }{}%
         53
         54 }
         55
         56 \def\@cpath@loop{%
                \IfSubStr\pathsuris@cpath@temp\@Slash{%
         57
                    \StrCut\pathsuris@cpath@temp\@Slash\pathsuris@cpath@temp@a\pathsuris@cpath@temp%
         58
                    \ifx\pathsuris@cpath@temp@a\@ToTop%
         59
                         \ifx\@CanPath\@empty%
         60
                             \edef\@CanPath{\@ToTop}%
         61
                        \else%
         62
                             \edef\@CanPath\@Slash\@ToTop}%
         63
                        \pi
         64
                         \@cpath@loop%
         65
                    \else%
         66
                    \ifx\pathsuris@cpath@temp@a\@Dot%
         67
                         \@cpath@loop%
         68
```

69

\else%

```
\IfBeginWith\pathsuris@cpath@temp\@ToTop{%
70
                                                       71
                                                       \IfBeginWith\pathsuris@cpath@temp\@Slash{%
72
                                                                      \edef\pathsuris@cpath@temp{\@CanPath\pathsuris@cpath@temp}%
73
                                                      }{%
74
75
                                                                      \ifx\@CanPath\@empty\else%
76
                                                                                      \edef\pathsuris@cpath@temp{\@CanPath\@Slash\pathsuris@cpath@temp}
                                                                      \fi%
77
                                                      }%
78
                                                       \def\@CanPath{}%
79
                                                       \@cpath@loop%
80
                                       }{%
81
                                                       \ifx\@CanPath\@empty%
82
                                                                      \edef\@CanPath{\pathsuris@cpath@temp@a}%
83
                                                       \else%
84
                                                                      \verb|\edge f @CanPath { @CanPath @Slash pathsuris@cpath@temp@a} % | A converge for the conve
85
                                                       \fi%
86
                                                       \@cpath@loop
87
                                       }%
88
89
                                        \fi\fi%
                        }{
90
                                        \ifx\@CanPath\@empty%
91
                                                       \edef\@CanPath{\pathsuris@cpath@temp}%
92
93
                                        \else%
                                                       \edef\@CanPath{\@CanPath\@Slash\pathsuris@cpath@temp}%
94
95
                                        \fi%
                        }%
96
97 }
```

Implement \cpath to print the canonicalized path.

\cpath

```
98 \newcommand\cpath[1]{%
99 \@cpath{#1}%
100 \@CanPath%
101 }
```

2.4 URIs

Various macros for dealing with URIs. To deal with empty URI components (scheme, authority, etc.), we use {\relax} to signify an non-existent component as oppsed to an empty one.

```
102 \def\makeuri@setempty#1{\def#1{\relax}}
103 \def\makeuri@empty{\relax}
104 \def\makeuri@test#1{%
105 \ifx#1\makeuri@empty\else#1\fi%
106 }
```

\makeuri \makeuri constructs a URI from scheme, authority, path, query and fragment separately.

```
107 \def\makeuri@uri{}
         108 \def\makeuri#1#2#3#4#5{
                 \edef\makeuri@scheme{#1}
         109
                 \edef\makeuri@authority{#2}
         110
                 \edef\makeuri@path{#3}
         111
         112
                 \ifx\makeuri@path\makeuri@empty\else
         113
                     \@cpath{#3}
                     \edef\makeuri@path{\@CanPath}
         114
                 \fi
         115
                 \edef\makeuri@query{#4}
         116
                 \edef\makeuri@fragment{#5}
         117
         118
                 \ifx\makeuri@scheme\makeuri@empty\else
         119
                     \edef\makeuri@scheme\@Colon}
         120
                 \ifx\makeuri@authority\makeuri@empty\else
         121
                     \edef\makeuri@authority{\@Slash\@Slash\makeuri@authority}
         122
                     \ifx\makeuri@path\makeuri@empty\else
         123
                         \IfBeginWith\makeuri@path\@Slash{}{
         124
         125
                             \edef\makeuri@path{\@Slash\makeuri@path}
         126
                     \fi
         127
                 \fi
         128
                 \ifx\makeuri@query\makeuri@empty\else
         129
                     \edef\makeuri@query{\@QuestionMark\makeuri@query}
         130
         131
         132
                 \ifx\makeuri@fragment\makeuri@empty\else
                     \edef\makeuri@fragment{\@Fragment\makeuri@fragment}
         133
         134
                 \edef\makeuri@uri{%
         135
                     \makeuri@test\makeuri@scheme%
         136
                     \makeuri@test\makeuri@authority%
         137
         138
                     \makeuri@test\makeuri@path%
         139
                     \makeuri@test\makeuri@query%
                     \makeuri@test\makeuri@fragment%
         140
         141
                }
         142 }
\seturi@
         143 \newif\if@pathsuris@done@
         144 \def\seturi@[#1]#2{%
                 \@pathsuris@done@false%
         145
                 \def\pathsuris@prefix@temp{#1}
         146
         147
                 \edef\pathsuris@curruri{#2}%
                 \edef\pathsuris@curruri{\expandafter\detokenize\expandafter{\pathsuris@curruri}}
         148
                 \let\pathsuris@temp\pathsuris@curruri%
         149
                 \makeuri@setempty\pathsuris@curruri@scheme%
         150
                 \makeuri@setempty\pathsuris@curruri@authority%
         151
                 \makeuri@setempty\pathsuris@curruri@path%
         152
         153
                 \makeuri@setempty\pathsuris@curruri@query%
         154
                 \makeuri@setempty\pathsuris@curruri@fragment%
```

```
% scheme
155
       \IfSubStr{\pathsuris@temp}{\@Colon}{%
156
           \% TODO check for valid scheme
157
           \StrBefore{\pathsuris@temp}{\@Colon}[\pathsuris@curruri@scheme]%
158
           \StrBehind{\pathsuris@temp}{\@Colon}[\pathsuris@temp]%
159
160
       }{}%
161
       % authority
       \IfBeginWith{\pathsuris@temp}{\@Slash\@Slash}{%
162
           \StrBehind{\pathsuris@temp}{\@Slash\@Slash}[\pathsuris@temp]%
163
           \IfSubStr{\pathsuris@temp}{\@Slash}{%
164
               \StrBefore{\pathsuris@temp}{\@Slash}[\pathsuris@curruri@authority]%
165
               \StrBehind{\pathsuris@temp}{\@Slash}[\pathsuris@temp]%
166
               % TODO userinfo, host, port
167
           }{%
168
               \IfSubStr\pathsuris@temp\@QuestionMark{
169
                   170
                   \StrBehind{\pathsuris@temp}{\@QuestionMark}[\pathsuris@temp]%
171
                   \edef\pathsuris@temp{\@QuestionMark\pathsuris@temp}%
172
173
               }{
174
                   \IfSubStr\pathsuris@temp\@Fragment{
                       \StrBefore{\pathsuris@temp}{\@Fragment}[\pathsuris@curruri@authority]%
175
                       \StrBehind{\pathsuris@temp}{\@Fragment}[\pathsuris@temp]%
176
                       \edef\pathsuris@temp{\@Fragment\pathsuris@temp}%
177
                   }{
178
179
                       \edef\pathsuris@curruri@authority{\pathsuris@temp}%
                       \@pathsuris@done@true%
180
                   }
181
               }
182
           }%
183
       }{}%
184
       % path, query, fragment
185
       \if@pathsuris@done@\else%
186
187
           \IfSubStr{\pathsuris@temp}{\@QuestionMark}{%
188
               \StrBefore{\pathsuris@temp}{\@QuestionMark}[\pathsuris@curruri@path]%
189
               \@cpath\pathsuris@curruri@path%
190
               \edef\pathsuris@curruri@path{\@CanPath}%
191
               \StrBehind{\pathsuris@temp}{\@QuestionMark}[\pathsuris@temp]%
192
               % query, fragment
193
               \IfSubStr{\pathsuris@temp}{\@Fragment}{%
194
195
                   \StrBefore{\pathsuris@temp}{\@Fragment}[\pathsuris@curruri@query]%
                   \StrBehind{\pathsuris@temp}{\@Fragment}[\pathsuris@curruri@fragment]%
196
               }{%
197
                   \edef\pathsuris@curruri@query{\pathsuris@temp}%
198
199
               }%
200
           }{%
201
               % path, fragment
202
               \IfSubStr{\pathsuris@temp}{\@Fragment}{%
                   \StrBefore{\pathsuris@temp}{\@Fragment}[\pathsuris@curruri@path]%
203
                   \@cpath\pathsuris@curruri@path%
204
```

```
\edef\pathsuris@curruri@path{\@CanPath}%
        205
                            \StrBehind{\pathsuris@temp}{\@Fragment}[\pathsuris@curruri@fragment]%
        206
                        }{%
        207
                            \edef\pathsuris@curruri@path{\pathsuris@temp}%
        208
                        }%
        209
        210
                   }%
        211
                \fi%
        212
                \makeuri\pathsuris@curruri@scheme\pathsuris@curruri@authority\pathsuris@curruri@path\pathsu
                \let\pathsuris@curruri@uri\makeuri@uri
        213
               %drop trailing slash of path
        214
               %\IfEndWith{\pathsuris@curruri@path}{\@Slash}{%
        215
        216
                     \StrGobbleRight{\pathsuris@curruri@path}{1}[\pathsuris@curruri@path]
               %}{}%
        217
        218
               %\edef\pathsuris@curruri@path{\cpath{\pathsuris@curruri@path}}%
        219
                \ifx\pathsuris@prefix@temp\@empty\else%
        220
                    \expandafter\let\csname \pathsuris@prefix@temp scheme\endcsname\pathsuris@curruri@schem
        221
                    \expandafter\let\csname \pathsuris@prefix@temp authority\endcsname\pathsuris@curruri@au
        222
        223
                    \expandafter\let\csname \pathsuris@prefix@temp path\endcsname\pathsuris@curruri@path%
        224
                    \expandafter\let\csname \pathsuris@prefix@temp query\endcsname\pathsuris@curruri@query%
                    \expandafter\let\csname \pathsuris@prefix@temp fragment\endcsname\pathsuris@curruri@fra
        225
                    \expandafter\let\csname \pathsuris@prefix@temp uri\endcsname\pathsuris@curruri@uri%
        226
        227
                \fi%
        228 }
\seturi
        229 \newrobustcmd\seturi[1][]{%
                \pathsuris@setcatcodes%
                \expandafter\pathsuris@resetcatcodes\seturi@[#1]%
        231
        232 }
         \asuri{macroname}{uri} generates \macroname[optional new macro name]{action},
         that allows for modifying uri in various ways.
        233
        234 \def\asuri#1{%
        235
                \pathsuris@setcatcodes%
        236
                \expandafter\pathsuris@resetcatcodes\@asuri[#1]%
        237 }
        238
        239 \def\@asuri[#1]#2{
                \@cpath{#2}
        240
                \expandafter\def\csname #1\endcsname{}
        241
        242
                \expandafter\edef\csname #1uri\endcsname{\@CanPath}
                \seturi[#1]{\@CanPath}
        243
                \expandafter\renewcommand\csname #1\endcsname[1][]{%
        244
        245
                    \pathsuris@setcatcodes%
        246
                    \@@asuri@[##1]{#1}%
        247
               }%
        248 }
```

```
249
250 \protected\def\@@asuri@[#1]#2#3{
       \pathsuris@resetcatcodes
251
       \@@asuri[#1]{#2}{#3}
252
253 }
254
255 \newif\if@asuri@changed@
256 \protected\def\@@asuri[#1]#2#3{
       \@asuri@changed@false
257
       \edef\@@asuri@command{#3}
258
       \trimstring\@@asuri@command
259
       \IfBeginWith\@@asuri@command{drop}{
260
            \StrBehind{\@@asuri@command}{drop}[\@@asuri@command]
261
           \trimstring\@@asuri@command
262
           \IfStrEq\@@asuri@command{query}{
263
                \makeuri{\csname #2scheme\endcsname}%
264
                    {\csname #2authority\endcsname}%
265
                    {\csname #2path\endcsname}%
266
267
                    \makeuri@empty%
268
                    {\csname #2fragment\endcsname}%
                \@asuri@changed@true
269
270
           }{
           \IfStrEq\@@asuri@command{fragment}{
271
                \makeuri{\csname #2scheme\endcsname}%
272
                    {\csname #2authority\endcsname}%
273
274
                    {\csname #2path\endcsname}%
                    {\csname #2query\endcsname}%
275
                    \makeuri@empty%
276
                \@asuri@changed@true
277
           }{
278
           \IfStrEq\@@asuri@command{extension}{
279
280
                \edef\@asuri@oldpath{\csname #2path\endcsname}
281
                \StrCount\@asuri@oldpath.[\@asuri@lastdot]
                \ifnum\@asuri@lastdot>0
282
                    \StrBehind[\@asuri@lastdot]\@asuri@oldpath.[\@asuri@extension]
283
                    \IfSubStr\@asuri@extension\@Slash{}{
284
                        \StrBefore[\@asuri@lastdot]\@asuri@oldpath.[\@asuri@oldpath]
285
                    }
286
287
                \fi
                \makeuri{\csname #2scheme\endcsname}%
288
                    {\csname #2authority\endcsname}%
289
                    \@asuri@oldpath%
290
                    \makeuri@empty%
291
                    \makeuri@empty%
292
293
                \@asuri@changed@true
294
           }{}}}
295
       }{
296
       \IfBeginWith\@@asuri@command{\@Slash}{
297
           \@cpath{\csname #2path\endcsname\@@asuri@command}
           \makeuri{\csname #2scheme\endcsname}%
298
```

```
{\csname #2authority\endcsname}%
299
                {\CanPath}%
300
                \makeuri@empty%
301
                \makeuri@empty%
302
            \@asuri@changed@true
303
304
305
        \IfBeginWith\@@asuri@command{\@QuestionMark}{
            \expandafter\ifx\csname #2query\endcsname\makeuri@empty
306
                \StrBehind\@@asuri@command\@QuestionMark[\@@asuri@command]
307
                \edef\@@asuri@nquery{\@@asuri@command}
308
            \else
309
                \edef\@@asuri@nquery{\csname #2query\endcsname\@@asuri@command}
310
            \fi
311
            \makeuri{\csname #2scheme\endcsname}%
312
                {\csname #2authority\endcsname}%
313
                {\csname #2path\endcsname}%
314
                {\@@asuri@nquery}%
315
                \makeuri@empty%
316
317
            \@asuri@changed@true
318
        \IfBeginWith\@@asuri@command{\@Fragment}{
319
            \expandafter\ifx\csname #2fragment\endcsname\makeuri@empty
320
                \StrBehind\@@asuri@command\@Fragment[\@@asuri@command]
321
                \edef\@@asuri@nfrag{\@@asuri@command}
322
323
            \else
324
                \edef\@@asuri@nfrag{\csname #2fragment\endcsname\@@asuri@command}
325
            \makeuri{\csname #2scheme\endcsname}%
326
                {\csname #2authority\endcsname}%
327
                {\csname #2path\endcsname}%
328
                {\csname #2query\endcsname}%
329
330
                {\@@asuri@nfrag}%
331
            \@asuri@changed@true
       }{}
332
333
       }}}
        \edef\@@asuri@ncs{#1}
334
        \if@asuri@changed@
335
            \ifx\@@asuri@ncs\@empty
336
337
                \asuri{#2}\makeuri@uri
            \else
338
339
                 \asuri\@@asuri@ncs\makeuri@uri
340
            \fi
        \fi
341
342 }
343
    auxiliary code:
344 \ensuremath{\mbox{def}\ensuremath{\mbox{\sc gasce}}} 
345 \ensuremath{\mbox{def}\mbox{trimstring#1}}
        \edef\pathsuris@trim@temp{#1}
```

```
347
       \IfBeginWith\pathsuris@trim@temp\@Space{
           \StrGobbleLeft\pathsuris@trim@temp1[#1]
348
           \trimstring{#1}
349
       }{
350
           \IfEndWith\pathsuris@trim@temp\@Space{
351
352
                \StrGobbleRight\pathsuris@trim@temp1[#1]
353
                \trimstring{#1}
           }{
354
                \edef#1{\pathsuris@trim@temp}
355
           }
356
       }
357
358 }
359
360 % windows paths
361
362 \catcode '\.=0
363 .catcode'.\=12
364 .let.@BackSlash\
365 .catcode'.\=0
366 \catcode '\.=12
367
368 \newif\if@windowstopath@inpath@
369 \def\windows@to@path#1{
       \@windowstopath@inpath@false
370
       \def\windows@temp{}
371
372
       \edef\windows@path{#1}
373
       \ifx\windows@path\@empty\else
           \expandafter\windows@path@loop\windows@path\windows@path@end
374
375
       \let#1\windows@temp
376
377 }
378 \def\windows@path@loop#1#2\windows@path@end{
379
       \def\windows@temp@b{#2}
       \ifx\windows@temp@b\@empty
380
381
           \def\windows@continue{}
382
       \else
           \def\windows@continue{\windows@path@loop#2\windows@path@end}
383
384
       \fi
385
       \if@windowstopath@inpath@
           \ifx#1\@BackSlash
386
387
                \edef\windows@temp\@Slash}
388
           \else
                \edef\windows@temp{\windows@temp#1}
389
           \fi
390
391
       \else
392
           \int 1:
393
                \edef\windows@temp{\@Slash\windows@temp}
394
                \@windowstopath@inpath@true
395
           \else
                \edef\windows@temp{\windows@temp#1}
396
```

```
397
                                                  \fi
398
                                 \fi
                                 \windows@continue
399
400 }
401
402 \ensuremath{\def\path@to@windows\#1{}}
403
                                \@windowstopath@inpath@false
                                 \def\windows@temp{}
404
                                 \edef\windows@path{#1}
405
                                \edef\windows@path{\expandafter\@gobble\windows@path}
406
                                \ifx\windows@path\@empty\else
407
                                                  \expandafter\path@windows@loop\windows@path\windows@path@end
408
409
                                \fi
                                \let#1\windows@temp
410
411 }
412 \end{2mm} windows @loop #1#2 \end{2mm} and $0 = 12 \end{2mm} windows @path @end{3mm} windows @path @end{3mm} windows @path @end{3mm} and $0 = 12 \end{3mm} windows @path @end{3mm} windows @pa
                                \def\windows@temp@b{#2}
413
                                \ifx\windows@temp@b\@empty
414
415
                                                  \def\windows@continue{}
416
                                \else
                                                  \def\windows@continue{\path@windows@loop#2\windows@path@end}
417
418
                                \if@windowstopath@inpath@
419
                                                  \int ifx#1/
420
                                                                    \edef\windows@temp{\windows@temp\@BackSlash}
421
422
                                                  \else
423
                                                                    \edef\windows@temp{\windows@temp#1}
                                                  \fi
424
                                \else
425
                                                  \int ifx#1/
426
                                                                    \edef\windows@temp{\windows@temp:\@BackSlash}
427
428
                                                                    \@windowstopath@inpath@true
429
                                                  \else
                                                                     \edef\windows@temp{\windows@temp#1}
430
431
                                                  \fi
                                \fi
432
                                \windows@continue
433
434 }
435
436 % kpsewhich
438 \newif\if@iswindows@\@iswindows@false
439 \fileExists{nul:}{\lifFileExists{/dev/null}{}}{\lifFileExists{/dev/null}{}}{\lifFileExists{/dev/null}{}}{\lifFileExists{/dev/null}{}}{\lifFileExists{nul:}}{\lifFileExists{/dev/null}{}}{\lifFileExists{nul:}}{\lifFileExists{nul:}}{\lifFileExists{nul:}}{\lifFileExists{null:}}{\lifFileExists{null:}}{\lifFileExists{null:}}{\lifFileExists{null:}}{\lifFileExists{null:}}{\lifFileExists{null:}}{\lifFileExists{null:}}{\lifFileExists{null:}}{\lifFileExists{null:}}{\lifFileExists{null:}}{\lifFileExists{null:}}{\lifFileExists{null:}}{\lifFileExists{null:}}{\lifFileExists{null:}}{\lifFileExists{null:}}{\lifFileExists{null:}}{\lifFileExists{null:}}{\lifFileExists{null:}}{\lifFileExists{null:}}{\lifFileExists{null:}}{\lifFileExists{null:}}{\lifFileExists{null:}}{\lifFileExists{null:}}{\lifFileExists{null:}}{\lifFileExists{null:}}{\lifFileExists{null:}}{\lifFileExists{null:}}{\lifFileExists{null:}}{\lifFileExists{null:}}{\lifFileExists{null:}}{\lifFileExists{null:}}{\lifFileExists{null:}}{\lifFileExists{null:}}{\lifFileExists{null:}}{\lifFileExists{null:}}{\lifFileExists{null:}}{\lifFileExists{null:}}{\lifFileExists{null:}}{\lifFileExists{null:}}{\lifFileExists{null:}}{\lifFileExists{null:}}{\lifFileExists{null:}}{\lifFileExists{null:}}{\lifFileExists{null:}}{\lifFileExists{null:}}{\lifFileExists{null:}}{\lifFileExists{null:}}{\lifFileExists{null:}}{\lifFileExists{null:}}{\lifFileExists{null:}}{\lifFileExists{null:}}{\lifFileExists{null:}}{\lifFileExists{null:}}{\lifFileExists{null:}}{\lifFileExists{null:}}{\lifFileExists{null:}}{\lifFileExists{null:}}{\lifFileExists{null:}}{\lifFileExists{null:}}{\lifFileExists{null:}}{\lifFileExists{null:}}{\lifFileExists{null:}}{\lifFileExists{null:}}{\lifFileExists{null:}}{\lifFileExists{null:}}{\lifFileExists{null:}}{\lifFileExists{null:}}{\lifFileExists{null:}}{\lifFileExists{null:}}{\lifFileExists{null:}}{\lifFileExists{null:}}{\lifFileExists{null:}}{\lifFileExists{null:}}{\lifFileExists{null:}}{\lifFileExists{null:}}{\lifFileExists{null:}}{\lifFileExists{null:}}{\lifFileExists{nul
440
441 \def\kpsewhich#1#2{\begingroup
442
                      \def\@Space{ }
443
                      \edef\kpsewhich@cmd{"|kpsewhich #2"}
444
                      \everyeof{\noexpand}
                      \catcode'\\=12
445
                      \edef#1{\@@input\kpsewhich@cmd\@Space}
446
```

```
\trimstring#1
447
     \global\let#1#1
448
449 \endgroup
450
451\,\text{\%} main directory
452
453 \edef\oldpercent catcode {\the\catcode'\%}
454 \catcode'\%=12
455 \let\percent%
456 \catcode'\%=\oldpercentcatcode
457
458 \edef\pwd@cmd{\if@iswindows@ -expand-var \percent CD\percent\else -var-value PWD\fi}
459 \kpsewhich\stex@maindir\pwd@cmd
460 \if@iswindows@\windows@to@path\stex@maindir\fi
461
462 \neq 1#2{
       \edef\filename@oldpath{#1}
463
       \verb|\StrCount| filename@oldpath|@Slash[\filename@lastslash]|
464
465
       \ifnum\filename@lastslash>0
466
            \verb|\StrBehind[\filename@lastslash] $$ filename@oldpath\\@Slash[\filename@oldpath] $$
            \csedef{#2}{\filename@oldpath}
467
       \else
468
            \csedef{#2}{\filename@oldpath}
469
       \fi
470
471 }
472
473 \def\path@dropextension#1#2{
       \path@filename{#1}{dropextension@temp}
474
       \StrCount\dropextension@temp\@Dot[\dropextension@lastdot]
475
       \ifnum\dropextension@lastdot>0
476
            \StrBehind[\dropextension@lastdot]\dropextension@temp\@Dot[\dropextension@ext]
477
478
            \StrLen\dropextension@ext[\dropextension@lastdot]
479
            \StrGobbleRight{#1}{\the\numexpr\dropextension@lastdot+1\@Space}[\dropextension@temp]
            \trimstring\dropextension@temp
480
            \csedef{#2}{\dropextension@temp}
481
482
       \else
            \csedef{#2}{#1}
483
484
       \fi
485 }
487 (/package)
```

Change History

| v1.0 | v1.1 |
|-----------------------------|--|
| | General: adding \baseURI from |
| General: First Version with | ${\tt omdoc.sty} \ {\tt and} \ {\tt \ defpath} \ {\tt from}$ |
| Documentation 1 | modules sty 1 |

References

[BFM05] Tim Berners-Lee, Roy T. Fielding, and Larry Masinter. *Uniform Resource Identifier (URI): Generic Syntax*. RFC 3986. Internet Engineering Task Force (IETF), 2005. URL: http://www.ietf.org/rfc/rfc3986.txt.