

pathsuris.sty: Paths and URIs for \TeX *

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

October 2, 2020

Abstract

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

Contents

1	User Interface	2
1.1	Base URIs	2
1.2	Using Absolute Paths	2
1.3	Path Canonicalization	2
1.4	URI splitting	2
2	The Implementation	4
2.1	Base URIs	4
2.2	Using Absolute Paths	4
2.3	Path Canonicalization	4
2.4	URIs	6

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

1 User Interface

1.1 Base URIs

`\baseURI` `\baseURI`¹

1.2 Using Absolute Paths

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 cname \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/LATEX` 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 L^AT_EXML transformation, which (if `\langle baseURI \rangle` is specified) resolves `\langle path \rangle` to an absolute URI according to [BerFieMas:05].

1.3 Path Canonicalization

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

path	canonicalized path	expected
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/.. /..		

1.4 URI splitting

By calling `\seturi[\meta{macroname}]{\langle path \rangle}`, the URI will be plit into `\macronamescheme`, `\macronameauthority`, `\macronamepath`, `\macronamequery`

¹EdNOTE: document it

and `\macronamefragment`, as in the following example. If the optional `macroname` is not provided, the default name is `pathsuris@curruri@`.

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

yields:

macro	value
<code>\myurischeme</code>	http
<code>\myuriauthority</code>	this.isatest
<code>\myuripath</code>	foo/bar/
<code>\myuriquery</code>	query
<code>\myurifragment</code>	fragment

2 The Implementation

```

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

```

2.1 Base URIs

`\baseURI` On the L^AT_EX 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 form 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/smgglom/sets}

```

will generate `/path/to/localmh/MathHub/source/smgglom/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'\#}%
11   \catcode'\#=12\relax%
12   \edef\pathsuris@oldcatcode@slash{\the\catcode'\/%}
13   \catcode'\/=12\relax%
14   \edef\pathsuris@oldcatcode@colon{\the\catcode'\:%}
15   \catcode'\:=12\relax%
16   \edef\pathsuris@oldcatcode@qm{\the\catcode'\?}%
17   \catcode'\?=12\relax%
18 }
19 \def\pathsuris@resetcatcodes{%
20   \catcode'\#\pathsuris@oldcatcode@hash\relax%
21   \catcode'\/>\pathsuris@oldcatcode@slash\relax%
22   \catcode'\:\pathsuris@oldcatcode@colon\relax%
23   \catcode'\?\pathsuris@oldcatcode@qm\relax%
24 }

```

We define some macros for later comparison.

```

25 \def\@ToTop{..}
26 \def\@Slash{/}
27 \def\@Colon{:}
28 \def\@QuestionMark{?}
29 \def\@ToHere{.}
30
31 \pathsuris@setcatcodes
32 \def\@Fragment{#}
33 \pathsuris@resetcatcodes

```

Implement \@cpath.

\@cpath

```

34 \def\@cpath#1{%
35   \edef\pathsuris@temp{#1}%
36   \def\@CanPath{}%
37   \IfBeginWith\pathsuris@temp\@Slash{%
38     \@cpath@loop%
39     \edef\@CanPath{\@Slash\@CanPath}%
40   }{%
41     \@cpath@loop%
42   }%
43   \IfEndWith\@CanPath\@Slash{%
44     \ifx\@CanPath\@Slash\else%
45       \StrGobbleRight\@CanPath1[\@CanPath]%
46     \fi%
47   }{}%
48 }
49
50 \def\@cpath@loop{%
51   \IfSubStr\pathsuris@temp\@Slash{%
52     \StrCut\pathsuris@temp\@Slash\pathsuris@temp@a\pathsuris@temp%
53     \ifx\pathsuris@temp@a\@ToTop%
54       \ifx\@CanPath\@empty%
55         \edef\@CanPath{\@ToTop}%
56       \else%
57         \edef\@CanPath{\@CanPath\@Slash\@ToTop}%
58       \fi%
59     \@cpath@loop%
60   \else%
61     \IfBeginWith\pathsuris@temp\@ToTop{%
62       \StrBehind{\pathsuris@temp}{\@ToTop}[\pathsuris@temp]%
63       \IfBeginWith\pathsuris@temp\@Slash{%
64         \edef\pathsuris@temp{\@CanPath\pathsuris@temp}%
65       }{%
66         \ifx\@CanPath\@empty\else%
67           \edef\pathsuris@temp{\@CanPath\@Slash\pathsuris@temp}
68         \fi%
69       }%

```

```

70         \def\@CanPath{}%
71         \@cpath@loop%
72     }{%
73         \ifx\@CanPath\@empty%
74             \edef\@CanPath{\pathsuris@temp@a}%
75         \else%
76             \edef\@CanPath{\@CanPath\@Slash\pathsuris@temp@a}%
77         \fi%
78         \@cpath@loop
79     }%
80 \fi%
81 }{
82     \ifx\@CanPath\@empty%
83         \edef\@CanPath{\pathsuris@temp}%
84     \else%
85         \edef\@CanPath{\@CanPath\@Slash\pathsuris@temp}%
86     \fi%
87 }%
88 }

```

Implement `\cpath` to print the canonicalized path.

`\cpath`

```

89 \newcommand\cpath[1]{%
90     \@cpath{#1}%
91     \@CanPath%
92 }

```

2.4 URI splitting

`\seturi@`

```

93 \newif\if@pathsuris@done@
94
95 \def\seturi@[#1]#2{%
96     \@pathsuris@done@false%
97     \def\pathsuris@prefix@temp{#1}%
98     \edef\pathsuris@curruri{#2}%
99     \edef\pathsuris@temp{\pathsuris@curruri}%
100    \def\pathsuris@curruri@scheme{}%
101    \def\pathsuris@curruri@authority{}%
102    \def\pathsuris@curruri@path{}%
103    \def\pathsuris@curruri@query{}%
104    \def\pathsuris@curruri@fragment{}%
105    % scheme
106    \IfSubStr{\pathsuris@temp}{\@Colon}{%
107        % TODO check for valid scheme
108        \StrBefore{\pathsuris@temp}{\@Colon}[\pathsuris@curruri@scheme]%
109        \StrBehind{\pathsuris@temp}{\@Colon}[\pathsuris@temp]%
110    }{}%

```

```

111 % authority
112 \IfBeginWith{\pathsuris@temp}{\@Slash\@Slash}{%
113     \StrBehind{\pathsuris@temp}{\@Slash\@Slash}[\pathsuris@temp]%
114     \IfSubStr{\pathsuris@temp}{\@Slash}{%
115         \StrBefore{\pathsuris@temp}{\@Slash}[\pathsuris@curruri@authority]%
116         \StrBehind{\pathsuris@temp}{\@Slash}[\pathsuris@temp]%
117         % TODO userinfo,host,port
118     }{%
119         % TODO query,fragment
120         \edef\pathsuris@curruri@authority{\pathsuris@temp}%
121         \@pathsuris@done@true%
122     }%
123 }{}%
124 % path, query, fragment
125 \if@pathsuris@done@else%
126     \IfSubStr{\pathsuris@temp}{\@QuestionMark}{%
127         % path
128         \StrBefore{\pathsuris@temp}{\@QuestionMark}[\pathsuris@curruri@path]%
129         \StrBehind{\pathsuris@temp}{\@QuestionMark}[\pathsuris@temp]%
130         % query,fragment
131         \IfSubStr{\pathsuris@temp}{\@Fragment}{%
132             \StrBefore{\pathsuris@temp}{\@Fragment}[\pathsuris@curruri@query]%
133             \StrBehind{\pathsuris@temp}{\@Fragment}[\pathsuris@curruri@fragment]%
134         }{%
135             \edef\pathsuris@curruri@query{\pathsuris@temp}%
136         }%
137     }{%
138         % path,fragment
139         \IfSubStr{\pathsuris@temp}{\@Fragment}{%
140             \StrBefore{\pathsuris@temp}{\@Fragment}[\pathsuris@curruri@path]%
141             \StrBehind{\pathsuris@temp}{\@Fragment}[\pathsuris@curruri@fragment]%
142         }{%
143             \edef\pathsuris@curruri@path{\pathsuris@temp}%
144         }%
145     }%
146 \fi%
147 %drop trailing slash of path
148 %\IfEndWith{\pathsuris@curruri@path}{\@Slash}{%
149 %    \StrGobbleRight{\pathsuris@curruri@path}{1}[\pathsuris@curruri@path]
150 %}%{}%
151 %
152 %\edef\pathsuris@curruri@path{\cpath{\pathsuris@curruri@path}}%
153 \ifx\pathsuris@prefix@temp\empty\else%
154     \expandafter\edef\csname \pathsuris@prefix@temp scheme\endcsname{\pathsuris@curruri@sch
155     \expandafter\edef\csname \pathsuris@prefix@temp authority\endcsname{\pathsuris@curruri@
156     \expandafter\edef\csname \pathsuris@prefix@temp path\endcsname{\pathsuris@curruri@path}
157     \expandafter\edef\csname \pathsuris@prefix@temp query\endcsname{\pathsuris@curruri@quer
158     \expandafter\edef\csname \pathsuris@prefix@temp fragment\endcsname{\pathsuris@curruri@f
159 \fi%
160 }

```

\seturi

```
161 \newrobustcmd\seturi[1][ ]{%
162   \pathsuris@setcatcodes%
163   \expandafter\pathsuris@resetcatcodes\seturi@[#1]%
164 }
```

auxiliary code:

```
165 \def\@Space{ }
166 \def\trimstring#1{
167   \edef\pathsuris@trim@temp{#1}
168   \IfBeginWith\pathsuris@trim@temp\@Space{
169     \StrGobbleLeft\pathsuris@trim@temp1[#1]
170     \trimstring{#1}
171   }{
172     \IfEndWith\pathsuris@trim@temp\@Space{
173       \StrGobbleRight\pathsuris@trim@temp1[#1]
174       \trimstring{#1}
175     }{
176       \edef#1{\pathsuris@trim@temp}
177     }
178   }
179 }
180
181 % windows paths
182
183 \catcode'\.=0
184 .catcode'\.=12
185 .let.\@BackSlash\
186 .catcode'\.=0
187 \catcode'\.=12
188
189 \newif\if@windowstopath@inpath@
190 \def\windows@to@path#1{
191   \@windowstopath@inpath@false
192   \def\windows@temp{}
193   \edef\windows@path{#1}
194   \ifx\windows@path\@empty\else
195     \expandafter\windows@path@loop\windows@path\windows@path@end
196   \fi
197   \let#1\windows@temp
198 }
199 \def\windows@path@loop#1#2\windows@path@end{
200   \def\windows@temp@b{#2}
201   \ifx\windows@temp@b\@empty
202     \def\windows@continue{}
203   \else
204     \def\windows@continue{\windows@path@loop#2\windows@path@end}
205   \fi
206   \if@windowstopath@inpath@
```



```

207         \ifx#1\@BackSlash
208             \edef\windows@temp{\windows@temp\@Slash}
209         \else
210             \edef\windows@temp{\windows@temp#1}
211         \fi
212     \else
213         \ifx#1:
214             \edef\windows@temp{\@Slash\windows@temp}
215             \@windowstopath@inpath@true
216         \else
217             \edef\windows@temp{\windows@temp#1}
218         \fi
219     \fi
220     \windows@continue
221 }
222
223 \def\path@to@windows#1{
224     \@windowstopath@inpath@false
225     \def\windows@temp{}
226     \edef\windows@path{#1}
227     \edef\windows@path{\expandafter\@gobble\windows@path}
228     \ifx\windows@path\@empty\else
229         \expandafter\path@windows@loop\windows@path\windows@path@end
230     \fi
231     \let#1\windows@temp
232 }
233 \def\path@windows@loop#1#2\windows@path@end{
234     \def\windows@temp@b{#2}
235     \ifx\windows@temp@b\@empty
236         \def\windows@continue{}
237     \else
238         \def\windows@continue{\path@windows@loop#2\windows@path@end}
239     \fi
240     \if@windowstopath@inpath@
241         \ifx#1/
242             \edef\windows@temp{\windows@temp\@BackSlash}
243         \else
244             \edef\windows@temp{\windows@temp#1}
245         \fi
246     \else
247         \ifx#1/
248             \edef\windows@temp{\windows@temp:\@BackSlash}
249             \@windowstopath@inpath@true
250         \else
251             \edef\windows@temp{\windows@temp#1}
252         \fi
253     \fi
254     \windows@continue
255 }
256

```

257 `</package>`