`pathsuris.sty`: Paths and URIs for sTeX*

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

October 15, 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

---

*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
\defpath   supplies the \defpath macro: \defpath[⟨baseURI⟩]{⟨cname⟩}{⟨path⟩} defines a command, so that \⟨csname⟩{⟨name⟩} expands to ⟨path⟩/⟨name⟩. 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 ⟨baseURI⟩ is for the LATEXML transformation, which (if ⟨baseURI⟩ is specified) resolves ⟨path⟩ to an absolute URI according to [BFM05, section 5.2].

1.3 Path Canonicalization

By calling \@cpath{⟨path⟩}, the canonicalized path will be stored in \@CanPath.
\cpath   To print a canonicalized path, simply use \cpath{⟨path⟩}. 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}]{⟨path⟩}, the URI will be plit into \macronamescheme, \macronameauthority, \macronamepath, \macronamequery

---

[1] EDNOTE: 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
`\myurischeme`	http
`\myuriauthority`	this.isatest
`\myuripath`	foo/bar/
`\myuriquery`	query
`\myurifragment`	fragment

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

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

`\seturi`

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

auxiliary code:

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

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

256 ⟨/package⟩

Change History

References

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