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

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

September 30, 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]

EdN: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[`⟨*baseURI*⟩`]{`⟨*cname*⟩`}{`⟨*path*⟩`}` defines a command, so that `\`⟨*csname*⟩`{`⟨*name*⟩`}` expands to ⟨*path*⟩`/`⟨*name*⟩. So we could have used

\defpath

```
\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 [**BerFieMas:05**].

1.3 Path Canonicalization

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

\cpath

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

```
⟨*package⟩
\RequirePackage{stex-base}
\RequirePackage{xstring}
\RequirePackage{etoolbox}
```

2.1 Base URIs

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

```
\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`.

```
\newrobustcmd\defpath[3][]{%
  \expandafter\newcommand\csname #2\endcsname[1]{#3/##1}%
}%
```

2.3 Path Canonicalization

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

```
\def\pathsuris@setcatcodes{%
    \edef\pathsuris@oldcatcode@hash{\the\catcode`\#}%
    \catcode`\#=12\relax%
    \edef\pathsuris@oldcatcode@slash{\the\catcode`\/}%
    \catcode`\/=12\relax%
    \edef\pathsuris@oldcatcode@colon{\the\catcode`\:}%
    \catcode`\:=12\relax%
    \edef\pathsuris@oldcatcode@qm{\the\catcode`\?}%
    \catcode`\?=12\relax%
}
\def\pathsuris@resetcatcodes{%
    \catcode`\#\pathsuris@oldcatcode@hash\relax%
    \catcode`\/\pathsuris@oldcatcode@slash\relax%
    \catcode`\:\pathsuris@oldcatcode@colon\relax%
    \catcode`\?\pathsuris@oldcatcode@qm\relax%
}
```

We define some macros for later comparison.

```
\def\@ToTop{..}
\def\@Slash{/}
\def\@Colon{:}
\def\@QuestionMark{?}
\def\@ToHere{.}

\pathsuris@setcatcodes
\def\@Fragment{#}
\pathsuris@resetcatcodes
```

Implement `\@cpath`.

`\@cpath`

```
\def\@cpath#1{%
    \edef\pathsuris@temp{#1}%
    \def\@CanPath{}%
    \@cpath@loop
}

\def\@cpath@loop{%
    \IfSubStr\pathsuris@temp\@Slash{%
        \StrCut\pathsuris@temp\@Slash\pathsuris@temp@a\pathsuris@temp%
        \ifx\pathsuris@temp@a\@ToTop%
            \ifx\@CanPath\@empty%
                \edef\@CanPath{\@ToTop}%
            \else%
                \edef\@CanPath{\@CanPath\@Slash\@ToTop}%
            \fi%
            \@cpath@loop%
        \else%
        \IfBeginWith\pathsuris@temp\@ToTop{%
            \StrBehind{\pathsuris@temp}{\@ToTop}[\pathsuris@temp]%
            \IfBeginWith\pathsuris@temp\@Slash{%
                \edef\pathsuris@temp{\@CanPath\pathsuris@temp}%
            }{%
                \ifx\@CanPath\@empty\else%
                    \edef\pathsuris@temp{\@CanPath\@Slash\pathsuris@temp}
                \fi%
            }%
            \def\@CanPath{}%
            \@cpath@loop%
        }{%
            \ifx\@CanPath\@empty%
                \edef\@CanPath{\pathsuris@temp@a}%
            \else%
                \edef\@CanPath{\@CanPath\@Slash\pathsuris@temp@a}%
            \fi%
            \@cpath@loop
        }%
        \fi%
    }{
        \ifx\@CanPath\@empty%
            \edef\@CanPath{\pathsuris@temp}%
        \else%
            \edef\@CanPath{\@CanPath\@Slash\pathsuris@temp}%
        \fi%
    }%
}
```

Implement `\cpath` to print the canonicalized path.

`\cpath`

```
\newcommand\cpath[1]{%
    \@cpath{#1}%
    \@CanPath%
}
```

2.4 URI splitting

`\seturi@`

```
\newif\if@pathsuris@done@

\def\seturi@[#1]#2{%
    \@pathsuris@done@false%
    \def\pathsuris@prefix@temp{#1}
    \edef\pathsuris@curruri{#2}%
    \edef\pathsuris@temp{\pathsuris@curruri}%
    \def\pathsuris@curruri@scheme{}%
    \def\pathsuris@curruri@authority{}%
    \def\pathsuris@curruri@path{}%
    \def\pathsuris@curruri@query{}%
    \def\pathsuris@curruri@fragment{}%
    % scheme
    \IfSubStr{\pathsuris@temp}{\@Colon}{%
        % TODO check for valid scheme
        \StrBefore{\pathsuris@temp}{\@Colon}[\pathsuris@curruri@scheme]%
        \StrBehind{\pathsuris@temp}{\@Colon}[\pathsuris@temp]%
    }{}%
    % authority
    \IfBeginWith{\pathsuris@temp}{\@Slash\@Slash}{%
        \StrBehind{\pathsuris@temp}{\@Slash\@Slash}[\pathsuris@temp]%
        \IfSubStr{\pathsuris@temp}{\@Slash}{%
            \StrBefore{\pathsuris@temp}{\@Slash}[\pathsuris@curruri@authority]%
            \StrBehind{\pathsuris@temp}{\@Slash}[\pathsuris@temp]%
            % TODO userinfo,host,port
        }{%
            % TODO query,fragment
            \edef\pathsuris@curruri@authority{\pathsuris@temp}%
            \@pathsuris@done@true%
        }%
    }{}%
    % path, query, fragment
    \if@pathsuris@done@\else%
        \IfSubStr{\pathsuris@temp}{\@QuestionMark}{%
            % path
            \StrBefore{\pathsuris@temp}{\@QuestionMark}[\pathsuris@curruri@path]%
            \StrBehind{\pathsuris@temp}{\@QuestionMark}[\pathsuris@temp]%
            % query,fragment
            \IfSubStr{\pathsuris@temp}{\@Fragment}{%
                \StrBefore{\pathsuris@temp}{\@Fragment}[\pathsuris@curruri@query]%
                \StrBehind{\pathsuris@temp}{\@Fragment}[\pathsuris@curruri@fragment]%
            }{%
                \edef\pathsuris@curruri@query{\pathsuris@temp}%
            }%
        }{%
            % path,fragment
            \IfSubStr{\pathsuris@temp}{\@Fragment}{%
                \StrBefore{\pathsuris@temp}{\@Fragment}[\pathsuris@curruri@path]%
                \StrBehind{\pathsuris@temp}{\@Fragment}[\pathsuris@curruri@fragment]%
            }{%
                \edef\pathsuris@curruri@path{\pathsuris@temp}%
            }%
        }%
    \fi%
    %drop trailing slash of path
    %\IfEndWith{\pathsuris@curruri@path}{\@Slash}{%
    %    \StrGobbleRight{\pathsuris@curruri@path}{1}[\pathsuris@curruri@path]
    %}{}%
    %
    %\edef\pathsuris@curruri@path{\cpath{\pathsuris@curruri@path}}%
    \ifx\pathsuris@prefix@temp\@empty\else%
        \expandafter\edef\csname \pathsuris@prefix@temp scheme\endcsname{\pathsuris@curruri@sch
        \expandafter\edef\csname \pathsuris@prefix@temp authority\endcsname{\pathsuris@curruri@
        \expandafter\edef\csname \pathsuris@prefix@temp path\endcsname{\pathsuris@curruri@path}
        \expandafter\edef\csname \pathsuris@prefix@temp query\endcsname{\pathsuris@curruri@quer
        \expandafter\edef\csname \pathsuris@prefix@temp fragment\endcsname{\pathsuris@curruri@f
    \fi%
}
```

\seturi

```
\newrobustcmd\seturi[1][]{%
    \pathsuris@setcatcodes%
    \expandafter\pathsuris@resetcatcodes\seturi@[#1]%
}
⟨/package⟩
```