

pathsuris.sty: Paths and URIs for \TeX *

Jinbo Zhang, Michael Kohlhase
Jacobs University, Bremen

February 18, 2016

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
2	The Implementation	3
2.1	Base URIs	3
2.2	Using Absolute Paths	3
2.3	Path Canonicalization	3

*Version v1.1 (last revised 2016/02/18)

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[<baseURI>]{<cname>}{<path>}` defines a command, so that `\<cname>{<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 L^ATeXML 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`. 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/../..		

¹EdNOTE: document it

2 The Implementation

```

1 <*package>
2 \RequirePackage{xstring}
3 \RequirePackage{forloop}
4 \RequirePackage{calc}
5 \RequirePackage{etoolbox}

```

2.1 Base URIs

`\baseURI` On the L^AT_EX side we do nothing (for the moment).

```

6 \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`.

```

7 \newrobustcmd\defpath[3] [] {%
8   \expandafter\newcommand\csname #2\endcsname[1]{#3/##1}%
9 }%

```

2.3 Path Canonicalization

We first create some counters. `AddrNum` will count the number of sections in the input path, `iLoop` will be used as the loop iterator, `iName` will be used for generating names such as `Addri`, `Addrii`, `RealAddrNum` will count the number of sections in the canonicalized path, `Cutable` will count the number of sections besides ...

```

10 \newcounter{AddrNum}
11 \newcounter{iLoop}
12 \newcounter{iName}
13 \newcounter{RealAddrNum}
14 \newcounter{Cutable}

```

We define two macros for later comparison.

```

15 \def\@ToTop{..}
16 \def\@Slash{/}

```

Then we split the input path.

```

17 \def\@MultiAddrs#1/#2\@nil{%
18   \def\CurArg{#1}%
19   \def\NextArg{#2}%

```

```

20 \ifx\@empty\CurArg% for the first one
21 \else%
22   \stepcounter{AddrNum}%
23   \expandafter\edef\csname Addr\roman{AddrNum}\endcsname{#1}% storing
24 \fi%
25 \ifx\@empty\NextArg% for the last one
26   \let\next@gobble%
27 \fi%
28 \next#2\@nil% recursion
29 }%

Implement \@cpath.
30 \def\@cpath#1{%
31   \let\next\@MultiAddrs%
32   \setcounter{AddrNum}{0}%
33   \setcounter{iLoop}{0}%
34   \setcounter{iName}{0}%
35   \setcounter{RealAddrNum}{0}%
36   \setcounter{Cutable}{0}%
37   \def\@CurrPath{}%
38   \def\@CanPath{}%
39   \def\@TempPath{}%
40   \def\@Rubbish{}%
41   \expandafter\next#1/\@nil% recursion starts
42   \forloop{iLoop}{0}{\value{iLoop} < \value{AddrNum}}{%
43     \stepcounter{iName}%
44     \edef\@CurrPath{\csname Addr\roman{iName}\endcsname}%
45     \ifx\@CurrPath\@ToTop%
46     \ifnum\value{Cutable} = 0%
47       \edef\@CanPath{\@CanPath\csname Addr\roman{iName}\endcsname/}%
48       \stepcounter{RealAddrNum}%
49     \else%
50       % cut the last part, and add a slash at the end
51       \StrCut[\value{RealAddrNum}]{/\@CanPath}{/}\@TempPath\@Rubbish%
52       \StrCut[1]{\@TempPath/}{/}\@Rubbish\@CanPath%
53       \addtocounter{RealAddrNum}{-1}%
54       \addtocounter{Cutable}{-1}%
55     \fi%
56   \else%
57     \edef\@CanPath{\@CanPath\csname Addr\roman{iName}\endcsname/}%
58     \stepcounter{RealAddrNum}%
59     \stepcounter{Cutable}%
60   \fi%
61 }%
62 \StrCut[\value{RealAddrNum}]{\@CanPath}{/}\@CanPath\@Rubbish% cut last /
63 }%

Implement \cpath to print the canonicalized path.
64 \newcommand\cpath[1]{% print canonical path
65 \@cpath{#1}%
66 \@CanPath%

```

```
67 }%
68 </package>
```

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>.