

MathHub Support for \LaTeX^*

Michael Kohlhasse
FAU Erlangen-Nürnberg
<http://kwarc.info/kohlhasse>

October 13, 2019

Abstract

The `sref` package is part of the \LaTeX collection, a version of $\text{\TeX}/\text{\LaTeX}$ that allows to markup $\text{\TeX}/\text{\LaTeX}$ documents semantically without leaving the document format, essentially turning $\text{\TeX}/\text{\LaTeX}$ into a document format for mathematical knowledge management (MKM).

The `mathhub` packages extend \LaTeX with support for MathHub file system layout, which has co-evoled with the MathHub.info portal for active documents, but is useful for organizing collections of \LaTeX documents in its own right.

Contents

1	Introduction	3
2	The User Interface	3
2.1	<code>mathhub.sty</code> : General Infrastructure	3
2.2	<code>omdoc--mh.sty</code> : MH Document Infrastructure	4
2.3	<code>modules-mh.sty</code> : MH Variants for Modules	4
2.4	<code>omtext-mh.sty</code> : MH Variants for OMText	5
2.5	<code>smultiling-mh.sty</code> : MH Variants for Multilinguality	6
2.6	<code>structview-mh.sty</code> : MH Variants for Structures and Views	6
2.7	<code>mikoslides-mh.sty</code> : Support for MiKo Slides	6
2.8	<code>problem-mh.sty</code> : Support for Problems	7
2.9	<code>hwexam-mh.sty</code> : Support for Assignments	7
2.10	<code>lstmh.sty</code> : Support for Listings	7
3	Limitations	7

*Version ? (last revised ?)

4	Implementation	8
4.1	<code>mathhub.sty</code> : General Infrastructure	8
4.2	<code>omdoc--mh.sty</code> : MH Document Infrastructure	9
4.3	<code>modules-mh.sty</code> : MH Variants for Modules	9
4.4	<code>omtext-mh.sty</code> : MH Variants for OMText	10
4.5	<code>smultiling-mh.sty</code> : MH Variants for Multilinguality	11
4.6	<code>structview-mh.sty</code> : MH Variants for Structures and Views	12
4.7	<code>mikoslides-mh.sty</code> : Support for MiKo Slides	13
4.8	<code>problem-mh.sty</code> : Support for Problems	14
4.9	<code>hwexam-mh.sty</code> : Support for Assignments	14
4.10	<code>tikzinput-mh.sty</code> : Support for Assignments	14
4.11	<code>lstmh.sty</code> : Support for Listings	15

1 Introduction

As \LaTeX files tend to be highly interlinked semantically one of the most important practical problems to solve for managing larger collections is the management of (relative or absolute) paths. The `mathhub` package provides an infrastructure for supporting a regular \sim manageable file system layout schema that has co-evolved with the MathHub.info portal for active documents, but is useful for organizing collections of \LaTeX documents in its own right. In particular, since the layout scheme is supported by the `lmh` tool (local `mathhub`) [lmh], which automates many management tasks.

MathHub (<http://MathHub.info>), is a portal and archive for flexiformal mathematics. It hosts much of the \LaTeX content MathHub on GIT repositories (public and private escrow) for mathematical documentation projects. MathHub supports online and offline (via `lmh`) authoring and document development infrastructure, and a rich, interactive reading interface.

The **MathHub file system layout** has a **MathHub root folder** (e.g. `~/localmh/MathHub`) which contains all \LaTeX sources. These are organized in a two-level folder system that is compatible by GIT repository managers like GitHub [GH] and GitLab [GL]. Even though it is not necessary for the `mathhub` package we will assume that these are GIT repositories, which have names of the form $\langle group \rangle / \langle repo \rangle$, where $\langle group \rangle$ is a MathHub-unique repository group and $\langle repo \rangle$ a repository name that is $\langle group \rangle$ -unique. The file and directory structure of a repository is arbitrary – except that it starts with the directory `source` because they are Math Archives in the sense of [Hor+11]. But this structure can be hidden from the \LaTeX author with MathHub-enabled versions of the \LaTeX macros (let’s call them **mh-variants**), which are defined in the `mh`-packages of the `mathhub` bundle, which we document in this manual.

1

2 The User Interface

2.1 mathhub.sty: General Infrastructure

Generally, the \TeX formatter `pdflatex` needs to know the file system paths of the referenced \LaTeX files: usually long relative paths. The `pathsuris` package [KGA18] from the \LaTeX bundle makes this somewhat more palatable by supplying the `\defpath` macro, which we can use to set the MathHub root path, e.g. by `\defpath{MathHub}{/user/foo/localmh/MathHub}` (we will assume this setting for all examples below).

The next information needed for referencing in the MathHub file system layout is the repository of the referencing file and that of the referenced one. The latter will be part of the mh-variants of the referencing commands introduced below, the former, can be declared by the `\mhcurrentrepos` macro relative to the MathHub

`\mhcurrentrepos`

¹EdNOTE: document lmh here, how to install, what it does (build system)

root path. Thus a typical top-level \TeX file will have lines like the following in the preamble:

```
\defpath{MathHub}{/user/foo/localmh/MathHub}
\mhcurrentrepos{group/repos}
```

They declare that it resides at the path `/user/foo/localmh/MathHub/group/repos` and declares the `MathHub` root path. But this fixed declaration makes the \TeX files less mobile, therefore it is a better idea to externalize these declarations into an external file (usually called `localpaths.tex` that is not under GIT control since it contains system-specific path information) and `\input` that in the \TeX file instead. Indeed `lmh` can generate these files automatically, which simplifies the management significantly.

Given a systematic grouping in the `MathHub` file layout scheme, \TeX files in the same repository (and often even in the same group) share much of the preamble material. Thus it makes sense to centralize that in external (shared) files and situate it at the group and repository levels: at the group level. For the group level, the `MathHub` file system layout uses a special repository `<group>/meta-inf/lib` and at the repository level we use `<group>/<repos>/lib` for such files. The `\libinput` macro supports this practice: `\libinput{<filename>}` macro inputs the files `<group>/meta-inf/lib/<filename>` and then `<group>/<repos>/lib/<filename>` if they exist. Thus a typical top-level \TeX file has the following lines in the preamble:

`\libinput`

```
\input{localpaths}
\libinput{preamble}
```

`\libusepackage` The `\libusepackage` is analogous. ²

EdN:2

2.2 omdoc--mh.sty: MH Document Infrastructure

`\addmhbibresource`

The `\addmhbibresource` macro is a variant of `\addbibresource` from `bibLaTeX` with repository support. Concretely, `\addmhbibresource[<repos>]{<path>}` expands to `\addbibresource{<MathHub>/<reponame>/<path>}`, where `<reponame>` is `<repo>` if that is non-empty and the current repository else. Note that in contrast to the other MH variants, this does not add the `/source/` into the path, since `bibTeX` files are often put into the `lib` directory parallel to `source`.

2.3 modules-mh.sty: MH Variants for Modules

`\importmhmodule`

The `\importmhmodule` macro is a variant of `\importmodule` with repository support. Instead of writing

```
\importmodule[load=\MathHub{fooMH/bar/source/baz/foobar}]{foobar}
```

²EdNOTE: explain the `pre.tex` and `post.tex` setup for `sTeX` modules (see the `omdoc` package)

we can simply write (assuming that `\MathHub` is defined as above)

```
\importmhmodule[repos=fooMH/bar,path=baz/foobar]{foobar}
```

Note that the `\importmhmodule` form is more semantic, which allows more advanced document management features in `MathHub`.

If `baz/foobar` is the “current module”, i.e. if we are on the `MathHub` path `...MathHub/fooMH/bar...`, then stating the repository in the first optional argument is redundant, so we can just use

```
\importmhmodule[path=baz/foobar]{foobar}
```

if no file needs to be loaded, `\importmhmodule` is the same as `\importmodule`.

`\mhcurrentrepos` Of course, neither `LATEX` nor `LATEXML` know about the repositories when they are called from a file system, so we can use the `\mhcurrentrepos` macro to tell them. But this is only needed to initialize the infrastructure in the driver file. In particular, we do not need to set it in each module, since the `\importmhmodule` macro sets the current repository automatically.

`\usemhmodule` The `\usemhmodule` is the analog to `\usemodule`.

`\mhinputref` For this, the `modules` package supplies the mh-variants `\mhinputref` and
`\mhinput` `\mhinput` of the `\inputref` macro introduced above and normal `LATEX` `\input` macro.

Caveat if you want to use the `MathHub` support macros, then every time a module is imported or a document fragment is included from another repos, the mh-variant `\importmhmodule` must be used, so that the “current repository” is set accordingly. To be exact, we only need to use mh-variants, if the imported module or included document fragment use mh-variants.

2.4 omtex-mh.sty: MH Variants for OMText

`\mhgraphics` The `\mhgraphics` macro is a variant of `\mycgraphics` with repository support. Instead of writing

```
\mycgraphics{\MathHub{fooMH/bar/source/baz/foobar}}
```

we can simply write (assuming that `\MathHub` is defined as above)

```
\mhgraphics[fooMH/bar]{baz/foobar}
```

Note that the `\mhgraphics` form is more semantic, which allows more advanced document management features in `MathHub`.

2.5 smultiling-mh.sty: MH Variants for Multilinguality

`mhmodsig` The `mhmodsig` and `mhmodnl` environments are the MH variants of the `modsig` and `modnl` environments from the `smultiing` package. Just as in the other MH packages, `mhmodnl` takes additional `repos` and `path` keys and combine them to load key of `modnl`. Instead of writing

```
\begin{modnl}[load=\MathHub{fooMH/bar/source/baz/foobar}]{foobar}{en}
```

we can simply write (assuming that `\MathHub` is defined as above)

```
\begin{modnl}[repos=fooMH/bar,path=baz/foobar]{foobar}{en}
```

`mhmodsig` is just a notational variant of `modsig` that allows to keep the sources uniform.

2.6 structview-mh.sty: MH Variants for Structures and Views

EdN:3

3

2.7 mikoslides-mh.sty: Support for MiKo Slides

`\mhframeimage` The `\mhframeimage` macro is a variant of `\frameimage` with repository support. Instead of writing

```
\frameimage{\MathHub{fooMH/bar/source/baz/foobar}}
```

we can simply write (assuming that `\MathHub` is defined as above)

```
\mhframeimage[fooMH/bar]{baz/foobar}
```

Note that the `\mhframeimage` form is more semantic, which allows more advanced document management features in `MathHub`.

If `baz/foobar` is the “current module”, i.e. if we are on the `MathHub` path `...MathHub/fooMH/bar...`, then stating the repository in the first optional argument is redundant, so we can just use

```
\mhframeimage{baz/foobar}
```

`\mhinputref*` If we want to transclude a the contents of a file as a note, we can use the `\mhinputref*` macro: `\mhinputref*[foo]{bar}` is equivalent to

```
\begin{note}
\mhinputref[foo]{bar}
\end{note}
```

³EdNOTE: needs to be documented

2.8 problem-mh.sty: Support for Problems

`\includemhproblem` The `\includemhproblem` macro is a variant of `\includeproblem` with repository support. Instead of writing

```
\includeproblem[pts=7]{\MathHub{fooMH/bar/source/baz/foobar}}
```

we can simply write (assuming that `\MathHub` is defined as above)

```
\includemhproblem[mhrepos=fooMH/bar,pts=7]{baz/foobar}
```

Note that the `\importmhproblem` form is more semantic, which allows more advanced document management features in MathHub.

2.9 hwexam-mh.sty: Support for Assignments

`\includemhassignment` The `\includemhassignment` macro is a variant of `\includeassignment` with repository support. Instead of writing

```
\includeassignment[number=3]{\MathHub{fooMH/bar/source/baz/foobar}}
```

we can simply write (assuming that `\MathHub` is defined as above)

```
\includemhassignment[mhrepos=fooMH/bar,number=3]{baz/foobar}
```

2.10 lstmh.sty: Support for Listings

`\lstinputmhlisting` The `\lstinputmhlisting` macro is a variant of `\lstinputlisting` with repository support. Instead of writing

```
\lstinputlisting[language=XML]{\MathHub{fooMH/bar/source/baz/foobar.xml}}
```

we can simply write (assuming that `\MathHub` is defined as above)

```
\lstinputmhlisting[mhrepos=fooMH/bar,language=XML]{baz/foobar.xml}
```

3 Limitations

In this section we document known limitations. If you want to help alleviate them, please feel free to contact the package author. Some of them are currently discussed in the `sTeX` GitHub repository [sTeX].

1. none reported yet.

4 Implementation

We need to set up the packages by requiring the `metakeys` package [Koh18] to be loaded (in the right version).

```
1 <*package>
2 \RequirePackage{keyval}
```

4.1 mathhub.sty: General Infrastructure

`\mhcurrentrepos` `\mhcurrentrepos` is used to initialize the current repository. If the repos has changed, it writes a call to the internal macro `\@mhcurrentrepos` for the aux file and calls it. So that the `\importmodule` calls there work with the correct repos.

```
3 \newcommand\mhcurrentrepos[1]{%
4   \edef\@test{#1}%
5   \ifx\@test\mh@currentrepos% if new dir = old dir
6     \relax% no need to change
7   \else%
8     \protected@write\@auxout{}\string\@mhcurrentrepos{#1}%
9   \fi%
10  \@mhcurrentrepos{#1}% define mh@currentrepos
11 }%
12 \newcommand\@mhcurrentrepos[1]{\edef\mh@currentrepos{#1}}%
```

`\libinput` the `\libinput` macro inputs from the `lib` directory of the MathHub repository and then the `meta-inf/lib` repos of the group, if they exist.

```
13 \def\modules@@first#1/#2;{#1}
14 \newcommand\libinput[1]{%
15   \edef\@group{\expandafter\modules@@first\mh@currentrepos;}
16   \edef\@infile{\MathHub{\@group/meta-inf/lib/#1}}
17   \IfFileExists\@infile{\input\@infile}{}%
18   \def\@libfile{\MathHub{\mh@currentrepos/lib/#1}}%
19   \IfFileExists\@infile {\IfFileExists\@libfile}{}%
20   {\PackageError{mathhub}
21     {Library file missing; cannot input #1.tex\MessageBreak%
22     Both \@libfile.tex\MessageBreak and \@infile.tex\MessageBreak%
23     do not exist}%
24   {Check whether the file name is correct}}}%
25 \IfFileExists\@libfile{\input\@libfile}{}}
```

`\libusepackage` the `\libusepackage` is analogous to `\libinput`

```
26 \newcommand\libusepackage[2][1]{%
27   \edef\@group{\expandafter\modules@@first\mh@currentrepos;}
28   \edef\@infile{\MathHub{\@group/meta-inf/lib/#2}}
29   \IfFileExists\@infile.sty{\usepackage[#1]{\@infile}}{%
30   \def\@libfile{\MathHub{\mh@currentrepos/lib/#2}}%
31   \IfFileExists {\@infile.sty}{\IfFileExists{\@libfile.sty}{}%
32   {\PackageError{mathhub}
33     {Library file missing; cannot use package #2.sty\MessageBreak%
```



```

34 Both \@libfile.sty\MessageBreak and \@inffile.sty\MessageBreak%
35 do not exist}%
36 {Check whether the file name is correct}}}}
37 \IfFileExists{\@libfile.sty}{\usepackage[#1]{\@libfile}}{}}
38 \</package>

```

4.2 omdoc--mh.sty: MH Document Infrastructure

```

39 <*omdoc>
40 \ProvidesPackage{omdoc-mh}[2019/03/20 v1.1 MathHub support for OMDoc Documents]
41 \RequirePackage{mathhub}

```

\addmhbibresource

```

42 \newcommand\addmhbibresource[2] [] {%
43   \def\@repos{#1}%
44   \ifx\@repos\empty%
45     \addbibresource{\MathHub{\mh@currentrepos/#2}}%
46   \else
47     \addbibresource{\MathHub{\@repos/#2}}%
48   \fi%
49   \ignorespaces}%
50 \</omdoc>

```

4.3 modules-mh.sty: MH Variants for Modules

```

51 <*modules>
52 \ProvidesPackage{modules-mh}[2019/03/20 v1.1 MathHub support for the sTeX modules package]
53 \RequirePackage{mathhub}

```

\importmhmodule

The `\importmhmodule[<key=value list>]{module}` saves the current value of `\mh@currentrepos` in a local macro `\mh@@repos`, resets `\mh@currentrepos` to the new value if one is given in the optional argument, and after importing resets `\mh@currentrepos` to the old value in `\mh@@repos`. We do all the `\ifx` comparison with an `\expandafter`, since the values may be passed on from other key bindings. Parameters will be passed to `\importmodule`.

```

54 \srefaddidkey{importmhmodule}%
55 \addmetakey{importmhmodule}{repos}% saves the repo's path. E.g: smglom/numberfield
56 \addmetakey{importmhmodule}{path}% saves the module name. E.g: naturalnumbers
57 \addmetakey[sms]{importmhmodule}{ext}% saves the extension: E.g: tex
58 \addmetakey[false]{importmhmodule}{conservative}[true]%
59 \newcommand\importmhmodule[2] [] {%
60   \metasetkeys{importmhmodule}{#1}%
61   \ifx\importmhmodule@path\empty% if module name is not set
62     \importmodule[ext=\importmhmodule@ext,id=\importmhmodule@id]{#2}%
63   \else%
64     \edef\mh@@repos{\mh@currentrepos}% remember so that we can reset it.
65     \ifx\importmhmodule@repos\empty% if in the same repos
66       \relax% no need to change mh@currentrepos, i.e, current directory.
67     \else%

```

```

68     \mhcurrentrepos{\importmhmodule@repos}% change it.
69     \fi%
70     \importmodule[load=\MathHub{\mh@currentrepos/source/\importmhmodule@path},%
71     ext=\importmhmodule@ext,id=\importmhmodule@id]{#2}%
72     \mhcurrentrepos{\mh@repos}% after importing, reset to old value
73     \fi%
74     \ignorespaces%
75 }%

```

and now the analogs

`\usemhmodule`

```

76 \newcommand\usemhmodule[2][]{%
77   \metasetkeys{importmhmodule}{#1}%
78   \ifx\importmhmodule@path\empty%
79     \usemodule[ext=\importmhmodule@ext,id=\importmhmodule@id]{#2}%
80   \else%
81     \edef\mh@repos{\mh@currentrepos}%
82     \ifx\importmhmodule@repos\empty%
83     \else%
84       \mhcurrentrepos{\importmhmodule@repos}%
85     \fi%
86     \usemodule[load=\MathHub{\mh@currentrepos/source/\importmhmodule@path},ext=\importmhmodule@
87     \mhcurrentrepos\mh@repos%
88   \fi%
89   \ignorespaces%
90 }%

```

`\mhinputref`

```

91 \newcommand\mhinputref[2][]{%
92   \def\@repos{#1}%
93   \edef\mh@repos{\mh@currentrepos}%
94   \ifx\@repos\empty%
95   \else%
96     \mhcurrentrepos{#1}%
97   \fi%
98   \inputref{\MathHub{\mh@currentrepos/source/#2}}%
99   \mhcurrentrepos\mh@repos%
100   \ignorespaces%
101 }%

```

`\mhinput`

```

102 \let\mhinput\mhinputref%
103 </modules>

```

4.4 omtex-mh.sty: MH Variants for OMTex

```

104 <*omtext>
105 \ProvidesPackage{omtext-mh}[2019/03/20 v1.1 MathHub support for the sTeX omtex package]
106 \RequirePackage{mathhub}

```

`\mh*graphics` Use the current value of `\mh@currentrepos` or the value of the `mhrepos` key if it is given in `\my*graphics`.

```

107 \def\Gin@mhrepos{}
108 \define@key{Gin}{mhrepos}{\def\Gin@mhrepos{#1}}
109 \newcommand\mhgraphics[2][]{\setkeys{Gin}{#1}}%
110 \edef\mh@crepos{\mh@currentrepos}%
111 \ifx\Gin@mhrepos\empty\mygraphics[#1]{\MathHub{\mh@currentrepos/source/#2}}%
112 \else\mygraphics[#1]{\MathHub{\Gin@mhrepos/source/#2}}\fi
113 \def\Gin@mhrepos{\mhcurrentrepos\mh@crepos}
114 \newcommand\mhgraphics[2][]{\begin{center}\mhgraphics[#1]{#2}\end{center}}
115 \newcommand\mhgraphics[2][]{\fbox{\mhgraphics[#1]{#2}}}
116 \newcommand\mhcbgraphics[2][]{\begin{center}\fbox{\mhgraphics[#1]{#2}}\end{center}}
117 \end{omtext}

```

4.5 smultiling-mh.sty: MH Variants for Multilinguality

```

118 \smultiling
119 \ProvidesPackage{smultiling-mh}[2019/03/20 v1.1 MathHub support for the sTeX smultiling package]
120 \RequirePackage{mathhub}

```

`mhmodsig`

```

121 \newenvironment{mhmodsig}{\begin{modsig}}{\end{modsig}}

```

`mhmodnl:*`

```

122 \addmetakey{mhmodnl}{repos}
123 \addmetakey{mhmodnl}{path}
124 \addmetakey*{mhmodnl}{title}
125 \addmetakey*{mhmodnl}{creators}
126 \addmetakey*{mhmodnl}{contributors}
127 \addmetakey{mhmodnl}{srccite}
128 \addmetakey{primary}{mhmodnl}[yes]

```

`mhmodnl` The `mhmodnl` environment is just a layer over the `module` environment and the `\importmhmodule` macro with the keys and language suitably adapted.

```

129 \newenvironment{mhmodnl}[3][]{\metasetkeys{mhmodnl}{#1}\def\@test{#1}}%
130 \edef\@crepos{\ifx\mhmodnl@repos\empty\mh@currentrepos\else\mhmodnl@repos\fi}%
131 \edef\@load{\MathHub{\@crepos/source/\ifx\mhmodnl@path\empty #2\else\mhmodnl@path\fi}}%
132 \ifx\@test\empty\begin{modnl}[load=\@load]{#2}{#3}\else\begin{modnl}[load=\@load,#1]{#2}{#3}\fi
133 \ignorespacesandpars}
134 {\end{modnl}\ignorespacesandparsafterend}

```

`mhviewsig` The `mhviewsig` environment is just a layer over the `mhview` environment with the keys suitably adapted.

```

135 \newenvironment{mhviewsig}[4][]{% keys, id, from, to
136 \def\@test{#1}\ifx\@test\empty%
137 \begin{mhview}[id=#2,ext=tex]{#3}{#4}\else%
138 \begin{mhview}[id=#2,#1,ext=tex]{#3}{#4}\fi%
139 \ignorespacesandpars}
140 {\end{mhview}\ignorespacesandparsafterend}

```

`mhviewnl` The `mhviewnl` environment is just a layer over the `mhview` environment with the keys and language suitably adapted.⁴

```

141 \newenvironment{mhviewnl}[5][]{% keys, id, lang, from, to
142 \def\@test{#1}\ifx\@test\@empty%
143 \begin{mhview}[id=#2.#3,ext=tex]{#4}{#5}\else%
144 \begin{mhview}[id=#2.#3,#1,ext=tex]{#4}{#5}\fi%
145 \ignorespacesandpars}
146 {\end{mhview}\ignorespacesandparsafterend}
147 \</smultiling>

```

4.6 structview-mh.sty: MH Variants for Structures and Views

```

148 \<structview>
149 \ProvidesPackage{structview-mh}[2019/03/20 v1.1 MathHub support for the sTeX structview package]
150 \RequirePackage{mathhub}

```

`mhstructure`

```

151 \newenvironment{mhstructure}[3][]{%
152 \gdef\@doit{\importmhmodule[#1]{#3}}%
153 \ifmod@show\par\noindent structure import "#2" from module #3 \@doit\fi%
154 \ignorespacesandpars%
155 }{%
156 \aftergroup\@doit\ifmod@show end import\fi%
157 \ignorespacesandparsafterend%
158 }%

```

`importmhmodulevia` this is now deprecated, we give an error

```

159 \newenvironment{importmhmodulevia}[2][]{%
160 {\PackageError{structview-mh}%
161 {The {importmhmodulevia} environment is deprecated}{use the {mhstructure} instead!}%
162 \begin{mhstructure}[#1]{missing}{#2}}
163 {\end{mhstructure}}}

164 \srefaddidkey{mhview}
165 \addmetakey{mhview}{display}
166 \addmetakey{mhview}{creators}
167 \addmetakey{mhview}{contributors}
168 \addmetakey{mhview}{src cite}
169 \addmetakey*{mhview}{title}
170 \addmetakey{mhview}{type}
171 \addmetakey{mhview}{fromrepos}
172 \addmetakey{mhview}{torepos}
173 \addmetakey{mhview}{frompath}
174 \addmetakey{mhview}{topath}
175 \addmetakey[sms]{mhview}{ext}

```

⁴EdNOTE: MK: we have to do something about the `if@langfiles` situation here. But this is non-trivial, since we do not know the current path, to which we could append `.\lang`!

mhview the MathHub version

```

176 \newenvironment{mhview}[3][]{% keys, from, to
177   \metasetkeys{mhview}{#1}%
178   \sref@target%
179   \begin{@mhview}{#2}{#3}%
180   \view@heading{#2}{#3}{\mhview@display}{\mhview@title}%
181   \ignorespacesandpars%
182 }{%
183   \end{@mhview}%
184   \ignorespacesandparsafterend%
185 }%
186 \ifmod@show\surroundwithmdframed{mhview}\fi

```

@mhview The @mhview does the actual bookkeeping at the module level.

```

187 \newenvironment{@mhview}[2]{%from, to
188   \importmhmodule[repos=\mhview@fromrepos,path=\mhview@frompath,ext=\mhview@ext]{#1}%
189   \importmhmodule[repos=\mhview@torepos,path=\mhview@topath,ext=\mhview@ext]{#2}%
190 }{}%

```

mhviewsketch The mhviewsketch environment is deprecated, we give an error

```

191 \newenvironment{mhviewsketch}[3][]{%
192   {\PackageError{structview}%
193     {The {mhviewsketch} environment is deprecated}{use the {mhview} instead!}%
194   \begin{mhview}[#1]{#2}{#3}}
195 {\end{mhview}}
196 \</structview>

```

4.7 mikoslides-mh.sty: Support for MiKo Slides

```

197 <*mikoslides>
198 \ProvidesPackage{mikoslides-mh}[2019/03/20 v1.1 MathHub support for the sTeX mikoslides package]
199 \RequirePackage{mathhub}

```

\mhframeimage Use the current value of \mh@currentrepos or the value of the mhrepos key if it is given in \frameimage.

```

200 \def\Gin@mhrepos{}
201 \define@key{Gin}{mhrepos}{\def\Gin@mhrepos{#1}}
202 \newcommand\mhframeimage[2][]{%
203   \setkeys{Gin}{#1}%
204   \edef\mh@@repos{\mh@currentrepos}%
205   \ifx\Gin@mhrepos\@empty%
206     \frameimage[#1]{\MathHub{\mh@currentrepos/source/#2}}%
207   \else%
208     \frameimage[#1]{\MathHub{\Gin@mhrepos/source/#2}}%
209   \fi%
210 }%

```

\mhinputref*

```

211 \let\orig@mhinputref\mhinputref

```

```

212 \def\mhinputref{\@ifstar\nmhinputref\orig@mhinputref}
213 \newcommand\nmhinputref[2] [] {\ifnotes\orig@mhinputref[#1]{#2}\fi}
214 \</mikoslides>

```

4.8 problem-mh.sty: Support for Problems

```

215 <*problem>
216 \ProvidesPackage{problem-mh}[2019/03/20 v1.1 MathHub support for the sTeX problem package]
217 \RequirePackage{mathhub}

\includemhproblem The \includemhproblem saves the current value of \mh@currentrepos in a local
macro \mh@@repos, resets \mh@currentrepos to the new value if one is given in
the optional argument, and after importing resets \mh@currentrepos to the old
value in \mh@@repos.

218 \newcommand\includemhproblem[2] [] {\metasetkeys{inclprob}{#1}%
219 \edef\mh@@repos{\mh@currentrepos}%
220 \ifx\inclprob@mhrepos\@empty\else\mhcurrentrepos\inclprob@mhrepos\fi%
221 \input{\MathHub{\mh@currentrepos/source/#2}}}%
222 \mhcurrentrepos\mh@@repos\clear@inclprob@keys}
223 </problem>

```

4.9 hwexam-mh.sty: Support for Assignments

```

224 <*hwexam>
225 \ProvidesPackage{hwexam-mh}[2019/03/20 v1.1 MathHub support for the sTeX hwexam package]
226 \RequirePackage{mathhub}

\includemhassignment The \includemhassignment saves the current value of \mh@currentrepos in a
local macro \mh@@repos, resets \mh@currentrepos to the new value if one is given
in the optional argument, and after importing resets \mh@currentrepos to the old
value in \mh@@repos.

227 \newcommand\includemhassignment[2] [] {\metasetkeys{inclassig}{#1}%
228 \edef\mh@@repos{\mh@currentrepos}%
229 \ifx\inclassig@mhrepos\@empty\else\mhcurrentrepos\inclassig@mhrepos\fi%
230 \includeassignment[#1]{\MathHub{\mh@currentrepos/source/#2}}}%
231 \mhcurrentrepos\mh@@repos\clear@inclassig@keys}

\inputmhassignment analogous

232 \newcommand\inputmhassignment[2] [] {\metasetkeys{inclassig}{#1}%
233 \edef\mh@@repos{\mh@currentrepos}%
234 \ifx\inclassig@mhrepos\@empty\else\mhcurrentrepos\inclassig@mhrepos\fi%
235 \inputassignment[#1]{\MathHub{\mh@currentrepos/source/#2}}}%
236 \mhcurrentrepos\mh@@repos\clear@inclassig@keys}
237 </hwexam>

```

4.10 tikzinput-mh.sty: Support for Assignments

```

238 <*tikzinput>
239 \ProvidesPackage{tikzinput-mh}[2019/03/20 v1.1 MathHub support for the sTeX tikzinput package]

```

```

240 \RequirePackage{mathhub}
241 \RequirePackage{pathsuris}

242 \define@key{Gin}{mhrepos}{\def\Gin@mhrepos{#1}}
243 \newcommand\mhtikzinput[2][]{\def\Gin@mhrepos{}\setkeys{Gin}{#1}%
244 \edef\mh@crepos{\mh@currentrepos}%
245 \ifx\Gin@mhrepos\empty\tikzinput[#1]{\MathHub{\mh@currentrepos/source/#2}}}%
246 \else\mhcurrentrepos\Gin@mhrepos\tikzinput[#1]{\MathHub{\Gin@mhrepos/source/#2}}\fi
247 \def\Gin@mhrepos{}\mhcurrentrepos\mh@crepos}
248 \newcommand\cmhtikzinput[2][]{\begin{center}\mhtikzinput[#1]{#2}\end{center}}
249 \end{tikzinput}

```

4.11 lstmh.sty: Support for Listings

```

250 \*lst
251 \ProvidesPackage{lstmh}[2019/03/20 v1.1 MathHub support for the listings package]
252 \RequirePackage{mathhub}
253 \RequirePackage{pathsuris}

254 \define@key{lst}{mhrepos}{\def\lst@mhrepos{#1}}
255 \newcommand\lstinputmhlisting[2][]{\def\lst@mhrepos{}\setkeys{lst}{#1}%
256 \edef\mh@crepos{\mh@currentrepos}%
257 \ifx\lst@mhrepos\empty\lstinputlisting[#1]{\MathHub{\mh@currentrepos/source/#2}}}%
258 \else\lstinputlisting[#1]{\MathHub{\lst@mhrepos/source/#2}}\fi
259 \def\lst@mhrepos{}\mhcurrentrepos\mh@crepos}
260 \newcommand\clstinputmhlisting[2][]{\begin{center}\lstinputmhlisting[#1]{#2}\end{center}}
261 \end{lst}

```

Change History

v1.0			<code>libinput</code> to input first the
General: Deprecated			<code>meta-inf</code> -level and then
<code>mhviewsketch</code>	1		repos-level file; this allows
moved all MH functionality into			more sharing and does not
one DTX file	1		break MathHub content (only
v1.1			one of them currently exists) . . 1
General: Changed the semantics of			

References

- [GH] *GitHub: Build software better, together*. URL: <http://github.com> (visited on 02/24/2014).
- [GL] *The first single application for the entire DevOps lifecycle – GitLab*. URL: <http://gitlab.com> (visited on 01/12/2019).
- [Hor+11] Fulya Horozal et al. “Combining Source, Content, Presentation, Narration, and Relational Representation”. In: *Intelligent Computer Mathematics*. Ed. by James Davenport et al. LNAI 6824. Springer Verlag, 2011, pp. 212–227. ISBN: 978-3-642-22672-4. URL: http://kwarc.info/frabe/Research/HIJKR_dimensions_11.pdf.
- [KGA18] Michael Kohlase, Deyan Ginev, and Rares Ambrus. *modules.sty: Semantic Macros and Module Scoping in sTeX*. Tech. rep. 2018. URL: <https://github.com/sLaTeX/sTeX/raw/master/sty/pathsuris/pathsuris.pdf>.
- [Koh18] Michael Kohlase. *metakeys.sty: A generic framework for extensible Metadata in L^AT_EX*. Tech. rep. 2018. URL: <https://github.com/sLaTeX/sTeX/raw/master/sty/metakeys/metakeys.pdf>.
- [lmh] *lmh: A cross-repository administration tool for the local authoring in MathHub.info*. URL: <https://github.com/KWARC/localmh> (visited on 02/05/2017).
- [sTeX] *sTeX: A semantic Extension of TeX/LaTeX*. URL: <https://github.com/sLaTeX/sTeX> (visited on 05/15/2015).