

# MathHub Support for $\text{\S T E X}^*$

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

October 15, 2020

## Abstract

The `mathhub` package collection is part of the  $\text{\S T E X}$  collection, a version of  $\text{\T E X}/\text{\L A T E X}$  that allows to markup  $\text{\T E X}/\text{\L A T E X}$  documents semantically without leaving the document format, essentially turning  $\text{\T E X}/\text{\L A T E X}$  into a document format for mathematical knowledge management (MKM).

The `mathhub` packages extend  $\text{\S T E X}$  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  $\text{\S T E X}$  documents in its own right.

## Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>The User Interface</b>	<b>4</b>
2.1	<code>mathhub.sty</code> : General Infrastructure . . . . .	4
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
<b>3</b>	<b>Limitations</b>	<b>7</b>

---

\*Version v1.2 (last revised 2010/10/01)

<b>4</b>	<b>Implementation</b>	<b>8</b>
4.1	<code>mathhub.sty</code> : General Infrastructure . . . . .	8
4.2	<code>omdoc--mh.sty</code> : MH Document Infrastructure . . . . .	12
4.3	<code>modules-mh.sty</code> : MH Variants for Modules . . . . .	12
4.4	<code>omtext-mh.sty</code> : MH Variants for OMText . . . . .	13
4.5	<code>smultiling-mh.sty</code> : MH Variants for Multilinguality . . . . .	14
4.6	<code>structview-mh.sty</code> : MH Variants for Structures and Views . . . . .	15
4.7	<code>mikoslides-mh.sty</code> : Support for MiKo Slides . . . . .	16
4.8	<code>problem-mh.sty</code> : Support for Problems . . . . .	17
4.9	<code>hwexam-mh.sty</code> : Support for Assignments . . . . .	17
4.10	<code>tikzinput-mh.sty</code> : Support for Assignments . . . . .	18
4.11	<code>lstmh.sty</code> : Support for Listings . . . . .	18

# 1 Introduction

As  $\text{\LaTeX}$  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  $\rightsquigarrow$  manageable file system layout schema that has co-evolved with the MathHub.info portal for active documents, but is useful for organizing collections of  $\text{\LaTeX}$  documents in its own right. In particular, since the layout scheme is supported by the `lmh` (local `mathhub` []), and `make` (a build system [] for MathHub archives) in MMT [MMT] which automates many management tasks. For instance, after installing the `mmt.jar`, the shell command `mmt lmh install <group>/<arch>` installs the MathHub archive `<group>/<arch>` together with all of its dependencies and `mmt make pdflatex <file>` generates PDF for the file `<file>` (and generates/updates all auxiliary files necessary along the way).

MathHub (<http://MathHub.info>), is a portal and archive for flexiformal mathematics. It hosts much of the  $\text{\LaTeX}$  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  $\text{\LaTeX}$  sources, which are in turn organized in **MathHub archives** [Hor+11]. 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 `<group>/<arch>`, where `<group>` is a MathHub-unique repository group and `<arch>` a MathHub archive name that is `<group>`-unique.

The MathHub archives have a prescribed structure; see [Hor+11] for details. For our purposes, we only need two aspects:

- the  $\text{\LaTeX}$  sources are all in a top-level subdirectory `source`,
- there is a top-level sub-directory `META-INF` with a manifest file `MANIFEST.MF` which consists of lines of the form `<key>: <values>`.

For the purposes of the `mathhub` package we assume that the `MANIFEST.MF` file has at least the `id` key specified and the value is exactly `<group>/<arch>`. Furthermore, we assume that the `MATHHUB` environment variable is set with the system path to the MathHub root folder.

With this information the mechanics of the MathHub archive structure can be hidden from the  $\text{\LaTeX}$  author with MathHub-enabled versions of the  $\text{\LaTeX}$  macros (let's call them **mh-variants**), which are defined in the `mh`-packages of the `mathhub` bundle, which we document in this manual. The mechanics of the `mathhub` bundle is as follows: For most  $\text{\LaTeX}$  package `<pack>.sty` there is a `mh`-variant `<pack>-mh.sty`, and `<pack>.sty` takes the option `mh`. When that is given (by calling `\usepackage[mh]{<pack>}`), then `<pack>.sty` inputs `<pack>-mh.sty` from the `mathhub` bundle, which augments the `<pack>` package with MathHub functionality.

## 2 The User Interface

We now document mh-variants of the  $\text{\TeX}$  packages that have MathHub functionality.

### 2.1 mathhub.sty: General Infrastructure

For the generation of absolute file paths, the `mathhub` package keeps track of the current archive. If this ever needs to be set manually, it can be declared by the `\mhcurrentrepos` macro relative to the MathHub root path. `\mhcurrentrepos{group/repos}` declare that it resides at the path `/user/foo/localmh/MathHub/group/repos` given that the MathHub root path is `/user/foo/localmh/mathhub`.

Given a systematic grouping in the MathHub file layout scheme,  $\text{\TeX}$  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  $\text{\TeX}$  file has the following lines in the preamble:

```
\libinput{preamble}
```

`\libusepackage` The `\libusepackage` is analogous it allows to share  $\text{\LaTeX}$  package between MathHub archives.

### 2.2 omdoc-mh.sty: MH Document Infrastructure

`\addmhbibresource` The `\addmhbibresource` macro is a variant of `\addbibresource` from `bib $\text{\LaTeX}$`  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 `bib $\text{\TeX}$`  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}
```

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

```
\importmhmodule[mhrepos=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 one of the following forms

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

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

`\mhcurrentrepos` Of course, neither `LATEX` nor `LATEX ML` 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 repository, 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 OMTex

`\cmhgraphics` The `\cmhgraphics` 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)

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

Note that the `\cmhgraphics` 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 `mhrepos` 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}[mhrepos=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:1

1

## 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}
```

---

<sup>1</sup>EdNOTE: 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 [Koh20] to be loaded (in the right version).

```
1 \langle *package\rangle
2 \ProvidesPackage{mathhub}[2010/10/01 v1.2 Basic MathHub functionality]
3 \RequirePackage{keyval}
4 \RequirePackage{pathsuris}
```

### 4.1 mathhub.sty: General Infrastructure

`\mhcurrentrepos` `\mhcurrentrepos` is used to initialize the current repository. If the repository 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 repository.<sup>2</sup>

```
5 \newcommand\mhcurrentrepos[1]{%
6   % \edef\@test{#1}%
7   % \ifx\@test\mhcurrentrepos% if new dir = old dir
8   %   \relax% no need to change
9   % \else%
10  %   \protected@write\@auxout{}\string\@mhcurrentrepos{#1}%
11  % \fi%
12  \@mhcurrentrepos{#1}% define mhcurrentrepos
13 }%
14 \newcommand\@mhcurrentrepos[1]{\edef\mhcurrentrepos{#1}}%
```

`\libinput` the `\libinput` macro inputs from the `lib` directory of the MathHub repository and then the `meta-inf/lib` repository of the group, if they exist. Since in practice nested libinputs may occur, we make sure that we stash the old values of `\mh@inffile` and `\mh@libfile` and restore them at the end.

```
15 \def\modules@@first#1/#2;{#1}
16 \newcommand\libinput[1]{%
17   \message{^^Jmhcurrentrepos: \mhcurrentrepos^^J}
18   \edef\@mh@group{\expandafter\modules@@first\mhcurrentrepos;}
19   \let\orig@inffile\mh@inffile\let\orig@libfile\mh@libfile
20   \def\mh@inffile{\MathHub{\@mh@group/meta-inf/lib/#1}}
21   \def\mh@libfile{\MathHub{\mhcurrentrepos/lib/#1}}%
22   \IfFileExists\mh@inffile{\input\mh@inffile}{}%
23   \IfFileExists\mh@inffile{\IfFileExists\mh@libfile}{%
24     {\PackageError{mathhub}
25       {Library file missing; cannot input #1.tex\MessageBreak%
26       Both \mh@libfile.tex\MessageBreak and \mh@inffile.tex\MessageBreak%
27       do not exist}%
28     {Check whether the file name is correct}}}%
29   \IfFileExists\mh@libfile{\input\mh@libfile\relax}{}
```

<sup>2</sup>EdNOTE: MK: It seems that we do not need the `currentrepos` in the aux file, leaving the commenting out just in case we do. We can consolidate this further if it turns out that we do not need this.



```

30 \let\mh@inffile\orig@inffile\let\mh@libfile\orig@libfile}

\libusepackage the \libusepackage is analogous to \libinput
31 \newcommand\libusepackage[2][]{%
32 \message{^^Jmh@currentrepos: \mh@currentrepos^^J}
33 \edef\@mh@group{\expandafter\modules@@first\mh@currentrepos;}
34 \let\orig@inffile\mh@inffile\let\orig@libfile\mh@libfile
35 \edef\mh@inffile{\MathHub{\@mh@group/meta-inf/lib/#2}}
36 \edef\mh@libfile{\MathHub{\mh@currentrepos/lib/#2}}%
37 \IfFileExists{\mh@inffile.sty}{\usepackage[#1]{\mh@inffile}}{}%
38 \IfFileExists {\mh@inffile.sty}{\IfFileExists{\mh@libfile.sty}{}%
39 {\PackageError{mathhub}
40 {Library file missing; cannot use package #2.sty\MessageBreak%
41 Both \mh@libfile.sty\MessageBreak and \mh@inffile.sty\MessageBreak%
42 do not exist}}%
43 {Check whether the file name is correct}}}%
44 \IfFileExists\mh@libfile{\input\mh@libfile\relax}{}
45 \let\mh@inffile\orig@inffile\let\mh@libfile\orig@libfile}

```

Generally, the  $\text{\TeX}$  formatter `pdflatex` needs to know the file system paths of the referenced  $\text{\LaTeX}$  files: usually long relative paths. The `pathsuris` package [KGA20] from the  $\text{\LaTeX}$  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). Fortunately, we can compute this automatically.

We parse the `MATHHUB` environment variable via `kpsewhich` ( $\text{\LaTeX}$  can run this even in paranoid mode) and then set the MathHub path using `\defpath`.

```

46 \newif@if@iswindows@@@iswindows@false
47 \IfFileExists{nul:}{\IfFileExists{/dev/null}}{\@iswindows@true}}{}
48
49 \def\kpsewhich#1#2{\begingroup
50 \def\@Space{ }
51 \edef\kpsewhich@cmd{"|kpsewhich #2"}
52 \everyeof{\noexpand}
53 \catcode'\=12
54 \edef#1{\@input\kpsewhich@cmd\@Space}
55 \trimstring#1
56 \global\let#1#1
57 \endgroup}
58
59 \kpsewhich\mathhub@path{--var-value MATHHUB}
60 \if@iswindows@\windows@to@path\mathhub@path\fi
61 \ifx\mathhub@path\@empty\else\defpath{MathHub}{\mathhub@path}\fi
62

```

Next we find the manifest of the MathHub archive: a file `MANIFEST.MF` up and over in the file system tree.

```

63 \def\findmanifest#1{
64 \cpath{#1}

```

```

65 \ifx\@CanPath\@Slash
66 \def\manifest@mf{}
67 \else\ifx\@CanPath\@empty
68 \def\manifest@mf{}
69 \else
70 \edef\@findmanifest@path{\@CanPath/MANIFEST.MF}
71 \if@iswindows@path@to@windows\@findmanifest@path\fi
72 \IfFileExists{\@findmanifest@path}{
73 \message{^^J MANIFEST.MF found at \@findmanifest@path^^J}
74 \edef\manifest@mf{\@findmanifest@path}
75 }{
76 \edef\@findmanifest@path{\@CanPath/META-INF/MANIFEST.MF}
77 \if@iswindows@path@to@windows\@findmanifest@path\fi
78 \IfFileExists{\@findmanifest@path}{
79 \message{^^J MANIFEST.MF found at \@findmanifest@path^^J}
80 \edef\manifest@mf{\@findmanifest@path}
81 }{
82 \edef\@findmanifest@path{\@CanPath/meta-inf/MANIFEST.MF}
83 \if@iswindows@path@to@windows\@findmanifest@path\fi
84 \IfFileExists{\@findmanifest@path}{
85 \message{^^J MANIFEST.MF found at \@findmanifest@path^^J}
86 \edef\manifest@mf{\@findmanifest@path}
87 }{
88 \findmanifest{\@CanPath/..}
89 }}}
90 \fi\fi
91 }

```

the next macro is a helper function for parsing MANIFEST.MF

```

92 \def\split@manifest@key{
93 \IfSubStr{\manifest@line}{\@Colon}{
94 \StrBefore{\manifest@line}{\@Colon}[\manifest@key]
95 \StrBehind{\manifest@line}{\@Colon}[\manifest@line]
96 \trimstring\manifest@line
97 \trimstring\manifest@key
98 }{
99 \def\manifest@key{}
100 }
101 }

```

the next helper function iterates over lines in MANIFEST.MF

```

102 \def\parse@manifest@loop{
103 \ifeof\@manifest
104 \else
105 \read\@manifest to \manifest@line\relax
106 \edef\manifest@line{\expandafter\detokenize\expandafter{\manifest@line}}
107 \split@manifest@key
108 % id
109 \IfStrEq\manifest@key{\detokenize{id}}{
110 \message{^^J archive id: \manifest@line^^J}

```

```

111     \xdef\manifest@mf@id{\manifest@line}
112   }{
113   % narration-base
114   \IfStrEq\manifest@key{\detokenize{narration-base}}{
115     \message{^^J archive narration-base: \manifest@line^^J}
116     \xdef\manifest@mf@narr{\manifest@line}
117   }{
118   % namespace
119   \IfStrEq\manifest@key{\detokenize{source-base}}{
120     \message{^^J archive source-base: \manifest@line^^J}
121     \xdef\manifest@mf@ns{\manifest@line}
122   }{
123   \IfStrEq\manifest@key{\detokenize{ns}}{
124     \message{^^J archive ns: \manifest@line^^J}
125     \xdef\manifest@mf@ns{\manifest@line}
126   }{
127   % dependencies
128   \IfStrEq\manifest@key{\detokenize{dependencies}}{
129     \message{^^J archive dependencies: \manifest@line^^J}
130     \xdef\manifest@mf@deps{\manifest@line}
131   }{
132
133   }}}}
134   \parse@manifest@loop
135 \fi
136 }

```

and finally, we find path of main file

```

137
138 \edef\oldpercentcatcode{\the\catcode'\%}
139 \catcode'\%=12
140 \let\percent%
141 \catcode'\%=\oldpercentcatcode
142
143 \edef\pwd@cmd{\if@iswindows@ -expand-var \percent CD\percent\else -var-value PWD\fi}
144 \kpsewhich\mathhub@maindir\pwd@cmd
145 \if@iswindows@\windows@to@path\mathhub@maindir\fi
146 \message{^^J PWD: \mathhub@maindir^^J}

```

We read in and parse MANIFEST.MF

```

147 \findmanifest{\mathhub@maindir}
148 \begingroup
149   \gdef\manifest@mf@id{}
150   \gdef\manifest@mf@narr{}
151   \gdef\manifest@mf@ns{}
152   \gdef\manifest@mf@deps{}
153   \newread\@manifest
154   \openin\@manifest\manifest@mf
155   \parse@manifest@loop
156   \closein\@manifest

```

```

157 \endgroup
158 \if@iswindows@{\windows@to@path\manifest@mf\fi

```

Finally – and that is the ultimate goal of all of the above, we set the current repos.

```

159 \mhcurrentrepos{\manifest@mf@id}
160 \</package>

```

## 4.2 omdoc--mh.sty: MH Document Infrastructure

```

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

```

\addmhbibresource

```

164 \newcommand\addmhbibresource[2] [] {%
165   \def\@repos{#1}%
166   \ifx\@repos\@empty%
167     \addbibresource{\MathHub{\mh@currentrepos/#2}}%
168   \else
169     \addbibresource{\MathHub{\@repos/#2}}%
170   \fi%
171   \ignorespacesandpars}%
172 \</omdoc>

```

## 4.3 modules-mh.sty: MH Variants for Modules

```

173 <*modules>
174 \ProvidesPackage{modules-mh}[2019/03/20 v1.1 MathHub support for the sTeX modules package]
175 \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.

```

176 \srefaddidkey{importmhmodule}%
177 \addmetakey{importmhmodule}{mhrepos}%
178 \addmetakey{importmhmodule}{path}%
179 \addmetakey{importmhmodule}{dir}%
180 \addmetakey[false]{importmhmodule}{conservative}[true]%
181 \newcommand\importmhmodule[2] [] {%
182   \usemodule@maybesetcodes
183   \metasetkeys{importmhmodule}{#1}%
184   \ifx\importmhmodule@dir\@empty%
185     \edef\@path{\importmhmodule@path}%
186   \else\edef\@path{\importmhmodule@dir/#2}\fi%
187   \ifx\@path\@empty% if module name is not set

```

```

188 \importmodule[id=\importmhmodule@id]{#2}%
189 \else%
190 \edef\mh@@repos{\mh@currentrepos}% remember so that we can reset it.
191 \ifx\importmhmodule@mhrepos\@empty% if in the same repos
192 \relax% no need to change mh@currentrepos, i.e., current directory.
193 \else%
194 \mhcurrentrepos{\importmhmodule@mhrepos}% change it.
195 \fi%
196 \importmodule[load=\MathHub{\mh@currentrepos/source/\@path},
197               id=\importmhmodule@id]{#2}%
198 \mhcurrentrepos{\mh@@repos}% after importing, reset to old value
199 \fi%
200 \ignorespacesandpars}

```

and now the analogs

`\usemhmodule`

```

201 \newcommand\usemhmodule[2][]{%
202 \metasetkeys{importmhmodule}{#1}%
203 \ifx\importmhmodule@dir\@empty%
204 \edef\@path{\importmhmodule@path}%
205 \else\edef\@path{\importmhmodule@dir/#2}\fi%
206 \ifx\@path\@empty%
207 \usemodule[id=\importmhmodule@id]{#2}%
208 \else%
209 \edef\mh@@repos{\mh@currentrepos}%
210 \ifx\importmhmodule@mhrepos\@empty%
211 \else\mhcurrentrepos{\importmhmodule@mhrepos}\fi%
212 \usemodule[load=\MathHub{\mh@currentrepos/source/\@path},
213           id=\importmhmodule@id]{#2}%
214 \mhcurrentrepos\mh@@repos%
215 \fi%
216 \ignorespacesandpars}

```

`\mhinputref`

```

217 \newcommand\mhinputref[2][]{%
218 \def\@repos{#1}%
219 \edef\mh@@repos{\mh@currentrepos}%
220 \ifx\@repos\@empty\else\mhcurrentrepos{#1}\fi%
221 \inputref{\MathHub{\mh@currentrepos/source/#2}}%
222 \mhcurrentrepos\mh@@repos%
223 \ignorespacesandpars}

```

`\mhinput`

```

224 \let\mhinput\mhinputref%
225 </modules>

```

#### 4.4 omtex-mh.sty: MH Variants for OMTex

```

226 <omtext>
227 \ProvidesPackage{omtext-mh}[2019/03/20 v1.1 MathHub support for the sTeX omtext package]
228 \RequirePackage{mathhub}

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

229 \def\Gin@mhrepos{}
230 \define@key{Gin}{mhrepos}{\def\Gin@mhrepos{#1}}
231 \newcommand\mhgraphics[2] [] {\setkeys{Gin}{#1}}%
232 \edef\mh@crepos{\mh@currentrepos}%
233 \ifx\Gin@mhrepos\empty\includegraphics[#1]{\MathHub{\mh@currentrepos/source/#2}}%
234 \else\includegraphics[#1]{\MathHub{\Gin@mhrepos/source/#2}}\fi
235 \def\Gin@mhrepos{\mhcurrentrepos\mh@crepos}
236 \newcommand\cmhgraphics[2] [] {\begin{center}\mhgraphics[#1]{#2}\end{center}}

```

The following macros are deprecated.

```

237 \newcommand\mhcgraphics[2] [] {\begin{center}\mhgraphics[#1]{#2}\end{center}}
238 \PackageWarning{omtext-mh}{\protect\mhcgraphics\space is deprecated, use \protect\cmhgraphics}
239 \newcommand\mhgraphics[2] [] {\fbox{\mhgraphics[#1]{#2}}}
240 \PackageWarning{omtext-mh}{\protect\mhgraphics\space is deprecated, use
241 \protect\mhgraphics\space and {center} instead}}
242 \newcommand\mhcbgraphics[2] [] {\begin{center}\fbox{\mhgraphics[#1]{#2}\end{center}}
243 \PackageWarning{omtext-mh}{\protect\mhcbgraphics\space is deprecated, use
244 \protect\mhgraphics,\space {center}, and \protect\fbox\space instead}}
245 </omtext>

```

## 4.5 smultiling-mh.sty: MH Variants for Multilinguality

```

246 <smultiling>
247 \ProvidesPackage{smultiling-mh}[2019/03/20 v1.1 MathHub support for the sTeX smultiling package]
248 \RequirePackage{mathhub}

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

mhmodnl:*
250 \addmetakey{mhmodnl}{mhrepos}
251 \addmetakey{mhmodnl}{path}
252 \addmetakey*{mhmodnl}{title}
253 \addmetakey*{mhmodnl}{creators}
254 \addmetakey*{mhmodnl}{contributors}
255 \addmetakey{mhmodnl}{srccite}
256 \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.

257 \newenvironment{mhmodnl}[3] [] {\metasetkeys{mhmodnl}{#1}\def\@test{#1}}%
258 \edef\@crepos{\ifx\mhmodnl@mhrepos\empty\mh@currentrepos\else\mhmodnl@mhrepos\fi}%
259 \edef\@load{\MathHub{\@crepos/source/\ifx\mhmodnl@path\empty #2\else\mhmodnl@path\fi}}%

```

```

260 \ifx\@test\@empty\begin{modnl}[load=\@load]{#2}{#3}\else\begin{modnl}[load=\@load,#1]{#2}{#3}\f
261 \ignorespacesandpars}
262 {\end{modnl}}\ignorespacesandparsafterend}

```

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

```

263 \newenvironment{mhviewsig}[4][]{% keys, id, from, to
264 \def\@test{#1}\ifx\@test\@empty%
265 \begin{mhview}[id=#2]{#3}{#4}\else%
266 \begin{mhview}[id=#2,#1]{#3}{#4}\fi%
267 \ignorespacesandpars}
268 {\end{mhview}}\ignorespacesandparsafterend}

```

**mhviewnl** The `mhviewnl` environment is just a layer over the `mhview` environment with the keys and language suitably adapted.<sup>3</sup>

```

269 \newenvironment{mhviewnl}[5][]{% keys, id, lang, from, to
270 \def\@test{#1}\ifx\@test\@empty%
271 \begin{mhview}[id=#2.#3]{#4}{#5}\else%
272 \begin{mhview}[id=#2.#3,#1]{#4}{#5}\fi%
273 \ignorespacesandpars}
274 {\end{mhview}}\ignorespacesandparsafterend}
275 \</smultiling>

```

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

```

276 \<*structview>
277 \ProvidesPackage{structview-mh}[2019/03/20 v1.1 MathHub support for the sTeX structview package
278 \RequirePackage{mathhub}

```

**mhstructure**

```

279 \newenvironment{mhstructure}[3][]{%
280 \gdef\@doit{\importmhmodule[#1]{#3}}%
281 \ifmod@show\par\noindent structure import "#2" from module #3 \@doit\fi%
282 \ignorespacesandpars}
283 {\aftergroup\@doit\ifmod@show end import\fi%
284 \ignorespacesandparsafterend}

```

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

```

285 \newenvironment{importmhmodulevia}[2][]{%
286 {\PackageError{structview-mh}%
287 {The {importmhmodulevia} environment is deprecated}{use the {mhstructure} instead!}%
288 \begin{mhstructure}[#1]{missing}{#2}}
289 {\end{mhstructure}}}

290 \srefaddidkey{mhview}
291 \addmetakey{mhview}{display}

```

<sup>3</sup>EDNOTE: 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`!

```

292 \addmetakey{mhview}{creators}
293 \addmetakey{mhview}{contributors}
294 \addmetakey{mhview}{srccite}
295 \addmetakey*{mhview}{title}
296 \addmetakey{mhview}{type}
297 \addmetakey{mhview}{fromrepos}
298 \addmetakey{mhview}{torepos}
299 \addmetakey{mhview}{frompath}
300 \addmetakey{mhview}{topath}

mhview the MathHub version
301 \newenvironment{mhview}[3][]{% keys, from, to
302 {\metasetkeys{mhview}{#1}%
303 \sref@target%
304 \begin{@mhview}{#2}{#3}%
305 \view@heading{#2}{#3}{\mhview@display}{\mhview@title}%
306 \ignorespacesandpars}
307 {\end{@mhview}\ignorespacesandparsafterend}
308 \ifmod@show\surroundwithmdframed{mhview}\fi

@mhview The @mhview does the actual bookkeeping at the module level.
309 \newenvironment{@mhview}[2]{%from, to
310 \importmhmodule[mhrepos=\mhview@fromrepos,path=\mhview@frompath]{#1}%
311 \importmhmodule[mhrepos=\mhview@torepos,path=\mhview@topath]{#2}%
312 }{}%

mhviewsketch The mhviewsketch environment is deprecated, we give an error
313 \newenvironment{mhviewsketch}[3][]{%
314 {\PackageError{structview}%
315 {The {mhviewsketch} environment is deprecated}{use the {mhview} instead!}%
316 \begin{mhview}[#1]{#2}{#3}}
317 {\end{mhview}}}

mhinlineView Analogous modification to inlineView
318 \newenvironment{mhinlineView}[2][]{% keys, source
319 {\metasetkeys{mhview}{#1}\sref@target%
320 \importmhmodule[mhrepos=\mhview@fromrepos,path=\mhview@frompath]{#2}%
321 \ignorespacesandpars}
322 {\ignorespacesandpars}

mhinlineview
323 \newcommand\mhinlineview[3][]{\begin{mhinlineView}[#1]{#2}{\mod@id#3}\end{mhinlineView}}
324 \end{structview}

```

## 4.7 mikoslides-mh.sty: Support for MiKo Slides

```

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

```



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

```

328 \def\Gin@mhrepos{}
329 \define@key{Gin}{mhrepos}{\def\Gin@mhrepos{#1}}
330 \newcommand\mhframeimage[2][]{\%
331   \setkeys{Gin}{#1}%
332   \edef\mh@@repos{\mh@currentrepos}%
333   \ifx\Gin@mhrepos\@empty%
334     \frameimage[#1]{\MathHub{\mh@currentrepos/source/#2}}%
335   \else%
336     \frameimage[#1]{\MathHub{\Gin@mhrepos/source/#2}}%
337   \fi%
338 }%
```

`\mhinputref*`

```

339 \let\orig@mhinputref\mhinputref
340 \def\mhinputref{\@ifstar\nmhinputref\orig@mhinputref}
341 \newcommand\nmhinputref[2][]{\ifnotes\orig@mhinputref[#1]{#2}\fi}
342 \</mikoslides>
```

## 4.8 `problem-mh.sty`: Support for Problems

```

343 <*problem>
344 \ProvidesPackage{problem-mh}[2019/03/20 v1.1 MathHub support for the sTeX problem package]
345 \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`.

```

346 \addmetakey{inclprob}{mhrepos}
347 \newcommand\includemhproblem[2][]{\metasetkeys{inclprob}{#1}%
348 \edef\mh@@repos{\mh@currentrepos}%
349 \ifx\inclprob@mhrepos\@empty\else\mhcurrentrepos\inclprob@mhrepos\fi%
350 \input{\MathHub{\mh@currentrepos/source/#2}}%
351 \mhcurrentrepos\mh@@repos\clear@inclprob@keys}
352 </problem>
```

## 4.9 `hwexam-mh.sty`: Support for Assignments

```

353 <*hwexam>
354 \ProvidesPackage{hwexam-mh}[2019/03/20 v1.1 MathHub support for the sTeX hwexam package]
355 \RequirePackage{mathhub}
```

`\inputmhassignment` The `\inputmhassignment` 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`.

```

356 \newcommand\inputmhassignment[2][]{\metasetkeys{inclassig}{#1}%
```

```

357 \edef\mh@@repos{\mh@currentrepos}%
358 \ifx\inclassig@mhrepos@empty\else\mhcurrentrepos\inclassig@mhrepos\fi%
359 \inputassignment[#1]{\MathHub{\mh@currentrepos/source/#2}}%
360 \mhcurrentrepos\mh@@repos\clear@inclassig@keys}
361 \newcommand\includemhassignment[2][\newpage\inputmhassignment[#1]{#2}]
362 \hwexam

```

## 4.10 tikzinput-mh.sty: Support for Assignments

```

363 \tikzinput
364 \ProvidesPackage{tikzinput-mh}[2019/03/20 v1.1 MathHub support for the sTeX tikzinput package]
365 \RequirePackage{mathhub}
366 \RequirePackage{pathsuris}

367 \define@key{Gin}{mhrepos}{\def\Gin@mhrepos{#1}}
368 \newcommand\mhtikzinput[2][\def\Gin@mhrepos{}\setkeys{Gin}{#1}]%
369 \edef\mh@@repos{\mh@currentrepos}%
370 \ifx\Gin@mhrepos@empty\tikzinput[#1]{\MathHub{\mh@currentrepos/source/#2}}%
371 \else\mhcurrentrepos\Gin@mhrepos\tikzinput[#1]{\MathHub{\Gin@mhrepos/source/#2}}\fi
372 \def\Gin@mhrepos{}\mhcurrentrepos\mh@@repos}
373 \newcommand\cmhtikzinput[2][\begin{center}\mhtikzinput[#1]{#2}\end{center}]
374 \tikzinput

```

## 4.11 lstmh.sty: Support for Listings

```

375 \lst
376 \ProvidesPackage{lstmh}[2019/03/20 v1.1 MathHub support for the listings package]
377 \RequirePackage{mathhub}
378 \RequirePackage{pathsuris}
379 \RequirePackage{listings}

380 \define@key{lst}{mhrepos}{\def\lst@mhrepos{#1}}
381 \newcommand\lstinputmhlisting[2][\def\lst@mhrepos{}\setkeys{lst}{#1}]%
382 \edef\mh@@repos{\mh@currentrepos}%
383 \ifx\lst@mhrepos@empty\lstinputlisting[#1]{\MathHub{\mh@currentrepos/source/#2}}%
384 \else\lstinputlisting[#1]{\MathHub{\lst@mhrepos/source/#2}}\fi
385 \def\lst@mhrepos{}\mhcurrentrepos\mh@@repos}
386 \newcommand\clstinputmhlisting[2][\begin{center}\lstinputmhlisting[#1]{#2}\end{center}]
387 \lst

```

## Change History

v1.0		libinput to input first the
General: Deprecated		meta-inf-level and then
mhviewsketch . . . . . 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

- [ ] *The Local MathHub Tool (LMH)*. URL: <https://uniformal.github.io/doc/archives/Mathhub/lmh.html> (visited on 10/02/2020).
- [ ] *The MMT Language and System; Building Documents*. URL: <http://uniformal.github.io/doc/applications/building> (visited on 10/02/2020).
- [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: [https://kwarc.info/frabe/Research/HIJKR\\_dimensions\\_11.pdf](https://kwarc.info/frabe/Research/HIJKR_dimensions_11.pdf).
- [KGA20] Michael Kohlhase, Deyan Ginev, and Rares Ambrus. *modules.sty: Semantic Macros and Module Scoping in sTeX*. Tech. rep. 2020. URL: <https://github.com/sLaTeX/sTeX/raw/master/sty/pathsuris/pathsuris.pdf>.
- [Koh20] Michael Kohlhase. *metakeys.sty: A generic framework for extensible Metadata in L<sup>A</sup>T<sub>E</sub>X*. Tech. rep. 2020. URL: <https://github.com/sLaTeX/sTeX/raw/master/sty/metakeys/metakeys.pdf>.
- [MMT] *MMT – Language and System for the Uniform Representation of Knowledge*. Project web site. URL: <https://uniformal.github.io/> (visited on 01/15/2019).
- [sTeX] *sTeX: A semantic Extension of TeX/LaTeX*. URL: <https://github.com/sLaTeX/sTeX> (visited on 05/11/2020).