

MathHub Support for \S T E X^*

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

October 28, 2020

Abstract

The `mathhub` package collection is part of the \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 \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 \S T E X documents in its own right.

Contents

1	Introduction	3
2	The User Interface	4
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
3	Limitations	7

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

4	Implementation	8
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 \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` (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 \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, 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 \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 \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. The mechanics of the `mathhub` bundle is as follows: For most \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 \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, \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{preamble}
```

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

2.2 omdoc-mh.sty: MH Document Infrastructure

`\addmhbibresource` The `\addmhbibresource` macro is a variant of `\addbibresource` from `bib \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 \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 `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 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}
```

¹EdNOTE: 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.
5 \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 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.
6 \def\modules@@first#1/#2;{#1}
7 \newcommand\libinput[1]{%
8 \ifcsvoid{mh@currentrepos}{%
9 \PackageError{mathhub}{current MathHub repository not found}{}}%
10 {}
11 \edef\@mh@group{\expandafter\modules@@first\mh@currentrepos;}
12 \let\orig@inffile\mh@inffile\let\orig@libfile\mh@libfile
13 \def\mh@inffile{\MathHub{\@mh@group/meta-inf/lib/#1}}
14 \def\mh@libfile{\MathHub{\mh@currentrepos/lib/#1}}%
15 \IfFileExists\mh@inffile{\input\mh@inffile}{}%
16 \IfFileExists\mh@inffile{\IfFileExists\mh@libfile}{%
17 {\PackageError{mathhub}
18 {Library file missing; cannot input #1.tex\MessageBreak%
19 Both \mh@libfile.tex\MessageBreak and \mh@inffile.tex\MessageBreak%
20 do not exist}%
21 {Check whether the file name is correct}}}%
22 \IfFileExists\mh@libfile{\input\mh@libfile\relax}{}
23 \let\mh@inffile\orig@inffile\let\mh@libfile\orig@libfile}

\libusepackage the \libusepackage is analogous to \libinput
24 \newcommand\libusepackage[2][{}]{%
25 \edef\@mh@group{\expandafter\modules@@first\mh@currentrepos;}
26 \let\orig@inffile\mh@inffile\let\orig@libfile\mh@libfile
27 \edef\mh@inffile{\MathHub{\@mh@group/meta-inf/lib/#2}}
28 \edef\mh@libfile{\MathHub{\mh@currentrepos/lib/#2}}%
29 \IfFileExists{\mh@inffile.sty}{\usepackage[#1]{\mh@inffile}}{}%
30 \IfFileExists {\mh@inffile.sty}{\IfFileExists{\mh@libfile.sty}{}%
31 {\PackageError{mathhub}
32 {Library file missing; cannot use package #2.sty\MessageBreak%
33 Both \mh@libfile.sty\MessageBreak and \mh@inffile.sty\MessageBreak%
```



```

34     do not exist}%
35   {Check whether the file name is correct}}}}
36 \IfFileExists{mh@libfile{\input{mh@libfile\relax}}{}
37 \let{mh@inffile\orig@inffile\let{mh@libfile\orig@libfile}

```

Generally, the \TeX formatter `pdflatex` needs to know the file system paths of the referenced \TeX files: usually long relative paths. The `pathsuris` package [KGA20] from the \TeX 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` (\LaTeX can run this even in paranoid mode) and then set the MathHub path using `\defpath`.

```

38
39 \kpsewhich{mathhub@path}{--var-value MATHHUB}
40 \if@iswindows@{windows@to@path}{mathhub@path}{fi
41 \ifx{mathhub@path}{empty}%
42   \PackageError{mathhub}{MATHHUB system variable not found or wrongly set}
43 \else\defpath{MathHub}{\mathhub@path}\fi
44

```

`\findmanifest` `\findmanifest{<path>}` searches for a file `MANIFEST.MF` up and over `<path>` in the file system tree.

```

45 \def\findmanifest#1{
46   \cpath{#1}
47   \ifx{@CanPath}{Slash}
48     \def\manifest@mf{}
49   \else\ifx{@CanPath}{empty}
50     \def\manifest@mf{}
51   \else
52     \edef\@findmanifest@path{\@CanPath/MANIFEST.MF}
53     \if@iswindows@{path@to@windows}{\@findmanifest@path}{fi
54     \IfFileExists{\@findmanifest@path}{
55       \%message{MANIFEST.MF found at \@findmanifest@path}
56       \edef\manifest@mf{\@findmanifest@path}
57       \xdef\temp@archive@dir{\expandafter\detokenize\expandafter{\@CanPath}}
58     }{
59     \edef\@findmanifest@path{\@CanPath/META-INF/MANIFEST.MF}
60     \if@iswindows@{path@to@windows}{\@findmanifest@path}{fi
61     \IfFileExists{\@findmanifest@path}{
62       \%message{MANIFEST.MF found at \@findmanifest@path}
63       \edef\manifest@mf{\@findmanifest@path}
64       \xdef\temp@archive@dir{\expandafter\detokenize\expandafter{\@CanPath}}
65     }{
66     \edef\@findmanifest@path{\@CanPath/meta-inf/MANIFEST.MF}
67     \if@iswindows@{path@to@windows}{\@findmanifest@path}{fi
68     \IfFileExists{\@findmanifest@path}{
69       \%message{MANIFEST.MF found at \@findmanifest@path}
70       \edef\manifest@mf{\@findmanifest@path}

```

```

71     \xdef\temp@archive@dir{\expandafter\detokenize\expandafter{\@CanPath}}
72   }{
73     \findmanifest{\@CanPath/..}
74   }}
75 \fi\fi
76 }

```

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

```

77 \def\split@manifest@key{
78   \IfSubStr{\manifest@line}{\@Colon}{
79     \StrBefore{\manifest@line}{\@Colon}[\manifest@key]
80     \StrBehind{\manifest@line}{\@Colon}[\manifest@line]
81     \trimstring\manifest@line
82     \trimstring\manifest@key
83   }{
84     \def\manifest@key{}
85   }
86 }

```

the next helper function iterates over lines in MANIFEST.MF

```

87 \def\parse@manifest@loop{
88   \ifeof\@manifest
89   \else
90     \read\@manifest to \manifest@line\relax
91     \edef\manifest@line{\expandafter\detokenize\expandafter{\manifest@line}}
92     \split@manifest@key
93     % id
94     \IfStrEq\manifest@key{\detokenize{id}}{
95       %\message{archive id: \manifest@line}
96       \xdef\manifest@mf@id{\manifest@line}
97     }{
98       % narration-base
99       \IfStrEq\manifest@key{\detokenize{narration-base}}{
100         %\message{archive narration-base: \manifest@line}
101         \xdef\manifest@mf@narr{\manifest@line}
102       }{
103         % namespace
104         \IfStrEq\manifest@key{\detokenize{source-base}}{
105           %\message{archive source-base: \manifest@line}
106           \xdef\manifest@mf@ns{\manifest@line}
107         }{
108           \IfStrEq\manifest@key{\detokenize{ns}}{
109             %\message{archive ns: \manifest@line}
110             \xdef\manifest@mf@ns{\manifest@line}
111           }{
112             % dependencies
113             \IfStrEq\manifest@key{\detokenize{dependencies}}{
114               %\message{archive dependencies: \manifest@line}
115               \xdef\manifest@mf@deps{\manifest@line}
116             }{

```

```

117     }}}}
118     \parse@manifest@loop
119   \fi
120 }

\parsemanifest \parsemanifest{<macroname>}{<path>} finds MANIFEST.MF via \findmanifest{<path>},
and parses the file, storing the individual fields (id, narr, ns and dependencies)
in <macroname>id, <macroname>narr, etc.
121 \newread\@manifest
122 \def\parsemanifest#1#2{
123   \gdef\temp@archive@dir{}
124   \findmanifest{#2}
125   \begingroup
126     \gdef\manifest@mf@id{}
127     \gdef\manifest@mf@narr{}
128     \gdef\manifest@mf@ns{}
129     \gdef\manifest@mf@deps{}
130     \openin\@manifest\manifest@mf
131     \parse@manifest@loop
132     \closein\@manifest
133   \endgroup
134   \if@iswindows@\windows@to@path\manifest@mf\fi
135   \cslet{#1id}\manifest@mf@id
136   \cslet{#1narr}\manifest@mf@narr
137   \cslet{#1ns}\manifest@mf@ns
138   \cslet{#1deps}\manifest@mf@deps
139   \cslet{#1dir}\temp@archive@dir
140 }

\setcurrentreposinfo \setcurrentreposinfo{<id>} sets the current repository to <id>, checks if the
MANIFEST.MF of this repository has already been read, and if not, find it, parses
it and stores the values in \currentrepos@<key>@<id> for later retrieval.
141 \def\setcurrentreposinfo#1{%
142   \ifcsdef{currentrepos@dir@#1}{%
143     \mhcurrentrepos{#1}%
144   }{%
145     \parsemanifest{mathhub@archive@}{\MathHub{#1}}%
146     \@setcurrentreposinfo%
147   }%
148 }
149
150 \def\@setcurrentreposinfo{%
151   \mhcurrentrepos{\mathhub@archive@id}%
152   \csxdef{currentrepos@dir@\mathhub@archive@id}{\mathhub@archive@dir}%
153   \csxdef{currentrepos@narr@\mathhub@archive@id}{\mathhub@archive@narr}%
154   \csxdef{currentrepos@ns@\mathhub@archive@id}{\mathhub@archive@ns}%
155   \csxdef{currentrepos@deps@\mathhub@archive@id}{\mathhub@archive@deps}%
156 }

```

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

```

157 \parsemanifest{mathhub@archive@}\stex@maindir
158 \@setcurrentreposinfo
159 \</package>

```

4.2 omdoc--mh.sty: MH Document Infrastructure

```

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

```

\addmhbibresource

```

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

```

4.3 modules-mh.sty: MH Variants for Modules

```

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

```

175 \srefaddidkey{importmhmodule}%
176 \addmetakey{importmhmodule}{mhrepos}%
177 \addmetakey{importmhmodule}{path}%
178 \addmetakey{importmhmodule}{dir}%
179 \addmetakey[false]{importmhmodule}{conservative}[true]%
180 \newcommand\importmhmodule[2] [] {%
181   \usemodule@maybesetcodes
182   \metasetkeys{importmhmodule}{#1}%
183   \ifx\importmhmodule@dir\@empty%
184     \edef\@path{\importmhmodule@path}%
185   \else\edef\@path{\importmhmodule@dir/#2}\fi%
186   \ifx\@path\@empty% if module name is not set
187     \importmodule[id=\importmhmodule@id]{#2}%
188   \else%
189     \edef\mh@@repos{\mh@currentrepos}% remember so that we can reset it.
190     \ifx\importmhmodule@mhrepos\@empty% if in the same repos

```

```

191     \relax% no need to change mh@currentrepos, i.e., current directory.
192   \else%
193     \setcurrentreposinfo{\importmhmodule@mhrepos}% change it.
194     \addto@thismodule{x\{noexpand\setcurrentreposinfo{\importmhmodule@mhrepos}}}%
195   \fi%
196   \importmodule[load=\MathHub{\mh@currentrepos/source/\@path},
197               id=\importmhmodule@id]{#2}%
198   \setcurrentreposinfo{\mh@@repos}% after importing, reset to old value
199   \addto@thismodule{x\{noexpand\setcurrentreposinfo{\mh@@repos}}}%
200 \fi%
201 \ignorespacesandpars%
202 }

```

and now the analogs

`\usemhmodule`

```

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

```

`\mhinputref`

```

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

```

`\mhinput`

```

226 \let\mhinput\mhinputref%
227 </modules>

```

4.4 omtex-mh.sty: MH Variants for OMTex

```

228 <*omtext>
229 \ProvidesPackage{omtext-mh}[2019/03/20 v1.1 MathHub support for the sTeX omtex package]

```

```

230 \RequirePackage{mathhub}

\mhgraphics Use the current value of \mh@currentrepos or the value of the mhrepos key if it
is given in \my*graphics.
231 \def\Gin@mhrepos{}
232 \define@key{Gin}{mhrepos}{\def\Gin@mhrepos{#1}}
233 \newcommand\mhgraphics[2][]{\setkeys{Gin}{#1}%
234 \edef\mh@crepos{\mh@currentrepos}%
235 \ifx\Gin@mhrepos\empty\includegraphics[#1]{\MathHub{\mh@currentrepos/source/#2}}%
236 \else\includegraphics[#1]{\MathHub{\Gin@mhrepos/source/#2}}\fi
237 \def\Gin@mhrepos{}\setcurrentreposinfo\mh@crepos}
238 \newcommand\cmhgraphics[2][]{\begin{center}\mhgraphics[#1]{#2}\end{center}}

```

The following macros are deprecated.

```

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

```

4.5 smultiling-mh.sty: MH Variants for Multilinguality

```

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

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

mhmodnl:*
252 \addmetakey{mhmodnl}{mhrepos}
253 \addmetakey{mhmodnl}{path}
254 \addmetakey*{mhmodnl}{title}
255 \addmetakey*{mhmodnl}{creators}
256 \addmetakey*{mhmodnl}{contributors}
257 \addmetakey{mhmodnl}{srccite}
258 \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.
259 \newenvironment{mhmodnl}[3][]{\metasetkeys{mhmodnl}{#1}\def\@test{#1}%
260 \edef\@crepos{\ifx\mhmodnl@mhrepos\empty\mh@currentrepos\else\mhmodnl@mhrepos\fi}%
261 \edef\@load{\MathHub{\@crepos/source/\ifx\mhmodnl@path\empty #2\else\mhmodnl@path\fi}}%
262 \ifx\@test\empty\begin{modnl}[load=\@load]{#2}{#3}\else\begin{modnl}[load=\@load,#1]{#2}{#3}\fi}
263 \ignorespacesandpars}
264 {\end{modnl}\ignorespacesandparsafterend}

```

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

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

`mhviewnl` The `mhviewnl` environment is just a layer over the `mhview` environment with the keys and language suitably adapted.²

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

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

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

`mhstructure`

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

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

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

292 \srefaddidkey{mhview}
293 \addmetakey{mhview}{display}
294 \addmetakey{mhview}{creators}
295 \addmetakey{mhview}{contributors}
296 \addmetakey{mhview}{srccite}
297 \addmetakey*{mhview}{title}
```

²EDNOTE: 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`!

```

298 \addmetakey{mhview}{type}
299 \addmetakey{mhview}{fromrepos}
300 \addmetakey{mhview}{torepos}
301 \addmetakey{mhview}{frompath}
302 \addmetakey{mhview}{topath}

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

@mhview The @mhview does the actual bookkeeping at the module level.
311 \newenvironment{@mhview}[2] {%from, to
312 \usemhmodule[mhrepos=\mhview@fromrepos,path=\mhview@frompath]{#1}%
313 \usemhmodule[mhrepos=\mhview@torepos,path=\mhview@topath]{#2}%
314 }{}%

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

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

mhinlineview
325 \newcommand\mhinlineview[3] [] {\begin{mhinlineView}[#1]{#2}{\module@id}#3\end{mhinlineView}}
326 \</structview>

4.7 mikoslides-mh.sty: Support for MiKo Slides
327 <*mikoslides>
328 \ProvidesPackage{mikoslides-mh}[2019/03/20 v1.1 MathHub support for the sTeX mikoslides package]
329 \RequirePackage{mathhub}

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

```



```

331 \define@key{Gin}{mhrepos}{\def\Gin@mhrepos{#1}}
332 \newcommand\mhframeimage[2][]{\%
333   \setkeys{Gin}{#1}%
334   \edef\mh@@repos{\mh@currentrepos}%
335   \ifx\Gin@mhrepos\empty%
336     \frameimage[#1]{\MathHub{\mh@currentrepos/source/#2}}%
337   \else%
338     \frameimage[#1]{\MathHub{\Gin@mhrepos/source/#2}}%
339   \fi%
340 }%

```

\mhinputref*

```

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

```

4.8 problem-mh.sty: Support for Problems

```

345 \<problem>
346 \ProvidesPackage{problem-mh}[2019/03/20 v1.1 MathHub support for the sTeX problem package]
347 \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`.

```

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

```

4.9 hwexam-mh.sty: Support for Assignments

```

355 \<hwexam>
356 \ProvidesPackage{hwexam-mh}[2019/03/20 v1.1 MathHub support for the sTeX hwexam package]
357 \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`.

```

358 \newcommand\inputmhassignment[2][]{\metasetkeys{inclassig}{#1}%
359 \edef\mh@@repos{\mh@currentrepos}%
360 \ifx\inclassig@mhrepos\empty\else\setcurrentreposinfo\inclassig@mhrepos\fi%
361 \inputassignment[#1]{\MathHub{\mh@currentrepos/source/#2}}%

```

```

362 \setcurrentreposinfo\mh@repos\clear@inclassig@keys}
363 \newcommand\includemhassignment[2] [] {\newpage\inputmhassignment[#1]{#2}}
364 \</hwexam>

```

4.10 tikzinput-mh.sty: Support for Assignments

```

365 < *tikzinput>
366 \ProvidesPackage{tikzinput-mh}[2019/03/20 v1.1 MathHub support for the sTeX tikzinput package]
367 \RequirePackage{mathhub}
368 \RequirePackage{pathsuris}

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

```

4.11 lstmh.sty: Support for Listings

```

377 < *lst>
378 \ProvidesPackage{lstmh}[2019/03/20 v1.1 MathHub support for the listings package]
379 \RequirePackage{mathhub}
380 \RequirePackage{pathsuris}
381 \RequirePackage{listings}

382 \define@key{lst}{mhrepos}{\def\lst@mhrepos{#1}}
383 \newcommand\lstinputmhlisting[2] [] {\def\lst@mhrepos{}\setkeys{lst}{#1}%
384 \edef\mh@repos{\mh@currentrepos}%
385 \ifx\lst@mhrepos\empty\lstinputlisting[#1]{\MathHub{\mh@currentrepos/source/#2}}}%
386 \else\lstinputlisting[#1]{\MathHub{\lst@mhrepos/source/#2}}\fi
387 \def\lst@mhrepos{}\setcurrentreposinfo\mh@repos}
388 \newcommand\clstinputmhlisting[2] [] {\begin{center}\lstinputmhlisting[#1]{#2}\end{center}}
389 < /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.
- [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^AT_EX*. 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).