MathHub Support for sTeX*

Michael Kohlhase
FAU Erlangen-Nürnberg
http://kwarc.info/kohlhase

December 31, 2020

Abstract

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

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

Contents

---

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

1 Introduction

As ſTEX 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 ⤳ manageable file system layout schema that has co-eveolved with the MathHub.info portal for active documents, but is useful for organizing collections of ſTEX documents in its own right. In particular, since the layout scheme is supported by the `lmh` (local m̲ath̲h̲ub []), 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 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 ſTEX 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 conibetains all ſTEX 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 ſTEX 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 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 ſTEX author with MathHub-enabled versions of the ſTEX 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 ſTEX 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 sTEX packages that have MathHub functionality.

2.1 `mathhub.sty`: General Infrastructure

\mhcurrentrepos

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, sTEX 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. Fort the group level, the MathHub file system layout uses a specical repository $\langle group\rangle$/`meta-inf/lib` and at the repository level we use $\langle group\rangle$/$\langle repos\rangle$/`lib` for such files. The

\libinput

`\libinput` macro supports this practice: `\libinput{`$\langle filename\rangle$`}` macro inputs the files $\langle group\rangle$/`meta-inf/lib/`$\langle filename\rangle$ and then $\langle group\rangle$/$\langle repos\rangle$/`lib/`$\langle filename\rangle$ if they exist. Thus a typical top-level sTEX 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 bibLATEX with repository support. Concretely, `\addmhbibresource[`$\langle repos\rangle$`]{`$\langle path\rangle$`}` expands to `\addbibresource{`$\langle MathHub\rangle$/$\langle reponame\rangle$/$\langle path\rangle$`}`, where $\langle reponame\rangle$ is $\langle repo\rangle$ 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}
```

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 loaded, `\importmhmodule` is the same as `\importmodule`.

Of course, LaTeX does not know about the repositories when they are called `\mhcurrentrepos` from a file system, so we can use the `\mhcurrentrepos` macro to tell them. But this is usually only needed to initialize the infrastructure in the driver file. In particular, we do not need to set it in 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 `omtext-mh.sty`: MH Variants for OMText

`\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

<div style="margin-left: auto; text-align: right">mhmodsig
mhmodnl</div>

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}
```

---

[1] 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 ⟨∗package⟩
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 \if@iswindows@%
16   \path@to@windows\mh@inffile%
17   \path@to@windows\mh@libfile%
18 \fi%
19 \IfFileExists\mh@inffile{\input\mh@inffile}{}%
20 \IfFileExists\mh@inffile{}{\IfFileExists\mh@libfile{}{%
21   {\PackageError{mathhub}%
22     {Library file missing; cannot input #1.tex\MessageBreak%
23     Both \mh@libfile.tex\MessageBreak and \mh@inffile.tex\MessageBreak%
24     do not exist}%
25   {Check whether the file name is correct}}}}%
26 \IfFileExists\mh@libfile{\input\mh@libfile\relax}{}%
27 \let\mh@inffile\orig@inffile\let\mh@libfile\orig@libfile}
```

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

```
28 \newcommand\libusepackage[2][]{%
29 \edef\@mh@group{\expandafter\modules@@first\mh@currentrepos;}%
30 \let\orig@inffile\mh@inffile\let\orig@libfile\mh@libfile%
31 \edef\mh@inffile{\MathHub{\@mh@group/meta-inf/lib/#2}}%
32 \edef\mh@libfile{\MathHub{\mh@currentrepos/lib/#2}}%
33 \if@iswindows@%
```

8

```
34    \path@to@windows\mh@inffile%
35    \path@to@windows\mh@libfile%
36  \fi%
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 TeX formatter pdflatex needs to know the file system paths of the referenced sTeX files: usually long relative paths. The pathsuris package [KGA20] from the sTeX 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 MathMub path using \defpath.

```
46
47  \kpsewhich\mathhub@path{--var-value MATHHUB}
48  \if@iswindows@\windows@to@path\mathhub@path\fi
49  \ifx\mathhub@path\@empty%
50  \PackageError{mathhub}
51    {MATHHUB system variable not found or wrongly set}
52    {use export MATHHUB="<path>", where <path> points your MathHub direcctory}
53  \else\defpath{MathHub}{\mathhub@path}\fi
54  \ifcsvoid{mathhub@path}{}{\message{^^JMATHHUB: \mathhub@path}}
55
```

\findmanifest \findmanifest{⟨path⟩} searches for a file MANIFEST.MF up and over ⟨path⟩ in the file system tree.

```
56  \def\findmanifest#1{%
57    \@cpath{#1}%
58    \ifx\@CanPath\@Slash%
59      \def\manifest@mf{}%
60    \else\ifx\@CanPath\@empty%
61        \def\manifest@mf{}%
62    \else%
63      \edef\@findmanifest@path{\@CanPath/MANIFEST.MF}%
64      \if@iswindows@\path@to@windows\@findmanifest@path\fi%
65      %\message{^^JHere: \@findmanifest@path^^J}
66      \IfFileExists{\@findmanifest@path}{%
67        %\message{MANIFEST.MF found at \@findmanifest@path}
68        \edef\manifest@mf{\@findmanifest@path}%
69        \xdef\temp@archive@dir{\expandafter\detokenize\expandafter{\@CanPath}}}%
70      }{%
```

```
71    \edef\@findmanifest@path{\@CanPath/META-INF/MANIFEST.MF}%
72    \if@iswindows@\path@to@windows\@findmanifest@path\fi%
73    \IfFileExists{\@findmanifest@path}{%
74      %\message{MANIFEST.MF found at \@findmanifest@path}
75      \edef\manifest@mf{\@findmanifest@path}%
76      \xdef\temp@archive@dir{\expandafter\detokenize\expandafter{\@CanPath}}%
77    }{%
78    \edef\@findmanifest@path{\@CanPath/meta-inf/MANIFEST.MF}%
79    \if@iswindows@\path@to@windows\@findmanifest@path\fi%
80    \IfFileExists{\@findmanifest@path}{%
81      %\message{MANIFEST.MF found at \@findmanifest@path}
82      \edef\manifest@mf{\@findmanifest@path}%
83      \xdef\temp@archive@dir{\expandafter\detokenize\expandafter{\@CanPath}}%
84    }{%
85      \findmanifest{\@CanPath/..}%
86    }}}%
87  \fi\fi%
88 }
```

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

```
89 \def\split@manifest@key{%
90   \IfSubStr{\manifest@line}{\@Colon}{%
91       \StrBefore{\manifest@line}{\@Colon}[\manifest@key]%
92       \StrBehind{\manifest@line}{\@Colon}[\manifest@line]%
93       \trimstring\manifest@line%
94       \trimstring\manifest@key%
95   }{%
96       \def\manifest@key{}%
97   }%
98 }
```

the next helper function iterates over lines in `MANIFEST.MF`

```
99  \def\parse@manifest@loop{%
100   \ifeof\@manifest%
101   \else%
102     \read\@manifest to \manifest@line\relax%
103     \edef\manifest@line{\expandafter\detokenize\expandafter{\manifest@line}}%
104     \split@manifest@key%
105     % id
106     \IfStrEq\manifest@key{\detokenize{id}}{%
107         %\message{archive id: \manifest@line}
108         \xdef\manifest@mf@id{\manifest@line}%
109     }{%
110     % narration-base
111     \IfStrEq\manifest@key{\detokenize{narration-base}}{%
112         %\message{archive narration-base: \manifest@line}
113         \xdef\manifest@mf@narr{\manifest@line}%
114     }{%
115     % namespace
116     \IfStrEq\manifest@key{\detokenize{source-base}}{%
```

```
117      % \message{archive source-base: \manifest@line}
118        \xdef\manifest@mf@ns{\manifest@line}%
119      }{%
120      \IfStrEq\manifest@key{\detokenize{ns}}{%
121          %\message{archive ns: \manifest@line}
122          \xdef\manifest@mf@ns{\manifest@line}%
123      }{%
124      % dependencies
125      \IfStrEq\manifest@key{\detokenize{dependencies}}{%
126          %\message{archive dependencies: \manifest@line}
127          \xdef\manifest@mf@deps{\manifest@line}%
128      }{%
129      }}}}}%
130      \parse@manifest@loop%
131    \fi%
132 }
```

\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.

```
133 \newread\@manifest
134 \def\parsemanifest#1#2{%
135    \gdef\temp@archive@dir{}%
136    \findmanifest{#2}%
137    \begingroup%
138      \gdef\manifest@mf@id{}%
139      \gdef\manifest@mf@narr{}%
140      \gdef\manifest@mf@ns{}%
141      \gdef\manifest@mf@deps{}%
142      \openin\@manifest\manifest@mf\relax%
143      \parse@manifest@loop%
144      \closein\@manifest%
145    \endgroup%
146    \if@iswindows@\windows@to@path\manifest@mf\fi%
147    \cslet{#1id}\manifest@mf@id%
148    \cslet{#1narr}\manifest@mf@narr%
149    \cslet{#1ns}\manifest@mf@ns%
150    \cslet{#1deps}\manifest@mf@deps%
151    \cslet{#1dir}\temp@archive@dir%
152 }
```

\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.

```
153 \def\setcurrentreposinfo#1{%
154    \ifcsdef{currentrepos@dir@#1}{%
155      \mhcurrentrepos{#1}%
156    }{%
157      \parsemanifest{mathhub@archive@}{\MathHub{#1}}%
```

```
158     \@setcurrentreposinfo%
159   }%
160   \ifcsvoid{mathhub@archive@dir}{\PackageError{mathhub}{No archive with %
161     name #1 found!}{make sure that #1 is directly in your MATHHUB folder %
162     and contains a MANIFEST.MF, either directly in #1 or in a meta-inf %
163     subfolder.}}{}%
164 }
165
166 \def\@setcurrentreposinfo{%
167   \mhcurrentrepos{\mathhub@archive@id}%
168   \csxdef{currentrepos@dir@\mathhub@archive@id}{\mathhub@archive@dir}%
169   \csxdef{currentrepos@narr@\mathhub@archive@id}{\mathhub@archive@narr}%
170   \csxdef{currentrepos@ns@\mathhub@archive@id}{\mathhub@archive@ns}%
171   \csxdef{currentrepos@deps@\mathhub@archive@id}{\mathhub@archive@deps}%
172 }
```

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

```
173 \parsemanifest{mathhub@archive@}\stex@maindir
174 \@setcurrentreposinfo
175 \message{^^J Current MathHub repository: \ifcsvoid{mh@currentrepos}{None}\mh@currentrepos}
176 ⟨/package⟩
```

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

```
177 ⟨∗omdoc⟩
178 \ProvidesPackage{omdoc-mh}[2019/03/20 v1.1 MathHub support for OMDoc Documents]
179 \RequirePackage{mathhub}
```

\addmhbibresource

```
180 \newcommand\addmhbibresource[2][]{%
181   \def\@repos{#1}%
182   \ifx\@repos\@empty%
183   \addbibresource{\MathHub{\mh@currentrepos/#2}}%
184   \else
185   \addbibresource{\MathHub{\@repos/#2}}%
186   \fi%
187   \ignorespacesandpars}%
188 ⟨/omdoc⟩
```

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

```
189 ⟨∗modules⟩
190 \ProvidesPackage{modules-mh}[2019/03/20 v1.1 MathHub support for the sTeX modules package]
191 \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`.

```
192 \srefaddidkey{importmhmodule}%
193 \addmetakey{importmhmodule}{mhrepos}%
194 \addmetakey{importmhmodule}{path}%
195 \addmetakey{importmhmodule}{ext}% why does this exist?
196 \addmetakey{importmhmodule}{dir}%
197 \addmetakey[false]{importmhmodule}{conservative}[true]%
198 \newcommand\importmhmodule[2][]{%
199   \usemodule@maybesetcodes
200   \metasetkeys{importmhmodule}{#1}%
201   \ifx\importmhmodule@dir\@empty%
202     \edef\@path{\importmhmodule@path}%
203   \else\edef\@path{\importmhmodule@dir/#2}\fi%
204   \ifx\@path\@empty% if module name is not set
205     \importmodule[id=\importmhmodule@id]{#2}%
206   \else%
207     \edef\mh@@repos{\mh@currentrepos}% remember so that we can reset it.
208     \ifx\importmhmodule@mhrepos\@empty% if in the same repos
209       \relax% no need to change mh@currentrepos, i.e, current directory.
210     \else%
211       \setcurrentreposinfo{\importmhmodule@mhrepos}% change it.
212       \addto@thismodulex{\noexpand\setcurrentreposinfo{\importmhmodule@mhrepos}}%
213     \fi%
214     \importmodule[load=\MathHub{\mh@currentrepos/source/\@path},
215                        id=\importmhmodule@id]{#2}%
216     \setcurrentreposinfo{\mh@@repos}% after importing, reset to old value
217     \addto@thismodulex{\noexpand\setcurrentreposinfo{\mh@@repos}}%
218   \fi%
219   \ignorespacesandpars%
220 }
```

and now the analogs

`\usemhmodule`

```
221 \newcommand\usemhmodule[2][]{%
222 \metasetkeys{importmhmodule}{#1}%
223 \ifx\importmhmodule@dir\@empty%
224   \edef\@path{\importmhmodule@path}%
225 \else\edef\@path{\importmhmodule@dir/#2}\fi%
226 \ifx\@path\@empty%
227   \usemodule[id=\importmhmodule@id]{#2}%
228 \else%
229   \edef\mh@@repos{\mh@currentrepos}%
230   \ifx\importmhmodule@mhrepos\@empty%
231   \else\setcurrentreposinfo{\importmhmodule@mhrepos}\fi%
232   \usemodule[load=\MathHub{\mh@currentrepos/source/\@path},%
233                        id=\importmhmodule@id]{#2}%
234   \setcurrentreposinfo\mh@@repos%
```

13

```
235 \fi%
236 \ignorespacesandpars}
```

\mhinputref

```
237 \newcommand\mhinputref[2][]{%
238 \def\@repos{#1}%
239 \edef\mh@@repos{\mh@currentrepos}%
240 \ifx\@repos\@empty\else\setcurrentreposinfo{#1}\fi%
241 \inputref{\MathHub{\mh@currentrepos/source/#2}}%
242 \setcurrentreposinfo\mh@@repos%
243 \ignorespacesandpars}
```

\mhinput

```
244 \let\mhinput\mhinputref%
245 ⟨/modules⟩
```

4.4   `omtext-mh.sty`: MH Variants for OMText

```
246 ⟨*omtext⟩
247 \ProvidesPackage{omtext-mh}[2019/03/20 v1.1 MathHub support for the sTeX omtext package]
248 \RequirePackage{mathhub}
```

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

```
249 \def\Gin@mhrepos{}
250 \define@key{Gin}{mhrepos}{\def\Gin@mhrepos{#1}}
251 \newcommand\mhgraphics[2][]{\setkeys{Gin}{#1}%
252 \edef\mh@@repos{\mh@currentrepos}%
253 \ifx\Gin@mhrepos\@empty\edef\temp@path{\MathHub{\mh@currentrepos/source/#2}}%\includegraphics[#
254 \else\edef\temp@path{\MathHub{\Gin@mhrepos/source/#2}}\fi%\includegraphics[#1]{\MathHub{\Gin@mh
255 \if@iswindows@\path@to@windows\temp@path\fi
256 \includegraphics[#1]{\temp@path}
257 \def\Gin@mhrepos{}\setcurrentreposinfo\mh@@repos}
258 \newcommand\cmhgraphics[2][]{\begin{center}\mhgraphics[#1]{#2}\end{center}}
```

The following macros are deprecated.

```
259 \newcommand\mhcgraphics[2][]{\begin{center}\mhgraphics[#1]{#2}\end{center}
260   \PackageWarning{omtext-mh}{\protect\mhcgraphics\space is deprecated, use \protect\cmhgraphics
261 \newcommand\mhbgraphics[2][]{\fbox{\mhgraphics[#1]{#2}}
262   \PackageWarning{omtext-mh}{\protect\mhbgraphics\space is deprecated, use
263     \protect\mhgraphics\space and {center} instead}}
264 \newcommand\mhcbgraphics[2][]{\begin{center}\fbox{\mhgraphics[#1]{#2}}\end{center}
265   \PackageWarning{omtext-mh}{\protect\mhcbgraphics\space is deprecated, use
266     \protect\mhgraphics,\space {center}, and \protect\fbox\space instead}}
267 ⟨/omtext⟩
```

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

268 ⟨∗smultiling⟩
269 \ProvidesPackage{smultiling-mh}[2019/03/20 v1.1 MathHub support for the sTeX smultiling package]
270 \RequirePackage{mathhub}

mhmodsig

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

mhmodnl:*

272 \addmetakey{mhmodnl}{mhrepos}
273 \addmetakey{mhmodnl}{path}
274 \addmetakey*{mhmodnl}{title}
275 \addmetakey*{mhmodnl}{creators}
276 \addmetakey*{mhmodnl}{contributors}
277 \addmetakey{mhmodnl}{srccite}
278 \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.

279 \newenvironment{mhmodnl}[3][]{\metasetkeys{mhmodnl}{#1}\def\@test{#1}%
280 \edef\@repos{\ifx\mhmodnl@mhrepos\@empty\mh@currentrepos\else\mhmodnl@mhrepos\fi}%
281 \edef\@load{\MathHub{\@repos/source/\ifx\mhmodnl@path\@empty #2\else\mhmodnl@path\fi}}%
282 \ifx\@test\@empty\begin{modnl}[load=\@load]{#2}{#3}\else\begin{modnl}[load=\@load,#1]{#2}{#3}\fi
283 \ignorespacesandpars}
284 {\end{modnl}\ignorespacesandparsafterend}

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

285 \newenvironment{mhviewsig}[4][]{% keys, id, from, to
286 \def\@test{#1}\ifx\@test\@empty%
287 \begin{mhview}[id=#2]{#3}{#4}\else%
288 \begin{mhview}[id=#2,#1]{#3}{#4}\fi%
289 \ignorespacesandpars}
290 {\end{mhview}\ignorespacesandparsafterend}

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

EdN:2 appears in left margin

291 \newenvironment{mhviewnl}[5][]{% keys, id, lang, from, to
292 \def\@test{#1}\ifx\@test\@empty%
293 \begin{mhview}[id=#2.#3]{#4}{#5}\else%
294 \begin{mhview}[id=#2.#3,#1]{#4}{#5}\fi%
295 \ignorespacesandpars}
296 {\end{mhview}\ignorespacesandparsafterend}
297 ⟨/smultiling⟩

---

[2]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*⟩!

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

```
298 ⟨∗structview⟩
299 \ProvidesPackage{structview-mh}[2019/03/20 v1.1 MathHub support for the sTeX structview package
300 \RequirePackage{mathhub}
```

mhstructure

```
301 \newenvironment{mhstructure}[3][]{%
302   \gdef\@@doit{\importmhmodule[#1]{#3}}%
303   \ifmod@show\par\noindent structure import "#2" from module #3 \@@doit\fi%
304   \ignorespacesandpars}
305 {\aftergroup\@@doit\ifmod@show end import\fi%
306   \ignorespacesandparsafterend}
```

importmhmodulevia this is now deprecated, we give an error

```
307 \newenvironment{importmhmodulevia}[2][]%
308 {\PackageError{structview-mh}%
309   {The {importmhmodulevia} environment is deprecated}{use the {mhstructure} instead!}%
310   \begin{mhstructure}[#1]{missing}{#2}}
311 {\end{mhstructure}}

312 \srefaddidkey{mhview}
313 \addmetakey{mhview}{display}
314 \addmetakey{mhview}{creators}
315 \addmetakey{mhview}{contributors}
316 \addmetakey{mhview}{srccite}
317 \addmetakey*{mhview}{title}
318 \addmetakey{mhview}{type}
319 \addmetakey{mhview}{fromrepos}
320 \addmetakey{mhview}{torepos}
321 \addmetakey{mhview}{frompath}
322 \addmetakey{mhview}{topath}
```

mhview the MathHub version

```
323 \newenvironment{mhview}[3][]% keys, from, to
324 {\metasetkeys{mhview}{#1}%
325   \sref@target%
326   \begin{@mhview}{#2}{#3}%
327   \view@heading{#2}{#3}{\mhview@display}{\mhview@title}%
328   \ignorespacesandpars}
329 {\end{@mhview}\ignorespacesandparsafterend}
330 \ifmod@show\surroundwithmdframed{mhview}\fi
```

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

```
331 \newenvironment{@mhview}[2]{%from, to
332   \usemhmodule[mhrepos=\mhview@fromrepos,path=\mhview@frompath]{#1}%
333   \usemhmodule[mhrepos=\mhview@torepos,path=\mhview@topath]{#2}%
334 }{}%
```

16

mhviewsketch The `mhviewsketch` environment is deprecated, we give an error

```
335 \newenvironment{mhviewsketch}[3][]%
336 {\PackageError{structview}%
337   {The {mhviewsketch} environment is deprecated}{use the {mhview} instead!}%
338   \begin{mhview}[#1]{#2}{#3}}
339 {\end{mhview}}
```

mhinlineView Analogous modification to `inlineView`

```
340 \newenvironment{mhinlineView}[2][]% keys, source
341 {\metasetkeys{mhview}{#1}\sref@target%
342   \importmhmodule[mhrepos=\mhview@fromrepos,path=\mhview@frompath]{#2}%
343   \ignorespacesandpars}
344 {\ignorespacesandpars}
```

mhinlineview

```
345 \newcommand\mhinlineview[3][]{\begin{mhinlineView}[#1]{#2}{\module@id}#3\end{mhinlineView}}
346 ⟨/structview⟩
```

4.7 mikoslides-mh.sty: Support for MiKo Slides

```
347 ⟨*mikoslides⟩
348 \ProvidesPackage{mikoslides-mh}[2020/12/06 v1.3 MathHub support for the sTeX mikoslides package
349 \RequirePackage{mathhub}
```

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

```
350 \def\Gin@mhrepos{}
351 \define@key{Gin}{mhrepos}{\def\Gin@mhrepos{#1}}
352 \newcommand\mhframeimage[2][]{%
353   \setkeys{Gin}{#1}%
354   \edef\mh@@repos{\mh@currentrepos}%
355   \ifx\Gin@mhrepos\@empty%
356     \edef\temp@path{\MathHub{\mh@currentrepos/source/#2}}%
357   \else%
358     \edef\temp@path{\MathHub{\Gin@mhrepos/source/#2}}%
359   \fi%
360   \if@iswindows@\path@to@windows\temp@path\fi%
361   \frameimage[#1]{\temp@path}%
362 }%
```

\mhinputref*

```
363 \let\orig@mhinputref\mhinputref
364 \def\mhinputref{\@ifstar\nmhinputref\orig@mhinputref}
365 \newcommand\nmhinputref[2][]{\ifnotes\orig@mhinputref[#1]{#2}\fi}
```

\mhexcursion

```
366 \newcommand\activatemhexcursion[2][]{\ifstrempty{#1}%
367   {\gappto\printexcursions{\mhinputref[\mh@currentrepos]{#2}}}%
368   {\gappto\printexcursions{\mhinputref[#1]{#2}}}}
```

```
369 \newcommand\mhexcursion[4][]{% repos, label, path, text
370    \ifnotes\activatemhexcursion[#1]{#3}\excursionref{#2}{#4}\fi}%
```

\mhexcursiongroup

```
371 \srefaddidkey{mhexcursiongroup}%
372 \addmetakey{mhexcursiongroup}{intro}%
373 \addmetakey{mhexcursiongroup}{mhrepos}%
374 \newcommand\mhexcursiongroup[1][]{%
375    \metasetkeys{mhexcursiongroup}{#1}%
376    \ifdefempty\printexcursions{}% only if there are excursions
377    {\begin{omgroup}[#1]{Excursions}%
378        \ifdefempty\mhexcursiongroup@intro{}%
379        {\ifdefempty\mhexcursiongroup@mhrepos%
380           {\mhinputref{\mhexcursiongroup@intro}}%
381           {\mhinputref[\mhexcursiongroup@mhrepos]{\mhexcursiongroup@intro}}}%
382        \printexcursions%
383     \end{omgroup}}}
384 ⟨/mikoslides⟩
```

4.8 **problem-mh.sty**: Support for Problems

```
385 ⟨∗problem⟩
386 \ProvidesPackage{problem-mh}[2019/03/20 v1.1 MathHub support for the sTeX problem package]
387 \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.

```
388 \addmetakey{inclprob}{mhrepos}
389 \newcommand\includemhproblem[2][]{\metasetkeys{inclprob}{#1}%
390 \edef\mh@@repos{\mh@currentrepos}%
391 \ifx\inclprob@mhrepos\@empty\else\setcurrentreposinfo\inclprob@mhrepos\fi%
392 \edef\temp@path{\MathHub{\mh@currentrepos/source/#2}}
393 \if@iswindows@\path@to@windows\temp@path\fi
394 \input{\temp@path}%
395 \setcurrentreposinfo\mh@@repos\clear@inclprob@keys}
396 ⟨/problem⟩
```

4.9 **hwexam-mh.sty**: Support for Assignments

```
397 ⟨∗hwexam⟩
398 \ProvidesPackage{hwexam-mh}[2019/03/20 v1.1 MathHub support for the sTeX hwexam package]
399 \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.

```
400 \newcommand\inputmhassignment[2][]{\metasetkeys{inclassig}{#1}%
401 \edef\mh@@repos{\mh@currentrepos}%
402 \ifx\inclassig@mhrepos\@empty\else\setcurrentreposinfo\inclassig@mhrepos\fi%
403 \edef\temp@path{\MathHub{\mh@currentrepos/source/#2}}%
404 \if@iswindows@\path@to@windows\temp@path\fi%
405 \inputassignment[#1]{\temp@path}%
406 \setcurrentreposinfo\mh@@repos\clear@inclassig@keys}%
407 \newcommand\includemhassignment[2][]{\newpage\inputmhassignment[#1]{#2}}
408 ⟨/hwexam⟩
```

4.10   tikzinput-mh.sty: Support for Assignments

```
409 ⟨∗tikzinput⟩
410 \ProvidesPackage{tikzinput-mh}[2019/03/20 v1.1 MathHub support for the sTeX tikzinput package]
411 \RequirePackage{mathhub}
412 \RequirePackage{pathsuris}

413 \define@key{Gin}{mhrepos}{\def\Gin@mhrepos{#1}}
414 \newcommand\mhtikzinput[2][]{\def\Gin@mhrepos{}\setkeys{Gin}{#1}%
415 \edef\mh@@repos{\mh@currentrepos}%
416 \ifx\Gin@mhrepos\@empty\edef\temp@path{\MathHub{\mh@currentrepos/source/#2}}%
417 \else\setcurrentreposinfo\Gin@mhrepos\edef\temp@path{\MathHub{\Gin@mhrepos/source/#2}}\fi%
418 \if@iswindows@\path@to@windows\temp@path\fi%
419 \tikzinput[#1]{\temp@path}
420 \def\Gin@mhrepos{}\setcurrentreposinfo\mh@@repos}
421 \newcommand\cmhtikzinput[2][]{\begin{center}\mhtikzinput[#1]{#2}\end{center}}
422 ⟨/tikzinput⟩
```

4.11   lstmh.sty: Support for Listings

```
423 ⟨∗lst⟩
424 \ProvidesPackage{lstmh}[2019/03/20 v1.1 MathHub support for the listings package]
425 \RequirePackage{mathhub}
426 \RequirePackage{pathsuris}
427 \RequirePackage{listings}

428 \define@key{lst}{mhrepos}{\def\lst@mhrepos{#1}}
429 \newcommand\lstinputmhlisting[2][]{\def\lst@mhrepos{}\setkeys{lst}{#1}%
430 \edef\mh@@repos{\mh@currentrepos}%
431 \ifx\lst@mhrepos\@empty\edef\temp@path{\MathHub{\mh@currentrepos/source/#2}}%
432 \else\edef\temp@path{\MathHub{\lst@mhrepos/source/#2}}\fi%
433 \if@iswindows@\path@to@windows\temp@path\fi%
434 \lstinputlisting[#1]{\temp@path}
435 \def\lst@mhrepos{}\setcurrentreposinfo\mh@@repos}
436 \newcommand\clstinputmhlisting[2][]{\begin{center}\lstinputmhlisting[#1]{#2}\end{center}}
437 ⟨/lst⟩
```

Change History

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 LaTeX*. 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).