MathHub Support for sTeX*

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

October 1, 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 ? (last revised ?)

1

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` tool (local <u>m</u>ath<u>h</u>ub) [lmh], which automates many management tasks.

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 contains all sTEX sources. These are organized in a two-level folder system that is compatible by GIT repository managers like GitHub [GH] and GitLab [GL]. Even though it is not necessary for the `mathhub` package we will assume that these are GIT repositories, which have names of the form ⟨*group*⟩/⟨*repo*⟩, where ⟨*group*⟩ is a MathHub-unique repository group and ⟨*repo*⟩ a repository name that is ⟨*group*⟩-unique. The file and directory structure of a repository is arbitrary – except that it starts with the directory `source` because they are Math Archives in the sense of [Hor+11]. But this structure can be hidden from the 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.

[1]

2   The User Interface

2.1   `mathhub.sty`: General Infrastructure

Generally, the TEXformatter `pdflatex` needs to know the file system paths of the referenced sTEX files: usually long relative paths. The `pathsuris` package [KGA18] 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).

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

---

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

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

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

Given a systematic grouping in the MathHub file layout scheme, SPEX 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` 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 SPEX file has the following lines in the preamble:

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

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

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

`\addmhbibresource`   The `\addmhbibresource` macro is a variant of `\addbibresource` from bibLATEX with repository support. Concretely, `\addmhbibresource[`$\langle repos \rangle$`]{`$\langle path \rangle$`}` 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}
```

---

[2] EDNOTE: explain the pre.tex and post.tex setup for sTeX modules (see the omdoc package)

4

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, neither LaTeX nor LaTeXMLknow about the repositories when they are called from a file system, so we can use the `\mhcurrentrepos` macro to tell `\mhcurrentrepos` them. But this is 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:2em">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:3

[3]

2.7 **mikoslides-mh.sty**: Support for MiKo Slides

\mhframeimage

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

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

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

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

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

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

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

\mhinputref*

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

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

---

[3]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 [Koh18] to be loaded (in the right version).

```
1 ⟨∗package⟩
2 \RequirePackage{keyval}
3 \RequirePackage{pathsuris}
4 \if@latexml\else
5 \RequirePackage{xparse}
6 \RequirePackage{expl3}
7 \fi
```

4.1 `mathhub.sty`: General Infrastructure

`\mhcurrentrepos`
`\@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.

```
 8 \newcommand\mhcurrentrepos[1]{%
 9   \edef\@test{#1}%
10   \ifx\@test\mh@currentrepos% if new dir = old dir
11     \relax% no need to change
12   \else%
13     \protected@write\@auxout{}{\string\@mhcurrentrepos{#1}}%
14   \fi%
15   \@mhcurrentrepos{#1}% define mh@currentrepos
16 }%
17 \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.

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

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

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

The next piece of code is adapted from https://tex.stackexchange.com/
questions/62010/can-i-access-system-environment-variables-from-latex-for-instance-home,
check there if it breaks. We use it to parse the `MATHHUB` environment variable via
`kpsewhich` (LaTeX can run this even in paranoid mode) and then set the `MathMub`
path using `\defpath`.

```
46 \ExplSyntaxOn
47 \sys_get_shell:nnN{kpsewhich ~ --var-value ~ MATHHUB} { } \MATHHUB
48 \tl_trim_spaces:N \MATHHUB
49 \ifx\MATHHUB\empty\else
50 \def\temp_def_path#1{\defpath{MathHub}{#1}}
51 \expandafter\temp_def_path\expandafter{\MATHHUB}
52 \fi
53 \ExplSyntaxOff
```

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

```
54 \def\findmanifest#1{
55   \@cpath{#1}
56   \ifx\@CanPath\@Slash
57     \def\manifest@mf{}
58   \else\ifx\@CanPath\@empty
59       \def\manifest@mf{}
60   \else
61     \IfFileExists{\@CanPath/MANIFEST.MF}{
62       \edef\manifest@mf{\@CanPath/MANIFEST.MF}
63     }{
64     \IfFileExists{\@CanPath/META-INF/MANIFEST.MF}{
65       \edef\manifest@mf{\@CanPath/META-INF/MANIFEST.MF}
66     }{
67     \IfFileExists{\@CanPath/meta-inf/MANIFEST.MF}{
68       \edef\manifest@mf{\@CanPath/meta-inf/MANIFEST.MF}
69     }{
70       \findmanifest{\@CanPath/..}
71     }}}
```

```
72    \fi\fi
73 }
```

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

```
74 \def\split@manifest@key{
75    \IfSubStr{\manifest@line}{\@Colon}{
76        \StrBefore{\manifest@line}{\@Colon}[\manifest@key]
77        \StrBehind{\manifest@line}{\@Colon}[\manifest@line]
78        \trimstring\manifest@line
79        \trimstring\manifest@key
80    }{
81        \def\manifest@key{}
82    }
83 }
```

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

```
84 \def\parse@manifest@loop{
85    \ifeof\@manifest
86    \else
87       \read\@manifest to \manifest@line\relax
88       \split@manifest@key
89       % id
90       \IfStrEq\manifest@key{id}{
91           \xdef\manifest@mf@id{\manifest@line}
92       }{
93       % narration-base
94       \IfStrEq\manifest@key{narration-base}{
95           \xdef\manifest@mf@narr{\manifest@line}
96       }{
97       % namespace
98       \IfStrEq\manifest@key{source-base}{
99           \xdef\manifest@mf@ns{\manifest@line}
100      }{
101      \IfStrEq\manifest@key{ns}{
102          \xdef\manifest@mf@ns{\manifest@line}
103      }{
104      % dependencies
105      \IfStrEq\manifest@key{dependencies}{
106          \xdef\manifest@mf@deps{\manifest@line}
107      }{
108
109      }}}}}
110      \parse@manifest@loop
111   \fi
112 }
```

and finally, we find path of main file

```
113 \begingroup
114    \edef\oldpercentcatcode{\the\catcode`\%}
115    \catcode`\%=12
```

```
116  \def\percent{%}
117  \catcode'\%=\oldpercentcatcode
118  \ExplSyntaxOn
119    \edef\windowsstring{\detokenize{windows}}
120    \edef\os_string{\expandafter\detokenize\expandafter{\c_sys_platform_str}}
121    \ifx\os_string\windowsstring
122      \edef\cmd_string{kpsewhich ~ -expand-var ~ \percent CD\percent}
123    \else
124      \edef\cmd_string{kpsewhich ~ -var-value ~ PWD}
125    \fi
126 \expandafter\sys_get_shell:nnN\expandafter{\cmd_string} { } \mainfile_dir
127      \tl_trim_spaces:N \mainfile_dir
128 \xdef\mainfile_dir{\expandafter\detokenize\expandafter{\mainfile_dir}}
129  \ExplSyntaxOff
130 \endgroup
131 \expandafter\let\expandafter\mathhub@maindir\csname mainfile_dir\endcsname
```

We read in and parse `MANIFEST.MF`

```
132 \findmanifest{\mathhub@maindir}
133 \begingroup
134   \gdef\manifest@mf@id{}
135   \gdef\manifest@mf@narr{}
136   \gdef\manifest@mf@ns{}
137   \gdef\manifest@mf@deps{}
138   \newread\@manifest
139   \openin\@manifest\manifest@mf
140   \parse@manifest@loop
141   \closein\@manifest
142 \endgroup
```

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

```
143 \mhcurrentrepos{\manifest@mf@id}
144 ⟨/package⟩
```

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

```
145 ⟨*omdoc⟩
146 \ProvidesPackage{omdoc-mh}[2019/03/20 v1.1 MathHub support for OMDoc Documents]
147 \RequirePackage{mathhub}
```

\addmhbibresource

```
148 \newcommand\addmhbibresource[2][]{%
149   \def\@repos{#1}%
150   \ifx\@repos\@empty%
151   \addbibresource{\MathHub{\mh@currentrepos/#2}}%
152   \else
153   \addbibresource{\MathHub{\@repos/#2}}%
154   \fi%
155   \ignorespacesandpars}%
156 ⟨/omdoc⟩
```

11

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

157 ⟨∗modules⟩
158 \ProvidesPackage{modules-mh}[2019/03/20 v1.1 MathHub support for the sTeX modules package]
159 \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 compar-
ison with an \expandafter, since the values may be passed on from other key
bindings. Parameters will be passed to \importmodule.

160 \srefaddidkey{importmhmodule}%
161 \addmetakey{importmhmodule}{mhrepos}%
162 \addmetakey{importmhmodule}{path}%
163 \addmetakey{importmhmodule}{dir}%
164 \addmetakey[sms]{importmhmodule}{ext}%
165 \addmetakey[false]{importmhmodule}{conservative}[true]%
166 \newcommand\importmhmodule[2][]{%
167 \metasetkeys{importmhmodule}{#1}%
168 \ifx\importmhmodule@dir\@empty%
169 \edef\@path{\importmhmodule@path}%
170 \else\edef\@path{\importmhmodule@dir/#2}\fi%
171 \ifx\@path\@empty% if module name is not set
172 \importmodule[id=\importmhmodule@id]{#2}%
173 \else%
174 \edef\mh@@repos{\mh@currentrepos}% remember so that we can reset it.
175 \ifx\importmhmodule@mhrepos\@empty% if in the same repos
176 \relax% no need to change mh@currentrepos, i.e, current directory.
177 \else%
178 \mhcurrentrepos{\importmhmodule@mhrepos}% change it.
179 \fi%
180 \importmodule[load=\MathHub{\mh@currentrepos/source/\@path},
181   ext=\importmhmodule@ext,id=\importmhmodule@id]{#2}%
182 \mhcurrentrepos{\mh@@repos}% after importing, reset to old value
183 \fi%
184 \ignorespacesandpars}

and now the analogs

\usemhmodule

185 \newcommand\usemhmodule[2][]{%
186 \metasetkeys{importmhmodule}{#1}%
187 \ifx\importmhmodule@dir\@empty%
188 \edef\@path{\importmhmodule@path}%
189 \else\edef\@path{\importmhmodule@dir/#2}\fi%
190 \ifx\@path\@empty%
191 \usemodule[id=\importmhmodule@id]{#2}%
192 \else%
193 \edef\mh@@repos{\mh@currentrepos}%

12

```
194 \ifx\importmhmodule@mhrepos\@empty%
195 \else\mhcurrentrepos{\importmhmodule@mhrepos}\fi%
196 \usemodule[load=\MathHub{\mh@currentrepos/source/\@path},
197                         ext=\importmhmodule@ext,id=\importmhmodule@id]{#2}%
198 \mhcurrentrepos\mh@@repos%
199 \fi%
200 \ignorespacesandpars}
```

\mhinputref

```
201 \newcommand\mhinputref[2][]{%
202 \def\@repos{#1}%
203 \edef\mh@@repos{\mh@currentrepos}%
204 \ifx\@repos\@empty\else\mhcurrentrepos{#1}\fi%
205 \inputref{\MathHub{\mh@currentrepos/source/#2}}%
206 \mhcurrentrepos\mh@@repos%
207 \ignorespacesandpars}
```

\mhinput

```
208 \let\mhinput\mhinputref%
209 ⟨/modules⟩
```

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

```
210 ⟨∗omtext⟩
211 \ProvidesPackage{omtext-mh}[2019/03/20 v1.1 MathHub support for the sTeX omtext package]
212 \RequirePackage{mathhub}
```

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

```
213 \def\Gin@mhrepos{}
214 \define@key{Gin}{mhrepos}{\def\Gin@mhrepos{#1}}
215 \newcommand\mhgraphics[2][]{\setkeys{Gin}{#1}%
216 \edef\mh@@repos{\mh@currentrepos}%
217 \ifx\Gin@mhrepos\@empty\includegraphics[#1]{\MathHub{\mh@currentrepos/source/#2}}%
218 \else\includegraphics[#1]{\MathHub{\Gin@mhrepos/source/#2}}\fi
219 \def\Gin@mhrepos{}\mhcurrentrepos\mh@@repos}
220 \newcommand\cmhgraphics[2][]{\begin{center}\mhgraphics[#1]{#2}\end{center}}
```

The following macros are deprecated.

```
221 \newcommand\mhcgraphics[2][]{\begin{center}\mhgraphics[#1]{#2}\end{center}
222   \PackageWarning{omtext-mh}{\protect\mhcgraphics\space is deprecated, use \protect\cmhgraphics
223 \newcommand\mhbgraphics[2][]{\fbox{\mhgraphics[#1]{#2}}
224   \PackageWarning{omtext-mh}{\protect\mhbgraphics\space is deprecated, use
225     \protect\mhgraphics\space and {center} instead}}
226 \newcommand\mhcbgraphics[2][]{\begin{center}\fbox{\mhgraphics[#1]{#2}}\end{center}
227   \PackageWarning{omtext-mh}{\protect\mhcbgraphics\space is deprecated, use
228     \protect\mhgraphics,\space {center}, and \protect\fbox\space instead}}
229 ⟨/omtext⟩
```

13

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

```
230 ⟨∗smultiling⟩
231 \ProvidesPackage{smultiling-mh}[2019/03/20 v1.1 MathHub support for the sTeX smultiling package]
232 \RequirePackage{mathhub}
```

mhmodsig

```
233 \newenvironment{mhmodsig}{\begin{modsig}}{\end{modsig}}
```

mhmodnl:*

```
234 \addmetakey{mhmodnl}{mhrepos}
235 \addmetakey{mhmodnl}{path}
236 \addmetakey*{mhmodnl}{title}
237 \addmetakey*{mhmodnl}{creators}
238 \addmetakey*{mhmodnl}{contributors}
239 \addmetakey{mhmodnl}{srccite}
240 \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.

```
241 \newenvironment{mhmodnl}[3][]{\metasetkeys{mhmodnl}{#1}\def\@test{#1}%
242 \edef\@repos{\ifx\mhmodnl@mhrepos\@empty\mh@currentrepos\else\mhmodnl@mhrepos\fi}%
243 \edef\@load{\MathHub{\@repos/source/\ifx\mhmodnl@path\@empty #2\else\mhmodnl@path\fi}}%
244 \ifx\@test\@empty\begin{modnl}[load=\@load]{#2}{#3}\else\begin{modnl}[load=\@load,#1]{#2}{#3}\fi
245 \ignorespacesandpars}
246 {\end{modnl}\ignorespacesandparsafterend}
```

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

```
247 \newenvironment{mhviewsig}[4][]{% keys, id, from, to
248 \def\@test{#1}\ifx\@test\@empty%
249 \begin{mhview}[id=#2,ext=tex]{#3}{#4}\else%
250 \begin{mhview}[id=#2,#1,ext=tex]{#3}{#4}\fi%
251 \ignorespacesandpars}
252 {\end{mhview}\ignorespacesandparsafterend}
```

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

```
253 \newenvironment{mhviewnl}[5][]{% keys, id, lang, from, to
254 \def\@test{#1}\ifx\@test\@empty%
255 \begin{mhview}[id=#2.#3,ext=tex]{#4}{#5}\else%
256 \begin{mhview}[id=#2.#3,#1,ext=tex]{#4}{#5}\fi%
257 \ignorespacesandpars}
258 {\end{mhview}\ignorespacesandparsafterend}
259 ⟨/smultiling⟩
```

---

[4]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

```
260 ⟨*structview⟩
261 \ProvidesPackage{structview-mh}[2019/03/20 v1.1 MathHub support for the sTeX structview package
262 \RequirePackage{mathhub}
```

mhstructure

```
263 \newenvironment{mhstructure}[3][]{%
264   \gdef\@@doit{\importmhmodule[#1]{#3}}%
265   \ifmod@show\par\noindent structure import "#2" from module #3 \@@doit\fi%
266   \ignorespacesandpars}
267 {\aftergroup\@@doit\ifmod@show end import\fi%
268   \ignorespacesandparsafterend}
```

importmhmodulevia   this is now deprecated, we give an error

```
269 \newenvironment{importmhmodulevia}[2][]%
270 {\PackageError{structview-mh}%
271   {The {importmhmodulevia} environment is deprecated}{use the {mhstructure} instead!}%
272   \begin{mhstructure}[#1]{missing}{#2}}
273 {\end{mhstructure}}

274 \srefaddidkey{mhview}
275 \addmetakey{mhview}{display}
276 \addmetakey{mhview}{creators}
277 \addmetakey{mhview}{contributors}
278 \addmetakey{mhview}{srccite}
279 \addmetakey*{mhview}{title}
280 \addmetakey{mhview}{type}
281 \addmetakey{mhview}{fromrepos}
282 \addmetakey{mhview}{torepos}
283 \addmetakey{mhview}{frompath}
284 \addmetakey{mhview}{topath}
285 \addmetakey[sms]{mhview}{ext}
```

mhview   the MathHub version

```
286 \newenvironment{mhview}[3][]% keys, from, to
287 {\metasetkeys{mhview}{#1}%
288   \sref@target%
289   \begin{@mhview}{#2}{#3}%
290   \view@heading{#2}{#3}{\mhview@display}{\mhview@title}%
291   \ignorespacesandpars}
292 {\end{@mhview}\ignorespacesandparsafterend}
293 \ifmod@show\surroundwithmdframed{mhview}\fi
```

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

```
294 \newenvironment{@mhview}[2]{%from, to
295   \importmhmodule[mhrepos=\mhview@fromrepos,path=\mhview@frompath,ext=\mhview@ext]{#1}%
296   \importmhmodule[mhrepos=\mhview@torepos,path=\mhview@topath,ext=\mhview@ext]{#2}%
297 }{}%
```

15

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

298 `\newenvironment{mhviewsketch}[3][]%`
299 `{\PackageError{structview}%`
300 `  {The {mhviewsketch} environment is deprecated}{use the {mhview} instead!}%`
301 `  \begin{mhview}[#1]{#2}{#3}}`
302 `{\end{mhview}}`

mhinlineView   Analogous modification to `inlineView`

303 `\newenvironment{mhinlineView}[2][]% keys, source`
304 `{\metasetkeys{mhview}{#1}\sref@target%`
305 `  \importmhmodule[mhrepos=\mhview@fromrepos,path=\mhview@frompath,ext=\mhview@ext]{#2}%`
306 `  \ignorespacesandpars}`
307 `{\ignorespacesandpars}`

mhinlineview

308 `\newcommand\mhinlineview[3][]{\begin{mhinlineView}[#1]{#2}{\mod@id}#3\end{mhinlineView}}`
309 ⟨/structview⟩

4.7   mikoslides-mh.sty: Support for MiKo Slides

310 ⟨∗mikoslides⟩
311 `\ProvidesPackage{mikoslides-mh}[2019/03/20 v1.1 MathHub support for the sTeX mikoslides package`
312 `\RequirePackage{mathhub}`

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

313 `\def\Gin@mhrepos{}`
314 `\define@key{Gin}{mhrepos}{\def\Gin@mhrepos{#1}}`
315 `\newcommand\mhframeimage[2][]{%`
316 `  \setkeys{Gin}{#1}%`
317 `  \edef\mh@@repos{\mh@currentrepos}%`
318 `  \ifx\Gin@mhrepos\@empty%`
319 `    \frameimage[#1]{\MathHub{\mh@currentrepos/source/#2}}%`
320 `  \else%`
321 `    \frameimage[#1]{\MathHub{\Gin@mhrepos/source/#2}}%`
322 `  \fi%`
323 `}%`

\mhinputref*

324 `\let\orig@mhinputref\mhinputref`
325 `\def\mhinputref{\@ifstar\nmhinputref\orig@mhinputref}`
326 `\newcommand\nmhinputref[2][]{\ifnotes\orig@mhinputref[#1]{#2}\fi}`
327 ⟨/mikoslides⟩

4.8   problem-mh.sty: Support for Problems

328 ⟨∗problem⟩
329 `\ProvidesPackage{problem-mh}[2019/03/20 v1.1 MathHub support for the sTeX problem package]`
330 `\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`.

```
331 \addmetakey{inclprob}{mhrepos}
332 \newcommand\includemhproblem[2][]{\metasetkeys{inclprob}{#1}%
333 \edef\mh@@repos{\mh@currentrepos}%
334 \ifx\inclprob@mhrepos\@empty\else\mhcurrentrepos\inclprob@mhrepos\fi%
335 \input{\MathHub{\mh@currentrepos/source/#2}}%
336 \mhcurrentrepos\mh@@repos\clear@inclprob@keys}
337 ⟨/problem⟩
```

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

```
338 ⟨∗hwexam⟩
339 \ProvidesPackage{hwexam-mh}[2019/03/20 v1.1 MathHub support for the sTeX hwexam package]
340 \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`.

```
341 \newcommand\inputmhassignment[2][]{\metasetkeys{inclassig}{#1}%
342 \edef\mh@@repos{\mh@currentrepos}%
343 \ifx\inclassig@mhrepos\@empty\else\mhcurrentrepos\inclassig@mhrepos\fi%
344 \inputassignment[#1]{\MathHub{\mh@currentrepos/source/#2}}%
345 \mhcurrentrepos\mh@@repos\clear@inclassig@keys}
346 \newcommand\includemhassignment[2][]{\newpage\inputmhassignment[#1]{#2}}
347 ⟨/hwexam⟩
```

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

```
348 ⟨∗tikzinput⟩
349 \ProvidesPackage{tikzinput-mh}[2019/03/20 v1.1 MathHub support for the sTeX tikzinput package]
350 \RequirePackage{mathhub}
351 \RequirePackage{pathsuris}

352 \define@key{Gin}{mhrepos}{\def\Gin@mhrepos{#1}}
353 \newcommand\mhtikzinput[2][]{\def\Gin@mhrepos{}\setkeys{Gin}{#1}%
354 \edef\mh@@repos{\mh@currentrepos}%
355 \ifx\Gin@mhrepos\@empty\tikzinput[#1]{\MathHub{\mh@currentrepos/source/#2}}%
356 \else\mhcurrentrepos\Gin@mhrepos\tikzinput[#1]{\MathHub{\Gin@mhrepos/source/#2}}\fi
357 \def\Gin@mhrepos{}\mhcurrentrepos\mh@@repos}
358 \newcommand\cmhtikzinput[2][]{\begin{center}\mhtikzinput[#1]{#2}\end{center}}
359 ⟨/tikzinput⟩
```

4.11 **lstmh.sty**: Support for Listings

```
360 ⟨∗lst⟩
361 \ProvidesPackage{lstmh}[2019/03/20 v1.1 MathHub support for the listings package]
```

```
362 \RequirePackage{mathhub}
363 \RequirePackage{pathsuris}
364 \RequirePackage{listings}

365 \define@key{lst}{mhrepos}{\def\lst@mhrepos{#1}}
366 \newcommand\lstinputmhlisting[2][]{\def\lst@mhrepos{}\setkeys{lst}{#1}%
367 \edef\mh@@repos{\mh@currentrepos}%
368 \ifx\lst@mhrepos\@empty\lstinputlisting[#1]{\MathHub{\mh@currentrepos/source/#2}}%
369 \else\lstinputlisting[#1]{\MathHub{\lst@mhrepos/source/#2}}\fi
370 \def\lst@mhrepos{}\mhcurrentrepos\mh@@repos}
371 \newcommand\clstinputmhlisting[2][]{\begin{center}\lstinputmhlisting[#1]{#2}\end{center}}
372 ⟨/lst⟩
```

Change History

References

[GH]   *GitHub: Build software better, together*. URL: `http://github.com` (visited on 02/24/2014).

[GL]   *The first single application for the entire DevOps lifecycle – GitLab*. URL: `http://gitlab.com` (visited on 01/12/2019).

[Hor+11] Fulya Horozal et al. "Combining Source, Content, Presentation, Narration, and Relational Representation". In: *Intelligent Computer Mathematics*. Ed. by James Davenport et al. LNAI 6824. Springer Verlag, 2011, pp. 212–227. ISBN: 978-3-642-22672-4. URL: `http://kwarc.info/frabe/Research/HIJKR_dimensions_11.pdf`.

[KGA18]   Michael Kohlhase, Deyan Ginev, and Rares Ambrus. *modules.sty: Semantic Macros and Module Scoping in sTeX*. Tech. rep. 2018. URL: `https://github.com/sLaTeX/sTeX/raw/master/sty/pathsuris/pathsuris.pdf`.

[Koh18]   Michael Kohlhase. *metakeys.sty: A generic framework for extensible Metadata in LaTeX*. Tech. rep. 2018. URL: `https://github.com/sLaTeX/sTeX/raw/master/sty/metakeys/metakeys.pdf`.

[lmh]   *lmh: A cross-repository administration tool for the local authoring in MathHub.info*. URL: `https://github.com/KWARC/localmh` (visited on 02/05/2017).

[sTeX]   *sTeX: A semantic Extension of TeX/LaTeX*. URL: `https://github.com/sLaTeX/sTeX` (visited on 05/15/2015).