MathHub Support for STEX*

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

June 19, 2021

Abstract

The mathhub package collection is part of the STEX collection, a version of TEX/IATEX that allows to markup TEX/IATEX documents semantically without leaving the document format, essentially turning TEX/IATEX 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

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

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

4	Implementation		8	
	4.1	mathhub.sty: General Infrastructure	8	
	4.2	omdocmh.sty: MH Document Infrastructure	12	
	4.3	modules-mh.sty: MH Variants for Modules	12	
	4.4	omtext-mh.sty: MH Variants for OMText	14	
	4.5	smultiling-mh.sty: MH Variants for Multilinguality	15	
	4.6	structview-mh.sty: MH Variants for Structures and Views	16	
	4.7	mikoslides-mh.sty: Support for MiKo Slides	17	
	4.8	problem-mh.sty: Support for Problems	18	
	4.9	hwexam-mh.sty: Support for Assignments	19	
	4.10	tikzinput-mh.sty: Support for Assignments	19	
	4.11	lstmh sty: Support for Listings	19	

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 \leadsto 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 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 $\langle group \rangle / \langle arch \rangle$ installs the installs the MathHub archive $\langle group \rangle / \langle arch \rangle$ together with all of its dependencies and mmt make pdflatex $\langle file \rangle$ generates PDF for the file $\langle file \rangle$ (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 $\langle group \rangle / \langle arch \rangle$, where $\langle group \rangle$ is a MathHub-unique repository group and $\langle arch \rangle$ a MathHub archive name that is $\langle group \rangle$ -unique.

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

- the STFX 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 \(\lambda e y \rangle : \lambda values \rangle . \)

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 $\langle group \rangle / \langle arch \rangle$. 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 $\langle pack \rangle$.sty there is a **mh-variant** $\langle pack \rangle$ -**mh.sty**, and $\langle pack \rangle$ -sty takes the option **mh**. When that is given (by calling \usepackage[mh] { $\langle pack \rangle$ }), then $\langle pack \rangle$ -sty inputs $\langle pack \rangle$ -mh.sty from the mathhub bundle, which augments the $\langle pack \rangle$ package with MathHub functionality.

2 The User Interface

We now document mh-variants of the SIEX 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 macro supports this practice: \libinput {\langle filename \rangle} \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

\libinput{preamble}

\libusepackage

The \libusepackage is analogous it allows to share IATEX package between Math-Hub 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.

\mhcurrentrepos

Of course, LATEX does not 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 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 \mhinputref \mhinput The \usemhmodule is the analog to \usemodule.

For this, the modules package supplies the mh-variants \mhinputref and \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

mhmodsig mhmodnl 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

1

2.7 mikoslides-mh.sty: Support for MiKo Slides

\mhframeimage

EdN:1

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 **Istmh.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).

4.1 mathhub.sty: General Infrastructure

\mhcurrentrepos

\libusepackage

\mhcurrentrepos is used to initialize the current repository.

7 \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.

```
8 \def\modules@@first#1/#2;{#1}
  9 \newcommand\libinput[1]{%
10 \ifcsvoid{mh@currentrepos}{%
            \PackageError{mathhub}{current MathHub repository not found}{}}%
13 \edef\@mh@group{\expandafter\modules@@first\mh@currentrepos;}%
14 \let\orig@inffile\mh@inffile\let\orig@libfile\mh@libfile%
15 \def\mh@inffile{\MathHub{\@mh@group/meta-inf/lib/#1}}%
16 \end{area} $$16 \end{area
17 \if@iswindows@%
            \path@to@windows\mh@inffile%
            \path@to@windows\mh@libfile%
19
21 \IfFileExists\mh@inffile{\input\mh@inffile}{}%
22 \IfFileExists\mh@inffile{}{\IfFileExists\mh@libfile{}{\%
            {\PackageError{mathhub}%
23
24
                   {Library file missing; cannot input #1.tex\MessageBreak%
25
                  Both \mh@libfile.tex\MessageBreak and \mh@inffile.tex\MessageBreak%
                  do not exist}%
         {Check whether the file name is correct}}}}%
28 \IfFileExists\mh@libfile{\input\mh@libfile\relax}{}%
29 \let\mh@inffile\orig@inffile\let\mh@libfile\orig@libfile}
the \libusepackage is analogous to \libinput
30 \newcommand\libusepackage[2][]{%
31 \edef\@mh@group{\expandafter\modules@@first\mh@currentrepos;}%
```

32 \let\orig@inffile\mh@inffile\let\orig@libfile\mh@libfile% 33 \edef\mh@inffile{\MathHub{\@mh@group/meta-inf/lib/#2}}%

```
34 \edgh{\mh@libfile{\mathbb{\mh@currentrepos/lib/#2}}}\%
35 \if@iswindows@%
    \path@to@windows\mh@inffile%
    \path@to@windows\mh@libfile%
37
38 \fi%
39 \IfFileExists{\mh@inffile.sty}{\usepackage[#1]{\mh@inffile}}{}%
40 \IfFileExists {\mh@inffile.sty}{}{\IfFileExists{\mh@libfile.sty}{}{\%
    {\PackageError{mathhub}%
      {Library file missing; cannot use package #2.sty\MessageBreak%
42
      Both \mh@libfile.sty\MessageBreak and \mh@inffile.sty\MessageBreak%
43
44
      do not exist}%
    {Check whether the file name is correct}}}}%
46 \IfFileExists\mh@libfile{\input\mh@libfile\relax}{}%
47 \let\mh@inffile\orig@inffile\let\mh@libfile\orig@libfile}
```

Generally, the T_EX formatter pdflatex needs to know the file system paths of the referenced STFX files: usually long relative paths. The pathsuris package [KGA20] from the STFX 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.

```
48 \ifcsvoid{mathhubpath}{
49 \ifcsvoid{mathhub@path}{
    \kpsewhich\mathhub@path{--var-value MATHHUB}
51 }{}
52 }{\let\mathhub@path\mathhubpath}
53 \if@iswindows@\windows@to@path\mathhub@path\fi
54 \ifx\mathhub@path\@empty%
55 \PackageError{mathhub}
    {MATHHUB system variable not found or wrongly set}
    {use export MATHHUB="<path>", where <path> points your MathHub directory}
58 \else\defpath{MathHub}{\mathhub@path}\fi
59 \ifcsvoid{mathhub@path}{}\message{^^JMATHHUB: \mathhub@path}}
findmanifest{\langle path \rangle} searches for a file MANIFEST.MF up and over \langle path \rangle in the
```

\findmanifest

file system tree.

```
61 \def\findmanifest#1{%
    \@cpath{#1}%
62
    \ifx\@CanPath\@Slash%
63
      \def\manifest@mf{}%
64
    \else\ifx\@CanPath\@empty%
65
        \def\manifest@mf{}%
66
    \else%
67
      \edef\@findmanifest@path{\@CanPath/MANIFEST.MF}%
68
      \if@iswindows@\path@to@windows\@findmanifest@path\fi%
69
      %\message{^^JHere: \@findmanifest@path^^J}
```

```
\IfFileExists{\@findmanifest@path}{%
71
72
        %\message{MANIFEST.MF found at \@findmanifest@path}
        \edef\manifest@mf{\@findmanifest@path}%
73
        \xdef\temp@archive@dir{\expandafter\detokenize\expandafter{\@CanPath}}%
74
75
       }{%
76
       \edef\@findmanifest@path{\@CanPath/META-INF/MANIFEST.MF}%
77
       \if@iswindows@\path@to@windows\@findmanifest@path\fi%
       \IfFileExists{\@findmanifest@path}{%
78
        %\message{MANIFEST.MF found at \@findmanifest@path}
79
        \edef\manifest@mf{\@findmanifest@path}%
80
        81
 82
       \edef\@findmanifest@path{\@CanPath/meta-inf/MANIFEST.MF}%
83
       \if@iswindows@\path@to@windows\@findmanifest@path\fi%
84
       \IfFileExists{\@findmanifest@path}{%
85
        %\message{MANIFEST.MF found at \@findmanifest@path}
86
        \edef\manifest@mf{\@findmanifest@path}%
87
        88
89
90
         \findmanifest{\@CanPath/..}%
91
      }}}%
     \fi\fi%
92
93 }
the next macro is a helper function for parsing MANIFEST.MF
94 \def\split@manifest@key{%
     \IfSubStr{\manifest@line}{\@Colon}{%
96
        \StrBefore{\manifest@line}{\@Colon}[\manifest@key]%
        \StrBehind{\manifest@line}{\@Colon}[\manifest@line]%
97
        \trimstring\manifest@line%
98
        \trimstring\manifest@key%
99
100
    }{%
         \def\manifest@key{}%
101
    }%
102
103 }
    the next helper function iterates over lines in MANIFEST.MF
104 \def\parse@manifest@loop{%
    \ifeof\@manifest%
105
     \else%
106
       \read\@manifest to \manifest@line\relax%
107
       \edef\manifest@line{\expandafter\detokenize\expandafter{\manifest@line}}%
108
       \split@manifest@key%
109
110
       \IfStrEq\manifest@key{\detokenize{id}}{%
111
          %\message{archive id: \manifest@line}
112
          \xdef\manifest@mf@id{\manifest@line}%
113
       }{%
114
115
      % narration-base
116
       \IfStrEq\manifest@key{\detokenize{narration-base}}{%
```

```
%\message{archive narration-base: \manifest@line}
                 117
                             \xdef\manifest@mf@narr{\manifest@line}%
                 118
                         }{%
                 119
                         % namespace
                 120
                         \IfStrEq\manifest@key{\detokenize{source-base}}{%
                 121
                 122
                            % \message{archive source-base: \manifest@line}
                 123
                             \xdef\manifest@mf@ns{\manifest@line}%
                         }{%
                 124
                         \IfStrEq\manifest@key{\detokenize{ns}}{%
                 125
                             %\message{archive ns: \manifest@line}
                 126
                             \xdef\manifest@mf@ns{\manifest@line}%
                 127
                         }{%
                 128
                 129
                         % dependencies
                         \IfStrEq\manifest@key{\detokenize{dependencies}}{%
                 130
                             %\message{archive dependencies: \manifest@line}
                 131
                             \xdef\manifest@mf@deps{\manifest@line}%
                 132
                         }{%
                 133
                         }}}}%
                 134
                 135
                         \parse@manifest@loop%
                 136
                       \fi%
                 137 }
                  \operatorname{parsemanifest}(\operatorname{macroname}) \{ (\operatorname{path}) \}  finds MANIFEST.MF via \operatorname{findmanifest}(\operatorname{path}) \},
\parsemanifest
                  and parses the file, storing the individual fields (id, narr, ns and dependencies)
                  in \langle macroname \rangleid, \langle macroname \ranglenarr, etc.
                 138 \newread\@manifest
                 139 \def\parsemanifest#1#2{%
                       \gdef\temp@archive@dir{}%
                       \findmanifest{#2}%
                 141
                       \begingroup%
                 142
                         \newlinechar=-1%
                 143
                         \endlinechar=-1%
                 144
                         \gdef\manifest@mf@id{}%
                 145
                         \gdef\manifest@mf@narr{}%
                 146
                         \gdef\manifest@mf@ns{}%
                 147
                         \gdef\manifest@mf@deps{}%
                 148
                         \openin\@manifest\manifest@mf\relax%
                 149
                         \parse@manifest@loop%
                 150
                         \closein\@manifest%
                 151
                       \endgroup%
                 152
                 153
                       \if@iswindows@\windows@to@path\manifest@mf\fi%
                       \cslet{#1id}\manifest@mf@id%
                 154
                       \cslet{#1narr}\manifest@mf@narr%
                 155
                       \cslet{#1ns}\manifest@mf@ns%
                 156
                       \cslet{#1deps}\manifest@mf@deps%
                 157
                 158
                       \cslet{#1dir}\temp@archive@dir%
                 159 }
```

\setcurrentreposinfo \setcurrentreposinfo{ $\langle id \rangle$ } sets the current repository to $\langle id \rangle$, checks if the MANIFEST.MF of this repository has already been read, and if not, find it, parses

```
it and stores the values in \colon currentrepos @\langle key \rangle @\langle id \rangle  for later retrieval.
                  160 \def\setcurrentreposinfo#1{%
                        \ifcsdef{currentrepos@dir@#1}{%
                  161
                          \mhcurrentrepos{#1}%
                  162
                  163
                        }{%
                          \parsemanifest{mathhub@archive@}{\MathHub{#1}}%
                  164
                  165
                          \@setcurrentreposinfo%
                  166
                        \ifcsvoid{mathhub@archive@dir}{\PackageError{mathhub}{No archive with %
                  167
                          name #1 found!}{make sure that #1 is directly in your MATHHUB folder %
                  168
                          and contains a MANIFEST.MF, either directly in #1 or in a meta-inf %
                  169
                          subfolder.}}{}%
                  170
                  171 }
                  172
                  173 \def\@setcurrentreposinfo{%
                        \mhcurrentrepos{\mathhub@archive@id}%
                  174
                        \csxdef{currentrepos@dir@\mathhub@archive@id}{\mathhub@archive@dir}\
                  175
                  176
                        \csxdef{currentrepos@narr@\mathhub@archive@id}{\mathhub@archive@narr}\%
                  177
                        \csxdef{currentrepos@ns@\mathhub@archive@id}{\mathhub@archive@ns}%
                        \csxdef{currentrepos@deps@\mathhub@archive@id}{\mathhub@archive@deps}\%
                  178
                  179 }
                   Finally – and that is the ultimate goal of all of the above, we set the current repos.
                  180 \parsemanifest{mathhub@archive@}\stex@maindir
                  181 \@setcurrentreposinfo
                  182 \message{^^J Current MathHub repository: \ifcsvoid{mh@currentrepos}{None}\mh@currentrepos}
                  183 (/package)
                          omdoc--mh.sty: MH Document Infrastructure
                   4.2
                  184 (*omdoc)
                  185 \ProvidesPackage{omdoc-mh}[2019/03/20 v1.1 MathHub support for OMDoc Documents]
                  186 \RequirePackage{mathhub}
\addmhbibresource
                  187 \newcommand\addmhbibresource[2][]{%
                  188
                        \def\@repos{#1}%
                        \ifx\@repos\@empty%
                  189
                        \addbibresource{\MathHub{\mh@currentrepos/#2}}%
                  190
                  191
                        \addbibresource{\MathHub{\@repos/#2}}%
                  192
                        \fi%
                       \ignorespacesandpars}%
                  195 (/omdoc)
                          modules-mh.sty: MH Variants for Modules
                   4.3
                  196 (*modules)
```

197 \ProvidesPackage{modules-mh}[2019/03/20 v1.1 MathHub support for the sTeX modules package]

198 \RequirePackage{mathhub}

235 \else%

The ∞ importmendule [$\langle key = value | list \rangle$] {module} saves the current value of \importmhmodule \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@crepos. 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. 199 \srefaddidkey{importmhmodule}% 200 \addmetakey{importmhmodule}{mhrepos}% 201 \addmetakey{importmhmodule}{path}% 202 \addmetakey{importmhmodule}{ext}% why does this exist? 203 \addmetakey{importmhmodule}{dir}% 204 \addmetakey[false]{importmhmodule}{conservative}[true]% 205 \newcommand\importmhmodule[2][]{% \usemodule@maybesetcodes 206 \metasetkeys{importmhmodule}{#1}% 207 208 \ifx\importmhmodule@dir\@empty% 209 \edef\@path{\importmhmodule@path}% \else\edef\@path{\importmhmodule@dir/#2}\fi% 210 \ifx\@path\@empty% if module name is not set 211 \importmodule[id=\importmhmodule@id]{#2}% 212 \else% 213 \edef\mh@@repos{\mh@currentrepos}% remember so that we can reset it. 214 215 \ifx\importmhmodule@mhrepos\@empty% if in the same repos 216\relax% no need to change mh@currentrepos, i.e, current directory. 217 218 \setcurrentreposinfo{\importmhmodule@mhrepos}% change it. 219 \addto@thismodulex{\noexpand\setcurrentreposinfo{\importmhmodule@mhrepos}}} 220 \importmodule[load=\MathHub{\mh@currentrepos/source/\@path}, 221 222 id=\importmhmodule@id]{#2}% \setcurrentreposinfo{\mh@@repos}% after importing, reset to old value 223 \addto@thismodulex{\noexpand\setcurrentreposinfo{\mh@@repos}}% 224 225 \fi% \ignorespacesandpars% 226 227 } and now the analogs \usemhmodule 228 \newcommand\usemhmodule[2][]{% 229 \metasetkeys{importmhmodule}{#1}% 230 \ifx\importmhmodule@dir\@empty% \edef\@path{\importmhmodule@path}% 232 \else\edef\@path{\importmhmodule@dir/#2}\fi% 233 \ifx\@path\@empty% \usemodule[id=\importmhmodule@id]{#2}% 234

\edef\mh@currentrepos}%

```
\ifx\importmhmodule@mhrepos\@empty%
             237
                  \else\setcurrentreposinfo{\importmhmodule@mhrepos}\fi%
             238
                  \usemodule[load=\MathHub{\mh@currentrepos/source/\@path},%
             239
                                        id=\importmhmodule@id]{#2}%
             240
                  \setcurrentreposinfo\mh@@repos%
             241
             242 \fi%
             243 \ignorespacesandpars}
\mhinputref
             244 \newcommand\mhinputref[2][]{%
             245 \def\@repos{#1}%
             246 \edef\mh@currentrepos}%
             247 \ifx\@repos\@empty\else\setcurrentreposinfo{#1}\fi%
             248 \inputref{\MathHub{\mh@currentrepos/source/#2}}%
             249 \setcurrentreposinfo\mh@@repos%
             250 \ignorespacesandpars}
   \mhinput
             251 \newcommand\mhinput[2][]{%
             252 \def\@repos{#1}%
             253 \edef\mh@currentrepos}%
             254 \ifx\@repos\@empty\else\setcurrentreposinfo{#1}\fi%
             255 \input{\MathHub{\mh@currentrepos/source/#2}}%
             256 \setcurrentreposinfo\mh@@repos%
             257 \ignorespacesandpars}
             258 (/modules)
                    omtext-mh.sty: MH Variants for OMText
              4.4
             259 (*omtext)
             260 \ProvidesPackage{omtext-mh}[2019/03/20 v1.1 MathHub support for the sTeX omtext package]
             261 \RequirePackage{mathhub}
             Use the current value of \mh@currentrepos or the value of the mhrepos key if it
\*mhgraphics
              is given in \my*graphics.
             262 \def\Gin@mhrepos{}
             263 \define@key{Gin}{mhrepos}{\def\Gin@mhrepos{#1}}
             264 \newcommand\mhgraphics[2][]{\setkeys{Gin}{\#1}%
             265 \edef\mh@currentrepos}%
             266 \ifx\Gin@mhrepos\@empty\edef\temp@path{\MathHub{\mh@currentrepos/source/#2}}%\includegraphics[#
             267 \else\edef\temp@path{\MathHub{\Gin@mhrepos/source/#2}}\fi%\includegraphics[#1]{\MathHub{\Gin@mh
             268 \if@iswindows@\path@to@windows\temp@path\fi
             269 \includegraphics [#1] {\temp@path}
             270 \def\Gin@mhrepos{}\setcurrentreposinfo\mh@@repos}
             271 \newcommand\cmhgraphics[2][]{\begin{center}\mhgraphics[#1]{#2}\end{center}}
                 The following macros are deprecated.
             272 \newcommand\mhcgraphics[2][]{\begin{center}\mhgraphics[#1]{#2}\end{center}
                  \PackageWarning{omtext-mh}{\protect\mhcgraphics\space is deprecated, use \protect\cmhgraphics
```

```
274 \newcommand\mhbgraphics[2][]{\fbox{\mhgraphics[#1]{#2}}
                \PackageWarning{omtext-mh}{\protect\mhbgraphics\space is deprecated, use
                  \protect\mhgraphics\space and {center} instead}}
          276
          277 \newcommand\mhcbgraphics[2][]{\begin{center}\fbox{\mhgraphics[#1]{#2}}\end{center}
                \PackageWarning{omtext-mh}{\protect\mhcbgraphics\space is deprecated, use
          279
                  \protect\mhgraphics,\space {center}, and \protect\fbox\space instead}}
          280 (/omtext)
           4.5
                  smultiling-mh.sty: MH Variants for Multilinguality
          281 (*smultiling)
          282 \ProvidesPackage{smultiling-mh}[2019/03/20 v1.1 MathHub support for the sTeX smultiling package
          283 \RequirePackage{mathhub}
 mhmodsig
          284 \newenvironment\{mhmodsig\}\{\begin\{modsig\}\}\{\newenvironment\{mhmodsig\}\}\}
mhmodnl:*
          285 \addmetakey{mhmodnl}{mhrepos}
          286 \texttt{\addmetakey{mhmodnl}{path}}
          287 \addmetakey*{mhmodnl}{title}
          288 \addmetakey*{mhmodnl}{creators}
          289 \addmetakey*{mhmodnl}{contributors}
          290 \addmetakey{mhmodnl}{srccite}
          291 \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.
          292 \newenvironment{mhmodnl}[3][]{\metasetkeys{mhmodnl}{#1}\def\@test{#1}%
          293 \edef\@repos{\ifx\mhmodnl@mhrepos\@empty\mh@currentrepos\else\mhmodnl@mhrepos\fi}%
          294 \edef\@load{\MathHub{\@repos/source/\ifx\mhmodnl@path\@empty #2\else\mhmodnl@path\fi}}%
          295 \left( \frac{295 \left( \frac{295 \left( \frac{2}{43} \right)}{15} \right)}{15} \right) \\
          296 \ignorespacesandpars}
          297 {\end{modnl}\ignorespacesandparsafterend}
mhviewsig The mhviewsig environment is just a layer over the mhview environment with the
           keys suitably adapted.
          298 \newenvironment{mhviewsig}[4][]{% keys, id, from, to
          299 \def\@test{#1}\ifx\@test\@empty%
          300 \begin{mhview}[id=#2]{#3}{#4}\else%
          301 \begin{mhview}[id=#2,#1]{#3}{#4}\fi%
          302 \ignorespacesandpars}
          303 {\end{mhview}\ignorespacesandparsafterend}
           The mhviewnl environment is just a layer over the mhview environment with the
           keys and language suitably adapted.<sup>2</sup>
```

²Ednote: MK: we have to do something about the if@langfiles situation here.

EdN:2

non-trivial, since we do not know the current path, to which we could append $.\langle lang \rangle !$

```
305 \def\@test{#1}\ifx\@test\@empty%
                  306 \ensuremath{\verb| begin{mhview}[id=\#2.\#3]{\#4}{\#5}\leq \ensuremath{|}}
                  307 \ensuremath{\mbox{begin}\{mhview\}[id=\#2.\#3,\#1]\{\#4\}\{\#5\}\fi\%}
                  308 \ignorespacesandpars}
                  309 {\end{mhview}\ignorespacesandparsafterend}
                  310 (/smultiling)
                          structview-mh.sty: MH Variants for Structures and
                          Views
                  311 (*structview)
                  312 \ProvidesPackage{structview-mh}[2019/03/20 v1.1 MathHub support for the sTeX structview package
                  313 \RequirePackage{mathhub}
      mhstructure
                  314 \newenvironment{mhstructure}[3][]{%
                       \gdef\@@doit{\importmhmodule[#1]{#3}}%
                       \ifmod@show\par\noindent structure import "#2" from module #3 \@@doit\fi%
                       \ignorespacesandpars}
                  317
                  318 {\aftergroup\@@doit\ifmod@show end import\fi%
                       \ignorespacesandparsafterend}
importmhmodulevia this is now deprecated, we give an error
                  320 \newenvironment{importmhmodulevia}[2][]%
                  321 {\PackageError{structview-mh}%
                       {The {importmhmodulevia} environment is deprecated}{use the {mhstructure} instead!}%
                       \begin{mhstructure}[#1]{missing}{#2}}
                  324 {\end{mhstructure}}
                  325 \srefaddidkey{mhview}
                  326 \addmetakey{mhview}{display}
                  327 \addmetakey{mhview}{creators}
                  328 \addmetakey{mhview}{contributors}
                  329 \addmetakey{mhview}{srccite}
                  330 \addmetakey*{mhview}{title}
                  331 \addmetakey{mhview}{type}
                  332 \addmetakey{mhview}{fromrepos}
                  333 \addmetakey{mhview}{torepos}
                  334 \addmetakey{mhview}{frompath}
                  335 \addmetakey{mhview}{topath}
           mhview the MathHub version
                  336 \newenvironment{mhview}[3][]% keys, from, to
                  337 {\metasetkeys{mhview}{\#1}%
                       \sref@target%
                  338
                  339
                       \begin{@mhview}{#2}{#3}%
                       340
                        \ignorespacesandpars}
                  342 {\end{@mhview}\ignorespacesandparsafterend}
```

304 \newenvironment{mhviewnl}[5][]{% keys, id, lang, from, to

```
343 \ifmod@show\if@latexml\else\surroundwithmdframed{mhview}\fi\fi
      Omhview The Omhview does the actual bookkeeping at the module level.
              344 \newenvironment{@mhview}[2]{%from, to
                   \usemhmodule[mhrepos=\mhview@fromrepos,path=\mhview@frompath]{#1}%
                   \usemhmodule[mhrepos=\mhview@torepos,path=\mhview@topath]{#2}%
              346
              347 }{}%
mhviewsketch The mhviewsketch environment is deprecated, we give an error
              348 \newenvironment{mhviewsketch}[3][]%
              349 {\PackageError{structview}%
              350 {The {mhviewsketch} environment is deprecated}{use the {mhview} instead!}%
                   \begin{mhview}[#1]{#2}{#3}}
              352 \{ \dnhview \} \}
mhinlineView Analogous modification to inlineView
              353 \newenvironment{mhinlineView}[2][]% keys, source
              354 {\metasetkeys{mhview}{#1}\sref@target%
                   \importmhmodule[mhrepos=\mhview@fromrepos,path=\mhview@frompath]{#2}%
                   \ignorespacesandpars}
              357 {\ignorespacesandpars}
mhinlineview
              358 \newcommand\mhinlineview[3][]{\begin{mhinlineView}} [#1]{#2}{\module@id}#3\end{mhinlineView}}
              359 (/structview)
                       mikoslides-mh.sty: Support for MiKo Slides
               4.7
              360 (*mikoslides)
              361 \ProvidesPackage{mikoslides-mh}[2020/12/06 v1.3 MathHub support for the sTeX mikoslides package{mikoslides-mh}]
              362 \RequirePackage{mathhub}
\mhframeimage Use the current value of \mh@currentrepos or the value of the mhrepos key if it
               is given in \frameimage.
              363 \def\Gin@mhrepos{}
              364 \define@key{Gin}{mhrepos}{\def\Gin@mhrepos{#1}}
              365 \newcommand\mhframeimage[2][]{%
                   \setkeys{Gin}{#1}%
                   \edef\mh@currentrepos}%
              367
                   \ifx\Gin@mhrepos\@empty%
              368
                     \edef\temp@path{\MathHub{\mh@currentrepos/source/#2}}%
              369
                   \else%
              370
                     \edef\temp@path{\MathHub{\Gin@mhrepos/source/#2}}%
              371
              372
                   \if@iswindows@\path@to@windows\temp@path\fi%
              373
                   \frameimage[#1]{\temp@path}%
              374
              375 }%
```

```
\mhinputref*
                  376 \let\orig@mhinputref\mhinputref
                  377 \def\mhinputref{\@ifstar\nmhinputref\orig@mhinputref}
                  378 \newcommand\nmhinputref[2][]{\ifnotes\orig@mhinputref[#1]{#2}\fi}
     \mhexcursion
                  379 \newcommand\activatemhexcursion[2][]{\ifstrempty{#1}%
                        {\gappto\printexcursions{\mhinputref[\mh@currentrepos]{#2}}}%
                        {\gappto\printexcursions{\mhinputref[#1]{#2}}}}
                  382 \newcommand\mhexcursion[4][]{% repos, label, path, text
                       \ifnotes\activatemhexcursion[#1]{#3}\excursionref{#2}{#4}\fi}%
\mhexcursiongroup
                  384 \srefaddidkey{mhexcursiongroup}%
                  385 \addmetakey{mhexcursiongroup}{intro}%
                  386 \addmetakey{mhexcursiongroup}{mhrepos}%
                  387 \newcommand\mhexcursiongroup[1][]{%
                        \metasetkeys{mhexcursiongroup}{#1}%
                  388
                        \ifdefempty\printexcursions{}% only if there are excursions
                  389
                  390
                        {\begin{omgroup}[#1]{Excursions}%
                  391
                            \ifdefempty\mhexcursiongroup@intro{}%
                            {\ifdefempty\mhexcursiongroup@mhrepos%
                  392
                              {\mhinputref{\mhexcursiongroup@intro}}%
                  393
                  394
                              {\mhinputref[\mhexcursiongroup@mhrepos]{\mhexcursiongroup@intro}}}%
                            \printexcursions%
                  395
                          \end{omgroup}}}
                  396
                  397 (/mikoslides)
                          problem-mh.sty: Support for Problems
                   4.8
                  398 (*problem)
                  399 \ProvidesPackage{problem-mh}[2019/03/20 v1.1 MathHub support for the sTeX problem package]
                  400 \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.
                  401 \addmetakey{inclprob}{mhrepos}
                  402 \newcommand\includemhproblem[2][]{\metasetkeys{inclprob}{#1}%
                  403 \edef\mh@@repos{\mh@currentrepos}%
                  404 \ifx\inclprob@mhrepos\@empty\else\setcurrentreposinfo\inclprob@mhrepos\fi%
                  405 \edef\temp@path{\MathHub{\mh@currentrepos/source/#2}}
                  406 \if@iswindows@\path@to@windows\temp@path\fi
                  407 \input{\temp@path}%
                  408 \setcurrentreposinfo\mh@@repos\clear@inclprob@keys}
```

409 (/problem)

4.9 hwexam-mh.sty: Support for Assignments

```
410 \langle *hwexam \rangle
                   411 \ProvidesPackage{hwexam-mh}[2019/03/20 v1.1 MathHub support for the sTeX hwexam package]
                   412 \RequirePackage{mathhub}
                    The \inputmhassignment saves the current value of \mh@currentrepos in a local
\inputmhassignment
                    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.
                   413 \newcommand\inputmhassignment[2][]{\metasetkeys{inclassig}{#1}%
                   414 \edef\mh@@repos{\mh@currentrepos}%
                   415 \ifx\inclassig@mhrepos\@empty\else\setcurrentreposinfo\inclassig@mhrepos\fi%
                   416 \edef\temp@path{\MathHub{\mh@currentrepos/source/#2}}%
                   417 \if@iswindows@\path@to@windows\temp@path\fi%
                   418 \inputassignment[#1] {\temp@path}%
                   419 \setcurrentreposinfo\mh@@repos\clear@inclassig@keys}%
                   420 \newcommand\includemhassignment[2][]{\newpage\inputmhassignment[#1]{#2}}
```

4.10 tikzinput-mh.sty: Support for Assignments

421 (/hwexam)

```
422 \*tikzinput\\
423 \ProvidesPackage{tikzinput-mh}[2019/03/20 v1.1 MathHub support for the sTeX tikzinput package]
424 \RequirePackage{mathhub}
425 \RequirePackage{pathsuris}
426 \define@key{Gin}{mhrepos}{\def\Gin@mhrepos{#1}}
427 \newcommand\mhtikzinput[2][]{\def\Gin@mhrepos{}\setkeys{Gin}{#1}%
428 \edef\mh@crepos{\mh@currentrepos}%
429 \ifx\Gin@mhrepos\@empty\edef\temp@path{\MathHub{\mh@currentrepos/source/#2}}%
430 \else\setcurrentreposinfo\Gin@mhrepos\edef\temp@path{\MathHub{\Gin@mhrepos/source/#2}}\fi%
431 \if@iswindows@\path@to@windows\temp@path\fi%
432 \tikzinput[#1]{\temp@path}
433 \def\Gin@mhrepos{}\setcurrentreposinfo\mh@crepos}
434 \newcommand\cmhtikzinput[2][]{\begin{center}\mhtikzinput[#1]{#2}\end{center}}
435 \/tikzinput\\
435 \/tikzinput\\
436 \/tikzinput\\
437 \newcommand\cmhtikzinput[2][]{\begin{center}\mhtikzinput[#1]{#2}\end{center}}
438 \/tikzinput\\
439 \leftarrentreposinfo\mhtikzinput[#1]{#2}\end{center}}
430 \leftarrentreposinfo\mhtikzinput[#1]{#2}\end{center}}
```

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

```
436 (*|st)
437 \ProvidesPackage{lstmh}[2019/03/20 v1.1 MathHub support for the listings package]
438 \RequirePackage{mathhub}
439 \RequirePackage{pathsuris}
440 \RequirePackage{listings}
441 \define@key{lst}{mhrepos}{\def\lst@mhrepos{#1}}
442 \newcommand\lstinputmhlisting[2][]{\def\lst@mhrepos{}\setkeys{lst}{#1}%
443 \edef\mh@@repos{\mh@currentrepos}%
444 \ifx\lst@mhrepos\@empty\edef\temp@path{\MathHub{\mh@currentrepos/source/#2}}%
445 \else\edef\temp@path{\MathHub{\lst@mhrepos/source/#2}}\fi%
446 \if@iswindows@\path@to@windows\temp@path\fi%
```

```
447 \lstinputlisting[#1] {\temp@path}  
448 \def\lst@mhrepos{}\setcurrentreposinfo\mh@@repos}  
449 \newcommand\clstinputmhlisting[2][] {\begin{center}\lstinputmhlisting[#1] {#2}\end{center}}  
450 \langle | \text{Ist} \rangle
```

Change History

v1.0	libi
General: Deprecated	meta
${\tt mhviewsketch}$ 1	repo
moved all MH functionality into	more
one DTX file $\dots 1$	break
v1.1	one o
General: Changed the semantics of	

libinput to input first the meta-inf-level and then repos-level file; this allows more sharing and does not break MathHub content (only one of them currently exists) . . 1

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).