

# MathHub Support for $\text{\LaTeX}^*$

Michael Kohlhasse  
Jacobs University, Bremen  
<http://kwarc.info/kohlhasse>

November 4, 2015

## Abstract

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

The `mathhub` packages extend  $\text{\LaTeX}$  with support for the MathHub.info portal

## Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>The User Interface</b>	<b>3</b>
2.1	Package Options . . . . .	3
2.2	<code>modules-mh</code> : MH Variants for Modules . . . . .	3
2.3	<code>omtext-mh</code> : MH Variants for OMText . . . . .	4
2.4	<code>smultiling-mh</code> : MH Variants for Multilinguality . . . . .	4
2.5	<code>mikoslides-mh</code> : Support for MiKo Slides . . . . .	4
2.6	<code>problem-mh</code> : Support for Problems . . . . .	5
2.7	<code>hwexam-mh</code> : Support for Assignments . . . . .	5
<b>3</b>	<b>Limitations</b>	<b>5</b>
<b>4</b>	<b>Implementation</b>	<b>6</b>
4.1	General Infrastructure . . . . .	6
4.2	<code>modules-mh</code> : MH Variants for Modules . . . . .	7
4.3	<code>omtext-mh</code> : MH Variants for OMText . . . . .	12
4.4	<code>smultiling-mh</code> : MH Variants for Multilinguality . . . . .	13
4.5	<code>mikoslides-mh</code> : Support for MiKo Slides . . . . .	15
4.6	<code>problem-mh</code> : Support for Problems . . . . .	16

---

\*Version v1.0 (last revised 2015/11/04)

4.7	hwexam-mh: Support for Assignments . . . . .	17
4.8	Finale . . . . .	18

# 1 Introduction

Much of the  $\text{\LaTeX}$  content is hosted on **MathHub** (<http://MathHub.info>), a portal and archive for flexiformal mathematics. **MathHub** offers GIT repositories (public and private escrow) for mathematical documentation projects, online and offline authoring and document development infrastructure, and a rich, interactive reading interface. The **modules** package supports repository-sensitive operations on **MathHub**.

Note that **MathHub** has two-level repository names of the form  $\langle group \rangle / \langle repo \rangle$ , where  $\langle group \rangle$  is a **MathHub**-unique repository group and  $\langle repo \rangle$  a repository name that is  $\langle group \rangle$ -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  $\text{\LaTeX}$  author with **MathHub**-enabled versions of the  $\text{\LaTeX}$  macros, which are defined in this package.

**Caveat** if you want to use the **MathHub** support macros (let’s call them **mh-variants**), then every time a module is imported or a document fragment is included from another repos, 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 The User Interface

### 2.1 Package Options

none so far

### 2.2 modules-mh: MH Variants for Modules

`\importmhmodule` The `\importmhmodule` macro is a variant of `\importmodule` with repository support. Instead of writing

```
\defpath{MathHub}{/user/foo/lmh/MathHub}
\importmodule[load=\MathHub{fooMH/bar/source/baz/foobar}]{foobar}
```

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

```
\importmhmodule[repos=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

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

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

`\mhcurrentrepos` Of course, neither  $\text{\LaTeX}$  nor  $\text{\LaTeXML}$  know about the repositories when they are called from a file system, so we can use the `\mhcurrentrepos` macro to tell them. But this is only needed to initialize the infrastructure in the driver file. In particular, we do not need to set it in each module, since the `\importmhmodule` macro sets the current repository automatically.

`\usemhmodule` The `\usemhmodule` and `\adoptmhmodule` macros are the analogs to `\usemodule` and `\adoptmodule`.

`\mhinputref` For this, the `modules` package supplies the mh-variants `\mhinputref` and `\mhinput` of the `\inputref` macro introduced above and normal  $\text{\LaTeX}$  `\input` macro.

## 2.3 omtext-mh: MH Variants for OMText

`\mhgraphics` The `\mhgraphics` macro is a variant of `\mycgraphics` with repository support. Instead of writing

```
\defpath{MathHub}{/user/foo/lmh/MathHub}
\mycgraphics{\MathHub{fooMH/bar/source/baz/foobar}}
```

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

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

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

## 2.4 smultiling-mh: MH Variants for Multilinguality

1 2

## 2.5 mikoslides-mh: Support for MiKo Slides

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

```
\defpath{MathHub}{/user/foo/lmh/MathHub}
\frameimage{\MathHub{fooMH/bar/source/baz/foobar}}
```

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

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

---

<sup>1</sup>EDNOTE: needs to be documented

<sup>2</sup>EDNOTE: mhmodsig seems to be missing what happened?

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}
```

## 2.6 problem-mh: Support for Problems

`\includemhproblem` The `\includemhproblem` macro is a variant of `\importmodule` with repository support. Instead of writing

```
\defpath{MathHub}{/user/foo/lmh/MathHub}
\includeproblem[pts=7]{\MathHub{fooMH/bar/source/baz/foobar}}
```

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

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

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

## 2.7 hwexam-mh: Support for Assignments

`\includemhassignment` The `\includemhassignment` macro is a variant of `\includeassignment` with repository support. Instead of writing

```
\defpath{MathHub}{/user/foo/lmh/MathHub}
\includeassignment[pts=7]{\MathHub{fooMH/bar/source/baz/foobar}}
```

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

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

## 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 **TeX** GitHub repository [sTeX].

1. none reported yet.

## 4 Implementation

The `sref` package generates two files: the  $\text{\LaTeX}$  package (all the code between `\package` and `\endpackage`) and the  $\text{\LaTeX}$ XML bindings (between `\beginltxml` and `\endltxml`). We keep the corresponding code fragments together, since the documentation applies to both of them and to prevent them from getting out of sync.

We first set up header information for the  $\text{\LaTeX}$ XML binding files and the base package.

```

1 \beginltxml | modules.ltxml | omtex.ltxml | smultiling.ltxml | mikoslides.ltxml | problem.ltxml | hwexam.ltxml
2 \package LaTeXXML::Package::Pool;
3 use strict;
4 use LaTeXXML::Package;
5 \endltxml | modules.ltxml | omtex.ltxml | smultiling.ltxml | mikoslides.ltxml | problem.ltxml | hwexam.ltxml
6 \package\ProvidesPackage{mathhub}[2015/11/04 v1.0 sTeX Support for MathHub.info]

7 \package
8 \DeclareOption*{}
9 \ProcessOptions
10 \endpackage
11 \beginltxml
12 \DeclareOption(undef,sub {});
13 \ProcessOptions();
14 \endltxml

```

Then we need to set up the packages by requiring the `metakeys` package [Koh15] to be loaded (in the right version).

```

15 \package
16 \RequirePackage{keyval}
17 \endpackage
18 \beginltxml
19 \RequirePackage('keyval');
20 \endltxml

```

### 4.1 General Infrastructure

`\mhcurrentrepos` `\mhcurrentrepos` is used to initialize the current repository. If the repos 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 repos.

```

21 \package
22 \newrobustcmd\mhcurrentrepos[1]{%
23   \edef\@test{#1}%
24   \ifx\@test\mh@currentrepos% if new dir = old dir
25     \relax% no need to change
26   \else%
27     \protected@write\@auxout{}{\string\@mhcurrentrepos{#1}}%
28     \fi%
29     \@mhcurrentrepos{#1}% define mh@currentrepos
30 }%
31 \newrobustcmd\@mhcurrentrepos[1]{\edef\mh@currentrepos{#1}}%

```

```

32 </package>
33 <*ltxml>
34 DefMacro('mhcurrentrepos{','\@mhcurrentrepos{#1}');
35 DefMacro('mhcurrentrepos{','\def\mh@currentrepos{#1}\@mhcurrentrepos{#1}');
36 DefConstructor('mhcurrentrepos{','',
37   afterDigest => sub{ AssignValue('current_repos',ToString($_[1]->getArg(1)),'global'); } );
38 </ltxml>#<
\libinput the \libinput macro inputs from the lib directory of the MathHub repository
or the meta-inf/lib repos of the group.
39 <ltxml>RaxTeX('
40 <*package | ltxml>
41 \def\modules@@first#1/#2;{#1}
42 \newcommand\libinput[1]{\def\@libfile{\MathHub{\mh@currentrepos/lib/#1}}%
43 \IfFileExists{\@libfile}{\input\@libfile}%
44 {\edef\@group{\expandafter\modules@@first\mh@currentrepos;}
45 \edef\@infile{\MathHub{\@group/meta-inf/lib/#1}}
46 \IfFileExists{\@infile}{\input{\@infile}}%
47 {\PackageError{modules}
48   {Library file missing, cannot input #1\MessageBreak%
49     Both \@libfile.tex\MessageBreak and \@infile.tex\MessageBreak do not exit}%
50   {Check whether the file name is correct}}}%
51 </package | ltxml>
52 <ltxml>');

```

## 4.2 modules-mh: MH Variants for Modules

We set up package options and pass them on to the modules package, which we also load.

```

53 <*modules>
54 \ProvidesPackage{modules-mh}[2015/11/04 v1.0 MathHub support for the sTeX modules package]
55 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{modules}}
56 \ProcessOptions
57 \RequirePackage{modules}
58 \RequirePackage{mathhub}
59 </modules>
60 <*modules.ltxml>
61 DeclareOption(undef,sub{PassOptions('modules','sty',ToString(Digest(T_CS('CurrentOption')))));
62 ProcessOptions();
63 RequirePackage('modules');
64 RequirePackage('mathhub');
65 </modules.ltxml>
\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.

```

```

66 <*modules>
67 \srefaddidkey{importmhmodule}%
68 \addmetakey{importmhmodule}{repos}% saves the repo's path. E.g: smglom/numberfield
69 \addmetakey{importmhmodule}{path}% saves the module name. E.g: naturalnumbers
70 \addmetakey[sms]{importmhmodule}{ext}% saves the extension: E.g: tex
71 \addmetakey[false]{importmhmodule}{conservative}[true]%
72 \newrobustcmd\importmhmodule[2] []{%
73   \metasetkeys{importmhmodule}{#1}%
74   \ifx\importmhmodule@path\@empty% if module name is not set
75     \importmodule[ext=\importmhmodule@ext,id=\importmhmodule@id]{#2}%
76   \else%
77     \edef\mh@@repos{\mh@currentrepos}% remember so that we can reset it.
78     \ifx\importmhmodule@repos\@empty% if in the same repos
79       \relax% no need to change mh@currentrepos, i.e, current dirctory.
80     \else%
81       \mhcurrentrepos{\importmhmodule@repos}% change it.
82     \fi%
83     \importmodule[load=\MathHub{\mh@currentrepos/source/\importmhmodule@path},%
84     ext=\importmhmodule@ext,id=\importmhmodule@id]{#2}%
85     \mhcurrentrepos{\mh@@repos}% after importing, reset to old value
86   \fi%
87   \ignorespaces%
88 }%

```

and now the analogs

`\usemhmodule`

```

89 \newrobustcmd\usemhmodule[2] []{%
90   \metasetkeys{importmhmodule}{#1}%
91   \ifx\importmhmodule@path\@empty%
92     \usemodule[ext=\importmhmodule@ext,id=\importmhmodule@id]{#2}%
93   \else%
94     \edef\mh@@repos{\mh@currentrepos}%
95     \ifx\importmhmodule@repos\@empty%
96     \else%
97       \mhcurrentrepos{\importmhmodule@repos}%
98     \fi%
99     \usemodule[load=\MathHub{\mh@currentrepos/source/\importmhmodule@path},ext=\importmhmodule@
100     \mhcurrentrepos\mh@@repos%
101   \fi%
102   \ignorespaces%
103 }%

```

`\adoptmhmodule`

```

104 \newrobustcmd\adoptmhmodule[2] []{%
105   \metasetkeys{importmhmodule}{#1}%
106   \ifx\importmhmodule@path\@empty
107     \adoptmodule[ext=\importmhmodule@ext,id=\importmhmodule@id]{#2}%
108   \else%
109     \edef\mh@@repos{\mh@currentrepos}%

```



```

110 \ifx\importmhmodule@repos\empty%
111 \else%
112 \mhcurrentrepos{\importmhmodule@repos}%
113 \fi%
114 \adoptmodule[load=\MathHub{\mh@currentrepos/source/\importmhmodule@path},ext=\importmhmodul
115 \mhcurrentrepos\mh@@repos%
116 \fi%
117 \ignorespaces%
118 }%

\mhinputref
119 \newrobustcmd\mhinputref[2][]{%
120 \def\@repos{#1}%
121 \edef\mh@@repos{\mh@currentrepos}%
122 \ifx\@repos\empty%
123 \else%
124 \mhcurrentrepos{#1}%
125 \fi%
126 \inputref{\MathHub{\mh@currentrepos/source/#2}}%
127 \mhcurrentrepos\mh@@repos%
128 \ignorespaces%
129 }%

\mhinput
130 \let\mhinput\mhinputref%

importmhmodulevia
131 \newenvironment{importmhmodulevia}[3][]{%
132 \gdef\@@doit{\importmhmodule[#1]{#2}{#3}}%
133 \ifmod@show\par\noindent importing module #2 via \@@doit\fi
134 }{%
135 \aftergroup\@@doit\ifmod@show end import\fi%
136 }%

137 \srefaddidkey{mhview}
138 \addmetakey{mhview}{display}
139 \addmetakey{mhview}{creators}
140 \addmetakey{mhview}{contributors}
141 \addmetakey{mhview}{srccite}
142 \addmetakey*{mhview}{title}
143 \addmetakey{mhview}{fromrepos}
144 \addmetakey{mhview}{torepos}
145 \addmetakey{mhview}{frompath}
146 \addmetakey{mhview}{topath}
147 \addmetakey[sms]{mhview}{ext}

mhview the MathHub version
148 \newenvironment{mhview}[3][]{% keys, from, to
149 \metasetkeys{mhview}{#1}%

```

```

150 \sref@target%
151 \begin{@mhview}{#2}{#3}%
152 \view@heading{#2}{#3}{\mhview@display}{\mhview@title}%
153 }{%
154 \end{@mhview}%
155 \ignorespaces%
156 }%
157 \ifmod@show\surroundwithmdframed{mhview}\fi

```

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

```

158 \newenvironment{@mhview}[2]{%from, to
159 \importmhmodule[repos=\mhview@fromrepos,path=\mhview@frompath,ext=\mhview@ext]{#1}%
160 \importmhmodule[repos=\mhview@torepos,path=\mhview@topath,ext=\mhview@ext]{#2}%
161 }{}%

```

**mhviewsketch** The mhviewsketch environment behaves like mhview, but only has text contents.

```

162 \newenvironment{mhviewsketch}[3][{}]{%
163 \metasetkeys{mhview}{#1}%
164 \sref@target%
165 \begin{@mhview}{#2}{#3}%
166 \view@heading{#2}{#3}{\mhview@display}{\mhview@title}%
167 }{%
168 \end{@mhview}%
169 \ignorespaces%
170 }%
171 \ifmod@show\surroundwithmdframed{mhviewsketch}\fi
172 </modules>

```

3

```

173 <*modules.ltxml>
174 DefKeyVal('mhview','id','Semiverbatim');
175 DefKeyVal('mhview','fromrepos','Semiverbatim');
176 DefKeyVal('mhview','torepos','Semiverbatim');
177 DefKeyVal('mhview','frompath','Semiverbatim');
178 DefKeyVal('mhview','topath','Semiverbatim');
179 DefKeyVal('mhview','title','Semiverbatim');
180 DefKeyVal('mhview','creators','Semiverbatim');
181 DefKeyVal('mhview','contributors','Semiverbatim');
182 DefKeyVal('mhview','display','Semiverbatim');
183 DefKeyVal('mhview','ext','Semiverbatim');
184 DefMacroI(T_CS('\begin{mhview}'),'OptionalKeyVals:mhview {}{}', sub {
185 my ($gullet, $keyvals, $from_arg, $to_arg) = @_;
186 my $from = ToString(Digest($from_arg));
187 my $to = ToString(Digest($to_arg));
188 AssignValue(from_module => $from);
189 AssignValue(to_module => $to);
190 my $from_repos = ToString(GetKeyVal($keyvals,'fromrepos'));
191 my $to_repos = ToString(GetKeyVal($keyvals,'torepos'));

```

---

<sup>3</sup>EDNOTE: MK: sort these into the rest.

```

292 my $repos = LookupValue('current_repos');
293 my $from_path = ToString(GetKeyVal($keyvals,'frompath'));
294 my $to_path = ToString(GetKeyVal($keyvals,'topath'));
295 my $ext = ToString(GetKeyVal($keyvals,'ext')) if $keyvals;
296 $ext = 'sms' unless $ext;
297 my $current_repos = LookupValue('current_repos');
298 if (!$from_repos) { $from_repos = $current_repos; }
299 if (!$to_repos) { $to_repos = $current_repos; }
300 return (
301     Tokenize("\\importMHmoduleI[repos=$from_repos,path=$from_path,ext=$ext]{$from}")->unlist,
302     Tokenize("\\importMHmoduleI[repos=$to_repos,path=$to_path,ext=$ext]{$to}")->unlist,
303     Invocation(T_CS('\\begin{viewenv}'),$keyvals,$from_arg,$to_arg)->unlist
304 );
305 });
306 DefMacroI('\\end{mhview}',undef,'\\end{viewenv}');
307
308 DefMacroI(T_CS('\\begin{mhviewsketch}'),'OptionalKeyVals:mhview {}{}', sub {
309     my ($gullet, $keyvals, $from_arg, $to_arg) = @_;
310     my $from = ToString(Digest($from_arg));
311     my $to = ToString(Digest($to_arg));
312     my $from_repos = ToString(GetKeyVal($keyvals,'fromrepos'));
313     my $to_repos = ToString(GetKeyVal($keyvals,'torepos'));
314     my $repos = LookupValue('current_repos');
315     my $from_path = ToString(GetKeyVal($keyvals,'frompath'));
316     my $to_path = ToString(GetKeyVal($keyvals,'topath'));
317     my $ext = ToString(GetKeyVal($keyvals,'ext')) if $keyvals;
318     $ext = 'sms' unless $ext;
319     my $current_repos = LookupValue('current_repos');
320     if (!$from_repos) { $from_repos = $current_repos; }
321     if (!$to_repos) { $to_repos = $current_repos; }
322     return (
323         Tokenize("\\importMHmoduleI[repos=$from_repos,path=$from_path,ext=$ext]{$from}")->unlist,
324         Tokenize("\\importMHmoduleI[repos=$to_repos,path=$to_path,ext=$ext]{$to}")->unlist,
325         Invocation(T_CS('\\begin{viewsketchenv}'),$keyvals,$from_arg,$to_arg)->unlist
326     );
327 });
328 DefMacroI('\\end{mhviewsketch}',undef,'\\end{viewsketchenv}');
329
330 DefConstructor('\\importmhmodule OptionalKeyVals:importmhmodule {}',
331     "<omdoc:imports "
332     . "from='?&GetKeyVal(#1,'load')(&canonical_omdoc_path(&GetKeyVal(#1,'load')))(###2'"
333     . "?&defined(&GetKeyVal(#1,'conservative'))(load='&GetKeyVal(#1,'conservative'))'"
334     afterDigest => \\importMHmoduleI);
335
336 DefConstructor('\\usemhmodule OptionalKeyVals:importmhmodule {}',
337     "<omdoc:uses from='?&GetKeyVal(#1,'load')(&canonical_omdoc_path(&GetKeyVal(#1,'load')))(###"
338     afterDigest => \\importMHmoduleI);
339
340 DefConstructor('\\adoptmhmodule OptionalKeyVals:importmhmodule {}',
341     "<omdoc:adopts from='?&GetKeyVal(#1,'load')(&canonical_omdoc_path(&GetKeyVal(#1,'load')))(#"

```

```

242   afterDigest => \&importMHmoduleI);
243
244 RawTeX('
245 \newcommand\mhinputref[2] [] {\def\@repos{#1}%
246 \edef\mh@@repos{\mh@currentrepos}%
247 \ifx\@repos\@empty\else\mhcurrentrepos{#1}\fi%
248 \inputref{\MathHub{\mh@currentrepos/source/#2}}}%
249 \mhcurrentrepos\mh@@repos}
250 \newcommand\mhinput[2] [] {\def\@repos{#1}%
251 \edef\mh@@repos{\mh@currentrepos}%
252 \ifx\@repos\@empty\else\mhcurrentrepos{#1}\fi%
253 \input{\MathHub{\mh@currentrepos/source/#2}}}%
254 \mhcurrentrepos\mh@@repos}
255 \newenvironment{importmhmodulevia}[3] [] {\def\@repos{#1}%
256 \edef\mh@@repos{\mh@currentrepos}%
257 \ifx\@repos\@empty\else\mhcurrentrepos{#1}\fi%
258 \gdef\@@doit{\importmhmodule[#1]{#2}{#3}}
259 \begin{importmoduleenv}[load=\MathHub{\mh@currentrepos/source/#2}]{#3}}
260 {\end{importmoduleenv}\aftergroup\@@doit}
261 ');
262 \modules.ltxml)

```

### 4.3 omtex-mh: MH Variants for OMTex

We set up package options and pass them on to the omtex package, which we also load.

```

263 (*omtex)
264 \ProvidesPackage{omtex-mh}[2015/11/04 v1.0 MathHub support for the sTeX omtex package]
265 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{omtex}}
266 \ProcessOptions
267 \RequirePackage{omtex}
268 \RequirePackage{mathhub}
269 \end{omtex}
270 (*omtex.ltxml)
271 DeclareOption(undef,sub{PassOptions('omtex','sty',ToString(Digest(T_CS('\CurrentOption')))); }
272 ProcessOptions();
273 RequirePackage('omtex');
274 RequirePackage('mathhub');
275 \end{omtex.ltxml}

```

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

```

276 (*omtex)
277 \addmetakey{Gin}{mhrepos}
278 \newcommand\mhgraphics[2] [] {\metasetkeys{Gin}{#1}%
279 \edef\mh@@repos{\mh@currentrepos}%
280 \ifx\Gin@mhrepos\@empty\mygraphics[#1]{\MathHub{\mh@currentrepos/source/#2}}}%
281 \else\mygraphics[#1]{\MathHub{\Gin@mhrepos/source/#2}}\fi
282 \def\Gin@mhrepos{\mhcurrentrepos\mh@@repos}

```

```

283 \newcommand\mhcgraphics[2] [] {\begin{center}\mhgraphics[#1]{#2}\end{center}}
284 \newcommand\mhgraphics[2] [] {\fbox{\mhgraphics[#1]{#2}}}
285 \newcommand\mhcbgraphics[2] [] {\begin{center}\fbox{\mhgraphics[#1]{#2}}\end{center}}
286 \omtext
287 \omtext.ltxml
288 sub mhgraphics {
289   my ($gullet,$keyval,$arg2) = @_ ;
290   my $repo_path;
291   if ($keyval) {
292     $repo_path = ToString(GetKeyVal($keyval,'mhrepos')); }
293   if (! $repo_path) {
294     $repo_path = ToString(Digest(T_CS('\mh@currentrepos'))); }
295   else {
296     $keyval->setValue('mhrepos',undef); }
297   my $mathhub_base = ToString(Digest('\MathHub{'}));
298   my $finalpath = $mathhub_base.$repo_path.'/source/'.ToString($arg2);
299   return Invocation(T_CS('\@includegraphicx'), $keyval, T_OTHER($finalpath)); }#$
300 DefKeyVal('Gin','mhrepos','Semiverbatim');
301 DefMacro('\mhgraphics OptionalKeyVals:Gin {}', \&mhgraphics);
302 DefMacro('\mhcgraphics [] {}', '\begin{center}\mhgraphics[#1]{#2}\end{center}');
303 DefMacro('\mhgraphics [] {}', '\fbox{\mhgraphics[#1]{#2}}');
304 \omtext.ltxml

```

#### 4.4 smultiling-mh: MH Variants for Multilinguality

We set up package options and pass them on to the `smultiling` package, which we also load.

```

305 \smultiling
306 \ProvidesPackage{smultiling-mh}[2015/11/04 v1.0 MathHub support for the sTeX smultiling package]
307 \DeclareOption*\PassOptionsToPackage{\CurrentOption}{smultiling}
308 \ProcessOptions
309 \RequirePackage{smultiling}
310 \RequirePackage{mathhub}
311 \smultiling
312 \smultiling.ltxml
313 DeclareOption(undef,sub{PassOptions('smultiling','sty',ToString(Digest(T_CS('\CurrentOption'))))}
314 ProcessOptions();
315 RequirePackage('smultiling');
316 RequirePackage('mathhub');
317 \smultiling.ltxml

```

`mhmodnl:`

```

318 \smultiling
319 \addmetakey{mhmodnl}{repos}
320 \addmetakey{mhmodnl}{path}
321 \addmetakey*{mhmodnl}{title}
322 \addmetakey*{mhmodnl}{creators}
323 \addmetakey*{mhmodnl}{contributors}
324 \addmetakey{mhmodnl}{srccite}

```

```

325 \addmetakey{primary}{mhmodnl}[yes]
326 \smultiling
327 \smultiling.ltxml
328 DefKeyVal('mhmodnl','title','Semiverbatim');
329 DefKeyVal('mhmodnl','repos','Semiverbatim');
330 DefKeyVal('mhmodnl','path','Semiverbatim');
331 DefKeyVal('mhmodnl','creators','Semiverbatim');
332 DefKeyVal('mhmodnl','contributors','Semiverbatim');
333 DefKeyVal('mhmodnl','primary','Semiverbatim');
334 \smultiling.ltxml

```

`mhmodnl` The `mhmodnl` environment is just a layer over the module environment and the `\importmhmodule` macro with the keys and language suitably adapted.

```

335 \smultiling
336 \newenvironment{mhmodnl}[3][\metasetkeys{mhmodnl}{#1}%
337 \def\@test{#1}\ifx\@test\empty\begin{module}[id=#2.#3]\else\begin{module}[id=#2.#3,#1]\fi%
338 \edef\@repos{\ifx\mhmodnl@repos\empty\mh@currentrepos\else\mhmodnl@repos}
339 \if@langfiles\importmhmodule[repos=\@repos,load=#2,ext=tex]{#2}\else
340 \ifx\mhmodnl@load\empty\importmodule{#2}\else\importmodule[ext=tex,load=\mhmodnl@load]{#2}\fi%
341 \fi}
342 \end{module}}
343 \smultiling
344 \smultiling.ltxml
345 DefEnvironment('mhmodnl OptionalKeyVals:mhmodnl {}{}',
346     "?#excluded()(<omdoc:theory xml:id='#2.#3' >"
347     . "?&defined(&GetKeyVal(#1,'creators'))(<dc:creator>&GetKeyVal(#1,'creators')</dc:cr
348     . "?&defined(&GetKeyVal(#1,'title'))(<dc:title>&GetKeyVal(#1,'title')</dc:title>())"
349     . "?&defined(&GetKeyVal(#1,'contributors'))(<dc:contributor>&GetKeyVal(#1,'contribut
350     . "<omdoc:imports from='?&GetKeyVal(#1,'load')(&canonical_omdoc_path(&GetKeyVal(#1,'
351     . "#body"
352     . "</omdoc:theory>)",
353     afterDigestBegin=>sub {
354     my ($stomach, $whatsit) = @_;
355     my $keyval = $whatsit->getArg(1);
356     my $signature = ToString($whatsit->getArg(2));
357     my $language = ToString($whatsit->getArg(3));
358     my $repos = ToString(GetKeyVal($keyval,'torepos'));
359     my $current_repos = LookupValue('current_repos');
360     if (!$repos) { $repos = $current_repos; }
361     my $defpaths = LookupValue('defpath');
362     my $load_path = ($$defpaths{MathHub}).$repos.'/source/'.$signature;
363
364     if ($keyval) {
365     # If we're not given load, AND the langfiles option is in effect,
366     # default to #2
367     if ((! $keyval->getValue('path')) && (LookupValue('smultiling_langfiles')) {
368     $keyval->setValue('load',$load_path); }
369     # Always load a TeX file
370     $keyval->setValue('ext','tex');
371     $keyval->setValue('id',"$signature.$language"); }

```

```

372     module_afterDigestBegin(@_);
373     importmoduleI(@_);
374     return; },
375     afterDigest=>sub {
376         module_afterDigest(@_); } });
377 </smultiling.ltxml>%$

```

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

```

378 <smultiling.ltxml>RawTeX(
379 <*smultiling | smultiling.ltxml>
380 \newenvironment{mhviewsig}[4] [] {\def@test{#1}\ifx@test\@empty%
381 \begin{mhview}[id=#2,ext=tex]{#3}{#4}\else\begin{mhview}[id=#2,#1,ext=tex]{#3}{#4}\fi}
382 {\end{mhview}}

```

**mhviewnl** The `mhviewnl` environment is just a layer over the `mhviewsketch` environment with the keys and language suitably adapted.<sup>4</sup>

```

383 \newenvironment{mhviewnl}[5] [] {\def@test{#1}\ifx@test\@empty%
384 \begin{mhviewsketch}[id=#2.#3,ext=tex]{#4}{#5}\else%
385 \begin{mhviewsketch}[id=#2.#3,#1,ext=tex]{#4}{#5}\fi}
386 {\end{mhviewsketch}}
387 </smultiling | smultiling.ltxml>
388 <smultiling.ltxml>');

```

## 4.5 mikosides-mh: Support for MiKo Slides

We set up package options and pass them on to the `mikosides` package, which we also load.

```

389 <*mikosides>
390 \ProvidesPackage{mikosides-mh}[2015/11/04 v1.0 MathHub support for the sTeX mikosides package]
391 \DeclareOption*{\PassOptionsToPackage{CurrentOption}{mikosides}}
392 \ProcessOptions
393 \RequirePackage{mikosides}
394 \RequirePackage{mathhub}
395 </mikosides>
396 <*mikosides.ltxml>
397 \DeclareOption(undef,sub{PassOptions('mikosides','sty',ToString(Digest(T_CS('CurrentOption'))))}
398 \ProcessOptions();
399 \RequirePackage('mikosides');
400 \RequirePackage('mathhub');
401 </mikosides.ltxml>

```

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

```

402 <mikosides>\addmetakey{Gin}{mhrepos}
403 <mikosides.ltxml>\DefKeyVal('Gin','mhrepos','Semiverbatim');

```

<sup>4</sup>EDNOTE: MK: we have to do something about the `if@langfiles` situation here. But this is non-trivial, since we do not know the current path, to which we could append `.(lang)!`

```

404 <mikoslides.ltxml>RawTeX(
405 <*mikoslides.ltxml | mikoslides>
406 \newcommand\mhframeimage[2] [] {%
407   \metasetkeys{Gin}{#1}%
408   \edef\mh@@repos{\mh@currentrepos}%
409   \ifx\Gin@mhrepos\empty%
410     \frameimage[#1]{\MathHub{\mh@currentrepos/source/#2}}%
411   \else%
412     \frameimage[#1]{\MathHub{\Gin@mhrepos/source/#2}}%
413   \fi%
414 }%
415 </mikoslides.ltxml | mikoslides>
416 <mikoslides.ltxml>');

```

## 4.6 problem-mh: Support for Problems

We set up package options and pass them on to the `problem` package, which we also load.

```

417 <*problem>
418 \ProvidesPackage{problem-mh}[2015/11/04 v1.0 MathHub support for the sTeX problem package]
419 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{problem}}
420 \ProcessOptions
421 \RequirePackage{problem}
422 \RequirePackage{mathhub}
423 </problem>
424 <*problem.ltxml>
425 \DeclareOption(undef,sub{PassOptions('problem','sty',ToString(Digest(T_CS('\CurrentOption'))));
426 \ProcessOptions();
427 \RequirePackage('problem');
428 \RequirePackage('mathhub');
429 </problem.ltxml>

```

`\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`.

```

430 <*problem>
431 \newcommand\includemhproblem[2] [] {\metasetkeys{inclprob}{#1}%
432 \edef\mh@@repos{\mh@currentrepos}%
433 \ifx\inclprob@mhrepos\empty\else\mh@currentrepos\inclprob@mhrepos\fi%
434 \input{\MathHub{\mh@currentrepos/source/#2}}%
435 \mh@currentrepos\mh@@repos\clear@inclprob@keys}
436 </problem>
437 <*problem.ltxml>
438 sub includemhproblem {
439   my ($gullet,$keyval,$arg2) = @_;
440   my $repo_path;
441   if ($keyval) {
442     $repo_path = ToString(GetKeyVal($keyval,'mhrepos')); }

```



```

443 if (! $repo_path) {
444   $repo_path = ToString(Digest(T_CS('\mh@currentrepos'))); }
445 else {
446   $keyval->setValue('mhrepos',undef); }
447 my $mathhub_base = ToString(Digest('\MathHub{'}));
448 my $finalpath = $mathhub_base.$repo_path.'/source/'.ToString($arg2);
449 return Invocation(T_CS('\includeproblem'), $keyval, T_OTHER($finalpath)); }#$
450 DefKeyVal('inclprob','mhrepos','Semiverbatim');
451 DefMacro('\includemhproblem OptionalKeyVals:inclprob {'', \&includemhproblem);
452 </problem.ltxml>

```

## 4.7 hwexam-mh: Support for Assignments

We set up package options and pass them on to the hwexam package, which we also load.

```

453 <*hwexam>
454 \ProvidesPackage{hwexam-mh}[2015/11/04 v1.0 MathHub support for the sTeX hwexam package]
455 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{hwexam}}
456 \ProcessOptions
457 \RequirePackage{hwexam}
458 \RequirePackage{mathhub}
459 </hwexam>
460 <*hwexam.ltxml>
461 DeclareOption(undef,sub{PassOptions('hwexam','sty',ToString(Digest(T_CS('\CurrentOption')))); }
462 ProcessOptions();
463 RequirePackage('hwexam');
464 RequirePackage('mathhub');
465 </hwexam.ltxml>

```

`\includemhassignment` The `\includemhassignment` 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`.

```

466 <*package>
467 \newcommand\includemhassignment[2][\metasetkeys{inclassig}{#1}%
468 \edef\mh@@repos{\mh@currentrepos}%
469 \ifx\inclassig@mhrepos\@empty\else\mhcurrentrepos\inclassig@mhrepos\fi%
470 \includeassignment[#1]{\MathHub{\mh@currentrepos/source/#2}}%
471 \mhcurrentrepos\mh@@repos\clear@inclassig@keys}
472 </package>
473 <*ltxml>
474 sub includemhassignment {
475   my ($gullet,$keyval,$arg2) = @_ ;
476   my $repo_path;
477   if ($keyval) {
478     $repo_path = ToString(GetKeyVal($keyval,'mhrepos')); }
479   if (! $repo_path) {
480     $repo_path = ToString(Digest(T_CS('\mh@currentrepos'))); }
481   else {

```

```

482     $keyval->setValue('mhrepos',undef); }
483   my $mathhub_base = ToString(Digest('\MathHub{'}));
484   my $finalpath = $mathhub_base.$repo_path.'/source/'.ToString($arg2);
485   return Invocation(T_CS('\includeassignment'), $keyval, T_OTHER($finalpath)); }##
486 DefKeyVal('inclprob','mhrepos','Semiverbatim');
487 DefMacro('\includemhassignment OptionalKeyVals:inclprob {}', \&includemhassignment);
488 \ltxml

```

\inputmhassignment analogous

```

489 \*package
490 \newcommand\inputmhassignment[2][\metasetkeys{inclassig}{#1}%
491 \edef\mh@@repos{\mh@currentrepos}%
492 \ifx\inclassig@mhrepos\empty\else\mhcurrentrepos\inclassig@mhrepos\fi%
493 \inputassignment[#1]{\MathHub{\mh@currentrepos/source/#2}}%
494 \mhcurrentrepos\mh@@repos\clear@inclassig@keys}
495 \*package
496 \ltxml
497 sub inputmhassignment {
498   my ($gullet,$keyval,$arg2) = @_;
499   my $repo_path;
500   if ($keyval) {
501     $repo_path = ToString(GetKeyVal($keyval,'mhrepos')); }
502   if (! $repo_path) {
503     $repo_path = ToString(Digest(T_CS('\mh@currentrepos'))); }
504   else {
505     $keyval->setValue('mhrepos',undef); }
506   my $mathhub_base = ToString(Digest('\MathHub{'}));
507   my $finalpath = $mathhub_base.$repo_path.'/source/'.ToString($arg2);
508   return Invocation(T_CS('\inputassignment'), $keyval, T_OTHER($finalpath)); }##
509 DefMacro('\inputmhassignment OptionalKeyVals:inclprob {}', \&inputmhassignment);
510 \ltxml

```

## 4.8 Finale

Finally, we need to terminate the file with a success mark for perl.

```

511 \ltxml
512 1;
513 \ltxml

```

## References

- [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](http://kwarc.info/frabe/Research/HIJKR_dimensions_11.pdf).
- [Koh15] Michael Kohlhasse. *metakeys.sty: A generic framework for extensible Metadata in L<sup>A</sup>T<sub>E</sub>X*. Tech. rep. Comprehensive T<sub>E</sub>X Archive Network (CTAN), 2015. URL: <http://www.ctan.org/tex-archive/macros/latex/contrib/stex/metakeys/metakeys.pdf>.
- [sTeX] *KWARC/sTeX*. URL: <https://svn.kwarc.info/repos/stex> (visited on 05/15/2015).