The `mathscinet` package

American Mathematical Society

Version 2.01, 2004/06/30

1 Introduction

The `mathscinet` packages provides definitions for certain commands that occasionally occur in bibliographic data exported from MathSciNet and that are not provided by LaTeX.

Warning: Although the macros provided in this package are probably acceptable if all you need to do is format an entry downloaded from MathSciNet, they should probably not be used for other purposes. In general, if you are trying to typeset material in any languages that require these characters, you are better off using specialized fonts and encodings for those languages.

All Unicode character references are taken from *The Unicode Standard, Version 3.0* (Addison-Wesley, 2000).

2 Implementation

Standard declaration of package name and date.

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{mathscinet}[2002/04/17 v1.05]
\RequirePackage{textcmds}\relax
```

`\setboxz@h` A useful abbreviation borrowed from `amsgen`.

```
\providecommand\setboxz@h{\setbox\z@\hbox}
```

2.1 Math font commands

`\bold` `\scr` These are simple aliases of core LaTeX commands.

```
\providecommand{\bold}{\mathbf}
\providecommand{\scr}{\mathcal}
```

`\germ` Since this doesn't correspond to a core LaTeX command, we generate an error if no appropriate definition is available.

```
\AtBeginDocument{%
    \@ifundefined{mathfrak}{%
        \providecommand{\germ}{%
            \PackageError{mathscinet}{To use the \string\germ\space
                command, please load the amsfonts package}\@ehc
        }%
```

```
    }{%
        \providecommand{\germ}{\mathfrak}%
    }%
}
```

\romsup The \tsup command comes from the textcmds package.
\asup

```
\providecommand{\romsup}{\tsup}
\providecommand{\asup}{\tsup}
```

\hslash Planck's constant over 2π: $\hbar$ (U+210F).

If the amssymb package isn't loaded, we just want this to be an alias for \hbar, which is defined in the LaTeX kernel (and then redefined by the amssymb package). If the amssymb package is loaded, we want to use its definition of \hslash. To prevent problems if amsrefs is loaded before amssymb, we defer our definition of \hslash until after all packages have been loaded.

```
\AtBeginDocument{\providecommand{\hslash}{\hbar}}
```

2.2 Arabic transliteration

\rasp Transliteration of the Arabic letter *hamza* (U+0621): ʾ (U+02BE).

```
\ProvideTextCommandDefault{\rasp}{\leavevmode\raise.45ex\hbox{$\rhook$}}
```

\lasp Transliteration of the Arabic letter *ain* (U+0639): ʿ (U+02BF).

```
\ProvideTextCommandDefault{\lasp}{\leavevmode\raise.45ex\hbox{$\lhook$}}
```

2.3 Latin Extended-A (latin1) characters

These are based on Barbara Beeton's definitions from amsclass.dtx.

\Dbar Latin capital letter D with stroke: Đ (U+0110). In the T1 encoding, we just use \DJ; otherwise we fake it.

```
\ProvideTextCommand{\Dbar}{T1}{\DJ}

\ProvideTextCommandDefault{\Dbar}{%
    \leavevmode\lower.5ex\rlap{\hskip-.07em\accent"16}D%
}
```

\dbar Latin lower letter d with stroke: đ (U+0111). In the T1 encoding, we just use \dj; otherwise we fake it.

```
\ProvideTextCommand{\dbar}{T1}{\dj}
```

If it looks like this is a small-caps font, we adjust the spacing appropriately.

```
\ProvideTextCommandDefault{\dbar}{%
    \begingroup
        \edef\@tempa{\scdefault}%
        \ifx\@tempa\f@shape
            \dimen@-.75ex
            \dimen@i-.08em
        \else
            \dimen@.02ex
```

```
            \dimen@i.1em
        \fi
        \leavevmode\raise\dimen@\rlap{\hskip\dimen@i\char"16}d%
    \endgroup
}
```

2.4 Cyrillic transliteration

\cprime Transliteration of the Cyrillic letter soft sign (U+042C): ′.

```
\ProvideTextCommandDefault{\cprime}{\tprime}
```

\cdprime Transliteration of the Cyrillic letter hard sign (U+042A): ″.

```
\ProvideTextCommandDefault{\cdprime}{\tprime\tprime}
```

\bud Ditto.

```
\ProvideTextCommandDefault{\bud}{\cdprime}
```

\cydot Vertically centered dot.

```
\ProvideTextCommandDefault{\cydot}{\leavevmode\raise.4ex\hbox{.}}
```

2.5 Miscellaneous diacritics (aka Frankenstein's diacritics)

\save@sf When putting together an accented character from bits and pieces, the \spacefactor of the base character often gets lost in the shuffle. We use essentialy the same technique as \add@accent to save and restore the spacefactor, but we wrap in in a pair of macros for convenience.

```
\def\save@sf{%
    \ifmmode\else\global\mathchardef\accent@spacefactor\spacefactor\fi
}
```

\restore@sf And here's the corresponding restore.

```
\def\restore@sf{\ifmmode\else\spacefactor\accent@spacefactor\fi}
```

\@underaccent This is perhaps the most interesting macro in this package (which admittedly isn't saying much). It attempts to convert a character (usually one of the standard above-letter diacritics like ¨) into an underhanging diacritic (like ¨̲). This is similar in spirit to the way that the OT1 \b command converts a macron into a bar-under accent, or the way that \d converts a period into an underhanging dot. However, the technique used here is a little more complicated and, hopefully, a little more general, in the sense of requiring fewer ad-hoc parameters. It only contains one magic constant (.2 ex), which seems to provide reasonable results for all of the Computer Modern fonts.

The basic algorithm is as follows:

1. Create a box B containing the base character at its natural height, depth, and width.
2. Create a box d consisting of the diacritic centered in a space equal to the width of B.

3. Lower box d by the sum of its height (to bring the top of d down to the baseline) plus the depth of B (to bring the top of d down to the bottom of B) plus .2 ex (to provide the spacing between the bottom of the letter and the top of the diacritic). Call the new box d'.
4. Create a new box C by superimposing boxes B and d'.
5. If the height of d was greater than 1 ex, reset the depth of box C to the sum of the depth of B and 1 ex less than the height of d. (See the appendix for a discussion of this step.)

```
\def\@underaccent#1#2#3{%
    \leavevmode
    \begingroup
        \ifmmode\let\@mathtoggle$\else\let\@mathtoggle\relax\fi
        \setboxz@h{\@mathtoggle#3\save@sf\@mathtoggle}%
        \setbox\@ne\hb@xt@\wd\z@{%
            \hss\fontshape\updefault\rmfamily#1\char#2\hss
        }%
        \dimen@\ht\@ne
        \advance\dimen@\dp\z@
        \advance\dimen@.2ex
        \setboxz@h{\lower\dimen@\rlap{\copy\@ne}\unhbox\z@}%
        \ifdim\ht\@ne>1ex
            \advance\dimen@-1.2ex
            \dp\z@\dimen@
        \fi
        \box\z@
        \restore@sf
    \endgroup
}
```

`\utilde` Tilde below (`U+0330`):

`\utilde{E}`	Ḛ	`U+1E1A`	`\utilde{e}`	ḛ	`U+1E1B`
`\utilde{I}`	Ḭ	`U+1E2C`	`\utilde{i}`	ḭ	`U+1E2D`
`\utilde{U}`	Ṵ	`U+1E74`	`\utilde{u}`	ṵ	`U+1E75`

```
\DeclareTextCommandDefault{\utilde}{\@underaccent\@empty{‘\~}}
```

`\uarc` Breve below (`U+032E`):

`\uarc{H}`	Ḫ	`U+1E2A`	`\uarc{h}`	ḫ	`U+1E2B`

```
\DeclareTextCommandDefault{\uarc}{\@underaccent\@empty{’025}}
```

`\lfhook` Comma below (`U+0326`):

`\lfhook{S}`	Ș	`U+0218`	`\lfhook{s}`	ș	`U+0219`
`\lfhook{T}`	Ț	`U+021A`	`\lfhook{t}`	ț	`U+021B`

```
\DeclareTextCommandDefault{\lfhook}{\@underaccent\supsize{‘\,}}
```

`\dudot` Diaeresis below (`U+0324`):

`\dudot{U}`	Ṳ	`U+1E72`	`\dudot{u}`	ṳ	`U+1E73`

```
\DeclareTextCommandDefault{\dudot}{\@underaccent\@empty{’177}}
```

\udot Dot below (U+0323): There are two options for implementing this: either map to the standard \d accent or define using \@underaccent. If we choose the former, we have two problems: (a) when applied to capital letters, the standard T1 and OT1 implementations of \d produce \spacefactors of 1000 instead of 999, and (b) the underspacing of the \udot accent will differ from that of the \dudot accent: x̤x̱. On the other hand, if we choose the latter, course, \d and \udot will differ: x̤x̤. Neither solution appeals, but it's easier to stick with \d, so that's what I'll do.

```
\DeclareTextCommandDefault{\udot}{\d}
```

\polhk Ogonek (Polish hook) (U+0328):

\polhk{A}	Ą	U+0104	\polhk{a}	ą	U+0105
\polhk{E}	Ę	U+0118	\polhk{e}	ę	U+0119
\polhk{I}	Į	U+012E	\polhk{i}	į	U+012F
\polhk{U}	Ų	U+0172	\polhk{u}	ų	U+0173
\polhk{O}	Ǫ	U+01EA	\polhk{o}	ǫ	U+01EB

The T1 and OT4 encodings implement the \k accent, so we just use it for most characters, although we will supplement them later with

```
\DeclareTextCommand{\polhk}{OT4}{\k}
\DeclareTextCommand{\polhk}{T1}{\k}

\DeclareTextCommand{\polhk}{OT1}[1]{\TextSymbolUnavailable{\k{#1}}#1}

\DeclareTextCompositeCommand{\polhk}{OT1}{a}{\msc@ogonek {.6}{.07} a}
\DeclareTextCompositeCommand{\polhk}{OT1}{A}{\msc@ogonek {.6}{.07} A}
\DeclareTextCompositeCommand{\polhk}{OT1}{e}{\msc@ogonek   0 {.06} e}
\DeclareTextCompositeCommand{\polhk}{OT1}{E}{\msc@ogonek{.35}{.07} E}
\DeclareTextCompositeCommand{\polhk}{OT1}{i}{\msc@ogonek {.2}{.07} i}
\DeclareTextCompositeCommand{\polhk}{OT1}{I}{\msc@ogonek {.2}{.07} I}
\DeclareTextCompositeCommand{\polhk}{OT1}{u}{\msc@ogonek {.6}{.07} u}
\DeclareTextCompositeCommand{\polhk}{OT1}{U}{\msc@ogonek   0 {.05} U}
\DeclareTextCompositeCommand{\polhk}{OT1}{o}{\msc@ogonek   0 {.07} o}
\DeclareTextCompositeCommand{\polhk}{OT1}{O}{\msc@ogonek   0 {.05} O}

\DeclareTextCompositeCommand{\polhk}{T1}{i}{\msc@ogonek@a   0  i}
\DeclareTextCompositeCommand{\polhk}{T1}{I}{\msc@ogonek@a   0  I}
\DeclareTextCompositeCommand{\polhk}{T1}{u}{\msc@ogonek@a {.6} u}
\DeclareTextCompositeCommand{\polhk}{T1}{U}{\msc@ogonek@a   0  U}
\DeclareTextCompositeCommand{\polhk}{T1}{o}{\msc@ogonek@a   0  o}
\DeclareTextCompositeCommand{\polhk}{T1}{O}{\msc@ogonek@a   0  O}

\DeclareTextCompositeCommand{\polhk}{OT4}{i}{\msc@ogonek {.2}{.07} i}
\DeclareTextCompositeCommand{\polhk}{OT4}{I}{\msc@ogonek {.2}{.07} I}
\DeclareTextCompositeCommand{\polhk}{OT4}{u}{\msc@ogonek {.6}{.07} u}
\DeclareTextCompositeCommand{\polhk}{OT4}{U}{\msc@ogonek   0 {.05} U}
\DeclareTextCompositeCommand{\polhk}{OT4}{o}{\msc@ogonek   0 {.07} o}
\DeclareTextCompositeCommand{\polhk}{OT4}{O}{\msc@ogonek   0 {.05} O}
```

\msc@ogonek

```
\def\msc@ogonek#1#2#3{%
```

```
        \begingroup
            \setboxz@h{#3\save@sf}%
            \dimen@\wd\z@
            \ooalign{%
                \unhbox\z@\crcr
                \hidewidth
                    \setboxz@h{\kern#1\dimen@\supsize$\lhook$}%
                    \dimen@\ht\z@
                    \advance\dimen@-#2ex\relax
                    \lower\dimen@\box\z@
                \hidewidth
            }%
            \restore@sf
        \endgroup
    }
```

`\msc@ogonek@a`

```
    \def\msc@ogonek@a#1#2{%
        \begingroup
            \ooalign{%
                #2\save@sf\crcr
                \hidewidth
                    \raise0.02ex\hbox{\kern#1ex\char'014}%
                \hidewidth
            }%
            \restore@sf
        \endgroup
    }
```

`\soft` Polish 'soft' letters T, D, L:

`\soft{T}`	Ť	`U+0164`	`\soft{t}`	ť	`U+0165`
`\soft{D}`	Ď	`U+010E`	`\soft{d}`	ď	`U+010f`
`\soft{L}`	Ľ	`U+013B`	`\soft{l}`	ľ	`U+013C`

```
    \DeclareTextCommand{\soft}{OT4}{\v}
    \DeclareTextCommand{\soft}{T1}{\v}
    \DeclareTextCommand{\soft}{OT1}{\v}

    \DeclareTextCompositeCommand{\soft}{OT1}{t}{\msc@soft{t}\@ne{.5ex}}
    \DeclareTextCompositeCommand{\soft}{OT1}{d}{\msc@soft{d}{.925}{.95ex}}
    \DeclareTextCompositeCommand{\soft}{OT1}{l}{\msc@soft{l}{.95}{.4ex}}
    \DeclareTextCompositeCommand{\soft}{OT1}{L}{\msc@soft{L}{.975}{.8ex}}

    \DeclareTextCompositeCommand{\soft}{OT4}{t}{\msc@soft{t}\@ne{.5ex}}
    \DeclareTextCompositeCommand{\soft}{OT4}{d}{\msc@soft{d}{.925}{.95ex}}
    \DeclareTextCompositeCommand{\soft}{OT4}{l}{\msc@soft{l}{.95}{.4ex}}
    \DeclareTextCompositeCommand{\soft}{OT4}{L}{\msc@soft{L}{.975}{.8ex}}
```

`\msc@soft`

```
    \def\msc@soft#1#2#3{%
```

```
    \leavevmode
    \begingroup
        \setboxz@h{#1}%
        \raise#2\ht\z@\rlap{\kern#3\supsize,}\unhbox\z@
    \endgroup
}
```

Appendix: Plumbing the depths of underhanging diacritics: Notes on the magic constant 0.2 ex and an exegesis of certain obscure corners of the `\@underaccent` macro

TEX assumes that a combining accent can be superimposed directly on top of any character whose height is 1 ex. If the actual height of the base character differs from 1 ex, the accent is shifted up or down to maintain the same vertical separation. Put another way, subtracting 1 ex from the height of a combining accent tells us how much a character's height is increased when that accent is added. Call this distance δ.

This distance can be analyzed into two pieces: $\delta = \sigma + \eta$, where σ is the separation between the top of the base character and the *bottom* of the accent, and η is the height of the bounding box of the accent.

If we now consider moving the accent to the bottom of a character, we have a new relation $\delta' = \sigma' + \eta$, where σ' is the distance between the bottom of the base and the *top* of the accent, and δ' is the amount by which the depth of the base character is increased.

If the placement of the underhanging version of an accent were strictly symmetrical with the overhanging version, then we would have $\sigma' = \sigma$ and thus $\delta' = \delta$. However, the placement is not symmetric since for aesthetic reasons we have chosen to set $\sigma' = .2$ ex, which seems in general to be a little less than the corresponding σ. Unfortunately, this means we can't calculate δ' accurately since there is no way to deduce η (or, equivalently, σ) from the font metric information available to us within TEX.[1]

This leaves us in a bit of a bind, since it means there is no way to calculate the depth accurately. In despair, I've decided just to pretend that $\delta' = \delta$ for now.

Finis

The usual `\endinput` to ensure that random garbage at the end of the file doesn't get copied by `docstrip`.

```
\endinput
```

[1] By exploiting symmetries present in some fonts, one can calculate η for some of the accents in those fonts, but this doesn't help us much since both σ and η vary between accents.