1 Using MetaCS

MetaCS provides commands for running and describing commands.

2 Running commands

2.1 \cc – run a parameterless command (switch)

A switch is a control sequence (command) with no arguments, e.g. `\itshape` which switches the current font to italic shape.

Format: `\cc{`⟨*command-name*⟩`}`, where ⟨*command-name*⟩ is without the backslash.

Example:

```
{\cc {itshape} This is italic.}
```
↦ *This is italic.*

How it works: `\cc` uses `\cs:w #1\cs_end:` to construct the control sequence expandably, add the backslash (\) at the front, and then run it.

If the resulting control sequence does not exist, it is no error.

```
\cc {xyz}
```
↦

Items passed in to the construction are expanded.

```
\newcommand \x {o}\newcommand \xxo {abc}\cc {xx\x }
```
↦ abc

Reference:

```
\cs:w #1\cs_end:
```

2.2 \cd – run a 1-argument command

`\cd` runs commands that take one argument, like `\textbf {Some text}`.

Format: `\cd{`⟨*command-name*⟩`}{`⟨*argument*⟩`}`, where ⟨*command-name*⟩ is the control sequence name without the backslash.

Example:

```
\cc {textbf}{Some text}
```
↦ **Some text**

```
\cd {textbf}{Some more text}
```
↦ **Some more text**

How it works: `\cd` builds on `\cc`'s method of constructing a command name by adding an argument.

It is no compilation error if the constructed command is not defined.

```
\cc {textbff}{Some text}
```
↦ Some text

Reference:

```
\tl_set:Nn \l_my_tl { #2 }
\cs:w #1\cs_end: { \tl_use:N \l_my_tl }
```

2.3 \cdr – print and run code

\cdr prints and runs whatever code is passed to it.

FORMAT: **\cdr**{⟨*code*⟩}

EXAMPLE:

```
\cdr {\sffamily \large \textsc {Some text}} ↦ \sffamily
\large \textsc {Some text} ↦ SOME TEXT
```

HOW IT WORKS: **\cdr** uses `\detokenize` to print its argument #1, then leaves argument #1 in the input stream.

REFERENCE: In effect

```
\detokenize{#1}
$\mapsto$
#1
```

2.4 \cdrq – print and run code in quotation environment

\cdrq does a **\cdr** inside a `\begin{quotation}` …`\end{quotation}` environment.

FORMAT: **\cdrq**{⟨*code*⟩}

EXAMPLE:

```
\cdrq {\ttfamily \tiny \color {red} Example} ↦
    \ttfamily \tiny \color {red} Example ↦ Example
```

EXAMPLE:

```
\cdrq {\textsf {\textit {Another \colorbox {blue!20}{example}}}} ↦
    \textsf {\textit {Another \colorbox {blue!20}{example}}} ↦ Another
example
```

2.5 \cdq – print code in quotation environment

\cdq prints code inside a `\begin{quotation}` …`\end{quotation}` environment.

FORMAT: **\cdq**{⟨*code*⟩}

EXAMPLE:

```
\cdq {\ttfamily \tiny \color {red} Example} ↦
    \ttfamily \tiny \color {red} Example
```

EXAMPLE:

```
\cdq {\textsf {\textit {Another \colorbox {blue!20}{example}}}} ↦
    \textsf {\textit {Another \colorbox {blue!20}{example}}}
```

2.6 \cdrd – print formatted code

\cdrd prints the code it is given. Variant typesetting formats, if desired, are selected via a key-value option. If no option is specified, then the plain style is used.

FORMAT: **\cdrd** [⟨*style-option*⟩] [⟨*option2*⟩] {⟨*code*⟩}

EXAMPLE: The heading for this subsection was produced with

```
\cdrd [format=subsection][print formatted code]{cdrd}
```

HOW IT WORKS: Values of the [format=] option key determine what formatting is applied.

List of values for the `format=` key:

Value	Example
⟨*None*⟩	`\cdrd {\xyz } ↦ \xyz`
head	`\cdrd [format=head]{\xyz } ↦` \xyz
custom	`\cdrd [format=custom][\bfseries \tiny \sffamily ]{\xyz } ↦` \xyz
section	See outside the table.
subsection	Similar to section
quote	`\cdrd [format=quote]{xyz demo This is a quote}`: See outside the table.
general	`\cdrd [format=general]{[format=xyz]} ↦ [format=xyz]`
detok	`\cdrd [format=detok]{\xyz } ↦ \xyz`

2.6.1 \cdrd: [format=general] key

This key runs the **\codegeneral** macro, which prints what it is given in `\ttfamily` format.

EXAMPLE:

```
\cdrd [format=general]{sample plain text} ↦ sample plain
text
```

2.6.2 \cdrd: [format=detok] key

This key runs the **\codedetok** macro, which detokenizes what it is given and then prints it in `\ttfamily` format.

EXAMPLE:

```
\cdrd [format=detok]{\bfseries sample bold text} ↦ \bfseries
sample bold text
```

2.6.3 \cdrd: [format=head] key

This key runs the **\cdrdhead** macro, which formats the command something like a heading.

EXAMPLE:

```
\cdrd [format=head]{\xyz } ↦ \xyz
```

2.6.4 \cdrd: [format=custom] key

This key runs the \cdrdcustom macro, which formats the command according to what was given to [⟨*option2*⟩] as switches.

EXAMPLE:

```
\cdrd [format=custom][\bfseries \tiny \sffamily ]{\xyz
} ↦ \xyz
```

2.6.5 \cdrd: [format=section] key

This key runs the \cdrdsection macro, which does \section{#3 -- #2} \label { sec:#3} in terms of the parameters received from the main function.

REFERENCE:

```
\group_begin:
\keys_set:nn { codef } { #1 }
\tl_use:N \l_codeformat_tl [ #2 ] { #3 }
\group_end:
```

codef receives the format option as #1 and assigns the corresponding macro to \l_codeformat_tl.

The codef key set is defined as follows:

```
\tl_new:N  \l_codeformat_tl
\tl_set:Nn \l_codeformat_tl { \cdrdplain }

\keys_define:nn { codef }
 {
  format .choice:,
  format / head   .code:n = \tl_set:Nn \l_codeformat_tl  { \cdrdhead },
  format / custom   .code:n = \tl_set:Nn \l_codeformat_tl  { \cdrdcustom
}, %user-code will be second option
  format / section   .code:n = \tl_set:Nn \l_codeformat_tl  { \
cdrdsection },
  format / subsection   .code:n = \tl_set:Nn \l_codeformat_tl  { \
cdrdsubsection },
  format / quote   .code:n = \tl_set:Nn \l_codeformat_tl  { \cdrdquote },
  format / listing   .code:n = \tl_set:Nn \l_codeformat_tl  { \
cdrdlisting },
  format / general   .code:n = \tl_set:Nn \l_codeformat_tl  { \
codegeneral },
  format / detok   .code:n = \tl_set:Nn \l_codeformat_tl  { \codedetok },
   }
```

[latexcode env from DOCTOOLS[2012] package, adapted]

2.6.6 \cdrd: [format=subsection] key

This key runs the \cdrdsubsection macro, which does the same as \cdrdsection but for subsections. See §??.

2.6.7 \cdrd: [format=quote] key

This key runs the \cdrdquote macro, which runs the \cdrdplain macro inside a \begin{quotation} …\end{quotation} environment.

EXAMPLE:

```
\cdrd [format=quote]{\xyz inside quote environment} ↦
    \xyz inside quote environment
```

2.6.8 \cdrd: no format key

If no format key is specified, the \cdrdplain macro runs, which just does a \detokenize formatted as ttfamily large blue text.

EXAMPLE:

```
\cdrd {\xyz plain formatting} ↦ \xyz plain formatting
```

2.7 \cdpr – print and run command + formatted argument

\cdpr prints, then runs any code given as #1 before the argument (given as #3) of the command (given as #2). The pre-argument insertion can be arbitrary code, but is intended to be formatting code, e.g., for emphasis, or for font commands for a different script to make the argument visible like it would be in a code editor window.

FORMAT: \cdrq[⟨*format-code*⟩]{⟨*command-name*⟩}{⟨*argument*⟩}

EXAMPLE: Argument is red bold

```
\cdpr [\bfseries \color {red}]{textit}{fgh mno} ↦ \textit{fgh
mno} ↦ fgh mno
```

EXAMPLE: Argument is visible Ugaritic glyphs

```
\cdpr [\ugfontb ]{sugtransrev}{  } ↦ \sugtransrev{𐎍𐎐𐎗} ↦ dlt
```

3 Control Sequence Meta Commands

The parts of a control sequence can be referred to in text using metacommands.

3.1 Meta Commands for emphasis

The following table lists the metacommands that can be used for emphasis.

Meta command	Example	Result
\cs{⟨*macro-name*⟩}	\cs {\test }	\test
\marg{⟨*mandatory argument*⟩}	\marg {test}	{⟨*test*⟩}
\margv{⟨*mandatory argument value*⟩}	\margv {test}	{test}
\oarg{⟨*optional argument*⟩}	\oarg {test}	[⟨*test*⟩]
\oargv{⟨*optional argument value*⟩}	\oargv {test}	[test]
\meta{⟨*meta value*⟩}	\meta {test}	⟨*test*⟩

3.2 Unemphasized Meta Commands

The following table lists the metacommands that can be used for unemphasized text.

Unemphasized Meta command	Example	Result
\css{⟨*cs-name*⟩}	\css {test}	\test
\margcss{⟨*mandatory argument*⟩}	\margcss {test}	{⟨*test*⟩}
\margvcss{⟨*mandatory argument value*⟩}	\margvcss {test}	{test}
\oargcss{⟨*optional argument*⟩}	\oargcss {test}	[⟨*test*⟩]
\oargvcss{⟨*optional argument value*⟩}	\oargvcss {test}	[test]
\meta{⟨*meta value*⟩}	\meta {test}	⟨*test*⟩

4 Examples

The \textit{} command is used on a run of inline text to produce italics.

EXAMPLE:

```
some text \textit {(a) some italic text} some more text.
```
↦ some text *(a) some italic text* some more text.

The \itshape switch switches the font over to italics. Its scope is restricted by the use of braces to form a group, or reversed by the \upshape switch.

EXAMPLE: braces

```
some text {\itshape (b) some italic text} some more text.
```
↦ some text *(b) some italic text* some more text.

EXAMPLE: \upshape

```
some text \itshape (c) some italic text \upshape some more text.
```
↦ some text *(c) some italic text* some more text.

The \cc{} command runs a switch.

EXAMPLE:

```
{ \cc {itshape}\cc {large} 123 }
```
↦ *123*

The \cd{} command runs a command plus argument.

EXAMPLE:

```
\cd {textit}{Sample}
```
↦ *Sample*

The \cdrd{} command prints its argument, and using its option-key settings, \cdrd[format=??]{}, can print in various formats.

EXAMPLE: plain style

```
\cdrd {\textit {Sample}}
```
↦ `\textit {Sample}`

EXAMPLE: heading style

`\cdrd [format=head]{\textit }` ↦ `\textit`

EXAMPLE: code style

`\cdrd [format=general]{define function x; print(x); }` ↦ `define function x; print(x);`

And so on.

A code listing is done with the `\begin {latexcode}` …`\end {latexcode}` envronment.

EXAMPLE:

```
\ee plain style
\cdrq{\cdrd{\textit{Sample}}}
\ee heading style
\cdrq{\cdrd[format=head]{\textit}}
\ee code style
\cdrq{\cdrd[format=general]{define function x; print(x); }}
```

[latexcode env from DOCTOOLS[2012] package, adapted]