

Latex mark-up

/Reference manual

1. Contents of this page

- Introduction
- ISO Standard mark-up
 - White space and comments
 - Mark-up directives
 - Greek letters
 - Other letters
 - Special characters
 - Symbol characters
 - Core words
 - Section header mark-up
 - Paragraph mark-up
- CADiZ-specific mark-up
 - Greek letters

1 Contents of this page

2 Introduction

3 ISO Standard...

3.1 White space and...

3.2 Mark-up directives

3.3 Greek letters

3.4 Other letters

3.5 Special characters

3.6 Symbol characters

3.7 Core words

3.8 Section header...

3.9 Paragraph mark-up

3.9.1 Axiomatic...

3.9.2 Generic...

3.9.3 Schema...

3.9.4 Generic...



Page 2 of 18



Go Back

Full Screen

Close

Quit

- Symbol characters
- Core words
- Paragraph mark-up
- Tool directives

2. Introduction

A \LaTeX mark-up for Z specifications is defined here by its conversion to a sequence of Z (UCS) characters. Cadiz performs this conversion to [Z characters](#), which it then [lexes](#). Only the core Z notation is considered here; the \LaTeX mark-up of symbols defined in the toolkit is documented [elsewhere](#).

3. ISO Standard mark-up

3.1. White space and comments

Spaces, tabs and newlines are soft space: they separate tokens of the mark-up, without generating any space characters to be lexed. Braces are also soft space, affecting the interpretation of neighbouring mark-up (unless escaped with \backslash).

The following mark-up is converted to spaces to be lexed.

1 Contents of this page

2 Introduction

3 ISO Standard...

3.1 White space and...

3.2 Mark-up directives

3.3 Greek letters

3.4 Other letters

3.5 Special characters

3.6 Symbol characters

3.7 Core words

3.8 Section header...

3.9 Paragraph mark-up

3.9.1 Axiomatic...

3.9.2 Generic...

3.9.3 Schema...

3.9.4 Comments

| L^AT_EX command | Rendering | Z character |
|--|----------------------|--------------------|
| <code>~</code> | interword space | <i>SPACE</i> |
| <code>\,</code> | thin space | <i>SPACE</i> |
| <code>\:</code> | medium space | <i>SPACE</i> |
| <code>\;</code> | thick space | <i>SPACE</i> |
| <code>\(space)</code> | interword space | <i>SPACE</i> |
| <code>\\</code> | newline | <i>NLCHAR</i> |
| <code>\t1</code> | tab stop 1 | <i>SPACE</i> |
| <code>\t2</code> | tab stop 2 | <i>SPACE</i> |
| <code>\t3</code> | tab stop 3 | <i>SPACE</i> |
| <code>\t4</code> | tab stop 4 | <i>SPACE</i> |
| <code>\t5</code> | tab stop 5 | <i>SPACE</i> |
| <code>\t6</code> | tab stop 6 | <i>SPACE</i> |
| <code>\t7</code> | tab stop 7 | <i>SPACE</i> |
| <code>\t8</code> | tab stop 8 | <i>SPACE</i> |
| <code>\t9</code> | tab stop 9 | <i>SPACE</i> |
| <code>\also</code> | small vertical space | <i>NLCHAR</i> |
| <code>\znewpage</code> | new page | <i>NLCHAR</i> |

A % character in the mark-up introduces a comment: all text from that % to the end of the line, and any following spaces and tabs at the beginning of the next line, are ignored.

1 Contents of this page

2 Introduction

3 ISO Standard...

3.1 White space and...

3.2 Mark-up directives

3.3 Greek letters

3.4 Other letters

3.5 Special characters

3.6 Symbol characters

3.7 Core words

3.8 Section header...

3.9 Paragraph mark-up

3.9.1 Axiomatic...

3.9.2 Generic...

3.9.3 Schema...

3.9.4 General...

3.2. Mark-up directives

Each `\LaTeXcommand` needs to have a mark-up directive to define its conversion to a sequence of Z characters. These mark-up directives are usually written immediately before the definition of the corresponding Z symbol. For example, those for toolkit symbols are in toolkit sections, and those for the core notation are in the prelude section. Hence you have to write mark-up directives only for new symbols that you define in your sections of your Z specification. The conversions defined by these mark-up directives should be consistent with the rendering produced by the \LaTeX typesetting tool. It is your responsibility to ensure that the rendering by \LaTeX is consistent with the conversion by CADiZ, as required by the Z standard.

The conversion of a `\LaTeXcommand` to a specific UCS character is defined by a mark-up directive like the following.

```
%%Zchar \Delta U+0394
%%Zchar \arithmos U-0001D538
```

If the `\LaTeXcommand` is to be used as an operator, spaces will be needed in its conversion according to its fixity.

```
%%Zinchar \rel U+2194
%%Zprechar \finset U-0001D53D
%%Zpostchar \rangle U+3009
```

1 Contents of this page

2 Introduction

3 ISO Standard...

3.1 White space and...

3.2 Mark-up directives

3.3 Greek letters

3.4 Other letters

3.5 Special characters

3.6 Symbol characters

3.7 Core words

3.8 Section header...

3.9 Paragraph mark-up

3.9.1 Axiomatic...

3.9.2 Generic...

3.9.3 Schema...

3.9.4 Conventions...

If a `\LaTeXcommand` is to be converted to more than one character, a directive like the following is used.

```
%%Zinword \partition partition
```

```
%%Zpreword \dom dom
```

```
%%Zpostword \star ^*
```

In the following sections, mark-up directives are given to define the conversions of all the `\LaTeXcommands` for the core notation. Any core notation for which no `\LaTeXcommand` is defined is ASCII and is marked-up as itself.

3.3. Greek letters

The Greek letters of the core Z language are converted according to the following mark-up directives.

| Z character | \LaTeX mark-up |
|-------------|-------------------------------------|
| Δ | <code>%%Zchar \Delta U+0394</code> |
| Ξ | <code>%%Zchar \Xi U+039E</code> |
| θ | <code>%%Zchar \theta U+03B8</code> |
| λ | <code>%%Zchar \lambda U+03BB</code> |
| μ | <code>%%Zchar \mu U+03BC</code> |

1 Contents of this page

2 Introduction

3 ISO Standard...

3.1 White space and...

3.2 Mark-up directives

3.3 Greek letters

3.4 Other letters

3.5 Special characters

3.6 Symbol characters

3.7 Core words

3.8 Section header...

3.9 Paragraph mark-up

3.9.1 Axiomatic...

3.9.2 Generic...

3.9.3 Schema...

3.9.4 Comments

3.4. Other letters

Other letters of the core Z language are converted according to the following mark-up directives.

| Z character | L ^A T _E X mark-up |
|--------------|---|
| \mathbb{A} | <code>%%Zchar \arithmos U-0001D538</code> |
| \mathbb{N} | <code>%%Zchar \nat U+2115</code> |
| \mathbb{P} | <code>%%Zprechar \power U+2119</code> |

3.5. Special characters

Special characters of the core Z language are converted according to the following mark-up directives.

| Z character | L ^A T _E X mark-up |
|------------------|---|
| $-$ | <code>%%Zchar _ U+005F</code> |
| $\{$ | <code>%%Zchar \{ U+007B</code> |
| $\}$ | <code>%%Zchar \} U+007D</code> |
| $\langle\langle$ | <code>%%Zchar \ldata U+300A</code> |
| $\rangle\rangle$ | <code>%%Zchar \rdata U+300B</code> |
| \lfloor | <code>%%Zchar \lblot U+2989</code> |
| \rfloor | <code>%%Zchar \rblot U+298A</code> |

Superscripts and subscripts are marked-up as follows.

1 Contents of this page

2 Introduction

3 ISO Standard...

3.1 White space and...

3.2 Mark-up directives

3.3 Greek letters

3.4 Other letters

3.5 Special characters

3.6 Symbol characters

3.7 Core words

3.8 Section header...

3.9 Paragraph mark-up

3.9.1 Axiomatic...

3.9.2 Generic...

3.9.3 Schema...

3.9.4 Generic...

L^AT_EX mark-up

 \wedge ⟨single L^AT_EX token⟩

 $\wedge\{\langle$ L^AT_EX tokens⟩ $\}$
 $_$ ⟨single L^AT_EX token⟩

 $_ \{ \langle$ L^AT_EX tokens⟩ $\}$

Z characters

 \nearrow ⟨Z character sequence⟩ \swarrow
 \nearrow ⟨Z character sequence⟩ \swarrow
 \searrow ⟨Z character sequence⟩ \nwarrow
 \searrow ⟨Z character sequence⟩ \nwarrow

For example, \exists_1 is marked-up as `\existsists_1`.

The L^AT_EX mark-up from which box characters are converted is discussed [below](#).

The L^AT_EX mark-up for *NLCHAR* and *SPACE* characters is discussed [above](#).

3.6. Symbol characters

Symbol characters of the core Z language are converted according to the following mark-up directives.

1 Contents of this page

2 Introduction

3 ISO Standard...

3.1 White space and...

3.2 Mark-up directives

3.3 Greek letters

3.4 Other letters

3.5 Special characters

3.6 Symbol characters

3.7 Core words

3.8 Section header...

3.9 Paragraph mark-up

3.9.1 Axiomatic...

3.9.2 Generic...

3.9.3 Schema...

3.9.4 Generic...

1 Contents of this page

2 Introduction

3 ISO Standard...

3.1 White space and...

3.2 Mark-up directives

3.3 Greek letters

3.4 Other letters

3.5 Special characters

3.6 Symbol characters

3.7 Core words

3.8 Section header...

3.9 Paragraph mark-up

3.9.1 Axiomatic...

3.9.2 Generic...

3.9.3 Schema...

| Z character | L ^A T _E X mark-up |
|-------------|---|
| ⊢ | %%Zchar \vdash U+22A2 |
| ∧ | %%Zinchar \land U+2227 |
| ∨ | %%Zinchar \lor U+2228 |
| ⇒ | %%Zinchar \implies U+21D2 |
| ⇔ | %%Zinchar \iff U+21D4 |
| ¬ | %%Zprechar \lnot U+00AC |
| ∀ | %%Zprechar \forall U+2200 |
| ∃ | %%Zprechar \exists U+2203 |
| × | %%Zinchar \cross U+00D7 |
| ∈ | %%Zinchar \in U+2208 |
| \ | %%Zinchar \hide U+29F9 |
| ⌋ | %%Zinchar \project U+2A21 |
| ⁰ | %%Zinchar \semi U+2A1F |
| >> | %%Zinchar \pipe U+2A20 |

The • character is marked-up as an at character (not formalized here due to typesetting problems).

3.7. Core words

Core words may be marked-up using the following commands, so that neighbouring spaces are converted implicitly.

| Z characters | L ^A T _E X mark-up |
|--------------|---|
| if | %%Zpreword \IF if |
| then | %%Zinword \THEN then |
| else | %%Zinword \ELSE else |
| let | %%Zpreword \LET let |
| pre | %%Zpreword \pre pre |
| function | %%Zinword \function function |
| generic | %%Zinword \generic generic |
| relation | %%Zinword \relation relation |
| leftassoc | %%Zinword \leftassoc leftassoc |
| rightassoc | %%Zinword \rightassoc rightassoc |
| , , | %%Zinword \listarg , , |
| – | %%Zinword \varg – |

3.8. Section header mark-up

Section headers are enclosed in a L^AT_EX `zsection` environment.

```
\begin{zsection}
\SECTION name \parents ...
\end{zsection}
```

These `\begin` and `\end` brackets are converted to *ZEDCHAR* and *ENDCHAR* respectively.

1 Contents of this page

2 Introduction

3 ISO Standard ...

3.1 White space and ...

3.2 Mark-up directives

3.3 Greek letters

3.4 Other letters

3.5 Special characters

3.6 Symbol characters

3.7 Core words

3.8 Section header ...

3.9 Paragraph mark-up

3.9.1 Axiomatic ...

3.9.2 Generic ...

3.9.3 Schema ...

3.9.4 Generic ...

1 Contents of this page

2 Introduction

3 ISO Standard . . .

3.1 White space and . . .

3.2 Mark-up directives

3.3 Greek letters

3.4 Other letters

3.5 Special characters

3.6 Symbol characters

3.7 Core words

3.8 Section header . . .

3.9 Paragraph mark-up

3.9.1 Axiomatic . . .

3.9.2 Generic . . .

3.9.3 Schema . . .

3.9.4 Generic . . .

Within a `zsection` environment, the only \LaTeX commands that are converted are those of white space, `_`, and the following.

Z characters \LaTeX mark-up

section $\%Z\text{preword} \backslash\text{SECTION}$ section

parents $\%Z\text{inword} \backslash\text{parents}$ parents

3.9. Paragraph mark-up

Each formal Z paragraph is enclosed in a \LaTeX environment, the `\begin` being converted to box characters (*ZEDCHAR* if the paragraph has no outline), and the `\end` being converted to an *ENDCHAR* character. Any middle line in an outlined paragraph may be marked-up using the `\where` command, which is converted to a `|` character with *SPACE* characters around it.

For each outlined paragraph, its mathematical representation, Z characters, and \LaTeX mark-up are given below.

3.9.1. Axiomatic description paragraph mark-up

Mathematical representation

| | |
|--|------------------|
| | <i>DeclPart</i> |
| | <i>Predicate</i> |

Z characters
AXCHAR

DeclPart

|

Predicate

ENDCHAR

Latex mark-up

`\begin{axdef}`

DeclPart

`\where`

Predicate

`\end{axdef}`

3.9.2. Generic axiomatic description paragraph mark-up

Mathematical representation



Z characters
GENCHAR AXCHAR [Formals]

1 Contents of this page

2 Introduction

3 ISO Standard...

3.1 White space and...

3.2 Mark-up directives

3.3 Greek letters

3.4 Other letters

3.5 Special characters

3.6 Symbol characters

3.7 Core words

3.8 Section header...

3.9 Paragraph mark-up

3.9.1 Axiomatic...

3.9.2 Generic...

3.9.3 Schema...

3.9.4 Generic...



Go Back

Full Screen

Close

Quit

```
DeclPart
|
Predicate
ENDCHAR
```

Latex mark-up

```
\begin{gendef}[Formals]
DeclPart
\where
Predicate
\end{gendef}
```

3.9.3. Schema definition paragraph mark-up

Mathematical representation

| | |
|------------------|-------|
| <i>NAME</i> | _____ |
| <i>DeclPart</i> | _____ |
| <i>Predicate</i> | _____ |

```
Z characters
SCHCHAR NAME
```

1 Contents of this page

2 Introduction

3 ISO Standard...

3.1 White space and...

3.2 Mark-up directives

3.3 Greek letters

3.4 Other letters

3.5 Special characters

3.6 Symbol characters

3.7 Core words

3.8 Section header...

3.9 Paragraph mark-up

3.9.1 Axiomatic...

3.9.2 Generic...

3.9.3 Schema...

3.9.4 Generic...



Go Back

Full Screen

Close

Quit

```
DeclPart
|
Predicate
ENDCHAR
```

Latex mark-up

```
\begin{schema}{NAME}
DeclPart
\where
Predicate
\end{schema}
```

3.9.4. Generic schema definition paragraph mark-up

Mathematical representation

| | |
|-----------------------|--|
| <i>NAME</i> [Formals] | |
| <i>DeclPart</i> | |
| <i>Predicate</i> | |

```
Z characters
GENSCH NAME [Formals]
```

1 Contents of this page

2 Introduction

3 ISO Standard...

3.1 White space and...

3.2 Mark-up directives

3.3 Greek letters

3.4 Other letters

3.5 Special characters

3.6 Symbol characters

3.7 Core words

3.8 Section header...

3.9 Paragraph mark-up

3.9.1 Axiomatic...

3.9.2 Generic...

3.9.3 Schema...

3.9.4 Generic...



Page 14 of 15



Go Back

Full Screen

Close

Quit

```
DeclPart
|
Predicate
END
```

Latex mark-up

```
\begin{schema}{NAME}[Formals]
DeclPart
\where
Predicate
\end{schema}
```

4. CADiZ-specific mark-up

CADiZ can convert the following additional \LaTeX mark-up. This is all defined in the prelude section, and is not noticed by the `-ws` option. *CADiZ would conform better to the standard if this mark-up were introduced in a separate CADiZ-specific section.*

4.1. Greek Letters

CADiZ recognises the usual \LaTeX mark-up of the whole of the rest of the Greek alphabet. Each is defined by a `%%Zchar` directive.

1 Contents of this page

2 Introduction

3 ISO Standard...

3.1 White space and...

3.2 Mark-up directives

3.3 Greek letters

3.4 Other letters

3.5 Special characters

3.6 Symbol characters

3.7 Core words

3.8 Section header...

3.9 Paragraph mark-up

3.9.1 Axiomatic...

3.9.2 Generic...

3.9.3 Schema...

3.9.4 Comments

4.2. Symbol characters

Symbol characters of the core Z language are converted according to the following mark-up directives.

| Z character | \LaTeX mark-up |
|-------------|---------------------------------------|
| • | <code>%%Zinchar \spot U+2981</code> |
| | <code>%%Zinchar \mid U+007C</code> |
| † | <code>%%Zinchar \dagger U+2020</code> |
| ⊕ | <code>%%Zinchar \zovr U+E01E</code> |
| ⊖ | <code>%%Zinchar \xor U+22BB</code> |

4.3. Core words

Core words may be marked-up using the following commands, so that neighbouring spaces are converted implicitly.

| Z characters | \LaTeX mark-up |
|----------------|--|
| <i>comment</i> | <code>%%Zword \comment comment</code> |
| <i>post</i> | <code>%%Zpreword \post post</code> |
| ⊢? | <code>%%Zinword \thrm \vdash?</code> |
| <i>undecor</i> | <code>%%Zpreword \undecor undecor</code> |

1 Contents of this page

2 Introduction

3 ISO Standard...

3.1 White space and...

3.2 Mark-up directives

3.3 Greek letters

3.4 Other letters

3.5 Special characters

3.6 Symbol characters

3.7 Core words

3.8 Section header...

3.9 Paragraph mark-up

3.9.1 Axiomatic...

3.9.2 Generic...

3.9.3 Schema...

3.9.4 Comments



Page 18 of 18



Go Back

Full Screen

Close

Quit

4.4. Paragraph mark-up

CADiZ recognises `\ST` as a synonym for `\where`.

For non-outlined paragraphs, CADiZ recognises an alternative to the `zed` environment called the `syntax` environment. They differ only in that in the `syntax` environment, the ampersand character is treated as white space (allowing \LaTeX to use it as mark-up of columns to be aligned). To mark-up an ampersand character in the `syntax` environment, precede it with a backslash.

4.5. Tool directives

The \LaTeX mark-up recognised by cadiz also includes that of various [tool directives](#).

IT 20-Jan-2002

- 1 Contents of this page
- 2 Introduction
- 3 ISO Standard...
- 3.1 White space and...
- 3.2 Mark-up directives
- 3.3 Greek letters
- 3.4 Other letters
- 3.5 Special characters
- 3.6 Symbol characters
- 3.7 Core words
- 3.8 Section header...
- 3.9 Paragraph mark-up
 - 3.9.1 Axiomatic...
 - 3.9.2 Generic...
 - 3.9.3 Schema...
 - 3.9.4 Generic...