

# Correcting errors in a Z specification

/Tutorial guides

This tutorial assumes that you have already prepared a Z specification for cadiz using either the [LaTeX mark-up](#) or the [troff mark-up](#), and have [run cadiz](#), giving it your Z specification and getting some error reports about it.

## 1. Contents of this page

- [Introduction](#)
- [Syntax reports](#)
- [Scope reports](#)
- [Type reports](#)

## 2. Introduction

Any reports produced by cadiz can be categorised as syntax, scope or type reports, and moreover are either errors or warnings. All warnings are distinguishable by starting with the word *Warning*, and although they point to possible mistakes, they can be ignored. All errors should be attended to, as they indicate violation

1 Contents of this page

2 Introduction

3 Syntax reports

4 Scope reports

5 Type reports

of the rules of Z. Attention should usually be focussed on the first error report, as fixing the problem that it relates to will often affect subsequent reports, either changing them or silencing them altogether.

## 3. Syntax reports

Syntax reports are distinguishable from other reports by the presence of a line of source mark-up followed by a pointer to a position in that line. For example, suppose the example in the [browsing tutorial](#) were typed in without the declaration of *door\_to\_door*, but with the comma that preceded it.

```
cheapest, cleanest, most_healthy, : vehicle
                                   ^
```

```
"bike.z", line 5: Expected another name after , in declaration list
```

This is nearly all the help that cadiz can give with syntax reports. (The [examine mode](#) shows where the syntax error is in relation to other paragraphs, but the paragraph in which the syntax error occurs is not shown; the report itself gives more precise context information.) The pointer shows how far cadiz had read before being unable to make any sense of the input. The file name and line number should enable you to find the line in your mark-up. The rest of the report explains what cadiz might have reasonably expected to find. If you do not know the cause of a syntax error and cannot guess, then you will have to refer to the [concrete syntax](#) and the particular [LaTeX mark-up](#) or [troff mark-up](#) of the

tokens of that concrete syntax. By checking what you have written against those documents, effectively trying to parse it yourself, the problem and its solution will be revealed.

## 4. Scope reports

Scope reports relate to undefined or multiply-defined names. The [browsing tutorial](#)'s example has an undefined name, which is reported like this.

"bike", paragraph ending at line 10:

Reference to undeclared name: Skateboard

Interactive modes, such as the [examine mode](#), provide an accurate presentation of where in the specification is the context of the report. So the less accurate description of context in these scope reports does not matter. Using the [examine mode](#), the scope error report can be selected and the [context](#) command chosen.

"bike", paragraph ending at line 10:  
Reference to undeclared name: Skateboard

context  
why not

This causes the particular undefined use of the name *Skateboard* to be selected in the specification. In the case of multiply-defined names, the [context](#) command

selects the new definition and the [prev context](#) command selects the original definition. That should provide sufficient information to decide whether a declaration has really been omitted, or a name has been misspelt, or whatever.

## 5. Type reports

Type reports are issued when expressions are composed in ways that violate the type rules of Z. They usually cite two types: one is the type of a particular expression, and the other is the type that the expression was expected to have given the context in which it is being used. The [browsing tutorial](#)'s example has a type error, which is reported like this.

```
"bike", paragraph ending at line 10:
  Type mismatch in relational predicate
  Relation's parameter's type:
    vehicle
  Argument's type:
    ps vehicle
  In expression:
    {Cycle}
```

As for scope reports, accurate context information is provided using the [examine mode](#) and the [context](#) command. Rather like for syntax reports, cadiz tries to make sense of as much as possible before issuing a report, which in this case means

1 Contents of this page

2 Introduction

3 Syntax reports

4 Scope reports

5 Type reports



Page 3 of 5



Go Back

Full Screen

Close

Quit

determining types for as many expressions as it can. The appropriate correction to a type error might therefore not be to the particular expression that browsing with cadiz shows to be the context of the report, but some other expression. It is often useful to browse the types of expressions neighbouring that context, to help decide which expression to change. It is also possible to browse the types in the type report itself, to check which type declarations are being named. The example type error could be fixed by removing the set brackets, or by changing the last equality to a membership, or by more substantial alterations.

*IT 1-Dec-2000*

*1 Contents of this page*

*2 Introduction*

*3 Syntax reports*

*4 Scope reports*

*5 Type reports*