

# introduce variable

/Reference manual/Z-related commands/Refinement commands

This command is part of the experimental [refinement editor](#).

The *introduce variable* command refines a specification statement to introduce a new local scope around it. It applies the following inference rule of the refinement calculus.

$$\frac{\begin{array}{l} \vdash? \forall sig(D) \bullet \Delta(sig(D) \cup F)[P, I, P \wedge (pred(D) \wedge C)] \\ \vdash? \forall sig(D) \bullet \Delta(sig(D) \cup F)[P, I \wedge (pred(D) \wedge C), Q] \end{array}}{\vdash? \Delta F[P, I, Q]}$$

where  $D \mid C$  is the new local scope.

The [code](#) that is implicitly generated by this refinement rule is the following block statement.

```
[| varD | C
  Δ(sig(D) ∪ F)[P, I, P ∧ (pred(D) ∧ C)];
  Δ(sig(D) ∪ F)[P, I ∧ (pred(D) ∧ C), Q]
|]
```

The *introduce variable* command is applicable when any specification statement  $\Delta F[P, I, Q]$  in a goal is inspected.

The local scope  $D \mid C$  is entered into a dialogue box using the syntax of a Z schema text, e.g.  $n : \mathbb{N} \mid n < 42$ . The default response is the previous response.

Alternatively, if a suitable schema text is displayed in the same window, that can have been selected first (crossed). The local scope is typechecked in the environment of the inspected specification statement.

## 1. Tactic example

*“introduce variable” “ $D \mid C$ ”  $p$*

This example applies the *introduce variable* command to specification statement  $p$  using the local scope  $D \mid C$ .

A tactic that applies the *introduce variable* command must be executed by *play tactic*; it is not applicable under *apply tactic* as the resulting code would not be accessible to the *code* command.

*IT 20-Nov-2000*