

new name

[/Reference manual/Z-related commands/In situ replacement commands](#)

Page 1 of 2



Go Back

Full Screen

Close

Quit

The *new name* command changes the name of a declaration. An existing variable declaration should be inspected, and the new name should be typed into the offered dialogue box. All references to the declaration are renamed.

The *new name* command can also be applied to schema inclusion declarations, in which case new names are requested for each individual component of the schema. (This is related to the *new decoration* command.)

The only declarations whose names can safely be changed are those of the hypothesis of a goal, those of quantified predicates, and those of set comprehension, definite description and let expressions. Declarations of schema components may not be renamed, because of the risk of changing the meaning of some schema calculus operations. Declarations that are global to a specification may not be renamed, because that would change the meaning of the specification. Declarations of operators may not be renamed, because of the lack of an operator template paragraph for the new name. Declarations of generic parameters may not be renamed.

Renaming a name that is declared more than once in a schema text, such as occurs on inclusion of schemas with overlapping signatures, causes all of its declarations to be so renamed.

The new name may not be the same as another existing declaration in the same schema.

1. Tactic example

“new name” “{(\ “old\”, \ “new\”)}” d

This example applies the *new name* command to the declaration *d*, renaming name *old* as name *new*. The second argument must be a set extension of pairs of string literals.

IT 31-Jan-2000