

## About Step Names

A step name consists of a level number in angle brackets (like  $\langle 2 \rangle$ ) followed by any non-empty string of digits, letters, and underscores (`_` characters). Naming steps with consecutive numbers is just a convention. The period (“.”) following the step name is optional and is not part of the name.

A proof can have hundreds or even thousands of steps, and it’s hopeless to try to give them all meaningful names. I find that a proof looks neater and is a little less forbidding when steps are numbered consecutively. When writing a proof, we are always inserting steps and moving steps around, so we may insert a step named  $\langle 3 \rangle 3b$  or  $\langle 3 \rangle 35$  between steps  $\langle 3 \rangle 3$  and  $\langle 3 \rangle 4$ . We can renumber those steps consecutively by placing the cursor on the step that they are proving (a level  $\langle 2 \rangle$  step in this example) and executing the **Renumber Proof** (`control+g control+r`) command. The command works only if the module has not been modified since it was successfully parsed.

Your proof may have a few important “lemma” steps that are used in places much later in the proof, so you’d like to give them names you can remember that don’t get changed by the **Renumber Proof** command. The **TLA<sup>+</sup> Module Editor** preferences page allows you to choose among several options for which steps get renumbered by the command. For example, you can choose to have it renumber only step names that begin with a number.

A step name is considered to be a (locally) defined symbol, so you can find a step from a use of its name with the **Goto Declaration** (**F3**) command or by holding down the *control* key and left-clicking on the name. You can also find all uses of the step by putting the cursor on the step name (where it appears in the step or on any use of the name) and executing the **Show Uses** (**F6**) command.