



Atelier B 4.6.3 is a **Maintenance Edition**, restricted to the owner of a maintenance agreement, who can download it through their dedicated section.

New Features

Atelier B 4.6.3 has been released on december 2020.

Supported OSes:

- Windows 10
- Linux: Ubuntu LTS 16.04

This version corrects, since version 4.5.5, 133 anomalies and improves the following features:

- AtelierB CSSP
 - B modelling and application programming for starter kits SK₀ and SK₁
- Interactive proof
 - Saved proofs can now be replayed until the first failing command
 - Userpass can now be saved in the patchprover file
- Project management
 - Improvement of the use of MANIFEST
 - MANIFEST files can now be used through the console
- Translator
 - C4B: add the translation of tables indexed by an enumeration.
 - B2C: control on table initialization
- Drudge
 - Use of real numbers

- It is now possible to add only rule validated by PP
- NPOG
 - Event-B: add the WITNESS keyword
 - Consolidation of PO generation
- Documentation
 - Add the proof obligation generator parametrization manual

Since version 4.5.5, an anomaly that existed since the first versions of the prover has been corrected. This anomaly is related to the instantiation of universal quantified hypothesis, in which protection against free variable capture was incomplete. This could lead to incorrect proofs in specific conditions.

AtelierB CSSP

Atelier CSSP, restricted to the owner of a CLEARSY Safety Platform (<https://www.clearsy.com/outils/clearsy-safety-platform/>), enables B modelling and application programming for starter kits SK₀ and SK₁.

Version 4.6 of **Atelier CSSP** contains the following features:

- A B project skeleton generator, compatible with CLEARSY Safety Platform
- A reworked HEX binary code string ([https://fr.wikipedia.org/wiki/HEX_\(Intel\)](https://fr.wikipedia.org/wiki/HEX_(Intel))), using security and utility functions library
- A binary code loading tool for smartcards SK₀ and SK₁
- A serial monitor to check the correct execution of the application on smartcards SK₀ and SK₁
- A specific user interface enabling compilation and programming in a single action
- Two projects examples – clock and combinatorial – allowing a first use of CLEARSY safety Platform.

WITNESS

In a system project with NPOG, the keyword WITNESS, that allows to suggest a value, can now be used.

It can be used specifically in the refinement of an ANY:

trcu.1

OP selectionn... ✕

nn : NAT
 =>
 real (nn) :
 REAL

```

1  /* test1
2  * Author:
3  * Creation date:
4  */
5  SYSTEM
6    test1
7  VARIABLES
8    time
9  INVARIANT
10 1/2    time : REAL
11 INITIALISATION
12 1/1    time := 0.0
13
14 EVENTS
15
16      trcu = ANY nn WHERE nn : NAT THEN
17        time := real (nn)
18      END
19
20
21 END
22
23

```

trcu.1

trcu.2

OP selectionn... ✕

real (1) = 1.0

```

1  /* test1_r
2  * Author:
3  * Creation date:
4  */
5  REFINEMENT test1_r
6  REFINES test1
7
8  VARIABLES
9  time
10
11 INITIALISATION
12    time := 0.0
13
14 OPERATIONS
15    trcu =
16    BEGIN
17 1/2      WITNESS nn = 1
18      THEN
19 0/2      time := 1.0
20
21    END
22  END
23
24 END

```

Interactive proof

When using the interactive prover, the user can now replay a proof until the last failing command. This feature can be accessed in the graphical interface with the new button:



Or from the command line with : st(Fail)

When saving in User Pass the current interactive proof, it can now be saved in the Patchprover file instead of the pmm file.

