

BST Reference Sheet

This document references all of the current actions and checkers available in BST.

This version of the document has been updated for **OpenBST 2.0 Beta 2**

This document is available under the CC-BY-NC-SA 4.0 International licence.



License details : <https://creativecommons.org/licenses/by-nc-sa/4.0/>

Website : <https://utybo.github.io/BST/>

More recent versions may be available here : <https://utybo.github.io/BST/#downloads>

Document version : 1.1.1

Changes since last version : Fix sizes

GENERAL INFORMATION

Most of the time, if an action, a checker or an NND expects you to provide a node, you can provide either a node ID, an alias (if you specified an alias before) or a variable that contains an alias or a node ID.

If an action or a checker expects you to provide a value, you can generally use a variable as well, as long as what is contained in the variable corresponds to the value expected

SYNTAX CHEATSHEET

Regular node beginning	1:
Logical node beginning	1:&
Virtual node beginning	1:>
Include a var. in a node	\${variable}
Incl. a v. node in a node	\${>1}
Incl. a l. node in a node	\${&1}
Option	:Option text nnd
Conditional option	:Option text nnd [checker:a,b]
Option with Scripts	:Option text nnd {action:a,b}{action:a,b}

Note : you can combine both the checker and the action(s) here. If you only have one checker or one action, you can omit the brackets

QUESTIONS? SUGGESTIONS? SUBMIT A TICKET AND WE'LL HELP!

<https://github.com/utybo/BST/issues>

In this document :

Syntax cheatsheet

Base actions reference sheet

Module actions reference sheet

Checkers reference sheet

BST REFERENCE TABLE : Actions in basic BST (vanilla, without modules)		
Name	Syntax	Description
add	add:<putIn>,<a>, add:<a>,	Does a mathematical addition. The numbers <i>a</i> and <i>b</i> are added (they can be variables), and the variable named <i>putIn</i> is set to the result. If <i>putIn</i> is not specified, then the result will be placed in <i>a</i> which has to be a variable in this case.
assert	assert:<checker>	If the checker returns <i>true</i> , nothing is done. However, if it returns <i>false</i> , make the story crash. Only available in test environments.
bound	bound:<variable>,<minimum>,<maximum>	Ensures that <i>variable</i> is within the bounds determined by the <i>minimum</i> and the <i>maximum</i> . While <i>variable</i> has to be a variable, <i>minimum</i> or <i>maximum</i> can either be a variable or a raw value. If <i>variable</i> is less than <i>minimum</i> , its value is set to <i>minimum</i> . If <i>variable</i> is greater than <i>maximum</i> , its value is set to <i>maximum</i> .
call	call:<logical node>	Executes the <i>logical node</i> specified, without actually « going to » it. Useful for subroutines.
clone	clone:<to clone>,<to set>	Duplicates a variable. The variable named <i>to set</i> will have its value set to the one of <i>to clone</i> – however, future modifications of <i>to clone</i> will have no effect on <i>to set</i> and vice versa.
crash	crash:	Throw an exception immediately, terminating the execution of the current test. Only available in test environments.
decr	decr:<variable>	Decrements <i>variable</i> by one.
div	div:<putIn>,<a>, div:<a>,	Does a mathematical euclidian division. The numbers <i>a</i> and <i>b</i> will be divided (they can be variables), and the variable named <i>putIn</i> is set to the result. If <i>putIn</i> is not specified, then the result will be placed in <i>a</i> which has to be a variable in this case. Note that this is an euclidian division : the result will be rounded.
exit	exit:	Closes the story instantly
fail	fail:	Equivalent to the crash action. Only available in test environments.
incr	incr:<variable>	Increments <i>variable</i> by one
input	input:<variable>,Text to show	Asks the user for a value, which is then passed to <i>variable</i> . The text to show to the user can contain commas.
mod	mod:<putIn>,<a>, mod:<a>,	Determine a modulo, which is the remainder of an euclidian division. The numbers <i>a</i> and <i>b</i> will be divided (they can be variables), and the variable named <i>putIn</i> is set to the remainder of the euclidian division. If <i>putIn</i> is not specified, then the remainder will be placed in <i>a</i> which has to be a variable in this case. Note that this is an euclidian division : the result will be rounded.
mul	mul:<putIn>,<a>, mul:<a>,	Does a mathematical multiplication. The numbers <i>a</i> and <i>b</i> are multiplied (they can be variables), and the variable named <i>putIn</i> is set to the result. If <i>putIn</i> is not specified, then the result will be placed in <i>a</i> which has to be a variable in this case.
rand	rand:<variable>,<maximum> rand:<variable>,<minimum>,<maximum>	Randomly pick a number between <i>minimum</i> (or 0 if no minimum is given) and <i>maximum</i> , inclusive, meaning that the result can be <i>minimum</i> , <i>maximum</i> , or any number inbetween.
set	set:<variable>,<value>	Sets the variable <i>variable</i> to the value <i>value</i>

Name	Syntax	Description
sub	sub:<putIn>,<a>, sub:<a>,	Does a mathematical subtraction. The numbers <i>a</i> and <i>b</i> are subtracted (they can be variables), and the variable named <i>putIn</i> is set to the result. If <i>putIn</i> is not specified, then the result will be placed in <i>a</i> which has to be a variable in this case.

BST REFERENCE TABLE : Actions in BST modules			
Module	Name	Syntax	Description
BDF	bdf_apply	bdf_apply:<name>,<prefix> bdf_apply:<name> bdf_apply:! bdf_apply:! bdf_apply:! bdf_apply:! bdf_apply:!	Import all the values from the bdf file named <i>name</i> , or named by the content of the variable <i>variable</i> . Each import is prefixed by <i>prefix</i> if specified.
BRM	brm_load	DEPRECATED	This action used to trigger the loading of all resources. This is now useless, as resources are loaded automatically.
HTB	htb_base64	htb_base64:<resource>,<variable>	Import the content of resource named <i>resource</i> in the variable named <i>variable</i> , encoded into Base64
HTB	htb_import	htb_import:<resource>,<variable>	Import the raw content of resource named <i>resource</i> in the variable named <i>variable</i> .
JSE	jse_autoimport	DEPRECATED	This action is deprecated, but is still recognized by OpenBST so that stories with it do not crash.
JSE	jse_eval	jse_eval:<variable>,<javascript>	Puts the result of the javascript operation <i>javascript</i> into the variable named <i>variable</i>
JSE	jse_import	DEPRECATED	This action is deprecated, but is still recognized by OpenBST so that stories with it do not crash.
JSE	jse_reset	DEPRECATED	This action is deprecated, but is still recognized by OpenBST so that stories with it do not crash.
SSB	ssb_ambient	ssb_ambient:<resource>	Plays the music resource named <i>resource</i> in background, as an ambient music. It is automatically looped.
SSB	ssb_play	ssb_play:<resource>	Plays the sound resource named <i>resource</i> as a one-shot sound.
SSB	ssb_stop	ssb_stop:<resource>	Stops the currently playing ambient sound, if any. Does not do anything if there are no ambient sounds currently playing.
UIB	uib_init	uib_init:	Initializes UIB
UIB	uib_set	uib_set:<component id>,<value>	Main entry point for setting component texts or values. Please refer to the manual for further details
UIB	uib_setprop	uib_setprop:<component id>,<property name>,<value>	Sets the property named <i>property name</i> of component <i>component id</i> to <i>value</i>
UIB	uib_setvisible	uib_setvisible:<boolean>	If <i>boolean</i> is true, show UIB. If <i>boolean</i> is false, hide UIB.
XBF	xbf_call	xbf_call:<name>,<node>	XBF equivalent of the action <i>call</i> , with the only addition being that the logical node is in the resource story named <i>name</i>
XSF	xsf_exec	xsf_exec:<name>,<function> OR xsf_exec:<name>,<function>,<putIn>	Launches the function named <i>function</i> in Javascript file named <i>name</i> and, if <i>putIn</i> is specified, puts the result from the function in the variable <i>putIn</i>

BST REFERENCE TABLE : Checkers in BST modules		
Name	Syntax	Description
greater	greater:<a>,	Checks if <i>a</i> is strictly greater than <i>b</i>
greaterequ	greaterequ:<a>,	Checks if <i>a</i> is greater than or equal to <i>b</i>
less	less:<a>,	Checks if <i>a</i> is strictly less than <i>b</i>
lessequ	lessequ:<a>,	Checks if <i>a</i> is less than or equal to <i>b</i>
equ	equ:<a>,	Checks if <i>a</i> is equal to <i>b</i>
not	not:<a>,	Checks if <i>a</i> is not equal to <i>b</i>
jse_eval	jse_eval:<javascript>	Checks if the <i>javascript</i> code returns true. If it does, than this checker returns true. If it returns false or any other value, the checker returns false.