

XVAN Library 1.4

-- verbs--

-- everything is a location, an object or a timer --

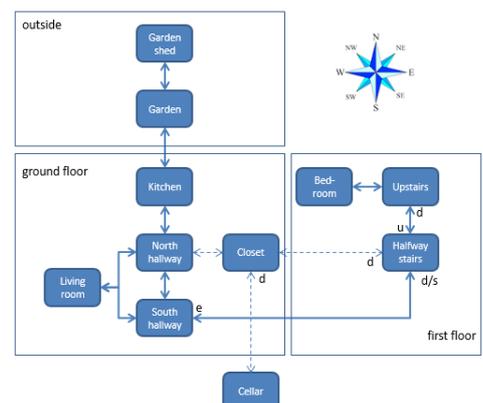


Table of Contents

What's new?	4
Introduction.....	5
ask.....	6
break.....	7
close.....	8
cry	9
drop	10
examine	11
get.....	12
give	13
go.....	14
hang.....	15
help.....	16
inventory	17
kill	18
knock	19
jump.....	20
listen	21
lock	22
look.....	23
move.....	24
open.....	25
poke.....	26
put	27
quit	29
read	30
restart.....	31
restore	32
save.....	33
say.....	34
score	35
sing	36
sit	37

smell	38
tell.....	39
testmode	40
throw	41
tie.....	42
touch.....	43
transcript	44
turn	45
undo	46
unlock	47
untie	48
unuse	49
use	50
verbose.....	51
wait.....	52
wear.....	53
xyzy	54
yell	55
Annex: verb template.....	56

What's new?

The XVAN Library version 1.3 has been tested with the XVAN IF Authoring System version 2.4.

In Library version 1.1: there are no new or changed verbs.

In Library version 1.2:

- Verbs “use” and “unused” were added (for use with IFI-XVAN);
- Verbs “sit”, “knock”, “poke”, put, jump, sing, cry, yell, say, touch and xyzy were added;
- Verb “restart” was added.
- Use of function notimers() for actions: save, restore, score, transcript and verbose;
- Some small changes in some actions;
- Some changes to o_all object;
- Some changes to o_it object;
- New file ifi-actions.lib for use with IFI-XVAN.

In Library version 1.3

- The Starter Kit is now called Library;
- Verbs “exits”, “goto”, “kick”, “remember”, “thank”, and “undo” were added;
- Common description d_remember was added;
- Common flags f_supporter and f_fixed were added;
- Common trigger t_reveal was added;
- Common attribute r_nr_to_reveal was added.

In Library version 1.4

- New implementation of the undo function.

Introduction

This document describes the verbs that come with the XVAN Library.

Verbs can be seen as actions that are initiated by the person playing the story. Examples are: take, drop, go, examine, ... Actions can be made more specific by adding nouns, adjectives, prepositions, etc: “put the garlic into the bag”.

In XVAN, actions are defined as verbs.

XVAN has no verbs hard coded in the interpreter because it wants to offer story authors the flexibility to define their own versions of verbs.

In order to get a head start, there is the XVAN Library. The Library has – among other things – a set of verbs that are used in most Interactive Fiction works. The verbs are available as ‘normal’ XVAN code, so they can easily be changed or removed and redefined when necessary. By inserting the library in the story file, the predefined verbs will be available in the story.

Verbs may also have logic built in. E.g. the verb might check if an item to be dropped is actually held by the player and if not print a message that the item cannot be dropped because it is not held by the player. The logic that is built into the verbs is intended to serve as sort of a default message to be executed when no code for the specific action is defined with the objects and locations. E.g. “turn book” will print something like “There’s no use in turning the book”. If “turn book” should trigger a chain of events specific for the story, than it should be coded with the book object. In this way, the verbs from the Library can be used in pretty much any story.

Verbs as “open”, “close”, “lock” and “unlock” are worked out quite extensively because these actions are likely to be the same in most stories.

Some of the verbs use XVAN artifacts like flags, attributes or triggers¹. E.g. “take” uses flag `f_takeable` to determine whether an item can be picked up.

Finally, verbs may contain instructions for the interpreter to solve ambiguity issues when it maps user input to object.

E.g. when the command is “drop cube” and the player is carrying a red cube and there’s a green cube on the floor, it is very likely that the red cube is meant. The verb’s ambiguity code will tell the interpreter to map the command to the red cube and not ask which cube is meant.

In this document, the verbs in the Library are described in terms of syntax, pre and post checks (prologue and epilogue), flags, attributes, triggers and ambiguity priorities.

At the end of the document, in the annex, a template for defining your own verbs is listed.

¹ For a description of flags, attributes and triggers, refer to the XVAN Introduction document.

ask

Purpose:

Ask someone about something.

Supported syntaxes:

Syntax	Normal behavior
"ask<subject1>about <subject2>"	Will print a message that subject2 is just an ordinary subject 2.

Checks done by the verb prologue

Check	Action
Is actor alive?	If not, abort.

Checks done by the verb epilogue

Check	Action
--	

Artifacts used

Artifact	Purpose
f_alive	Determine whether subject1 is something that is alive.

Ambiguity priority

Priority	Conditions
1 (subject1)	Subjects that are alive.
2 (subject1)	All other subjects.

Remarks:

- Ask has scope all_locs.
- Asking about must be handled by the subject itself, as this is too specific to handle all possible combinations in a verb. The verb will only print a message if nothing has been coded in the subject definition.
- If subject1 is the actor, it will print a message about talking to yourself.

break

Purpose:

Break a subject.

Supported syntaxes:

Syntax	Normal behavior
<code>"break"</code>	Will prompt what must be broken.
<code>"break <subject1>"</code>	Will print a message that breaking subject1 isn't particularly helpful.
<code>"break <subject1> with<subject2>"</code>	

Checks done by the verb prologue

Check	Action
Is actor alive?	If not, abort.
Does actor hold subject2?	If not, try to take subject2 and continue if succeeds.

Checks done by the verb epilogue

Check	Action
--	

Artifacts used

Artifact	Purpose
<code>f_takeable</code>	Determine whether subject1 can be taken if it is not held by the actor.

Ambiguity priority

Priority	Conditions
1 (subject2)	Subjects that are held by the actor.
2 (subject2)	All other subjects.

Remarks:

Breaking subjects must be handled by the subject itself, as this is too specific to handle all possible combinations in a verb. The verb will only print a message if nothing has been coded in the subject definition.

close

Purpose:

Close an item.

Supported syntaxes:

Syntax	Normal behavior
“close”	Will prompt what must be closed.
“close subject”	Will close the subject.

Checks done by the verb prologue

Check	Action
Is actor alive?	If not, abort.
Is subject openable?	If not, abort.
Is subject closed?	If yes, abort.

Checks done by the verb epilogue

Check	Action
Is it dark ?	If yes, print message about darkness.

Artifacts used

Artifact	Purpose
f_openable	Determine whether the subject can be opened and closed.
f_open	Determine whether the subject is already closed.
f_opaque	After closing, the subject is set to opaque.
f_lit	Check if the surroundings are still lit after closing the subject.

Ambiguity priority

Priority	Conditions
1	Subjects that are open.
2	All other subjects in scope.

Remarks:

The subject is assumed to be opaque when it's closed. For see-through subjects, the opaque flag must be cleared again in the subject's code.

cry

Purpose:

rying.

Supported syntaxes:

Syntax	Normal behavior
Everything, only has a default	Prints a message that the actor is not a crybaby.

Checks done by the verb prologue

Check	Action
Is actor alive?	If not, abort.

Checks done by the verb epilogue

Check	Action
--	

Artifacts used

Artifact	Purpose
--	

Ambiguity priority

Priority	Conditions
--	

Remarks:

--

drop

Purpose:

Drop an item.

Supported syntaxes:

Syntax	Normal behavior
<code>"drop"</code>	Will prompt what must be dropped.
<code>"drop <subject1>"</code>	Will drop subject1.
<code>"drop <subject1><in><subject2>"</code>	Will print default message that subject1 cannot be dropped in subject2. Note: <in> may be any preposition.

Checks done by the verb prologue

Check	Action
Is actor alive?	If not, abort.

Checks done by the verb epilogue

Check	Action
Is it dark ?	If yes, print message about darkness.

Artifacts used

Artifact	Purpose
<code>f_bypass</code>	Is cleared after dropping, so the dropped subject will not be exempt from visibility checks any longer.
<code>r_is</code>	Attribute that holds the correct conjugation of to be for the current actor.
<code>f_container</code>	To determine if <subject2> is a container

Ambiguity priority

Priority	Conditions
1 (subject1)	Subjects that are held by the actor.
2 (subject1)	All other subjects in scope.
1 (subject 2)	Subjects that are a container.

Remarks:

`"drop <subject1><in><subject2>"` must be handled by the subject itself, as this is too specific to handle all possible combinations in a verb. The verb will only print a message if nothing has been coded in the subject definition.

examine

Purpose:

Examine an item.

Supported syntaxes:

Syntax	Normal behavior
“examine”	Will prompt what must be examined.
“examine <subject>”	Will examine the subject.

Checks done by the verb prologue

Check	Action
Is actor alive?	If not, abort.

Checks done by the verb epilogue

Check	Action
--	

Artifacts used

Artifact	Purpose
d_exa	Description that holds the description for examining the subject

Ambiguity priority

Priority	Conditions
--	

Remarks:

“x” and “investigate” are synonyms for examine.

get

Purpose:

Pick up an item.

Supported syntaxes:

Syntax	Normal behavior
"get"	Will prompt what must be taken.
"get <subject1>"	Will drop subject1.
"get<subject1><from><subject2>"	Will print default message that subject1 cannot be taken from subject2. Note: <from> may be any preposition.

Checks done by the verb prologue

Check	Action
Is actor alive?	If not, abort.

Checks done by the verb epilogue

Check	Action
--	

Artifacts used

Artifact	Purpose
f_bypass	Is set after taking, so the dropped subject will be exempt from visibility checks from now on.
f_takeable	Determine whether the subject can be taken.
r_is	Attribute that holds the correct conjugation of to be for the current actor.

Ambiguity priority

Priority	Conditions
1 (subject1)	Subjects that can be picked up and are not held by the actor.
1 (subject1)	Subjects that can be picked up and are held by subject2
2 (subject1)	Subjects that can be picked up.
3 (subject1)	All other subjects.
1 (subject2)	Subjects that own subject1.

Remarks:

"take" and "grab" are synonyms for get.

give

Purpose:

Give a subject to another subject.

Supported syntaxes:

Syntax	Normal behavior
"give <subject1> to <subject2>"	Will print a message that the gift is refused (for living subject2) or that there's no use giving subject1 to subject2.

Checks done by the verb prologue

Check	Action
Is actor alive?	If not, abort.
Does actor have subject?	If not, abort.

Checks done by the verb epilogue

Check	Action
--	

Artifacts used

Artifact	Purpose
f_alive	Determine whether subject1 is alive.

Ambiguity priority

Priority	Conditions
1 (subject2)	Subjects that are alive.
2 (subject2)	All other subjects.

Remarks:

Giving subjects must be handled by the subject itself, as this is too specific to handle all possible combinations in a verb. The verb will only print a message if nothing has been coded in the subject definition.

go

Purpose:

Move through the game world.

Supported syntaxes:

Syntax	Normal behavior
"<direction>"	Move the player object from the current location to the location that is to the direction.
"go <direction>"	Move the player object from the current location to the location that is to the direction.
"go to <direction>"	Move the player object from the current location to the location that is to the direction.

Checks done by the verb prologue

Check	Action
Is actor alive?	If not, abort.

Checks done by the verb epilogue

Check	Action
--	

Artifacts used

Artifact	Purpose
--	

Ambiguity priority

Priority	Conditions
--	

Remarks:

--

hang

Purpose:

Hang a subject on/in/behind another subject.

Supported syntaxes:

Syntax	Normal behavior
"hang <subject1><on><subject2>"	Will print a message subject1 cannot be hung on subject2. Note: <on> may be any preposition.

Checks done by the verb prologue

Check	Action
Is actor alive?	If not, abort.
Does actor hold subject2?	If not, try to take subject2 and continue if succeeds.

Checks done by the verb epilogue

Check	Action
--	

Artifacts used

Artifact	Purpose
--	

Ambiguity priority

Priority	Conditions
--	

Remarks:

Hanging subjects must be handled by the subject itself, as this is too specific to handle all possible combinations in a verb. The verb will only print a message if nothing has been coded in the subject definition.

help

Purpose:

Switch between displaying verb syntax suggestions. Suggestions are displayed when the player enters a sentence the interpreter does not understand.

Supported syntaxes:

Syntax	Normal behavior
"help"	If verb help is on, turn it off. If verb help is off, turn it on.

Checks done by the verb prologue

Check	Action
--	

Checks done by the verb epilogue

Check	Action
--	

Artifacts used

Artifact	Purpose
<code>o_player.f_verb_help</code>	Determine whether verb help is on.

Ambiguity priority

Priority	Conditions
--	

Remarks:

- By default, verb help is on.
- Help prevents the timers to be fired after execution.

inventory

Purpose:

List the player's possessions.

Supported syntaxes:

Syntax	Normal behavior
"inventory"	List the player's possessions.

Checks done by the verb prologue

Check	Action
Is the actor the player?	If not, abort.

Checks done by the verb epilogue

Check	Action
--	

Artifacts used

Artifact	Purpose
r_preposition	Attribute that holds an item's preposition in case the item is contained in another object that the player holds.

Ambiguity priority

Priority	Conditions
--	

Remarks:

- "I" is synonym for inventory;
- The verb only prints "You are carrying:". For printing the descriptions for the items the player carries, trigger t_i must be called by the item itself.
This is accomplished by adding the instruction "inventory" ->t_i in the TRIGGERS section of the object.

kill

Purpose:

Kill a subject.

Supported syntaxes:

Syntax	Normal behavior
“kill”	Will prompt what must be killed.
“kill <subject1> with <subject2>”	Will print a message that killing is not the answer. In case the actor is subject1, it will say that’s it won’t commit suicide.

Checks done by the verb prologue

Check	Action
Is actor alive?	If not, abort.

Checks done by the verb epilogue

Check	Action
--	

Artifacts used

Artifact	Purpose
f_alive	Determine whether subject1 is alive.

Ambiguity priority

Priority	Conditions
1 (subject1)	Subjects that are alive.
2 (subject1)	All other subjects.

Remarks:

Killing subjects must be handled by the subject itself, as this is too specific to handle all possible combinations in a verb. The verb will only print a message if nothing has been coded in the subject definition.

knock

Purpose:

Knock on an item.

Supported syntaxes:

Syntax	Normal behavior
"knock"	Will prompt what must be knocked on.
"knock <subject>"	Will print a message that there is no answer.
"knock on <subject>"	

Checks done by the verb prologue

Check	Action
Is actor alive?	If not, abort.

Checks done by the verb epilogue

Check	Action
--	

Artifacts used

Artifact	Purpose
--	

Ambiguity priority

Priority	Conditions
--	

Remarks:

--

jump

Purpose:

Jumping or moving.

Supported syntaxes:

Syntax	Normal behavior
"jump"	Prints "Wheee".
"jump <direction>"	Same as "go <direction>".

Checks done by the verb prologue

Check	Action
Is actor alive?	If not, abort.

Checks done by the verb epilogue

Check	Action
--	

Artifacts used

Artifact	Purpose
--	

Ambiguity priority

Priority	Conditions
--	

Remarks:

--

listen

Purpose:

Listening.

Supported syntaxes:

Syntax	Normal behavior
"listen"	Will print a message that there's only environment sounds.
"listen to <subject>"	Will print a message that there is no sound from the subject.

Checks done by the verb prologue

Check	Action
Is actor alive?	If not, abort.

Checks done by the verb epilogue

Check	Action
--	

Artifacts used

Artifact	Purpose
--	

Ambiguity priority

Priority	Conditions
--	

Remarks:

listening and listening to subjects must be handled by the subject itself, as this is too specific to handle all possible combinations in a verb. The verb will only print a message if nothing has been coded in the subject definition.

lock

Purpose:

Lock an item.

Supported syntaxes:

Syntax	Normal behavior
"lock"	Will prompt what must be locked.
"lock <subject1>"	Will lock the subject provided the actor has the correct key.
"lock <subject1> with <subject2>"	Will lock subject1 using subject2 as the key.

Checks done by the verb prologue

Check	Action
Is actor alive?	If not, abort.
Is subject1 lockable?	If not, abort.
Is subject1 locked?	If yes, abort?

Checks done by the verb epilogue

Check	Action
--	

Artifacts used

Artifact	Purpose
f_lockable	Determine whether subject1 is something that can be locked
f_locked	Determine whether subject1 is already locked.
r_is	Attribute that holds the correct conjugation of to be for the current actor.
r_do	Attribute that holds the correct conjugation of to do for the current actor.
r_key	Attribute that holds the key to unlock subject1. Must be set in the object code. Default value is %none.

Ambiguity priority

Priority	Conditions
1 (subject1)	Subjects that are lockable and locked.
2 (subject1)	Subjects that are lockable.
3 (subject1)	All other subjects.

Remarks:

--

look

Purpose:

Describe the surroundings.

Supported syntaxes:

Syntax	Normal behavior
"look"	Call the entrance() function for the current location.
"look at <subject>"	Will examine the subject.
"look <in><subject>"	Will print default message that there is nothing special in the subject. Note: <in> may be any preposition.

Checks done by the verb prologue

Check	Action
Is actor alive?	If not, abort.
Is it dark ?	If yes, print message about darkness and abort.

Checks done by the verb epilogue

Check	Action
--	

Artifacts used

Artifact	Purpose
d_exa	Description that holds the description for examining the subject
f_seenbefore	Flag that denotes that the location was visited before. Is set to 0 by the verb to ensure that the long room description is printed by the entrance().
r_preposition	Attribute that holds the preposition in case the actor is contained in another object.

Ambiguity priority

Priority	Conditions
--	

Remarks:

"look <in><subject>" must be handled by the subject itself, as this is too specific to handle all possible combinations in a verb. The verb will only print a message if nothing has been coded in the subject definition.

move

Purpose:

Move a subject.

Supported syntaxes:

Syntax	Normal behavior
<code>"move"</code>	Will prompt what must be moved.
<code>"move<subject1>"</code>	Will prompt in which direction subject1 must be moved.
<code>"move <subject1> to <direction>"</code> <code>"move <subject1><direction>"</code>	Will print a message that it doesn't make sense to move subject1 to that direction. Note: <in> may be any preposition.
<code>"move <subject1><in><subject2>"</code>	Will print a message that moving subject1 in subject2 accomplishes nothing. Note: <in> may be any preposition.

Checks done by the verb prologue

Check	Action
Is actor alive?	If not, abort.
Is subject1 fixed in place?	If yes, abort.

Checks done by the verb epilogue

Check	Action
--	

Artifacts used

Artifact	Purpose
<code>f_fixed</code>	Determine whether the subject is fixed in place.

Ambiguity priority

Priority	Conditions
1 (subject1)	Subjects that are not fixed in place.
2(subject1)	All other subjects.

Remarks:

Moving subjects must be handled by the subject itself, as this is too specific to handle all possible combinations in a verb. The verb will only print a message if nothing has been coded in the subject definition.

open

Purpose:

Open an item.

Supported syntaxes:

Syntax	Normal behavior
“open”	Will prompt what must be opened.
“open subject”	Will open the subject. In case any visible items are in the subject, a list of these items will be printed.

Checks done by the verb prologue

Check	Action
Is actor alive?	If not, abort.
Is subject openable?	If not, abort.
Is subject locked?	If yes, abort.
Is subject open?	If yes, abort.

Checks done by the verb epilogue

Check	Action
--	

Artifacts used

Artifact	Purpose
f_openable	Determine whether the subject can be opened and closed.
f_open	Determine whether the subject is already closed.
f_opaque	After opening, the subject is no longer opaque.
f_any	Used to count the number of items that are contained in the subject.
r_nr_to_reveal	Number of items contained in the subject. Used to print the list of items.
f_first	Used to determine when to print the starting text of the item list.
t_reveal	Trigger that prints the list of items in the subject.

Ambiguity priority

Priority	Conditions
1	Subjects that are openable, not open and not locked.
2	Subjects that are openable and not locked.
3	Subjects that are openable
4	All other subjects.

Remarks:

--

poke

Purpose:

Poke (in) an item.

Supported syntaxes:

Syntax	Normal behavior
"poke"	Will prompt what must be poked
"poke <subject>"	Will print that there is nothing in the subject.
"poke in <subject>"	

Checks done by the verb prologue

Check	Action
Is actor alive?	If not, abort.

Checks done by the verb epilogue

Check	Action
--	

Artifacts used

Artifact	Purpose
--	

Ambiguity priority

Priority	Conditions
--	

Remarks:

--

put

Purpose:

Put an item in/on/behind/ another item.

Supported syntaxes:

Syntax	Normal behavior
“put”	Will prompt what must be put.
“put <subject1>”	Will ask where to put subject1.
“put <subject1><in><subject2>”	Will print default message that subject1 cannot be put in subject2. Note: <in> may be any preposition.

Checks done by the verb prologue

Check	Action
Is actor alive?	If not, abort.

Checks done by the verb epilogue

Check	Action
Is it dark ?	If yes, print message about darkness.

Artifacts used

Artifact	Purpose
f_bypass	Is cleared after dropping, so the dropped subject will not be exempt from visibility checks any longer.
f_takeable	If subject1 is not held, can it be taken?
f_container	Is subject2 a container (can something be put in it?)
f_supporter	Is subject2 a supporter (can something be put on it?)
r_is	Attribute that holds the correct conjugation of to be for the current actor.
f_container	To determine if <subject2> is a container

Ambiguity priority

Priority	Conditions
1 (subject1)	Subjects that are held by the actor.
2 (subject1)	All other subjects in scope.
1 (subject2)	Subjects that are a container.
1 (subject2)	Subjects that are a supporter.
1 (subject2)	Subjects that are not held by the actor.
2 (subject2)	All other subjects in scope.

Remarks:

“put <subject1><in><subject2>” can handle prepositions “in” and “on”. Other prepositions must be handled by the objects themselves. If not handled by the object, the verb will print a message that the subject cannot be put.

quit

Purpose:

Quit and exit the game.

Supported syntaxes:

Syntax	Normal behavior
"quit"	Will ask the player is he is sure and if yes, quit the game.

Checks done by the verb prologue

Check	Action
--	

Checks done by the verb epilogue

Check	Action
--	

Artifacts used

Artifact	Purpose
--	

Ambiguity priority

Priority	Conditions
--	

Remarks:

--

read

Purpose:

Read an item.

Supported syntaxes:

Syntax	Normal behavior
"read"	Will prompt what must be read.
"read subject"	Will print that no text is printed on the subject.

Checks done by the verb prologue

Check	Action
Is actor alive?	If not, abort.

Checks done by the verb epilogue

Check	Action
--	

Artifacts used

Artifact	Purpose
--	

Ambiguity priority

Priority	Conditions
--	

Remarks:

Reading must be handled by the subject itself.

restart

Purpose:

Restart the game.

Supported syntaxes:

Syntax	Normal behavior
"restart"	Will ask the player is he is sure and if yes, restart the game.

Checks done by the verb prologue

Check	Action
--	

Checks done by the verb epilogue

Check	Action
--	

Artifacts used

Artifact	Purpose
--	

Ambiguity priority

Priority	Conditions
--	

Remarks:

--

restore

Purpose:

Restore earlier saved progress.

Supported syntaxes:

Syntax	Normal behavior
"restore"	Restore the progress that was saved earlier.

Checks done by the verb prologue

Check	Action
--	

Checks done by the verb epilogue

Check	Action
--	

Artifacts used

Artifact	Purpose
--	

Ambiguity priority

Priority	Conditions
--	

Remarks:

- Restore prevents the timers to be fired after execution.

save

Purpose:

Save progress.

Supported syntaxes:

Syntax	Normal behavior
"save"	Save the current state of the story.

Checks done by the verb prologue

Check	Action
--	

Checks done by the verb epilogue

Check	Action
--	

Artifacts used

Artifact	Purpose
<code>o_player.f_may_save</code>	Check whether saving is allowed.

Ambiguity priority

Priority	Conditions
--	

Remarks:

- There are situations where saving the game is not allowed. E.g. in a maze or during a fight. By setting the flag `o_player.f_may_save`, the save verb will print a message that saving is not allowed at this point.
- Save does not save the undo history.
- Save prevents the timers to be fired after execution.

say

Purpose:

Addressing someone.

Supported syntaxes:

Syntax	Normal behavior
Everything, only has a default.	Prints a message to use “ask” or “tell” to speak to someone..

Checks done by the verb prologue

Check	Action
Is actor alive?	If not, abort.

Checks done by the verb epilogue

Check	Action
--	

Artifacts used

Artifact	Purpose
--	

Ambiguity priority

Priority	Conditions
--	

Remarks:

--

score

Purpose:

Print the current score.

Supported syntaxes:

Syntax	Normal behavior
"score"	Print the player's score and the maximum score.

Checks done by the verb prologue

Check	Action
--	

Checks done by the verb epilogue

Check	Action
--	

Artifacts used

Artifact	Purpose
<code>o_player.r_score</code>	The player's current score.
<code>o_player.r_max_score</code>	The maximum score for this game.

Ambiguity priority

Priority	Conditions
--	

Remarks:

- Increasing the player's score is done by adding a number to the `r_score` attribute;
- Score prevents the timers to be fired after execution.

sing

Purpose:

Singing.

Supported syntaxes:

Syntax	Normal behavior
Everything, only has a default	Prints a message that the actor is not in the mood for singing.

Checks done by the verb prologue

Check	Action
Is actor alive?	If not, abort.

Checks done by the verb epilogue

Check	Action
--	

Artifacts used

Artifact	Purpose
--	

Ambiguity priority

Priority	Conditions
--	

Remarks:

--

sit

Purpose:

Sit on things.

Supported syntaxes:

Syntax	Normal behavior
"sit <on><subject>"	Move the player to <subject> with preposition 'on'. Note: <on> may be any preposition

Checks done by the verb prologue

Check	Action
Is actor alive?	If not, abort.

Checks done by the verb epilogue

Check	Action
--	

Artifacts used

Artifact	Purpose
--	

Ambiguity priority

Priority	Conditions
--	

Remarks:

- 'sit' moves the player to <subject>. If <subject> is opaque, the player will not be visible, regardless of the preposition that was used.

smell

Purpose:

Smelling.

Supported syntaxes:

Syntax	Normal behavior
“smell”	Will print a message that you smell nothing special.
“smell <subject>”	Will print a message that it smells just like a subject.

Checks done by the verb prologue

Check	Action
Is actor alive?	If not, abort.

Checks done by the verb epilogue

Check	Action
--	

Artifacts used

Artifact	Purpose
--	

Ambiguity priority

Priority	Conditions
--	

Remarks:

Smelling and smelling subjects must be handled by the subject itself, as this is too specific to handle all possible combinations in a verb. The verb will only print a message if nothing has been coded in the subject definition.

tell

Purpose:

Tell someone about something.

Supported syntaxes:

Syntax	Normal behavior
"tell <subject1>about <subject2>"	Will print a message saying thanks for sharing.

Checks done by the verb prologue

Check	Action
Is actor alive?	If not, abort.

Checks done by the verb epilogue

Check	Action
--	

Artifacts used

Artifact	Purpose
f_alive	Determine whether subject1 is something that is alive.

Ambiguity priority

Priority	Conditions
1 (subject1)	Subjects that are alive.
2 (subject1)	All other subjects.

Remarks:

- Telling about must be handled by the subject itself, as this is too specific to handle all possible combinations in a verb. The verb will only print a message if nothing has been coded in the subject definition.
- If subject1 is the actor, it will print a message about talking to yourself.

testmode

Purpose:

Get user input from a file. This command is intended for regression testing.

Supported syntaxes:

Syntax	Normal behavior
<code>"testmode"</code>	Reads user input from a file. The file must be named "testinput.txt".

Checks done by the verb prologue

Check	Action
--	

Checks done by the verb epilogue

Check	Action
--	

Artifacts used

Artifact	Purpose
--	

Ambiguity priority

Priority	Conditions
--	

Remarks:

The file testinput.txt must be in the same directory as the interpreter.

throw

Purpose:

Throw a subject.

Supported syntaxes:

Syntax	Normal behavior
<code>"throw <subject1></code>	Drops subject1.
<code>"throw<subject1> to <direction>"</code> <code>"throw<subject1><direction>"</code>	Will print a message that the subject bounces to the wall and falls on the floor.
<code>"throw<subject1><at><subject2>"</code>	Will print a message that throwing subject1 at subject2 accomplishes nothing. Note: <at> may be any preposition.

Checks done by the verb prologue

Check	Action
Is actor alive?	If not, abort.
Does actor hold subject1?	If not, try to take subject1 and continue if succeeds.

Checks done by the verb epilogue

Check	Action
--	

Artifacts used

Artifact	Purpose
<code>f_takeable</code>	Determine whether subject1 can be taken if it is not held by the actor.

Ambiguity priority

Priority	Conditions
1 (subject1)	Subjects that are held by the actor.
2 (subject1)	Subjects that are not held but can be picked up.
3 (subject1)	All other subjects.

Remarks:

Throwing subjects must be handled by the subject itself, as this is too specific to handle all possible combinations in a verb. The verb will only print a message if nothing has been coded in the subject definition.

tie

Purpose:

Tie a subject to another subject.

Supported syntaxes:

Syntax	Normal behavior
"tie"	Will prompt what must be tied.
"tie <subject1>"	Will prompt to what subject1 must be tied to.
"tie <subject1> to <subject2>"	Will print a message that subject1 cannot be tied to subject 2.

Checks done by the verb prologue

Check	Action
Is actor alive?	If not, abort.

Checks done by the verb epilogue

Check	Action
--	

Artifacts used

Artifact	Purpose
--	

Ambiguity priority

Priority	Conditions
--	

Remarks:

Tying subjects must be handled by the subject itself, as this is too specific to handle all possible combinations in a verb. The verb will only print a message if nothing has been coded in the subject definition.

touch

Purpose:

Touching things.

Supported syntaxes:

Syntax	Normal behavior
"touch"	Will prompt what must be touched.
"touch <subject>"	Prints that it just feels like a <subject>.

Checks done by the verb prologue

Check	Action
Is actor alive?	If not, abort.

Checks done by the verb epilogue

Check	Action
--	

Artifacts used

Artifact	Purpose
--	

Ambiguity priority

Priority	Conditions
--	

Remarks:

--

transcript

Purpose:

Copy story input and output to a file.

Supported syntaxes:

Syntax	Normal behavior
"transcript"	Will copy user input and story output to file transcript.txt.

Checks done by the verb prologue

Check	Action
--	

Checks done by the verb epilogue

Check	Action
--	

Artifacts used

Artifact	Purpose
--	

Ambiguity priority

Priority	Conditions
--	

Remarks:

- The first time 'transcript' is entered, it will start copying I/O to the file. Upon the next 'transcript' command, the file will be closed.
- If the transcript file already exists, the output will be added to the end of the file;
- If the file transcript.txt already exists, it will be overwritten.
- Transcript prevents the timers to be fired after execution.

turn

Purpose:

Turn a subject.

Supported syntaxes:

Syntax	Normal behavior
<code>"turn"</code>	Will prompt what must be turned.
<code>"turn <subject>"</code>	Will print that turning the subject accomplishes nothing.
<code>"turn <subject><direction>"</code> <code>"turn <subject> to <direction>"</code>	Will print a message that it doesn't make sense turning the subject to the direction.
<code>"turn on <subject>"</code> <code>"turn <subject> on"</code>	Will print a message that the subject cannot be turned on.
<code>"turn off <subject>"</code> <code>"turn <subject> off"</code>	Will print a message that the subject cannot be turned off.

Checks done by the verb prologue

Check	Action
Is actor alive?	If not, abort.

Checks done by the verb epilogue

Check	Action
--	

Artifacts used

Artifact	Purpose
<code>f_fixed</code>	Determine whether the subject is fixed in place.

Ambiguity priority

Priority	Conditions
--	

Remarks:

- "rotate" is synonym for turn;
- Turning subjects must be handled by the subject itself, as this is too specific to handle all possible combinations in a verb. The verb will only print a message if nothing has been coded in the subject definition.

undo

Purpose:

Undo the last turn.

Supported syntaxes:

Syntax	Normal behavior
“undo”	Undo the previous turn.

Checks done by the verb prologue

Check	Action
--	

Checks done by the verb epilogue

Check	Action
--	

Artifacts used

Artifact	Purpose
<code>o_player.f_may_undo</code>	Check whether undoing is allowed.

Ambiguity priority

Priority	Conditions
--	

Remarks:

- There are situations where undoing is not allowed. E.g. in a maze or during a fight. By clearing the flag `o_player.f_may_undo`, the undo verb will print a message that undoing is not allowed at this point.
- Undo prevents the timers to be fired after execution.

unlock

Purpose:

Unlock an item.

Supported syntaxes:

Syntax	Normal behavior
"unlock"	Will prompt what must be unlocked.
"unlock <subject1>"	Will unlock the subject provided the actor has the correct key.
"unlock <subject1> with <subject2>"	Will unlock subject1 using subject2 as the key.

Checks done by the verb prologue

Check	Action
Is actor alive?	If not, abort.
Is subject1 lockable?	If not, abort.
Is subject locked?	If yes, abort.

Checks done by the verb epilogue

Check	Action
--	

Artifacts used

Artifact	Purpose
f_locked	Determine whether subject1 is already locked.
r_is	Attribute that holds the correct conjugation of to be for the current actor.
r_do	Attribute that holds the correct conjugation of to do for the current actor.
r_key	Attribute that holds the key to unlock subject1. Must be set in the object code. Default value is is %none.

Ambiguity priority

Priority	Conditions
1 (subject1)	Subjects that are locked.
2 (subject1)	All other subjects.

Remarks:

--

untie

Purpose:

untie a subject from another subject.

Supported syntaxes:

Syntax	Normal behavior
"untie"	Will prompt what must be untied.
"untie <subject1>"	Will prompt what subject1 must be untied from.
"untie <subject1> from <subject2>"	Will print a message that subject1 cannot be untied from subject 2.

Checks done by the verb prologue

Check	Action
Is actor alive?	If not, abort.

Checks done by the verb epilogue

Check	Action
--	

Artifacts used

Artifact	Purpose
--	

Ambiguity priority

Priority	Conditions
--	

Remarks:

Untying subjects must be handled by the subject itself, as this is too specific to handle all possible combinations in a verb. The verb will only print a message if nothing has been coded in the subject definition.

unuse

Purpose:

Drop a subject.

Supported syntaxes:

Syntax	Normal behavior
"unuse <subject>"	Drop the subject.

Checks done by the verb prologue

Check	Action
--	

Checks done by the verb epilogue

Check	Action
--	

Artifacts used

Artifact	Purpose
--	

Remarks:

The verb 'unuse' was added for IFI-XVAN. When an object from the sidebar (inventory) is dragged and dropped on a blank text area, the command "drop subject" is sent to back-end (see also the IFI-XVAN documentation).

use

Purpose:

Depending on the state of the game, perform a certain action on a subject.

Supported syntaxes:

Syntax	Normal behavior
"use <subject1>"	Perform an action on subject1. See remarks for possible actions.
"use <subject1> with <subject2>"	Perform an action with subject1 and subject 2. See remarks for possible actions.

Checks done by the verb prologue

Check	Action
--	

Checks done by the verb epilogue

Check	Action
--	

Artifacts used

Artifact	Purpose
f_alive	Determine whether a subject is alive.
f_examined	Determine whether a subject has been examined before.
f_locked	Determine whether a subject is locked.
f_key	Determine whether a subject is a key.
f_container	Determine whether a subject can contain other subjects.

Remarks:

- The verb 'use' was added for IFI-XVAN. When an object text hyperlink is clicked or an icon is dropped on an object hyperlink, the GUI sends a 'use' command to the back-end (see also the IFI-XVAN documentation).
- **use subject1:**
 - If subject1 is unexamined, the command "examine subject1" will be executed.
 - Otherwise the command "get subject1" will be executed.
- **use subject1 with subject2:**
 - if both subjects are alive, the command "ask subject2 about subject1" is executed.
 - If only subject2 is alive, the command "give subject1 to subject2" is executed.
 - If subject2 is locked and subject12 is a key, the command "unlock subject2 with subject1" is executed.
 - If subject2 is a container, the command "put subject1 in subject2" is executed.
 - Else, the command "put subject1 on subject2" is executed.

verbose

Purpose:

Switch between always printing full room descriptions and only printing the full description at first entrance.

Supported syntaxes:

Syntax	Normal behavior
"verbose"	If verbose is on, turn it off. If verbose is off, turn it on.

Checks done by the verb prologue

Check	Action
--	

Checks done by the verb epilogue

Check	Action
--	

Artifacts used

Artifact	Purpose
o_player.f_verbose	Determine whether verbose mode is on or off.

Ambiguity priority

Priority	Conditions
--	

Remarks:

- By default, verbose mode is turned off. Full room descriptions are only entered upon first entry of the room.
- Verbose prevents the timers to be fired after execution.

wait

Purpose:

Wait for one or more turns.

Supported syntaxes:

Syntax	Normal behavior
“wait”	Wait 1 turns
“wait <number>”	Wait the number of turns.

Checks done by the verb prologue

Check	Action
--	

Checks done by the verb epilogue

Check	Action
--	

Artifacts used

Artifact	Purpose
--	

Ambiguity priority

Priority	Conditions
--	

Remarks:

During the wait period, the timers will fire. All timed events will continue during wait.

wear

Purpose:

Wear an item.

Supported syntaxes:

Syntax	Normal behavior
"wear"	Will prompt what must be worn.
"wear subject"	Will print that the subject is now worn.

Checks done by the verb prologue

Check	Action
Is actor alive?	If not, abort.
Is subject wearable?	If not, abort.
Is subject worn?	If yes, abort.

Checks done by the verb epilogue

Check	Action
--	

Artifacts used

Artifact	Purpose
f_wearable	Determine whether the subject can be worn.
f_worn	Determine whether the subject is already worn.

Ambiguity priority

Priority	Conditions
1	Subjects that are wearable and not worn.
2	Subjects that are wearable and worn.
3	All other subjects.

Remarks:

--

xyzzzy

Purpose:

Courtesy to Colossal Cave adventure.

Supported syntaxes:

Syntax	Normal behavior
Everything, only has a default	Prints a message that this is not Colossal Cave.

Checks done by the verb prologue

Check	Action
Is actor alive?	If not, abort.

Checks done by the verb epilogue

Check	Action
--	

Artifacts used

Artifact	Purpose
--	

Ambiguity priority

Priority	Conditions
--	

Remarks:

--

yell

Purpose:

Yelling.

Supported syntaxes:

Syntax	Normal behavior
Everything, only has a default	Prints "Aaaargh".

Checks done by the verb prologue

Check	Action
Is actor alive?	If not, abort.

Checks done by the verb epilogue

Check	Action
--	

Artifacts used

Artifact	Purpose
--	

Ambiguity priority

Priority	Conditions
--	

Remarks:

--

Annex: verb template

Use this template to define additional verbs. Text starting with '#' are comments.

The simplest form of a verb is \$VERB name ENDVERB.

```
$VERB name SYNONYM name2
```

```
PROLOGUE
```

```
<actions to be performed before anything else for this verb is done>
```

```
EPILOGUE
```

```
<actions to be performed after everything for this verb has been done>
```

```
"user command to respond to"
```

```
<...code...>
```

```
"user command to respond to"
```

```
<...code...>
```

```
DEFAULT
```

```
<code to execute wen nothing else fired>
```

```
ENDVERB
```