XVAN 2.6

-- installation and user guide --

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

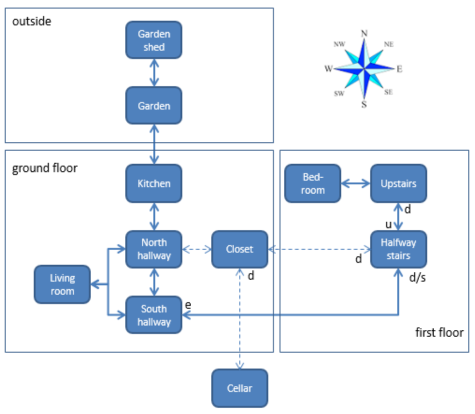

Table of Contents

Installing XVAN

XVAN consists of 2 executable files: Compiler and Interpreter. Depending on view settings, in Windows the files may display as Compiler.exe and Interpreter.exe.

Installation is easy: just copy the files Compiler and Interpreter to the directory where you want to use them.

XVAN does not make changes to the Windows Registry.

Using the XVAN Compiler

The compiler expects one or more text files as input. Use a real text editor like notepad, vi or textedit to create and edit these files. Filenames and extensions are arbitrary.

In the sample files that come with the installation, the story files have a .xvn extension:

Windows

In Windows, double-click the Compiler file to start the program and enter the input file and output file names when prompted.

Or - in a CMD window in the folder where the compiler program is located - type:

"Compiler -i <input file> -o <output file>". If you omit one or both filenames, the compiler will ask for it.

Linux

In Linux, from the directory where the compiler program is located, type:

"./Compiler -i <input file> -o <output file>". If you omit one or both filenames, the compiler will ask for it.

The compiler will now compile the story and vocabulary files and generate an output file with the filename that was entered earlier. This file is input for the interpreter.

macOS

To start the compiler (in case of problems see last section of this file):

1. From a terminal window
 - cd to the directory with the files.
- type "./Compiler -i <inputfile> -o <outputfile>" (without the quotes)
or
- type "./Compiler". The compiler will prompt for the file names.

2. From the Finder:
Navigate to the folder with the files.
The first time you start the compiler:
    Left-Control-click on the Compiler icon (or right-click)
    Select 'open' from the pop-up menu

Click 'open' in the window that shows up.
The compiler will prompt for the input and output file names.

The next times you want to start the compiler, just double-click the Compiler icon.

In case the compiler does not start but opens as a text file with rubbish do the following:
Mark the compiler file as executable:
- From a terminal window cd to the directory with the files
- type "chmod u=rwx Compiler" (without the quotes)

Now start the compiler again

Compiler – debug info

Adding the option '-d' (without the ') on the command line will make the compiler add additional debug information to the output file.
The interpreter will detect that the file contains debug information and it will change the prompt from '> ' to 'debug>' as a reminder that this is not a file for final release.

Using the XVAN Interpreter

The interpreter expects a binary file that was generated by the compiler (the output file that was entered with the compile command).

Console(terminal) version of the interpreter

Windows

In Windows, double-click the Interpreter file to start the program and enter the story file when prompted.

Or - in a CMD window in the folder where the compiler program is located - type:

"Interpreter -i <story file>". If you omit the filename, the interpreter will ask for it.

Linux

In Linux, from the directory where the interpreter program is located, type:

"./Interpreter -i <story file>". The program will open in a Console and load the story file. If you omit the filename, the interpreter will ask for it.

macOS

To start the interpreter (in case of problems see last section of this file):

3. From a terminal window
 - cd to the directory with the files.
- type "./Interpreter -i <inputfile> " (without the quotes)
or
- type "./Interpreter". The interpreter will prompt for the file names.

4. From the Finder:
Navigate to the folder with the files.
The first time you start the interpreter:

Left-Control-click on the Interpreter icon (or right-click)
Select 'open' from the pop-up menu
Click 'open' in the window that shows up.
The interpreter will prompt for the input file name

The next times you want to start the interpreter, just double-click the Interpretericon.

In case the interpreter does not start but opens as a text file with rubbish do the following:
Mark the interpreter file as executable:
- From a terminal window cd to the directory with the files
- type "chmod u=rwx Interpreter" (without the quotes)

Now start the interpreter again

Glk version of the XVAN Interpreter

The Glk API offers richer I/O functionality (I used it to create the status window).

The Glk Windows executable requires the following DLLs:

- Glk.dll
- ScaleGfx.dll

In case either one is missing, the Glk Interpreter will throw an error message when started.

Note: the GLK version of the interpreter does not support entering the filename(s) on the command line (Glk does not support argc, argv). The story file for the Glk Interpreter must always be named 'out.dat'.

Note: the GLK version is available for Windows and Linux. There is currently no Glk library for macOS.

IFI version of the XVAN Interpreter (IFI-XVAN)

IFI-XVAN offers a graphical user interface (GUI). In IFI-XVAN the XVAN back-end connects to the Brahman GUI from Strandgames.

IFI XVAN is best downloaded from the Brahman repository that is maintained by Strandgames. The repository also contains instructions how to build the application. New XVAN versions will also be merged in the Brahman repo.

Communication between XVAN back-end and IFI front-end can be handled by including the IFI library in the XVAN story source.

Note: there is only one version of the XVAN compiler: the console version. Story files produced by the compiler can be run by all interpreter versions.

Interpreter - general

The interpreter may generate the following files during play:

- save.dat to store game progress (save function);
- transcript.txt to log all input and output (transcript function).

Sample stories

XVAN 2.4 comes with 3 sample stories:

- Cloak of Darkness
  The Cloak of Darkness is sort of "Hello world" for text adventures. More info on Cloak of Darkness can be found at http://www.firthworks.com/roger/cloak/
- Escape!
  Escape! Started out as a set of ‘difficult’ situations that I wanted XVAN to be able to handle. Over time it evolved into a real (thin plot) story. I submitted Escape! In The Spring Thing 2019 and it received badges for best puzzles and best npc.
- Bronze
  An XVAN port of Emily Short's Bronze (http://inform7.com/learn/eg/bronze/index.html). Bronze is written in Inform 7; I made a port to demonstrate that XVAN can handle a full-sized game, that demonstrates a range of classic IF functionality.

Tutorial

A tutorial is included with the distribution. The tutorial teaches how to set up an XVAN text adventure game from scratch. It starts with setting up the (reusable) basics: basic verbs, default messages, a mechanisms for kicking off a game, mechanism for moving around, etc. Next, a simple map and player object are implemented, followed by the objects in the game. Whenever necessary, additional verbs are implemented. Finally, some style options, the difference between the console and Glk interpreter and disambiguation rules are explained.

XVAN Library

The XVAN Library contains ‘normal’ XVAN code: a number of verbs and artefacts to get a head start. The Library is a file that must be in the same directory as the compiler and story file(s). Check the sample stories escape.xvn or bronze.xvn to see how to include the Library in your own story (including is just 1 line of code).

IFI Library

The IFI Library contains ‘normal’ XVAN code to handle the communication between the XVAN back-end and the IFI GUI. The IFI Library is a file that must be in the same directory as the compiler and story file(s). Check the sample storie files escape.xvn or bronze.xvnto see how to include the Library in your own story.

Compatibility

The binary output file that is generated by the compiler is platform independent. I.e. an output file that is generated by a Linux compiler can be read by a Windows interpreter and vice versa. The same counts for the save game file "save.dat".

Testing

I have tested XVAN on the following platform versions:

- Windows 7 64 bit;
- Windows 7 32 bit;
- Windows 10 64 bit;
- Linux Mint 17.1 (Rebecca);
- macOS High Sierra.