

Set-Pos-In-Game v1.47

Real time object placing; by Faguss (ofp-faguss.com)

1. Requirements

any version of Fwatch
Recommended 1.15

<ftp://ftp.ofpr.info/ofpd/utls/fwatch.zip>
<http://ofp-faguss.com/fwatch/download>

2. Installation

- **Installing Fwatch:** extract "fwatch.zip" to the Operation Flashpoint folder
- Extract *Set-Pos-In-Game.rar* archive contents to the Operation Flashpoint folder

3. Usage

Quickstart:

- If you haven't used the installer then copy missions from the `\Set-Pos-In-Game\Demo missions\` to `\Users\`
- Launch game with the Fwatch enabled (double-click on *fwatch.exe*).
- Select Mission Editor on Nogova. Load mission *SPIG-Demo*.
- Select *Edit* from the action menu. See control list on page 3 on how to move objects.

SPIG in my mission - adding dialogs:

- **Automatically:** If you have Fwatch 1.14 or newer then just launch the script and an option to include dialog will appear.
- **Manually:** If mission folder does not have *Description.ext* then create it (plain text format).
In the *Description.ext* add line:

```
#include "Set-Pos-In-Game\Dialogs.hpp"
```

SPIG in my mission - adding ability to launch the script:

- In the script *Init.sqs* or in the player initialization field (in the Mission Editor) add line:

```
Player AddAction ["Edit", "..\Set-Pos-In-Game\Start.sqs"]
```

SPIG in my mission - adding pre-existing objects: (OPTIONAL)

- Define array SPIG_OBJS and fill it with objects' names. For example:

```
SPIG_OBJS = [a1, a2, a3, a4]
```

4. Custom settings

- **Number of lines in the object list**

Open Set-Pos-In-Game\Dialogs.hpp file in text editor and modify lines 199 and 200. Reload mission.

```
__EXEC(ObjectList_H = 0.8 + AR_modify);
```

- **Always disabled „Zero Velocity” option on start**

Open Set-Pos-In-Game\Start.sqs file in text editor. Modify line 44:

```
SPIG_VELOCITY = false
```

- **Always fully visible options on start**

Open Set-Pos-In-Game\Start.sqs file in text editor. Modify line 41:

```
SPIG_OPT = true
```

- **Change initial script speed**

Open Set-Pos-In-Game\Start.sqs file in text editor. Modify line 43:

```
SPIG_SPEED = 20
```

- **Disable auto on-line update check**

Open Set-Pos-In-Game\Start.sqs file in text editor. Put semicolon at the beginning of the line 77.

```
:[ ] exec "..\Set-Pos-In-Game\Version.sqs"
```

5. Key Bindings

W, A, S, D	Move object
Q, Z	Elevate, descend object
X, C	Rotate object
ALT, SHIFT, CTRL	Holding one of these will change speed rate
BACKSPACE	Undo last position/direction change
ARROWS, NUMPAD	Operate camera
+ and – (keypad), PG UP, PG DOWN	Change camera's distance to the object
F1, F2	Previous / next object (more with SHIFT and CTRL).
F3	Mark object (allows to move multiple objects at once)
F4	Hide / show all markings.
F5	Un / Mark all.
F12	Show / hide menu
TAB (in manual cam)	Show / hide dialog
SPACE (when marked objects; hold it)	Suspend moving multiple or to change direction separately
1 – 5	Quick camera position change
SHIFT or CTRL + 1 or 2	Increase / decrease script speed
V	Display version

6. Options

Load & Save	Open file manager. More info on loading and saving on page 5.
Clone Object	Create object of the same class (using <code>createCam</code> command).
Remove Object	Delete current object.
Insert Object	Enters „Insert Object“ menu. More information on page 4.
Manual Camera	Enter manual camera mode (it doesn't follow the object).
Unit properties	Edit unit properties (for exporting to sqm).
Zero Velocity	Set velocity to 0 for all objects in the array. Works even if left editing mode.
Zero Height	Set object's position to ground level.
Block Axis	Blocks changing position in one of the axis. Argument sets which one.

To **clone / remove multiple** mark at least two objects (with *F3*).

7. Insert Object

This option allows to browse through the list of objects and insert desired one to the game. Click on the *Exit* button to cancel. Button *Add* appends current item to your list and leaves the menu. Alternatively, hold SHIFT while pressing *Add* to remain and add another item.

There are object databases for 40 addons. Script automatically detects which addons you're using and loads the correct bases.

Addon	File name	Addon	File name
3WX Objects 1.1	F3WX_O1.pbo	MAPFACT JOF Objects v1.2	JOF_Objects1.pbo
ALs Training Objects	ASL-MOUTKIT.pbo	MAPFACT Mine Workers v1.2	MAP_MineWorkers.pbo
AGS Buildings 2.6	AGS_build.pbo	MAPFACT Mine Vehicles	MAP_MineVehicles.pbo
AGS Industrial 3.0	AGS_inds.pbo	MAPFACT Military Objects	MAP_MilObj-Pack.pbo
AGS Harbour 1.2	AGS_port.pbo	MAPFACT Military Oil 1.2	MAP_OilAddon.pbo
Berghoff's Meditterrain Nature 1.0	BRG_N4.pbo	MAPFACT Sheds	MAP_Shed.pbo
BFV Plants Pack 0.8	BFV_gecko.pbo	MAPFACT WW2 Armored	MAP_WW2_Panzer.pbo
Editor Update 102	editorupdate102.pbo	MAPFACT WW2 Bunker	MAP_WW2_Bunker.pbo
Editor Update 103	Editor103.pbo	MAPFACT WW2 Construction	MAP_WW2_Kulissen.pbo
KLA Construction Kit	Kla_ConKit.pbo	NIK's Earth Works	NKHole.pbo
KLA Power Lines	Kla_Pali.pbo	NOCH Bunkers	noch_bunker.pbo
L2K German Military Base	L2K_BwKaserne.pbo	NOCH Hesco Bastions	noch_hesc.pbo
L2K Lights	L2K_Rundumlicht.pbo	Project Farmland Objects 1.2	fml_objects.pbo
L2K Submarine Bunker	L2K_UbootBunker.pbo	RKSL Netpack US	RKSL-Netpack-US.pbo
L2K Submarine Bunker Objects	L2K_UbootZPack.pbo	RKSL Netpack UK	RKSL-Netpack-UK.pbo
Linker Split's Grass	WWII_grass_config.pbo	RKSL Hangar	RKSL-Hangers.pbo
MAPFACT Baracker	Baracken.pbo	RKSL HAS	RKSL-HASN2.pbo
MAPFACT Bunker Components	BauElementeV01.pbo	Sled88's Grass	erba.pbo
MAPFACT Editor Update	MAP_Editorupgrad.pbo	UWAR Grass 1.23	uwar_grass.pbo
MAPFACT Heaps v2.0	MAP_Heaps.pbo	Winter Nogojev 1.1	kegnoecain_snow.pbo

8. Load & Save Option

Position of the objects in the list may be stored in a text file. User may choose to save:

- exact X, Y, Z coordinates
- position relative to the current object (anchor)
- coordinates and properties for the sqm file

There is a limit of 20 files per each save type.

Loading script uses objects in the current list.

- Missing objects will be created (with `createCam` command).
- Exceeding objects will not be removed.
- Script ignores class mismatch.

Files are saved to `\watch\mdb\`. File name syntax is:

```
- spig_<type><ID>.txt
```

Where `<type>` may be `pos`, `rel` or `sqm` and `<id>` is the number of file. For example:

```
spig_pos10.txt
```

Each save file contains:

- `info` array – stores information about save (displayed in the file manager)
- `settings` array – global variables values (script options)
- `spig_vars` array – information about the save file used by `Spig_Load.sqs` script
- `spigX` arrays – where `X` is an identification number. Contains information about objects.

Array contents in `spig_pos`:

```
spigX = [ X, Y, Z, dir, class]
```

X, Y, Z – coordinates

Dir – direction

Class – class name

Array contents in `spig_rel`:

```
spigX = [ A, B, Z, dir, class]
```

A, B, Z – distance from the anchor

Anchor object has a little different array:

```
spigX = [ X, Y, Z, dir, class, "anchor"]
```

Array contents in *spig_sqm*:

There are 20 elements:

- 0 – Pos X
- 1 – Pos Y
- 2 – Pos Z
- 3 – Side
- 4 – Class
- 5 – Rank
- 6 – Player control
- 7 – Special
- 8 – Info age
- 9 – Vehicle lock
- 10 – Name
- 11 – Skill
- 12 – Initialization
- 13 – Health/Armor
- 14 – Fuel
- 15 – Ammunition
- 16 – Probability of presence
- 17 – Azimut
- 18 – Condition of presence
- 19 – Placement radius

9. Loading (*pos/rel*) save file in your mission

With Fwatch 1.11 or newer:

- Press button *Save & Convert* in the file manager menu.
- Copy `\Set-Pos-In-Game>Loading in mission\SPIG_Load.sqs` to the mission directory
- In the *Init.sqs* type: `<file name> exec "SPIG_Load.sqs"`

With older Fwatch versions:

- Copy `\Set-Pos-In-Game\exel\spigConvert.exe` to the `\fwatch\mdb\` directory
- Drag and drop save file onto *spigConvert.exe*
- Move converted save file to the mission directory
- Copy `\Set-Pos-In-Game>Loading in mission\SPIG_Load.sqs` to the mission directory
- In the *Init.sqs* type: `<file name> exec "SPIG_Load.sqs"`

spigConvert.exe program converts save file to a format readable by OFP. `SPIG_Load.sqs` reads the converted data and loads objects position. It does not require Fwatch, dialogs or the *SPIG* script.

To use pre-existing objects fill `SPIG_OBJJS` array before launching the script.

Script execution syntax is:

```
<file name> exec "SPIG_Load.sqs"
```

OR

```
[<file name>, <number>, <anchor pos>, "norefresh"] exec "SPIG_Load.sqs"
```

Arguments marked red are optional.

`<file name>` is the name of the converted save file in your mission directory.

`<number>` is the time interval between creating objects. Game may slow down if you'll try to load too many objects at once. The solution is a pause between creating objects.

`<anchor pos>` is an array `[X,Y,Z,DIR]` with custom anchor coordinates (works only with *rel* type). It must contain at least two items.

"norefresh" - with this string objects are created in relation to the original anchor position (position is not updated during loading). Works only with the *rel* type.

In the multiplayer `SPIG_Load.sqs` should be executed only by the server.

Objects loaded with this script are assigned to global variables:

```
spigXobj
```

Where *x* is object number (see save file). Example:

```
spig2obj setDamage 0.5
```

Script may create soldiers with `createUnit` command. To activate this function define `SPIG_SOLDIERS` array. It's syntax is:

```
SPIG_SOLDIERS = [ [ <id>, <group>, <rank> ], ... ]
```

Where `<id>` is variable number (see save file) and `<group>` is a group that a new soldier will join. Rank value is optional.

In OFP every created soldier must join an existing group. That means at least one unit have to be inserted through mission editor in order to spawn an army.

You don't have to define array for every soldier. Soldiers starting from `<id>` and next will join the same group. For example:

```
SPIG_SOLDIERS = [ [ 1, group mike ], [ 6, group player ] ]
```

Units from 1 to 5 will join mike's group and unit 6 and others will join player's group.

Player's group is the default one so if you'll leave this array empty...

```
SPIG_SOLDIERS = [ ]
```

...everyone will join your group.

During loading `SPIG_LOADING` variable is set to true. Scripts cannot work concurrently because they use the same global variables. So to load multiple save files use:

```
<file name> exec "SPIG_Load.sqs"  
  
@SPIG_LOADING  
  
@!SPIG_LOADING  
  
<file name> exec "SPIG_Load.sqs"
```


10. Exporting (*sqm*) save file to the mission.sqm

With Fwatch 1.11 or newer:

- Press button *Save & Export* in the file manager menu.
- Go back to Mission Editor.
- Load and save mission

With older Fwatch versions:

- Copy `\\Set-Pos-In-Game\\exe\\spig2sqm.exe` to the mission folder
- Drag and drop save file from the `\\fwatch\\mdb\\` onto `spig2sqm.exe`
- Load mission in the Mission Editor and then save it.

This method allows to add objects directly to the mission core file.

- Exporting soldiers is not supported.
- For each execution a backup file is created
- Do not use `.sqm` save file with `SPIG_Load.sqs`

In conclusion: use *pos/rel* type of save when you want to create objects in the middle of the mission. If you want to have the objects from the start – use `.sqm` type.

11. Credits

Thanks to Anguis, Macser and STGN for testing.

UWAR Grass, Linker Split's Grass, Sled88's Grass databases made by Rožek. *NOCH Bunkers, NOCH Hesco Bastions* databases made by Kenoxite.

Script uses dialogs (*RscDisplayCustomArcade, RscDisplayArcadeUnit*) taken from the original game resource file.

12. Version history

1.0 (01.06.10)

First release.

1.1 (09.06.10)

- improved dialog
- partially fixed issues with setting object height
- relative position updated when bringing back markings with *F4*
- *fixZ* is not stored in save files
- seamless switch between file manager and editing dialog
- script 'remembers' last selected file in file manager
- script is terminated when player dies
- **Zero Velocity** option is turned on by default

1.2 (17.06.10)

- added stuff which allows to load save in the mission
- fixed height issue when **Zero Velocity** option was disabled
- fixed issues with moving multiple objects
- fixed saving **Relative Position** issues
- *fixX, fixY* are not stored in save files
- improved file manager script
- saving relative camera direction (in „Relative Position” save type)
- *spig_vars* array now saved to file
- speed rate depends on benchmark
- *V* key quits manual camera mode

1.21 (22.06.10)

- fixed helicopter constant elevation when disabled **Zero Velocity** option
- fixed loading **Relative Position** problem (objects weren't properly placed)
- fixed loss of control problem caused by moving object over hilly terrain
- reduced camera warping

- script for moving multiple objects now handles object height when it's falling
- shortened message display time

1.3 (24.07.10)

- added **Insert Object** option
- can change script speed (*SHIFT,CTRL* + 1,2)
- changed installation: copy *Set-Pos-In-Game* to OFP root folder instead of mission folder
- cloned objects have same direction
- data are saved with `qwrite` instead of `write` (faster saving, stores font case)
- fixed bug with creating objects when loading coordinates
- fixed bug when removing marked objects
- implemented **Clone / Remove** multiple objects
- improved **Clone / Remove** object code
- manual camera mode variables aren't reset each time script is launched
- modified *spigConvert.exe* program
- no error message if there are no objects in *SPIG_OBJJS* array
- numpad also operates camera
- *PG UP* and *PG DOWN* works like + and – on numpad
- quick camera keys (1 – 5)
- removed delay in displaying messages
- script does not run if *Fwatch* is disabled
- *SHIFT* now affects camera speed
- single key press opens dialog in manual camera mode
- *SPACE* locks multiple objects (can change direction without changing position)
- *SPIG_OBJJS* array is created even if user has not defined it
- *SPIG_Load.sqs* assign global variable to loaded object (`spigXobj`)
- *SPIG_Load.sqs* script can now create soldiers with `createUnit` command
- *SPIG_Load.sqs* – fixed bug when anchor was the last object
- *SPIG_Load.sqs* – new execution syntax (passing time delay)
- *SPIG_Load.sqs* – no error message when starting editing after loading
- *SPIG_Load.sqs* – *SPIG_OBJJS* is created even if user has not defined it
- *TAB* opens dialog in manual camera mode (instead of *RIGHT WIN* key)
- *V* key checks script version

1.31 (07.10.10)

- added `bfv_gecko` and `fml_objects` databases
- auto version check at startup
- category name localized in databases for mapfact addons
- changed condition when to process databases (if no `ofpres` loaded)
- fixed message when removing first object from list
- fixed land vehicle constant elevation when disabled **Zero Velocity**
- implement widescreen solution for dialogs
- improved script displaying messages
- *Insert Object* - no problems with leaving with *ESC* key

- *Insert Object* - camera does not reset after adding object with *SHIFT*
- *Insert Object* - fixed wrong message when added multiple objects
- *SpigConvert.exe* – program now support drag & drop
- *SPIG_Load.sqs* – anchor coordinates are updated during loading relative position

1.32 (06.11.10)

- added 5:4 support

1.33 (27.11.10)

- added 12 new databases
- added document „How to create a database”
- optimization – less calls to *fwatch*

1.34 (14.02.11)

- added „mark all” option (F5)
- fixed category order in *BRG_N4* database
- new menu *Edit Unit*
- new program *save2sqm.exe*
- new save type

1.35 (23.02.11)

- added height information to *sqm* save type array
- added **Undo last move** feature
- fixed error message when there was no objects in the list
- fixed error with object creation during loading *sqm* type
- fixed error when entering **Edit Unit** menu after *pos/rel* load
- renamed *cla* save type to *sqm*
- *spig2sqm.exe* - now adds code to set object's height
- *spig2sqm.exe* - fixed – wasn't working with `class vehicles absence`
- *SPIG_Load.sqs* - fixed – temporary object wasn't removed when creating soldiers
- *SPIG_Load.sqs* - removed code to create *SPIG_HEIGHT* array
- *SPIG_Load.sqs* - removed `setFormDir` command

1.36 (11.03.11)

- added 3 grass addons databases (by Rožek)

1.37 (17.10.11)

- added 15:9 aspect ratio support
- added *SPIG-Blank.intro* demo mission
- auto version check every week if user is using *Fwatch* v1.1

- now it's possible to include dialogs in *Resource.hpp*
- *SPIG_Load.sqs* – added `SPIG_LOADING` var – only one script execution at a time
- updated *SPIG-Soldiers.noe* demo mission

1.38 (15.06.11)

- added database for updated Sled88's grass addon
- changed dialogs classnames so it doesn't collide with IGSE script
- updated internet address which script uses to check version

1.39 (06.05.13)

- added databases for NOCH addons
- added option for automatic export to SQM (requires Fwatch v1.11)
- saving (overwriting) is halted if current file couldn't be deleted
- *spig2sqm.exe* – implemented silent mode for launching program from the game
- *spig2sqm.exe* – backup files have number in their name

1.4 (28.05.13)

- added option to automatically include SPIG dialog in `description.ext` if not present
- File Manager wasn't storing last selected file of the sqm type - fixed
- fixed File Manager window background (enlarged)
- loading sqm files wasn't functioning – fixed
- removed version check
- *SPIG_Load.sqs* uses new arguments (custom anchor position and no refresh)
- *SPIG_Load.sqs* script was skipping part where anchor is assigned to global var – fixed
- *spigConvert.exe* - updated program so it can be launched from the game

1.41 (25.01.14)

- cannot save if there aren't any objects
- fixed bug with loading rel and then saving sqm
- fixed object misplacement after copying multiple
- fixed lockout bug from the previous version
- objects were slightly moved after exporting to sqm – fixed
- „unmark all” wasn't always working - fixed
- while holding SHIFT or CTRL you can scroll through objects faster

1.42 (29.08.14)

- script *IncludeDialog.sqs* now works only with the Fwatch 1.13 or newer

1.43 (25.04.15)

- added databases for addons: Mikero's Editor103 and MAPFACT Editor Upgrade

1.44 (30.05.15)

- *SPIG_Load.sqs* – does not load „pos” objects twice anymore
- *SPIG_Load.sqs* – formation dir is set to anchor’s direction in „rel” loading
- *SPIG_Load.sqs* – wouldn’t change existing anchor position to a custom one - fixed

1.45 (18.07.15)

- removed code that used `benchmark` command
- brought back version check

1.46 (01.06.16)

- script checking version now compatible with Fwatch 1.15
- script adding dialog include now compatible with Fwatch 1.15

1.47 (23.08.16)

- Automatically detects screen aspect ratio (requires Fwatch 1.15)
- Removed option to manually set aspect ratio in *Dialogs.hpp*
- Now including *Dialogs.hpp* in the *Resource.cpp* will not work
- Uses Fwatch input multi method (requires Fwatch 1.13)
- User can change height of the object list and number of lines will be adjusted automatically
- Including dialog – works even when `#include` macro is commented
- Including dialog – detects if mission (in Mission Editor) has not been saved
- Fixed bug in version check where it would constantly display message about new version
- Error message is displayed when trying to save / load without selecting position first
- Error message is displayed when trying to load non-existent file
- *SPIG_Load.sqs* – when loading relative with a custom position objects before the anchor were loaded to the original position and not the custom one – fixed
- *spig2sqm.exe* – saves file error information to `log.txt`