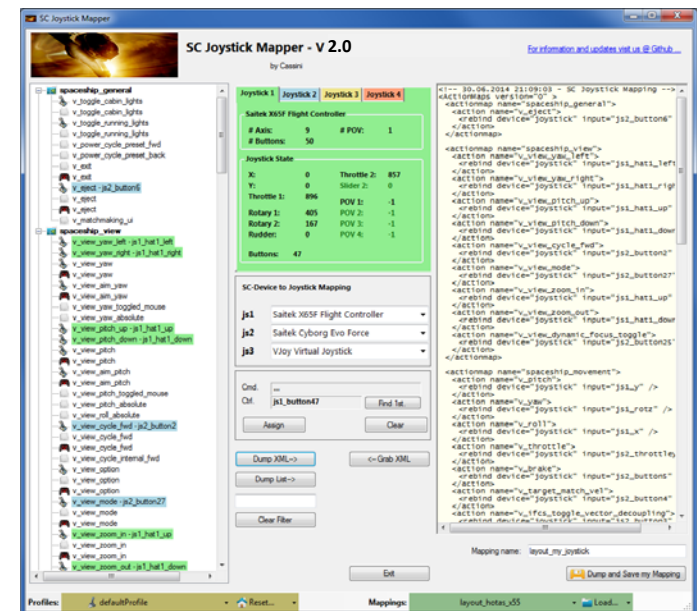


# SC Joystick Mapper Quick Reference Guide V 2.0

20140630 – Cassini  
ChangeLog: see ReadMe.txt

Disclaimer:  
Usual stuff – no warranty whatsoever..  
Freeware – made for the SC community  
Hope it helps and does not suck.  
Have fun in the verse ...



# Updating from V 1.x to V 2.0:

- As the action list taken from the game assets (GameData.pak)  
You have to **manually remove the 'defaultProfile.xml' file if it exists in the same folder as the program file.**  
If the program finds it there it is taken before the one from the game (which is may be not what you wanted)
- You may however use this priority for any purpose  
i.e. place a defaultProfile.xml file in the program directory and it will be taken as action list

# Workflow

- Connect the joystick devices to the PC
- Start from scratch or load an existing map from a file
- Make or refine mappings
- Save the new map to an XML file
- Use it in the game: e.g. `pp_rebindkeys C:\maps\layout_my_joystick`
- **V 2.0: You may load and save the map directly from your game folders so next time you just use `pp_rebindkeys layout_my_joystick`**

Note: the predefined actions are the ones found in the AC game default profile – it is likely that some of them will not work at all as the game is not finished. There is no proper description for which one does what – you may get help in SC Forums.

As I had my issues with missiles here a finding..

To reallocate the missile fire command you should map the following 2 actions to the same joystick button:

- `v_target_missile_lock_selected`
- `v_weapon_launch_missile`

BTW: if you copy e.g. “`pp_rebindkeys C:\maps\layout_my_joystick`” from notepad you may use Ctrl-V to paste it in-game into the console – saves you some typing...

# The GUI ...

Action tree and mappings

XML dump of the mappings used

The screenshot shows the SC Joystick Mapper V 2.0 interface. On the left is a tree view of actions, including categories like 'spaceship\_movement' and 'v\_target\_match\_vel'. The top center displays 'Detected Joystick devices (up to 8 are shown)' with a table for Joystick 1 through 4, listing properties like # Axis, # POV, X, Y, Throttle, Rotary, and Rudder. Below this is the 'SC-Device to Joystick Mapping' section with dropdown menus for js1, js2, and js3. The bottom left shows a 'Dump nice List' and a 'V2: filter the action tree' input. The bottom center has 'V2: Load from game folders' and 'V2: New Reset with options' buttons. The bottom right features 'V2: Save into game folders' and 'V2: Resize the window' buttons. On the right side, an XML dump is visible, with annotations for 'Joystick properties (greyed out ones are not available)', 'Joystick device map (the default is usually OK)', 'Current mapping', 'Action Mapping Buttons', and 'XML Area Buttons'. The bottom status bar shows 'Profiles: defaultProfile' and 'Mappings: layout\_hotas\_x55'.

Detected Joystick devices  
(up to 8 are shown)

Joystick properties  
(greyed out ones are not available)

Joystick device map  
(the default is usually OK)

Current mapping

Action Mapping Buttons

XML Area Buttons

Dump nice List

V2: filter the action tree

V2: Load from game folders

V2: New Reset with options

V2: Save into game folders

V2: Resize the window

# The Joystick Area...

The screenshot shows the SC Joystick Mapper interface. On the left is a tree view of game actions, with 'spaceship\_movement' expanded to show various axes like pitch, yaw, and roll. The main area is divided into sections for each joystick. The first joystick, 'Joystick 1', is a 'Satek X65F Flight Controller' with 9 axes and 50 buttons. The second joystick, 'Joystick 2', is a 'Satek Cyborg Evo Force' with X, Y, and Z axes, rotation, and buttons. The 'SC-Device to Joystick Mapping' section at the bottom shows 'js1' mapped to 'Satek X65F Flight Controller', 'js2' to 'Satek Cyborg Evo Force', and 'js3' to 'VJoy Virtual Joystick'.

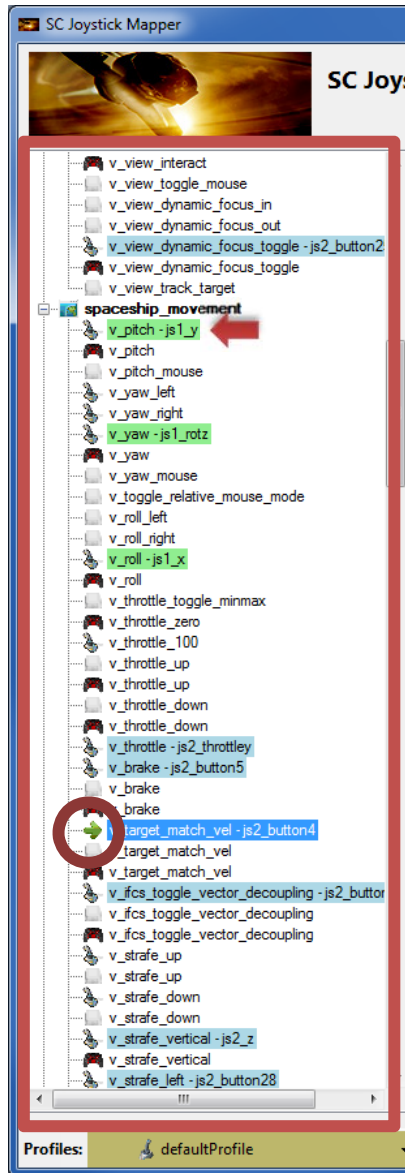
Here I pressed the Button 8 on the Cyborg Evo Joystick to capture the image

The tabs represent the joystick devices found connected to the PC also the number 1..8 shows the order the PC reports them which is crucial to the mapping as this will result in the js\_1, js\_2 .. Names used to build the command name. The elements are the ones the joystick seems to support – greyed ones are not available for this device.

The *SC-Device to Joystick Mapping* can be used if the default assignment “Joystick 1 -> js\_1” does not match what the CryEngine is using. – Usually the default should work. You may only remap js1..js3 - 4..8 will remain as detected.

Just hit any button, Axis and see how things are changing.

# The Action Tree ...



The tree is initially built from the known actions which are grouped along 'actionmaps' e.g. 'spaceship\_movement'.

Each action is either a predefined joystick or keyboard action – this is given by the SC default profile.

By 'rebinding' or mapping an action with a different control one does **replace** the default one i.e. **overwriting keyboard actions will result in not having them available on the keyboard once you load the map in the game!**

However no damage is done! This mapping is only valid until you exit the game or type `pp_rebindkeys` without a name

If actions are mapped (as shown) the color indicates to which joystick the mapping goes.

**v\_pitch – js1\_y** then means that the action v\_pitch (joystick per default) is rebound to the joystick 1 (green) and there the Y-axis control.

If the background is white - there is no current mapping given.

Unmapped actions are ignored.

Click on any action to make it the used action in the mapping area.

Once selected it is marked with the green arrow.



# The Mapping Area...

Whenever you click on an action in the Action Tree it is copied into Cmd. and can be mapped to a Control.

The Control (Ctrl.) is the last joystick item you activated on the currently shown joystick tab.

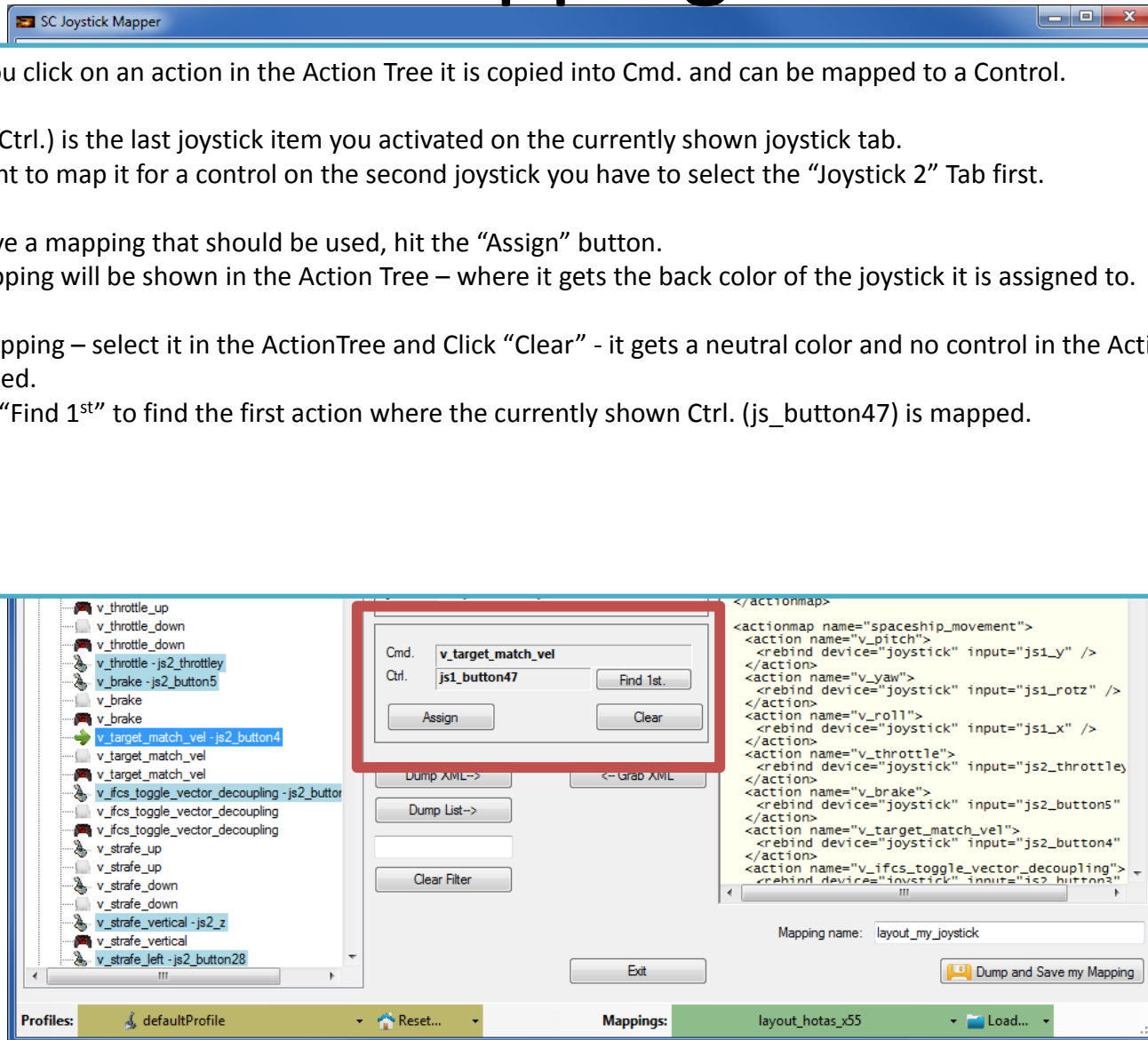
I.e. if you want to map it for a control on the second joystick you have to select the "Joystick 2" Tab first.

Once you have a mapping that should be used, hit the "Assign" button.

The new mapping will be shown in the Action Tree – where it gets the back color of the joystick it is assigned to.

To clear a mapping – select it in the ActionTree and Click "Clear" - it gets a neutral color and no control in the ActionTree – it is now unmapped.

You may use "Find 1st" to find the first action where the currently shown Ctrl. (js\_button47) is mapped.



# The XML Area...

Mappings are sent to the game using XML formatted files.  
The XML Area is where you may find the mapping after hitting the 'Dump' button.

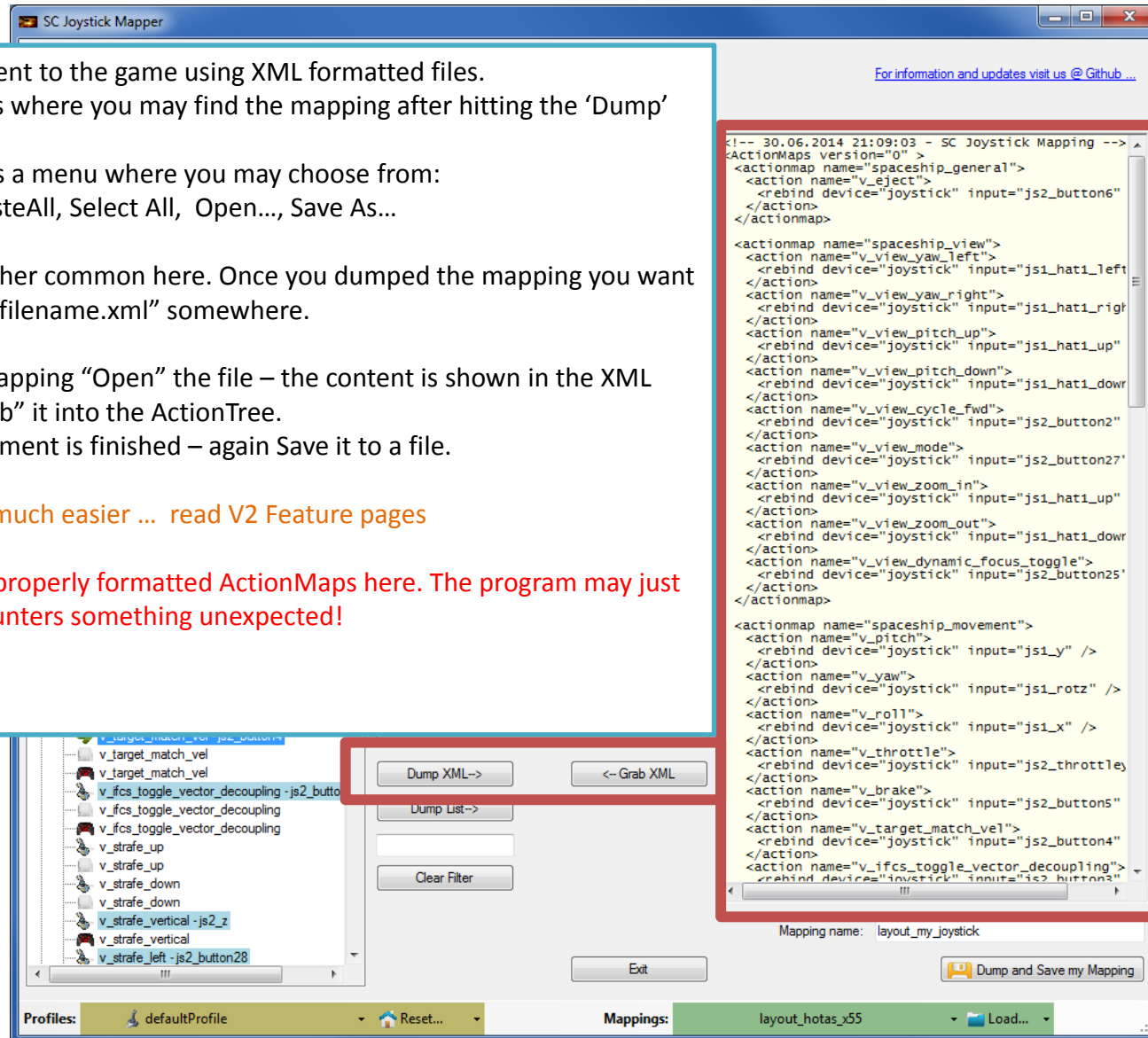
Rightclick opens a menu where you may choose from:  
Copy, Paste, PasteAll, Select All, Open..., Save As...

The usage is rather common here. Once you dumped the mapping you want to "Save" it as "filename.xml" somewhere.

To refine any mapping "Open" the file – the content is shown in the XML Area, then "Grab" it into the ActionTree.  
Once the refinement is finished – again Save it to a file.

Load and Save much easier ... read V2 Feature pages

Note: only use properly formatted ActionMaps here. The program may just break if it encounters something unexpected!





# The XML Area...

If you hit “Dump List” a formatted list of the mapped actions is written into the XML area.

You may use the “Save As..” menu to save it e.g. as TXT file.

The image shows a screenshot of the SC Joystick Mapper application. The main window displays a list of mapped actions on the left and a large XML area on the right. The XML area contains a formatted list of mapped actions, such as:

```
-- 30.06.2014 21:34:03 - SC Joystick Mapping --  
*** spaceship_general  
V_eject - js2_button6  
*** spaceship_view  
V_view_yaw_left - js1_hat1_le  
V_view_yaw_right - js1_hat1_ri  
V_view_pitch_up - js1_hat1_up  
V_view_pitch_down - js1_hat1_do  
V_view_cycle_fwd - js2_button2  
V_view_mode - js2_button2  
V_view_zoom_in - js1_hat1_up  
V_view_zoom_out - js1_hat1_do  
V_view_dynamic_focus_toggle - js2_button2  
*** spaceship_movement  
V_pitch - js1_y  
V_yaw - js1_rotz  
V_roll - js1_x  
V_throttle - js2_thrott1  
V_brake - js2_buttons5  
V_target_match_vel - js2_button4  
V_ifcs_toggle_vector_decoupling - js2_button3  
V_strafe_vertical - js2_z  
V_strafe_left - js2_button2  
V_strafe_right - js2_button2  
V_strafe_longitudinal - js2_rotx  
V_newtonian_yaw - js1_x  
V_newtonian_pitch - js1_y  
V_newtonian_brake - js2_buttons5  
V_ifcs_toggle_safety - js2_button2  
V_afterburner - js2_button1  
*** spaceship_targeting  
V_target_cycle_all_fwd - js1_button1  
V_target_cycle_all_back - js1_button8  
V_target_missile_lock_focused - js1_button2  
V_target_cycle_hostile_fwd - js1_button1  
V_target_nearest_hostile - js1_button1  
*** spaceship_weapons  
V_attack1_group1 - js1_button1  
V_attack1_group2 - js1_button4  
V_attack1_group3 - js1_buttons5  
V_attack1_group4 - js1_button4  
*** spaceship_missiles  
V_weapon_cycle_missile_fwd - js2_button2  
V_weapon_launch_missile - js1_button2  
*** spaceship_defensive  
V_weapon_launch_countermeasure - js1_button3  
V_weapon_cycle_countermeasure_fwd - js2_button2  
V_shield_boost_recharge - js1_button1  
V_shield_cycle_presets_fwd - js1_button9  
V_shield_cycle_presets_back - js1_button1  
*** spaceship_auto_weapons
```

A file save dialog is open, showing the file name "T2Mapping.txt" and the file type "Text files (\*.txt)". The dialog is titled "Speichern unter" and shows the file location as "Computer > 1\_APPLIC (E:) > G > StarCitizen > My".

The SC Joystick Mapper interface also shows a "Dump List" button highlighted in red, and a "Dump XML" button. The XML area is also highlighted in red. The mapping name is "layout\_my\_joystick".

# V2 – Features - 1

The screenshot shows the SC Joystick Mapper interface. On the left, a tree view displays various action categories. The 'spaceship\_movement' category is expanded, and a search filter 'thr' is applied, resulting in a list of actions containing 'thr'. The 'Clear Filter' button is highlighted with an orange box. Below the tree, there are buttons for 'Dump XML-->', '<-- Grab XML', 'Dump List-->', and 'Clear Filter'. The 'Clear Filter' button is also highlighted with an orange box. To the right, the XML output is displayed, showing actions like '<action name="v\_throttle">', '<action name="v\_brake">', and '<action name="v\_target\_match\_vel">'. The 'Mapping name' field is set to 'layout\_my\_joystick'. At the bottom, there are buttons for 'Exit' and 'Dump and Save my Mapping'. The status bar at the bottom shows 'Profiles: defaultProfile' and 'Mappings: layout\_hotas\_x55'.

SC Joystick Mapper

SC Joystick Mapper

You may filter the action tree now

Start typing and the tree is reduced to the actions and controls that contain the characters typed

e.g. I typed 'thr' to see my throttles only  
Try button and you get all your assigned buttons only etc.

Click 'Clear Filter' to get back to the complete list again.

Note: this will not change, remove or modify any of your mappings, it just reduces the tree to the ones you are interested in.

Dump XML--> <-- Grab XML

Dump List-->

thr

Clear Filter

```
<action name="v_throttle">
  <rebind device="joystick" input="js2_throttley" />
</action>
<action name="v_brake">
  <rebind device="joystick" input="js2_button5" />
</action>
<action name="v_target_match_vel">
  <rebind device="joystick" input="js2_button4" />
</action>
<action name="v_ifcs_toggle_vector_decoupling">
  <rebind device="joystick" input="js2_button3" />
</action>
```

Mapping name: layout\_my\_joystick

Exit

Dump and Save my Mapping

Profiles: defaultProfile

Mappings: layout\_hotas\_x55

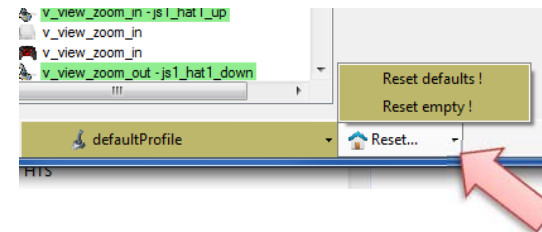
# V2 – Features - 2

## New working with profiles.

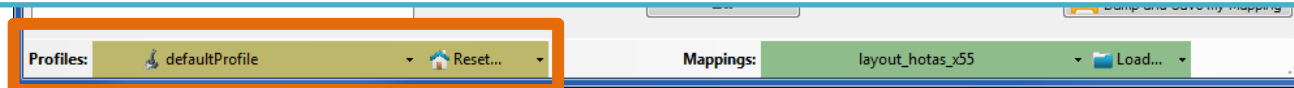
The program gets the actions from the real game asset – so you are always up to the actual values.

From here you may Reset the action list to the following

- RESET EMPTY reverts to just an action list without any mappings
- RESET DEFAULTS loads the Joystick actions mapped with what CIG is providing



Note: as CIG is providing a number of defaultProfiles you may chose one of those – however using the **defaultProfile** is usually the best option  
(This may be work in progress by CIG...)



# V2 – Features - 3

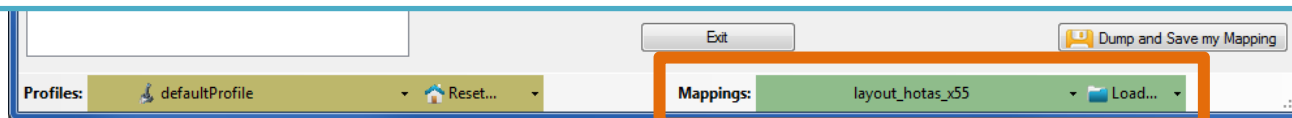
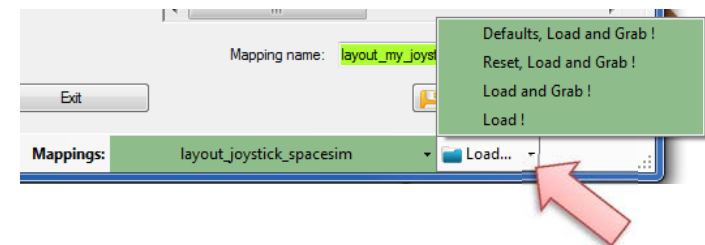
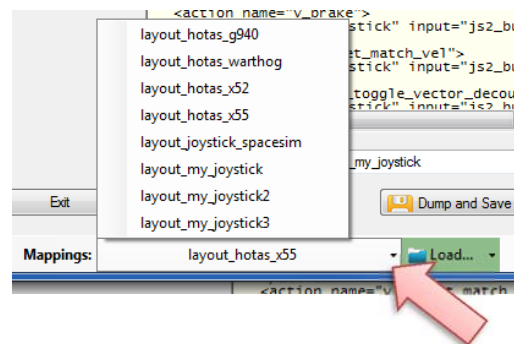
## New working with actionmaps (Maps, Mapping etc..)

The program gets the actionsmaps from the real game asset – so you are always up to the actual values.  
(..\StarCitizen\CitizenClient\Data\Controls\Mappings)

From here you may first chose a map, then 'Load' the actionmap – this will overwrite you XML window in any case

- LOAD loads the map into the XML window only
- LOAD and GRAB loads the map into the XML window and clicks Grab i.e. merges the existing mapping with the one loaded
- RESET, LOAD and GRAB first Reset (empty) the action list (all mappings cleared) then it loads and grabs the new map
- DEFAULT, LOAD and GRAB first Reset (defaults) the action list then it loads and grabs the new map and merges them with the defaults

See last page for some common workflows  
And how to handle them easily



# V2 – Features - 4

## New working with your own actionmaps

The program not only gets the actionsmaps from the real game asset – but also can save your maps there.

(...\StarCitizen\CitizenClient\Data\Control s\Mappings)

1. Type a name (limitations see note)
2. Hit the button – it will then Dump and Save your map into the game folder (well asking you to overwrite it if it exists)

NOTE: your map name has always to start with 'layout\_my\_' to prevent modifying CIGs own actionmaps

Lowercase only, no spaces, tabs allowed else you see the red flag ..

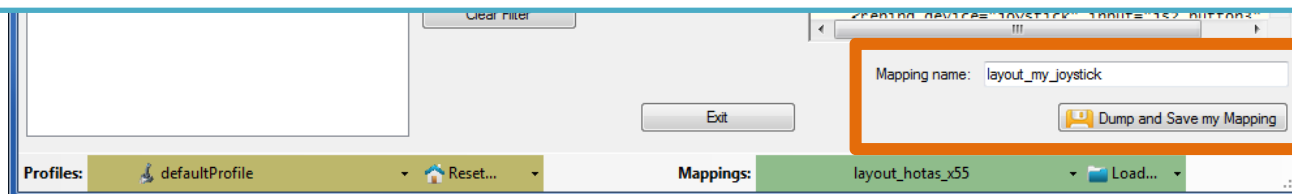
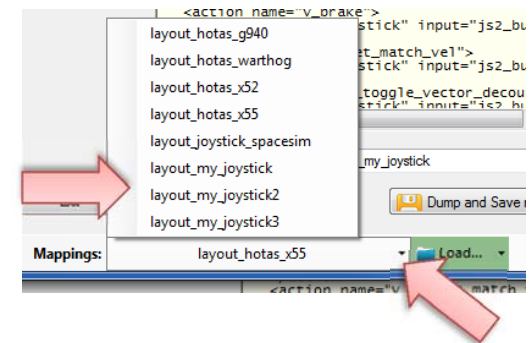
Mapping name: layout\_any

A successful Save will show the green flag

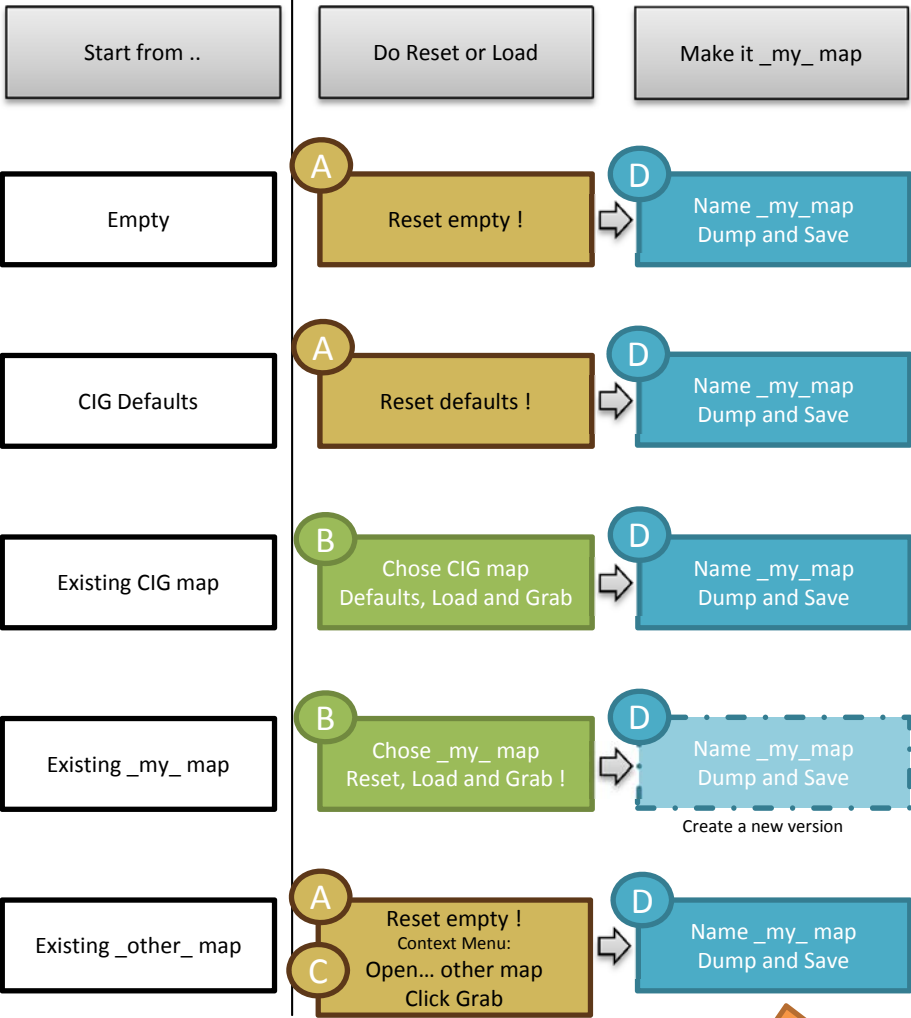
Mapping name: layout\_my\_joystick3  
Dump and Save my Mapping

Your own maps will then show up like the game provided maps  
pp\_rebindkeys | layout\_my\_joystick should load it into the game

Note: For your convenience each Save also makes a copy of into your personal  
"My Documents\SCJMapper" folder – no work is lost if there is an update  
that cleans the Mappings folder.



# SCJMapper V 2 – Common Workflows



The screenshots illustrate the following workflow steps:

- A:** Screenshot of the 'Reset defaults!' and 'Reset empty!' buttons in the software interface.
- B:** Screenshot of the 'Mapping name:' dialog box with a context menu open, showing options like 'Defaults, Load and Grab!', 'Reset, Load and Grab!', 'Load and Grab!', and 'Load!'.
- C:** Screenshot of a context menu over a code editor showing options like 'Copy', 'Paste', 'Paste (Replace all)', 'Select All', 'Open...', and 'Save as...'.
- D:** Screenshot of the 'Dump and Save my Mapping' dialog box.
- E:** Screenshot of the 'Assign Cntrl.' dialog box showing a command 'v\_eject' assigned to 'js1\_button7'.

Flow arrows connect these steps: A → B → C → D → E → D → E → D.