BMS for Cockpit Builders

The numbers of small cockpits being built by simmer increase yearly. And even if most of them are small desktop cockpits, they need specific support from the code to be functional. Since the early days of Falcon, cockpit builders could interact with the shared memory banks to interface their cockpit and BMS added a lot of different features in those banks while maintaining backward compatibility with other flavours of Falcon to ensure that third party hardware remained compatible throughout the updates.

BMS went further by adding cockpit display extraction. That new feature allows to externalize the cockpit display contents to secondary screens connected on the computer (network is not possible). The most obvious application for that (and even though formerly available through the MFD extractor application from "Lightning" for Open Falcon) is for the two Multi Function Displays, which were dearly lacking from our cockpits for the past 10 years. As added bonus the HUD, the DED, PFD & RWR can also be externalized.

Key callbacks for cockpit builders need also to be specific as all switch states need to be implemented, rather than toggles better suited for simple Hotas programming.

And finally, BMS also makes full use of multiple ID joysticks and offers more than a dozen analogue axes beside the normal HOTAS axis to implement functions such as COMMS, MSL and THREAT volumes, HUD and HMS brightness, PITCH, ROLL and YAW trims, etc etc.
The throttle axis can also be set the exact same way as the real throttle through the IDLE CUTOFF code which gets rid of the IDLE detent key which was never realistically implemented in Falcon.

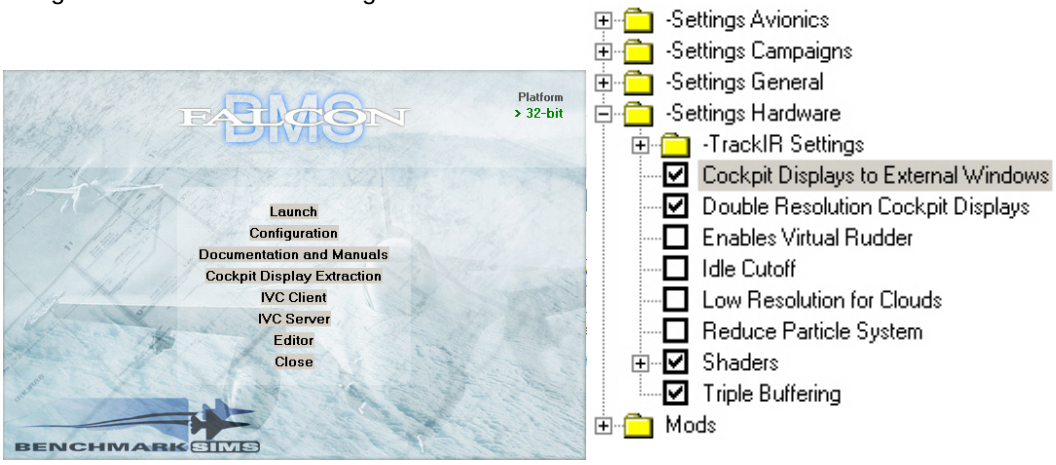
1. Shared Memory

Without the shared memory, there is no cockpit possible. Its purpose is to provide an area where third party cockpit software can fetch the information to reflect the Falcon cockpit events. The most obvious feature is to enable all cockpit lights, but all data for the instrument are also featured.
To explore the structure of the shared memory, open the **Flight Data.h** file which is located in your **Docs\Other Documentation** folder. All bits and their respective address are listed in this file. Hopefully, BMS will publish a technical document explaining all the shared memory at a later stage. In the meantime, you can still use the BMS 2.0 Technical Reference Binary.doc which is located in the **Docs\Falcon 4 Legacy Manuals\5 - BMS 2** folder, which is still the best reference about the shared memory.
The structure of the Falcon shared memory has not changed over the since BMS 2.0 to maintain backward compatibility. Yet some bits present but unsupported in BMS 2.0 have been renamed and are now fully implemented in BMS 4.0. Here's such a list which hopefully will fill the documentation gap until the BMS 4.0 technical manual is published:

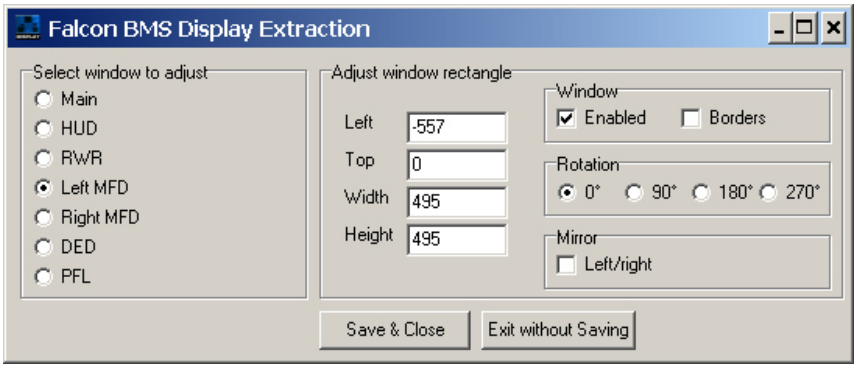
1. ALT (eyebrow) light bit replaced by EQUIP HOT caution panel (address = 0x8, lightbit1)
2. OBS (eyebrow) lightbit replaced by OXYBROW (0x4, lightbit1) that's OXY LOW on right eyebrow.
3. OXY_LOW (0x800000, lightbit2) refers to the OBOGS caution panel.
4. OIL lightbit (0x100, lightbit1) is replaced by FLCS_ ABCD (1light only for the 4 branches of the FLCS). That light is on the TEST panel.
5. DUAL lightbit (0x200, lightbit1) is replaced by the FLCS lightbit. That's the right eyebrow FLCS light.
6. ENG2FIRE (0x2000, lightbit3) is replaced by FLCS BIT RUN which is the upper part of the FLCS bit indicator. (RUN). That one is also used for the FLCS bit magnetic switch.
7. LOCK (0x4000, lightbit3) replaced by FLCS BIT FAIL which is the lower part of the FLCS bit indicator (FAIL) on the FLCS panel.
8. SHOOT (0x8000, lightbit3) bit is replaced by DBUWARN light bit which is the right eyebrow light for the Digital backup unit (DBU)

2. Display extraction

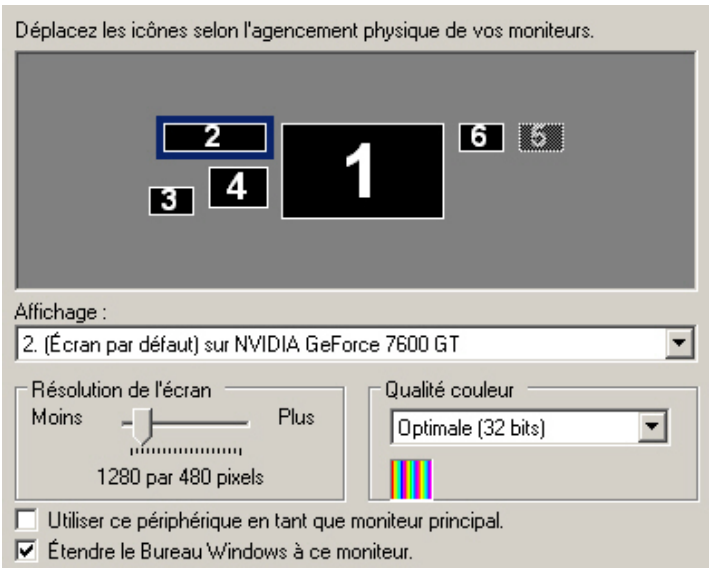
Display extraction is not activated by default. It must be enabled through the BMS Launcher application. Select Configuration to launch the config:



The Double resolution cockpit displays should be checked as well to give you better resolution for those displays. Once enabled, you need to tell BMS where you want to have those displays extracted! That is done with a specific application available from the launcher menu: Cockpit Display Extraction
Each display (MAIN, HUD, RWR, Left MFD, Right MFD, DED & PFL) can then be selected and its format and location adjusted: Window left & top coordinates, window width and height, borders enabled or not and you can also set a rotation angle in 90° steps or mirror the display. Please note that the MAIN window setting is also set through the application and that's your only way to disable borders for the main Falcon screen if you run in –window mode.



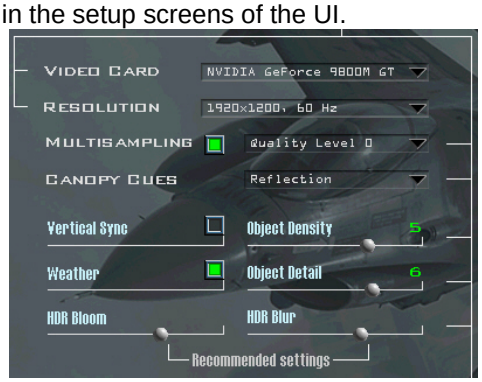
In a Windows multi screens environment, each screen has its own set of coordinates based on the position and resolution of the primary screen. The primary screen always has the 0.0 coordinates on its top left corner. Any screen entering that same coordinates system gets its own coordinates for top left corner (which can be obtained from the multi screen applet (desktop – properties – settings)



Knowing this, finding the correct spot to extract your displays to a specific screen and/or location becomes easy! It is advised not to follow the above picture example and keep your main Falcon screen (#1) on the right most position. Not doing so might create screen shifting problems when entering 3D, but that is mostly for full screen mode and if you activate the display extraction, you are running in windowed mode, so you should be safe.

BMS display extraction works in the 3D pit and there's no point to have it working in the 2D pit since it is not supported anymore. Extraction continues to work in external view but is not updated anymore until you come back to the 3D cockpit.

It is worth mentioning that BMS display extractions has a cost in FPS and it is advised to have Vertical Sync deactivated from the Falcon settings and have it application controlled from your video driver in Windows settings. V sync is ON by default in the setup screens of the UI.



If frame rate remains a problem, it is a known fact that WIN7 is better optimized and FPS results proved to be much better with the latest release of Windows, and especially with ATI drivers. That of course may change with time.

At the time of writing this article, MFDE application is also able to extract the instruments such as the DED, PFD, RWR & HUD (as well as additional flight instruments: ADI, ASI, AOA, VVI, HSI, …) but the application is not yet suitable for the MFDs. Hopefully that issue will be fixed with time. MFDE is network able though, while the BMS display extraction is not. Using a small network can prevent the frame rate issue some of us reported.

As a consequence, MFDE is a welcome addition for the cockpit builder and will be needed running alongside BMS (single computer or networked computers) if one needs to extract instruments.

3. Key Callbacks

Cockpit users don't use toggles switches because it creates synchronisation issues between the switches in the 3D cockpit and the real switches in their cockpit. To ensure that the physical switches always activate the right function, they need all switch states implemented in the key file so that moving the landing gear in the down position sends the gear down key and not the toggle gear key, for instance.
Open falcon had a Cockpit builder .key file with all switch states callbacks and BMS ensured that newly implemented switches have all their states implemented.
The default .key file doesn't have all those callbacks and very often cockpit builders will have to use their own custom key file. Doing so is a long process but a sure way of easing later maintenance on your keys. All it takes is to track the needed callbacks from the BMS callback spreadsheet, add it to the .key file with the empty key line (-1 0 0XFFFFFFFF 0 0 0 1) and add your comments within " "

Here's an example:
Imagine you need to add the Probe heat switch on the TEST panel in your key file. The switch has three positions: ON/OFF/TEST.

1. First find the Probe heat callbacks in the BMS callbacks spreadsheet (they are displayed panel by panel)
SimProbeHeatOn, SimProbeHeatOff, SimProbeHeatTest

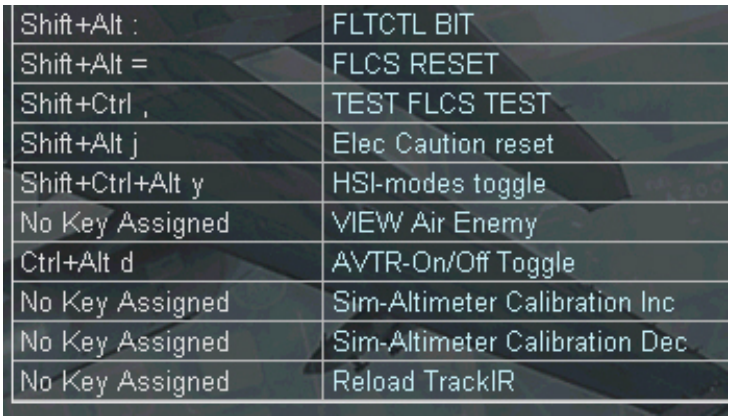
2. Open your key file and create three new lines, each starting with the callback:
SimProbeHeatOn
SimProbeHeatOff
SimProbeHeatTest

3. Add the empty key programming line:
SimProbeHeatOn -1 0 0XFFFFFFFF 0 0 0 1
SimProbeHeatOff -1 0 0XFFFFFFFF 0 0 0 1
SimProbeHeatTest -1 0 0XFFFFFFFF 0 0 0 1

4. Add " and your comments for the purpose of the callback and close the "
SimProbeHeatOn -1 0 0XFFFFFFFF 0 0 0 1 "TEST panel Probeheat ON"
SimProbeHeatOff -1 0 0XFFFFFFFF 0 0 0 1 "TEST panel Probeheat OFF"
SimProbeHeatTest -1 0 0XFFFFFFFF 0 0 0 1 "TEST panel Probeheat TEST"
Remember the comment are displayed in the UI keyfile window, ensure they are short and to the point if you want them to remain visible.

Let me make a further comment about the last digit in the empty key programming line:
-1 0 0XFFFFFFFF 0 0 0 **1**
1 mean display the key code and allow change. If you replace it with **0**, then the key code won't be displayed and if you replace it with a **-1**, the key code will be displayed and changes will not be allowed (that's a good way to protect your keyfile from the controller page.

5. Save the .key file, launch BMS and go to the controller setup page to display the key file. Scroll down until you see the probeheat entries. They should be labelled 'no key assigned'. Assign a keystroke to each entry as usual and save the new keyfile.



Shift+Alt :	FLTCTL BIT
Shift+Alt =	FLCS RESET
Shift+Ctrl ,	TEST FLCS TEST
Shift+Alt j	Elec Caution reset
Shift+Ctrl+Alt y	HSI-modes toggle
No Key Assigned	VIEW Air Enemy
Ctrl+Alt d	AVTR-On/Off Toggle
No Key Assigned	Sim-Altimeter Calibration Inc
No Key Assigned	Sim-Altimeter Calibration Dec
No Key Assigned	Reload TrackIR

Building a keyfile from scratch is a major task. Unfortunately neither the default bms.key, nor the keystrokes. Key or even the cockpitbuilder.key files are up to date with all latest BMS callbacks. Over the years I created mine which you are free to use. The package contains a BMS4.0 keyfile, a complete set for your Cougar based on real Hotas programming and shifted layers for sim specific functions. Feel free to download and modify the full package.
The key file need to be placed in your User\Config folder and loaded through the UI controller setup screen. The TMJ and TMM are as always loaded with Foxy. A graphical layout is also provided.

4. Analogue axis support

Beside the original HOTAS axis, some controls such as volume knobs, trims wheels, brightness knobs, … are not optimally interfaced with keystrokes. To overcome that, BMS is supporting more than a dozen analogue entries for specific cockpit controls:
COMM1, COMM2, MSL & THREAT volumes, PITCH, Roll &Yaw trims, HUD and HMS brightness, Field of View, Zoom, Reticule depression, second engine throttle, INTERCOM … all can be wired to regular 100Kohms potentiometers to give better response to these controls.
Please note that volumes (COM1, COM2, MSL, Threat & INTERCOM) pots are better interfaced with logarithmic pots instead of linear pots.
Note2: The INTERCOM volume is badly named and is rather a master volume.

As in the real cockpit some knobs will need an ON/OFF button at the CCW position. When that is the case, the BMS axis will also need it. The relevant ON/OFF keystrokes will need to be programmed on the pot switch. In BMS, the relevant axes are: HUD brightness, HMS brightness, COMM1 and COMM2 volumes.

A final relevant feature for the throttle control is the cutoff code. Since the early versions of Falcon the way the engine is started is wrong. To start the engine, it was required to depress the idle detent. This is not the case in the real F-16 where the throttle handle rest vertically in the cutoff position and simply needs to be advanced to the idle point, where it snaps back horizontally. The idle detent, better named the cutoff release handle is required to shut down the engine because it is a mechanical pin preventing the pilot to move the throttle back past the idle point and thus shutting down the engine in flight.
The idle cutoff code needs to be activated from the settings video/audio/hardware of the BMS config application. The next step is to define the idle point in the advanced Falcon controller setup. You do that the same way as setting the AB point but with the right mouse button. Position your throttle at idle, click on SET AB with the right mouse button to set the idle point, a red line is displayed at idle.
Move the throttle to Full MIL and click the set AB with the left mouse button. A green line is displayed at the AB point.
Once fully activated, simply moving the throttle past the idle point will start the engine, if the JFS was running properly. The idle detent keystroke is not required anymore.
If that feature can be used on a throttle with a strong cut off point, it is clear that if you lack any mechanical device preventing you to go lower than the idle point (like the default Cougar detents), the engine might shut down unexpectedly. It is better suited for those of us using a throttle rail replica or any comparable system.

5. List (non exhaustive) of known third parties cockpit applications working with BMS

- MFDE (MFD extractor) – Lightning' tools
- CPD (Centre Pedestal Display) – Lightning' tools
- AIC – ViperCore
- DED/UFC & EHSI – Ghost's tools
- UCC – USB Control Centre (XP only)