Elite Dangerous Cougar Display (ED:CD) User Manual

This user guide is designed to get you started as quickly as possible, with minimal clicks. If you're reading this for the first time, then the Quick Start Guide will get you up and running in a minute, and even if you don't read it then you'll get to see something working even if you don't know exactly what's going on!

For those who want to customise the layout screens and voice synthesis options, you should skip ahead.

Table of Contents

Quick Start Guide

I know, you just want to get started without any hassle in the fastest possible time. From reading this to starting Elite with ED:CD should take no more than 2 minutes.

Some things you may want to consider before starting

- You may want to change the Windows taskbar to auto hide.

- Launch Elite with standard user level permissions so that key presses can be sent between Cougar Display and Elite.

- Change the graphics settings within Elite to **Borderless Window** so that you have more control of window positioning and mouse/keyboard. I would recommend starting Elite at step 6 below, so that your main monitor doesn't get too cluttered before you have a chance to see what's going on.

- Plug in your Cougars and USB / additional monitors. Note – if you have touch screen monitors then the use of the Cougars are optional.

- Change your (control panel) region settings to use a period (.) as a decimal separator. Using commas can cause formatting issues when the app tries to convert lists of floating point number to and from string representations.

Now please read the following section: **Running ED:CD for the first time**

Some things you may want to consider doing at some point to get the most from ED:CD

To get the maximum benefit from ED:CD there are some tasks you may wish to consider doing, such as:

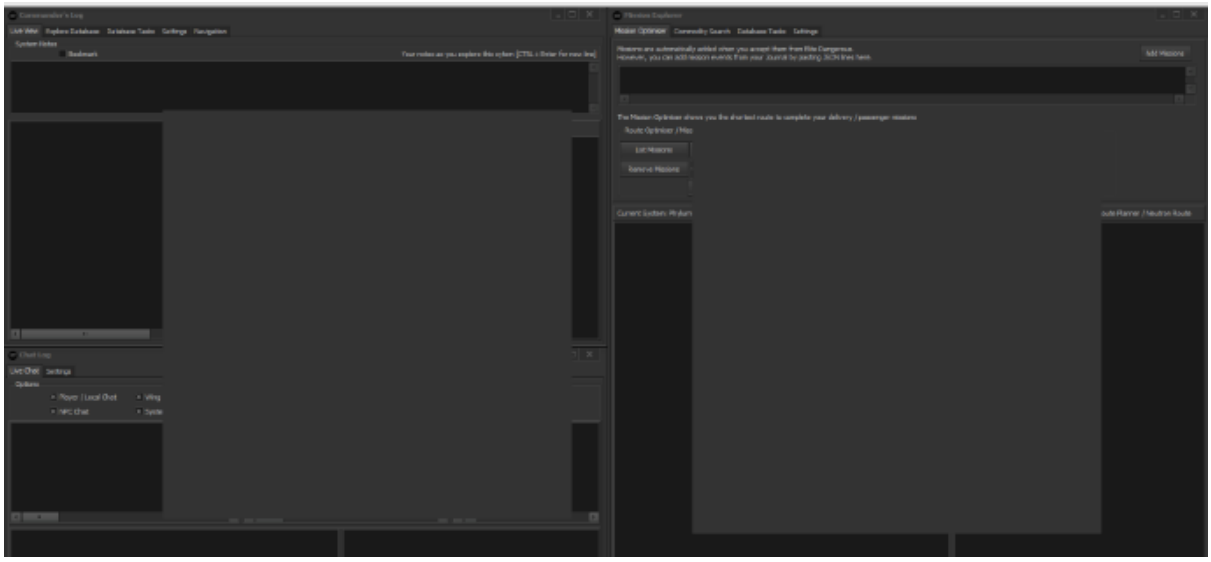
- Import your journal logs (from the Database tab on the **Commander's Log**). This will help with way point plotting (Mission Explorer) as previously visited systems will be in cache. Additionally, you can now review all the systems/planets you've visited and analyse the results.
- Import INARA files (**Mission Explorer**). This will help with the **Mission Optimiser** and **Commodity Search** functions.
- Select your Ship's COVAS assistant (main app window, **Speech Synthesis** tab).
- Set up TTS voices for use by pirates and system authority (main app window, **Air Traffic Control**) – this will allow you to be precise which voices are to be used.
- Set up custom speech phrases and enable/disable the events – **Speech Synthesis** tab
- Setup custom panels to display yet more useful information
- Install custom panels created by other commanders.

Known Issues / CubeSim Support

Cube Sim make fantastic hardware for flight simulators – these are 8” display panels surrounded with a Thrustmaster MFD Cougar joystick.

It has been reported by some users with that the application fails to start (initialise and render into the OpenGL windows), other than after the initial setup, when using Cubesim or tablets attached as additional monitors.

Typically, these OpenGL windows are initially created on the main monitor and then the user moves them to their preferred position which is often a secondary, tertiary or dedicated Cube Sim monitor. It is then, upon application restart, that the issue shows itself – as blank (solid grey) windows and appears to hang.



The exact cause for this is unknown but investigation appears to indicate that the tablet/cubesim drivers are unable to properly pass through OpenGL requests onto the main graphics card for rendering.

Workaround

If the OpenGL windows are moved back to the main monitor (before closing the app), then the app launches and runs on subsequent starts. To automatically perform this task, add the following to the **MFDCougar.ini** file:

```
[Cubesim]
MoveWindowsOnExit=0
MoveWindowsOnStart=1
```

The **`MoveWindowsOnStart`** entry attempts to create the OpenGL windows on the main monitor and then subsequently move them into their final destination position (as specified in the **MFD.ini** file). OpenGL double buffering is disabled with this option set, which seems to have resolved the issue of the application crashing when a tablet (or other device) is attached as a secondary monitor. The `MoveWindowsOnExit` entry causes the application to modify the **MFD.ini** file and set the positions of the OpenGL windows to ensure they are within the main monitor bounds when the application is shutdown. However, the user is then expected to move the window positions to the destination monitor(s) each and every time after application launches. This is depreciated and `MoveWindowsOnStart` is the preferred option.

Running ED:CD For the First Time

Running the application is as simple as the following 8 steps:

1. For 64-bit MFD Cougar Display – install Microsoft Access database engine 2010
2. Double-click the application (MFDCougar.EXE) to launch
3. Wait 1-3 minutes for the application to initialise (TTS voices, INARA downloads)
4. Ensure correct Elite folder locations are shown on the main form
5. Reposition Windows
6. Minimise to taskbar
7. Start Elite Dangerous (user level privileges) and load CMDR profile (Solo, Open, etc.)
8. Use Taskbar applet to “start” ED:CD
9. Configure Key bindings (optional)

Launching

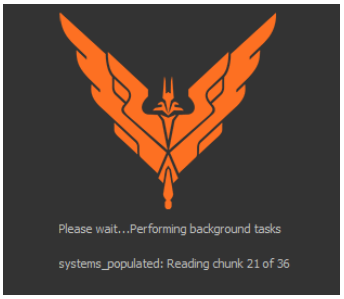
The first thing that happens when ED:CD runs is that multiple windows open (may take a few moments to completely initialise):

- Main Application screen
- Commander’s Log
- Mission Explorer
- Chat Log
- Splash screen (hopefully on top) detailing the current background task.

On the very first run, the following happens in the background:

- TTS/SAPI Initialisation
  Each and every (SAPI) TTS Voice in your system is tested and default voices for NPCs, System Authority NPCs and your ship are assigned. This task takes about 30 seconds, but is system dependant. You may want to consider adding more voices to your installation (covered later on).

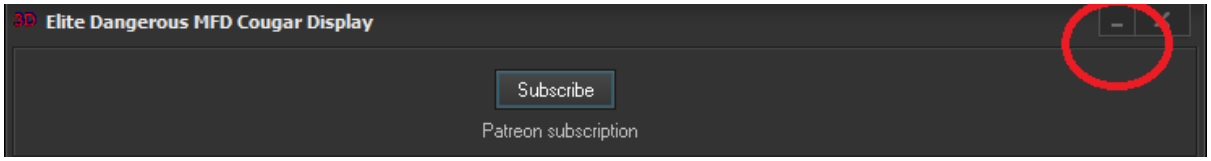
- Downloading Files / building database / Backups
  Additionally, since you’re are running for the first time, commodity updates from INARA are downloaded – for this a splash screen shows the progress of the task. This should take seconds and can be skipped via the settings tab (Mission Explorer) on future launches.



  The database import runs in a background thread and you are freely able to minimise the application to the system tray. **Mission Explorer** commodity searches will not be fully functional until the import process is complete.
  The Mission Explorer (settings tab) contains a schedule to auto backup the local database.

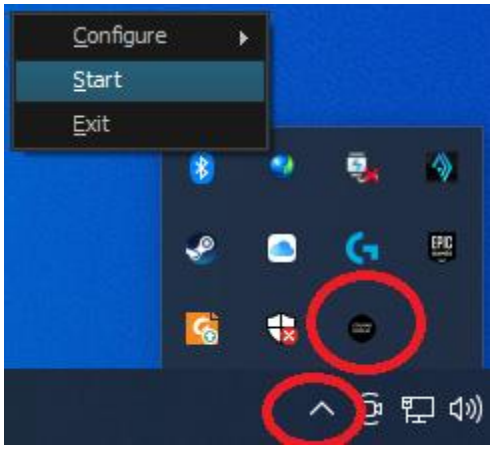
Minimise to taskbar

Activate the main window and click the minimise window icon.
At this point you should have move all other windows to their respective locations.



Use Taskbar Applet to Start

Left clicking the Cougar Display icon within the system tray brings up a menu, and clicking **Start** executes the rendering thread and makes the application respond to in-game events.



Note: Since Cougar Display reacts and tracks journal state, it is best to ensure the app is 'started' after the "loadout" event in the journal – i.e. just after selecting Open or Solo mode in Elite.

Clicking **Pause** temporarily freezes the rendering thread and displays the MFD Cougar windows as regular 'windows' so that you can adjust positions and dimensions as necessary.

That's it! You've finished the quick start guide and are hopefully enjoying the experience.

Note: Please ensure your **system clock is synchronised** with the internet. The app reads from Elite's journal files and relies upon an accurate system time in order to decode the events.

Note2: The application is setup to decode the journal files produced by Elite. Since Odyssey update #11 the journal file naming convention has changed and so you must configure the application to read either the Odyssey journals or the Legacy mode 3.8 journal files. See **Legacy Mode** and/or the **Troubleshooting** section at the end of this guide if you are using the older Elite 3.8 client.

Positioning and Sizing Windows

The next most important step is to position and resize the windows to their respective monitors and ensure the Cougar joysticks are placed over the monitors for the MFDs.

The very first time Cougar Display is executed the MFD Cougar windows are displayed with their windows titles showing – allowing you to move, size and place the windows on your desktop.

Upon subsequent launches, a file called "autostart" will have been created and the presence of this file prevents the window titles from showing. If, for some reason, you have issues with the pause/start procedure (below) then delete the "autostart" file and restart the application.
Note: pause/start menu items have no effect when the MFD Cougar windows are showing the window titles.

Positioning Procedure

Left click the system tray icon, and select **Pause** menu item. The MFD Cougar windows now have their window titles visible so that the positions can be set via mouse drag. Left click the system tray icon, and select **Resume** menu item to allow the application to resume.
These window settings are saved to an INI file in the application root folder "**MFD.ini**".

Note: If a monitor is unplugged, or a window is positioned such that it's partially off-screen, the next time Cougar Display is started, the window position is reset back to the main monitor. This is done on purpose so that windows are not "lost" in a region of desktop not covered by any monitor. Ensure all windows are fully inside the bounds of all monitors. The utility file (available on the website) **MFDWindowTest.exe** can be used to quickly display and position all the OpenGL windows the application creates.
Also note that even if a single pixel of the window is positioned beyond the monitor boundary then the application will move the window back to the main monitor. In some cases the window may appear to be fully within the monitor bounds, but actually it's overflowing. Try moving the window inwards when testing window positioning.

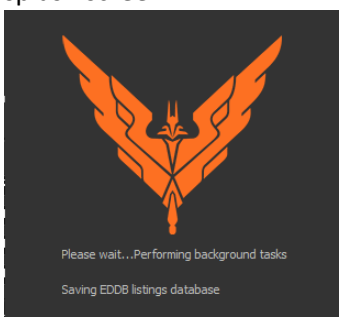
Exit/Quit Cougar Display

The Application will auto shutdown when Elite closes. This behaviour can be changed by setting the value to 0 in the MFDCougar.ini file.

```
[Shutdown]
AutoShutdown=1
```

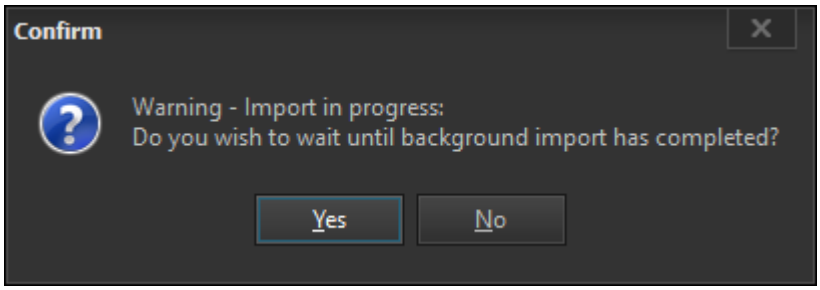
When the main application window is closed, the following may show:

- splash screen



- Warning – Import in progress

The INARA commodities are downloaded (from the internet) and imported on a schedule defined on the **Mission Explorer** (settings tab).



You can exit Cougar Display by clicking on the **No** button to force an early exit – but this is not recommended since the database import process is incomplete.

Key Bindings

Cougar Display is designed to run without any specific configuration and works 'out of the box'. However, by its very nature, the icons representing actions within Elite are dynamically assigned to the different screens according to the game status. Therefore, the same Cougar button press needs to be dynamically bound to different Elite key bindings. With twenty joystick buttons per Cougar, and 4 or 5 screens of icons, we are looking at hundreds of key bindings.
To simplify the whole process, the built-in layout designer has a **quick key** binding option to automatically map the virtual Cougar buttons/icons to Elite keyboard bindings. With once click, all the key assignments can be made by finding unused bindings for actions within the Elite key bindings file. If there are no unused bindings, then the application will overwrite the keyboard binding, starting with the secondary binding.

Note: Elite Dangerous must be set to use **custom** bindings rather than one of the pre-set joysticks, eg SaitekX52. To ensure Elite is using a custom bindings, edit one of the bindings and save the changes.

There is more information on the Key Binding process in http://cougardisplay.site/keybinds.html and http://cougardisplay.site/troubleshoot.html#bindings

English Language

There is an assumption made that the Windows OS language is English, English keyboard layout and Elite Dangerous language is English.
Key binding mechanism relies upon the Windows internal name for key presses (eg HOME, UP, PAGEUP, etc) and these names are written into XML files – the names must also match those used for Elite Dangerous. Issues can arise when there's a language mismatch between the 3 main components: Elite Dangerous Key bindings XML, Cougar Display XML files, and the Windows keyboard input locale. For foreign languages and input, you may have to translate the Cougar Display XML files manually (MFD_Layout.xml and MFD_Iconset.xml).

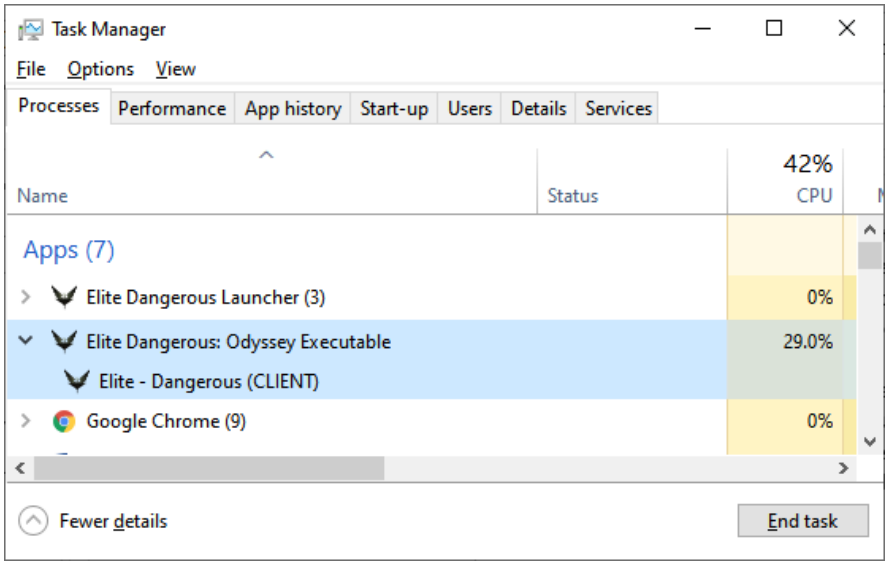
Elite Client

Ensure both Elite and MFDCougar.exe are launched with the same Windows Elevation level (e.g. standard user, or both elevated to Administrator - do not mix the elevation levels).

For key strokes to be sent to Elite, the window title of Elite must be known and matching the window title in **MFDCougar.ini** file. By default the title is set to:

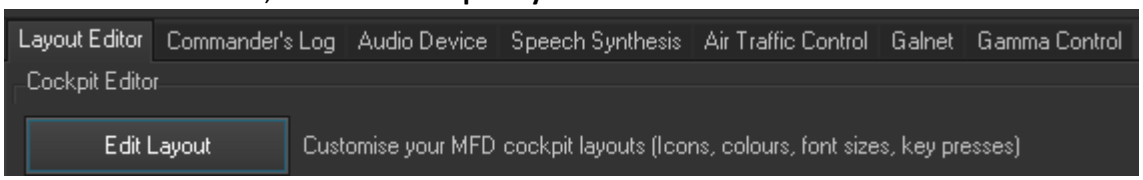
```
[EDClient]
Title=Elite - Dangerous (CLIENT)
```

If Frontier change the Elite client title, you'll need to amend the **INI** file accordingly. For example, during Alpha/Beta versions of Elite, the client title may need to be altered. This can be confirmed within Elite → Options → Graphics → Display → Fullscreen → Windowed mode. The title bar of the window is then displayed (and visible from task Manager)


Task Manager
File Options View
Processes Performance App history Start-up Users Details Services
42%
Name Status CPU
Apps (7)
Elite Dangerous Launcher (3) 0%
Elite Dangerous: Odyssey Executable 29.0%
Elite - Dangerous (CLIENT)
Google Chrome (9) 0%
Fewer details
End task

Layout Editor

From the main screen, locate the **Cockpit Layout Editor** and click **Edit.**



Elite Key Assignments

Elite uses an XML file to record which device to read in order for the user to perform an action within the game. For example, the snippet below is taken from the **Custom.4.0.binds** file (`C:\Users\xxx\AppData\Local\Frontier Developments\Elite Dangerous\Options\Bindings`) and would be used to launch Shield Cell Bank within Elite via a joystick button press.

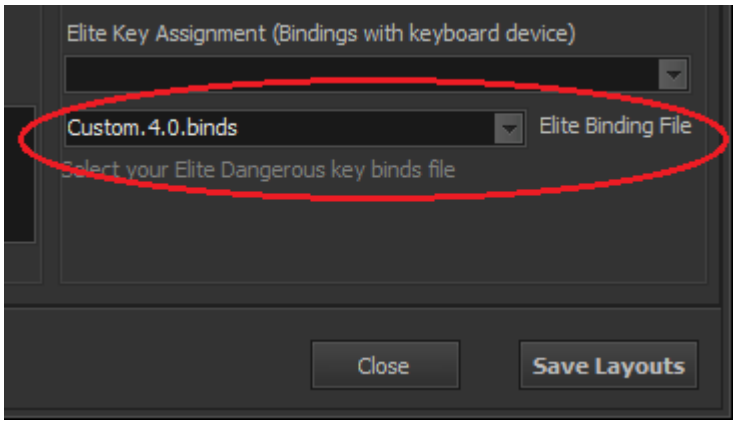
```
<UseShieldCell>
      <Primary Device="T16000M" Key="Joy_6" />
      <Secondary Device="{NoDevice}" Key="" />
</UseShieldCell>
```

Elite has two possible devices which can be used to capture input for any action. When the **MFD Layout Editor** window closes, the application will look for an unassigned device and replace the binding with an entry from the application's list of icon/button key assignments (MFD_Iconset.xml). By default, the app will try to assign the secondary device as a keyboard, but may fall back to assigning the primary device. The result of the mapping may look like the example below:

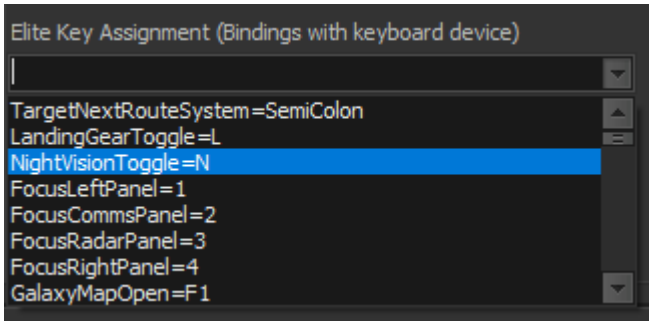
```
<UseShieldCell>
      <Primary Device="T16000M" Key="Joy_6" />
      <Secondary Device="Keyboard" Key="Key_C" >
            <Modifier Device="Keyboard" Key="Key_LeftAlt" />
      </Secondary>
</UseShieldCell>
```

Manual Key Assignment

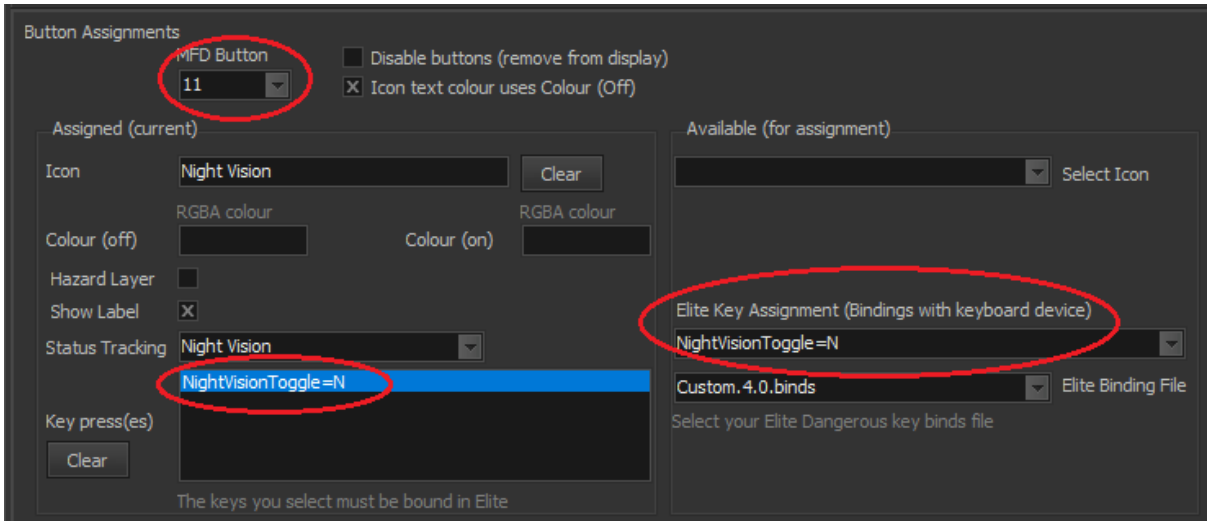
Select the **Elite binding xml File** from the **Elite Binding File** drop-down list.



When Elite's key binds file has been read successfully, the **Elite Key Assignment** drop-down list will contain all the Elite actions that have a keyboard device associated. You can then assign the Elite action to the button/icon in the MFD layout **Button Assignment** section of the interface.



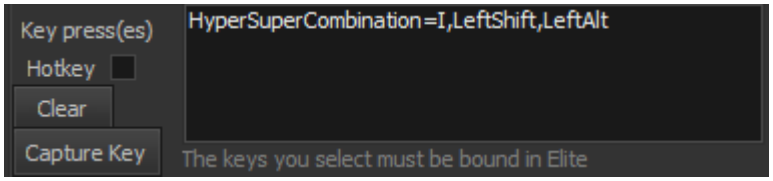
In order for you to assign an Elite action to the icon/button, the action must appear in the **Elite Key Assignment** drop-down list; if the action is not in the list, then the action does not have an associated key binding (it may be bound to a joystick, for example). You will need to either edit the Elite Key Binds file directly (quite simple to do once you are familiar with the process) or run Elite Dangerous and make the key binding change from there (and close Elite after). In either case re-open the **Layout Editor** to reload the modified Elite Binds file.



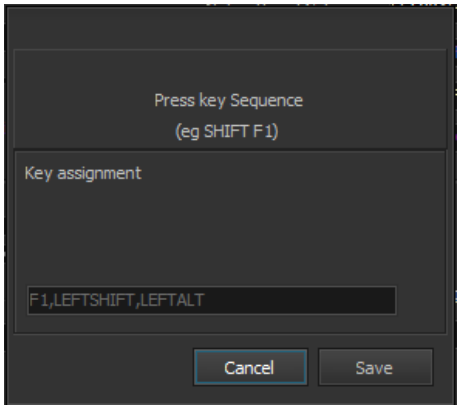
To remove the key sequence associated to the icon, double-click the key entry in the list.

Custom Key Binding Value

Choosing a value from the **Elite Key Assignment** dropdown sets the key press to the value contained in the **MFD_Iconset.xml** for the chosen Elite action. Sometimes, Elite just doesn't recognise the key sequence and you have to provide a new key binding. You can edit the key attribute in the **MFD_Iconset.xml** file, or click the **Capture Key** button to assign a key sequence to override the listed value.



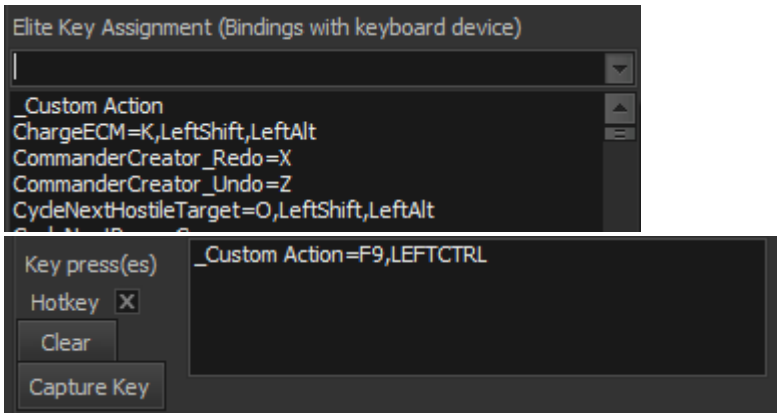
Clicking **Capture Key** brings up a key capture screen which will capture key strokes and modifier keys (Shift,Ctrl,Alt) and assign it to the selected key sequence. See troubleshoot.html for more information.



Setting the **HotKey** checkbox registers the keypress as a Windows hotkey. When the hotkey press is detected by Windows, the keypress is passed into the MFD Cougar application for processing and the MFD actions associated are performed; the key sequence is **NOT** passed to Elite Dangerous.

Additional Notes for HotKey Assignments

Keys assigned to the MFD icons in the layout editor are usually associated with an Elite action because these key sequences are being send to Elite to perform some function (the Elite Binding, such as toggle ship lights).
Whilst this is generally desirable, there are some very specific cases where we want to display a custom icon in the MFD grid but not send a key sequence to Elite; instead we want the icon to flash/animate and for some other action to occur. An example of this is the "**Radio Chatter**" icon feature - which tracks the status of the Air Traffic Control "ATC Simulation" on/off toggle. By setting the **hotkey** flag in the button UI, the keypress can be assigned to a "customAction" to perform the toggle.



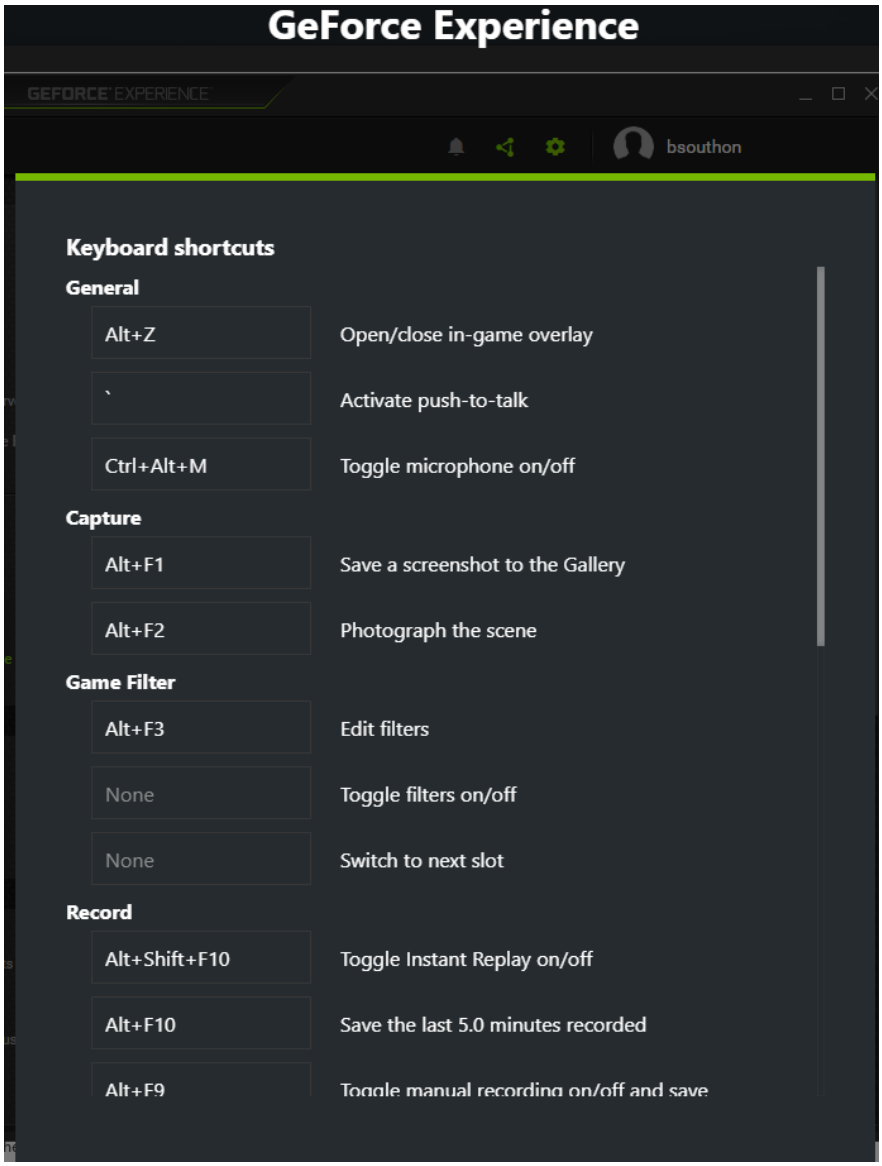
Note: Unlike regular Elite key press sequences hot keys have no distinction between left/right modifiers.

Note: F12 (without a modifier such as SHIFT) can't be used as a hotkey - see https://learn.microsoft.com/en-us/windows/win32/api/winuser/nf-winuser-registerhotkey
Also note that hotkeys already registered with the same key press sequence will **not** be overwritten. Several applications already include hot keys, such as AMD Control Panel, nVidia Performance Overlay Windows Accessibility, amongst others.

nVidia Overlay uses the following hotkeys by default, so you may wish to switch these off:

ALT+F1, ALT + F2, ALT + F3, ALT + F5, ALT + F6, ALT+F7, ALT + F8, ALT+ F9, ALT + F10, ALT + F12, ALT + Z, CTRL + ALT + M

(Start → GeForce Experience → settings (cog menu icon) → In-Game Overlay → Settings → Keyboard shortcuts)
Open/Close in-game overlay (ALT + Z) can't be unassigned, but the whole in-game overlay can be disabled.



Miscellaneous Button Properties

Button Latch

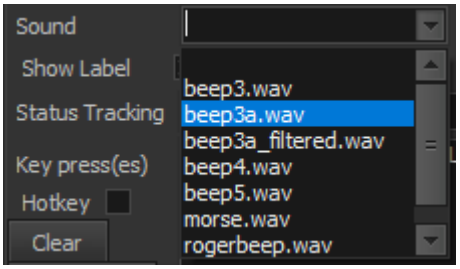
When a Thrustmaster MFD Cougar button is pressed, it often doesn't feel like the button press has 'registered' as these buttons feel quite soft and squidgy to the touch.

To combat pressing the buttons multiple times, a button latch slider can provide a delay so that button presses have to be several milliseconds apart to register (this prevents button spamming). The default is 1000ms or 1 second.

Each layout (Flight, Weapons, SLF, Supercruise, etc) in the **MFD Layout Editor** has one **button latch** value for all the icons/buttons.

Sound

When an icon is clicked or MFD button pressed, a sound can be played to give audio feedback that the button has been 'registered'. The choice of the audio file is listed in the drop-down box and the list is built from all the .wav/.mp3 files in the application **audio** folder.



Each button on the MFD layout can have a different sound assignment.

Note: It is recommended to use a utility such as Adobe Audition or (the excellent and free) WaveShop to 'normalise' the audio files to the same volume.

Label X: and **Label Y:**

The label positions around each icon on the main displays can be tweaked slightly to allow a more pleasing position on the screen, so that the icon text matches the icon position a bit better.

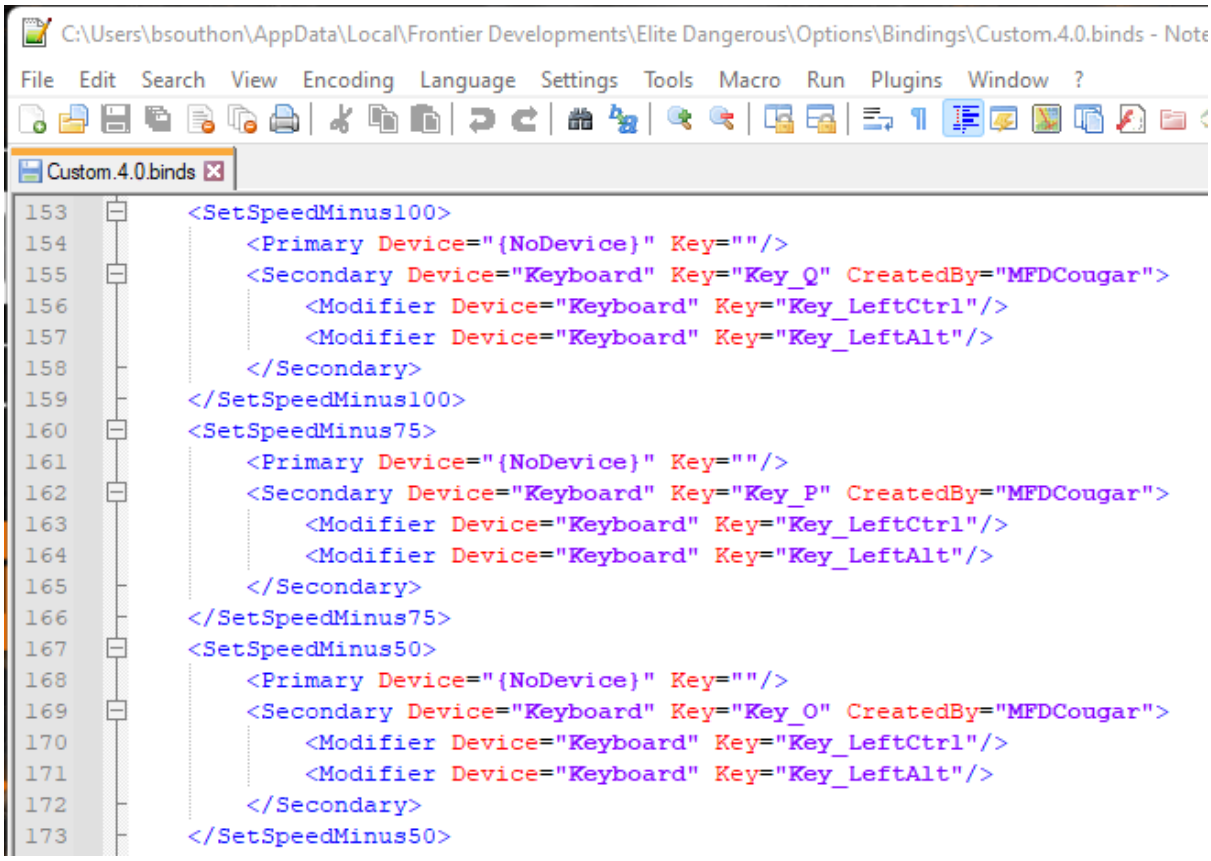
Manually Editing Elite's Binds File

In some cases, editing the XML file directly is very fast and efficient; however there is a certain amount of risk if you're not confident in performing this job. Always make a backup before you start!

Binding files are located in (`C:\Users\xxx\AppData\Local\Frontier Developments\Elite Dangerous\Options\Bindings`)

Backup the file and use **Notepad ++** to open **`Custom.X.0.binds`** (where X is 3 for Horizons and 4 for Odyssey). If this file does not exist, it's because you are using one of the pre-set binding files used by Elite – once you modify Elite's default key assignments, the file will be created for you.

Search for the Elite action you wish to assign a key press to and modify the XML entry.



Example

Adding a keyboard as a secondary device to **NightVisionToggle**

In this example, NightVisionToggle has been previously assigned to a joystick but now we want to also assign a key stroke (ALT N) to this action so that MFD Cougar Display can associate a button press to it.

We need to change the entry from:

```
<NightVisionToggle>
      <Primary Device="T16000M" Key="Joy_6" />
      <Secondary Device="{NoDevice}" Key="" />
</NightVisionToggle>
```

And change the entry to:

```
<NightVisionToggle>
      <Primary Device="T16000M" Key="Joy_6" />
      <Secondary Device="Keyboard" Key="Key_N">
            <Modifier Device="Keyboard" Key="Key_LeftAlt" />
      </Secondary>
</NightVisionToggle>
```

In the above example, we assigned (left) ALT N as a secondary key bind. In order to use the ALT key, Elite uses the `<Modifier>` XML element to specify either SHIFT, CTRL or ALT keys (one modifier element per modifier key) and is prefixed with either Left or Right.

Once the XML file has been saved using Notepad ++, then re-open the Layout Editor in MFD Cougar Display and the key binding can be assigned to the MFD Buttons/icons.

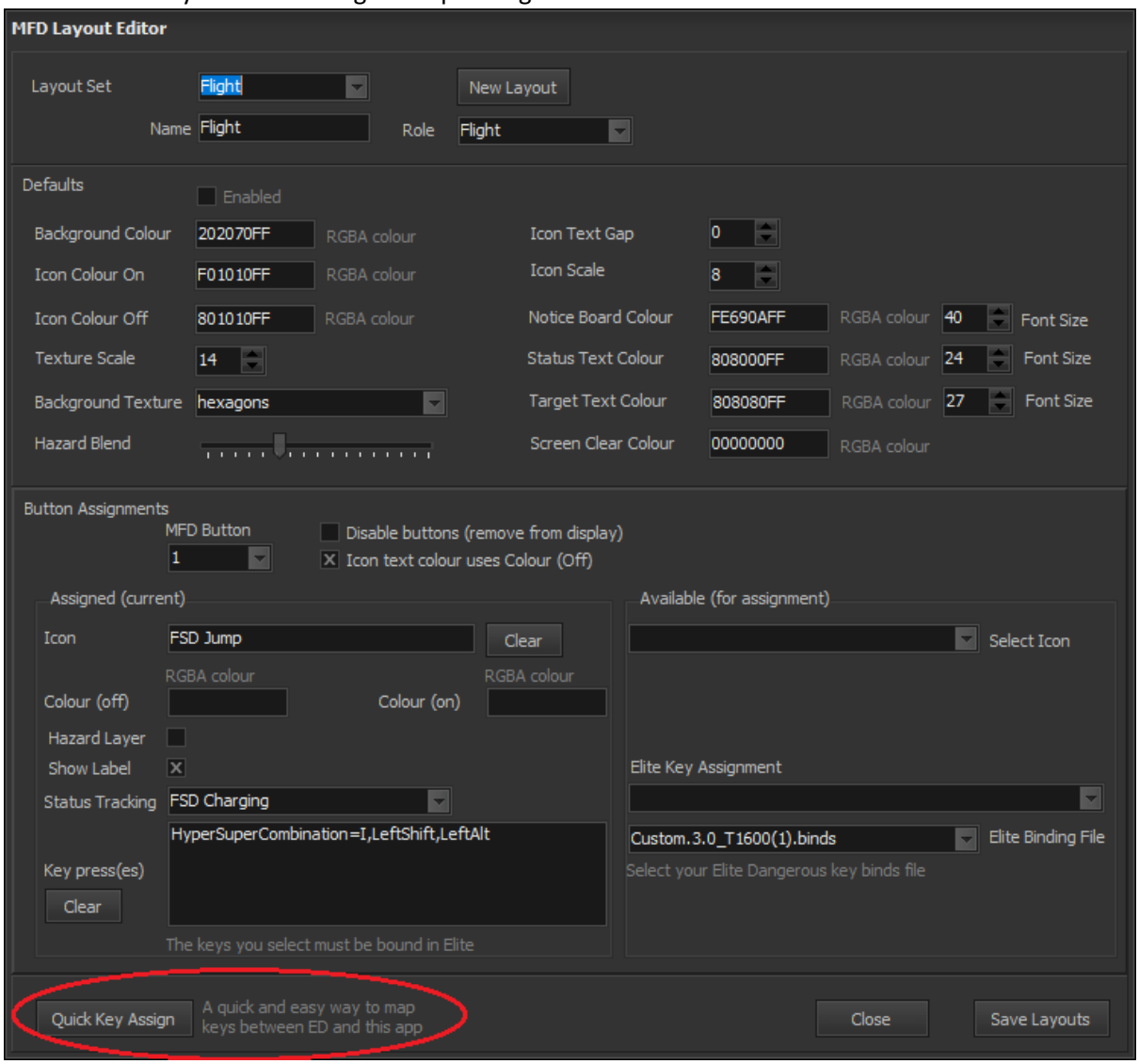
Note: Certain combinations of keys and modifiers just don't work. There is no explanation for this behaviour, other than legacy keyboard driver behaviour dating back to Windows 3.0.

Note2: Right ALT and right CTRL both insert 'hidden' extra key sequences when sent to Elite from MFD Cougar. Avoid using these modifiers if you intent to use the action with Cougar Display – conversely do use these modifiers for the keys you aren't using with Cougar Display. See http://cougardisplay.site/troubleshoot.html#bindings for more information and troubleshooting tools.

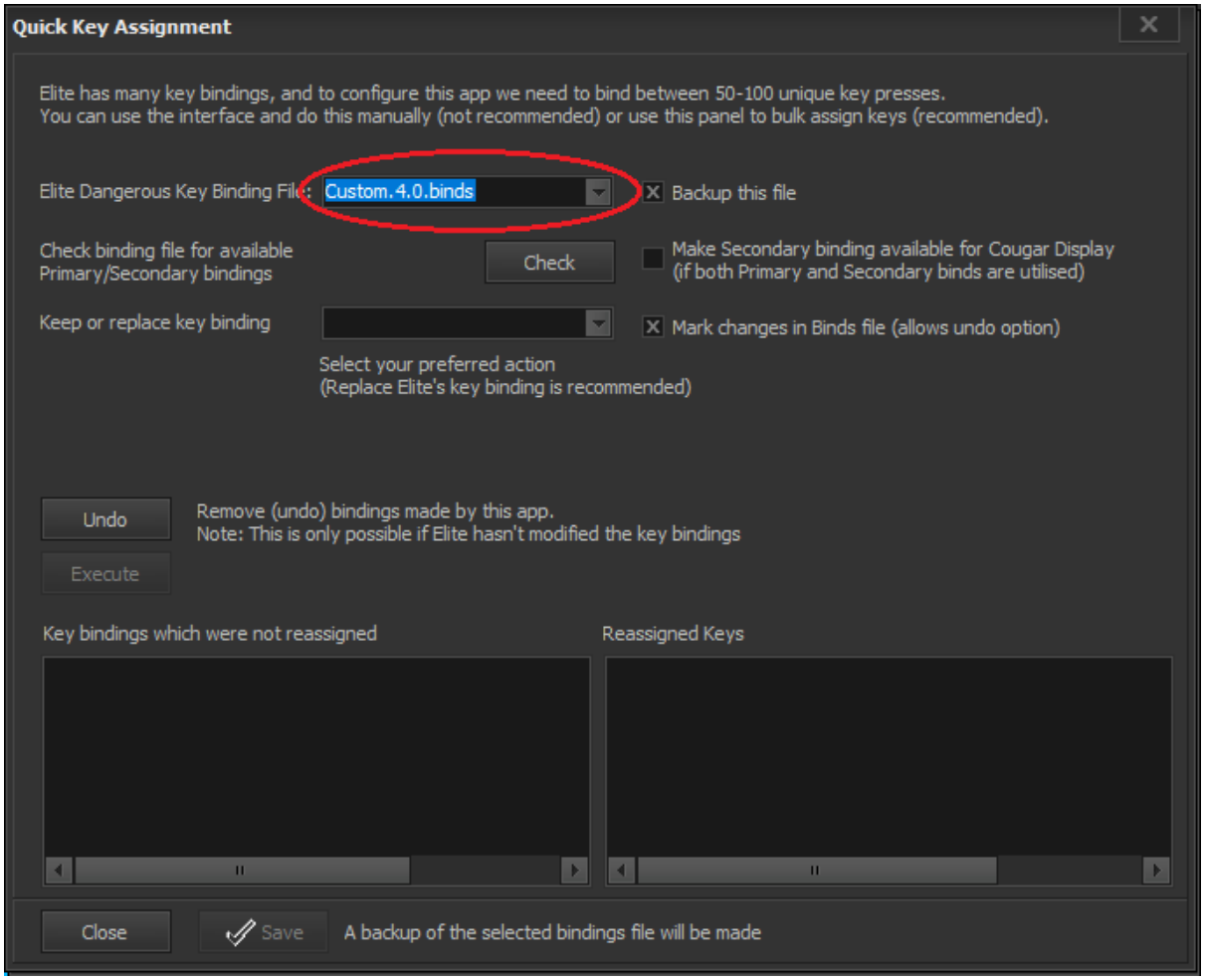
Quick Key Assignment

The **Quick Key** Assignment is designed to replace empty and available bindings within the Elite Binds file, with pre-defined key sequences associated with the MFD Icons (MFD_Iconset.xml).

Click the **Quick Key Assign** button. At this time, there is no need to alter any other setting. You will be notified if any unsaved changes are pending.



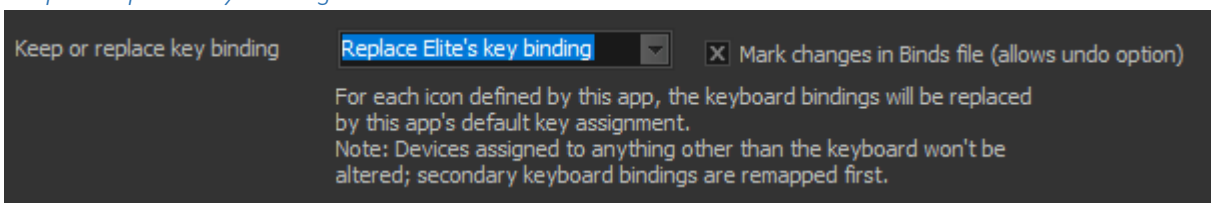
Select the Binds file you wish to modify. By default it's the same file as the Layout Editor was using, but you can change it to any you find.



Quick Key Assign attempts to find unused [device] bindings for the actions specified by the buttons/icons in the Layout Editor. You can check for possible issues by clicking the **Check** button – this will highlight the Elite actions where both the Primary and Secondary binds are occupied by joystick devices. You should review these binds and either un-assign a device in Elite, manually editing the Binds XML file (see manually editing Key Binds), or select the option **Make Secondary binding available for Cougar Display** (and click **Check** button again).

Assuming the check function reports no issues, you can proceed to select an action for **Keep or Replace key binding**.

Keep or Replace key Binding

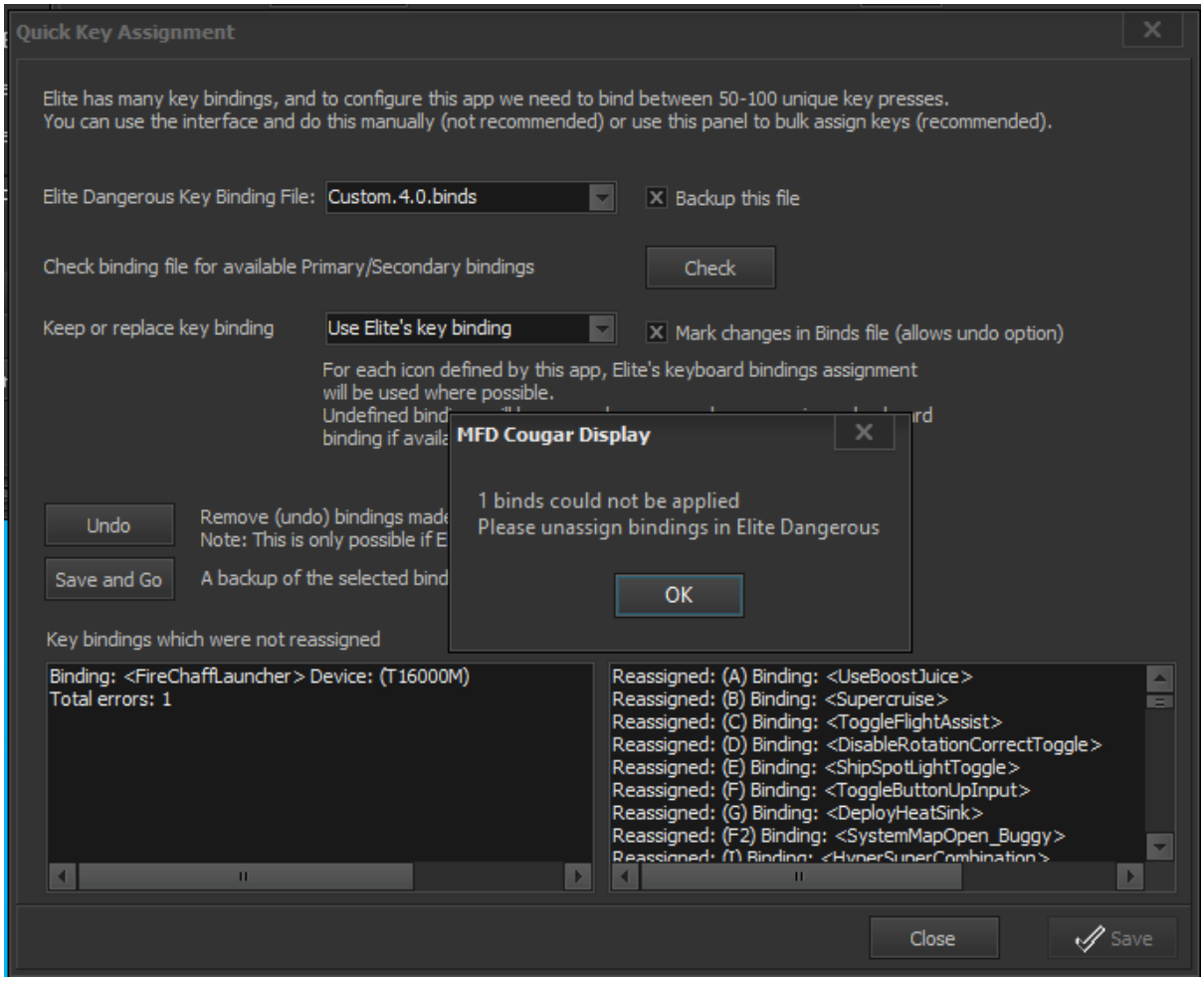


Select the action you want to occur when the app detects that there is already a keyboard device associated to the Elite action which needs binding.

There are two possible choices:

1. The first is to make no change to Elite's binding and instead change MFD Cougar to use that key sequence (instead of the value in MFD_Iconset.xml). This is not recommended because the key sequence may not be suitable for use via the windows API (MFD Cougar Display simulates key presses and sends them to Elite for processing).
2. The second choice is to change (overwrite) the Elite Binding with the key sequence specified in **MFD_Iconset.xml**. This file has a known-to-work set of keys and modifiers and is thus recommended.

Click **Execute** button.
Note: By default, a backup copy of the selected bindings file is performed.



Any issues found with the mapping will be listed on the left in the **Key bindings which were not reassigned**. Conversely, the remapped keys are shown on the right. This is for your information only in case you wish to manually inspect your Elite bindings file.

If you are happy, click the **Save** button (to commit changes), or click **Undo** to revert the changes. Changes can be undone if the option **Mark changes in Binds file** is set (this adds an attribute to all binds entries added/modified so that the application can quickly locate and remove changes).
The **Save** button (writing to the Binds XML file) will not be available if bindings could not be fully applied. **Note**: Uncheck **Mark changes in Binds file** if 3rd party apps need strict XML conformance.

Dependencies

No dependencies upon other programs, utilities or 3rd party DLLs. Only Elite Dangerous is required, along with an optional second/third/forth monitor and optional Thrustmaster MFD Cougar joysticks.

For 64-bit MFDCougar Display, Microsoft Access drivers are required (installation EXE is provided with the full .zip package, and a link on the web site).

Note: Open GL 4.5 or later is required. This is provided by your graphics card vendor and is in-built along-side DirectX 11/12.

MFD Window Guide

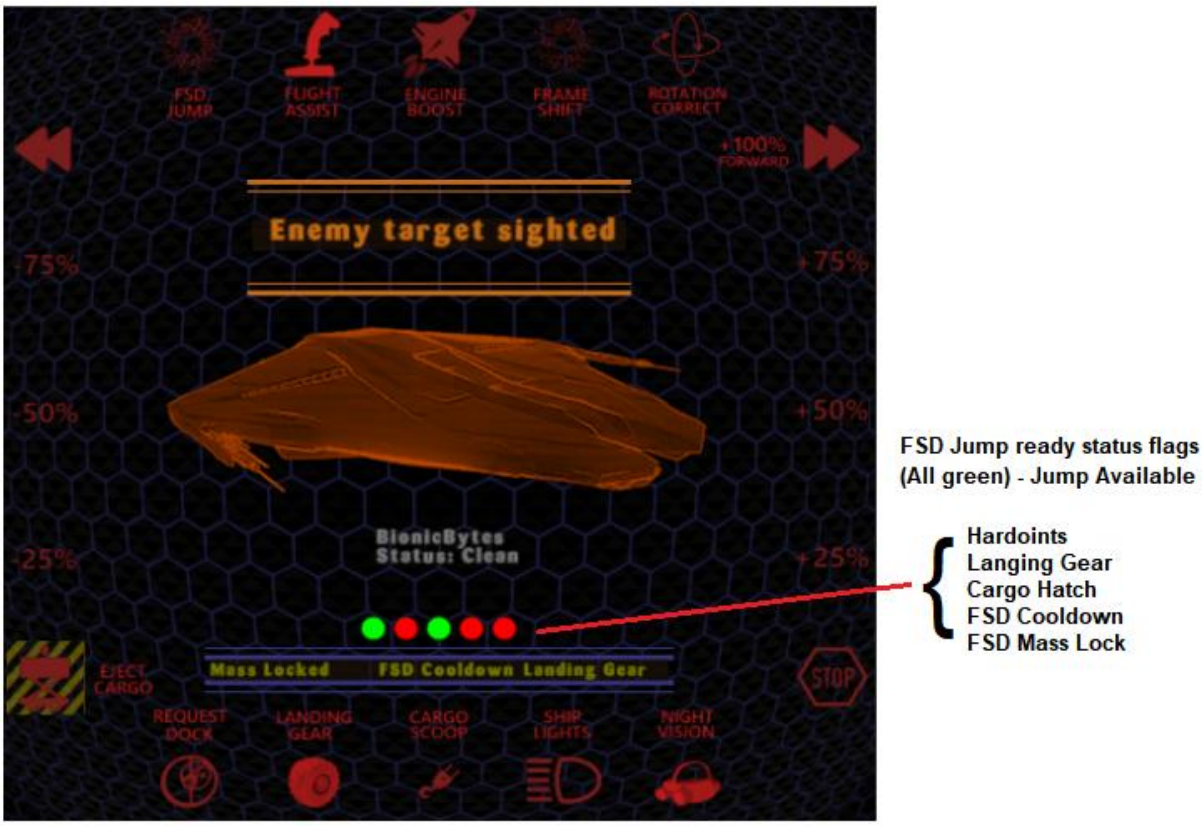
This section covers the images displayed on the Cougar Display MFD windows.

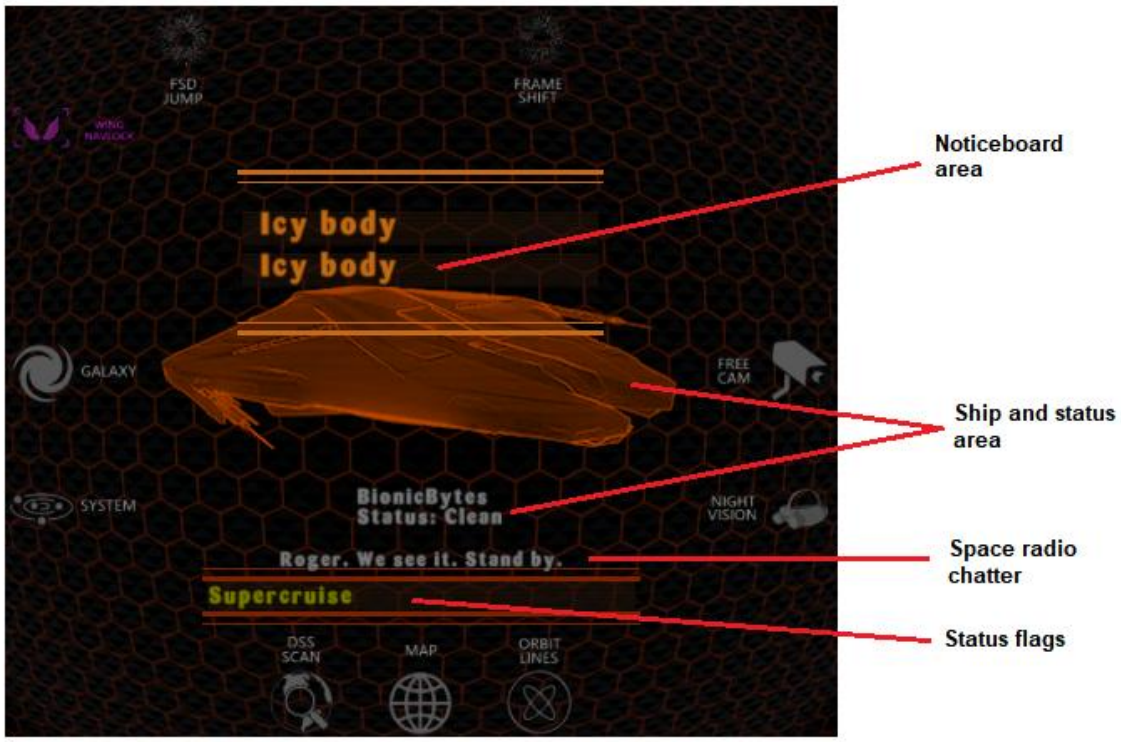
The 4 rocker switches on the Thrustmaster Cougar/WingWing/CubeSim MFDs are mapped as follows. Currently, there is no user interface to remap or configure these buttons.

The 20 buttons are configured via key bindings in the **Layout Editor**.

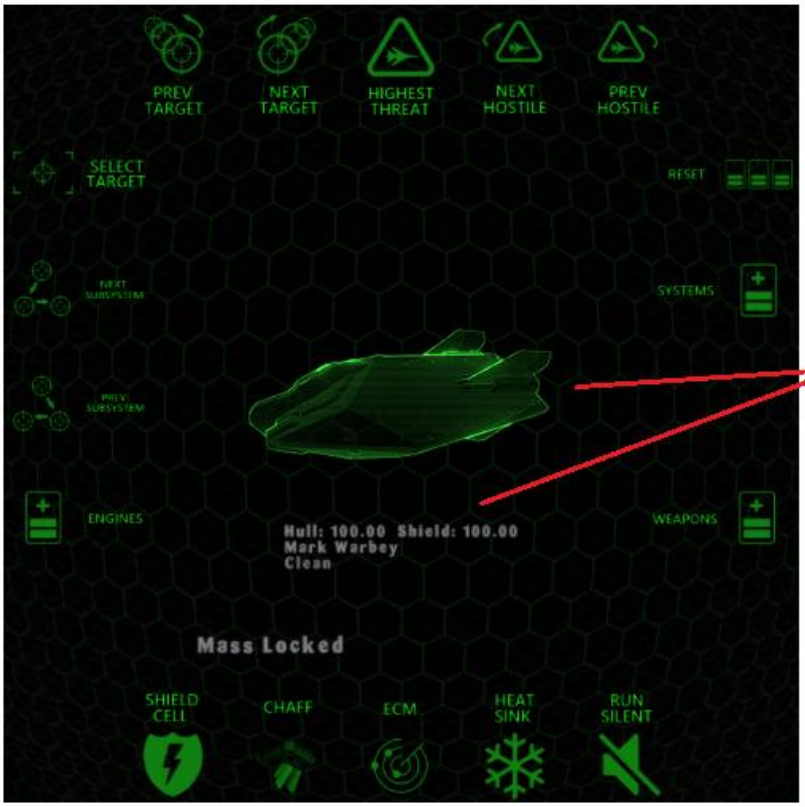
Other MFD devices such as WingWing and CubeSim are supported and their buttons are mapped as if a Thrustmaster MFD Cougar were attached.

Flight/Supercruise consoles are assigned to MFD Cougar 0

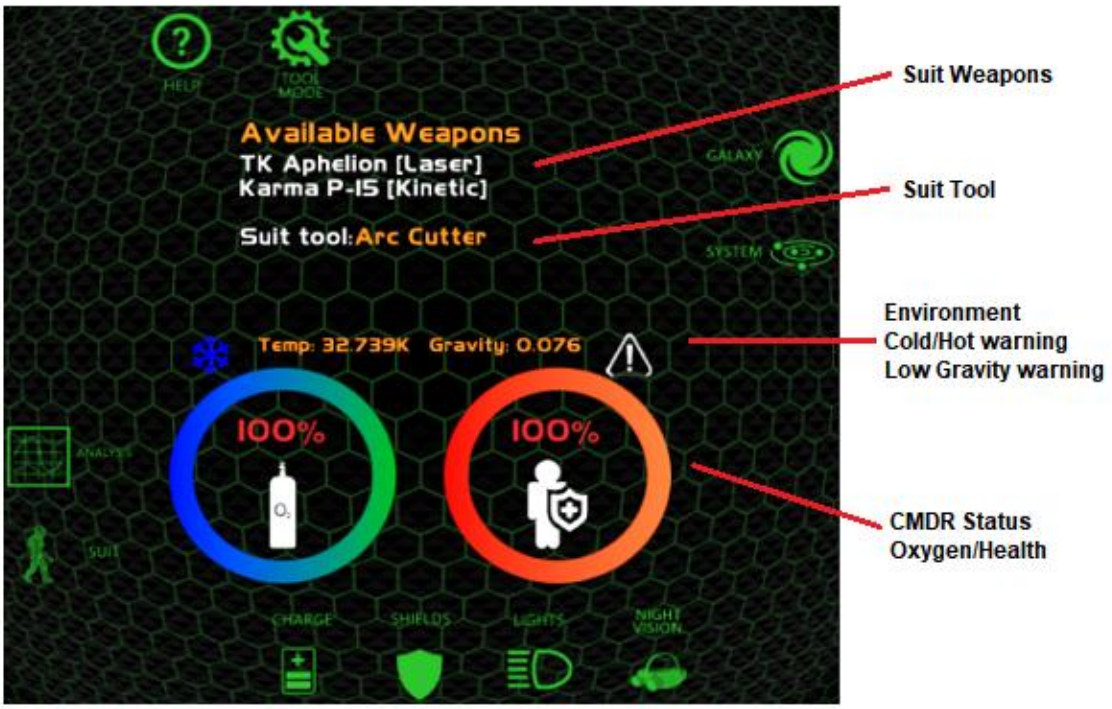




Weapons console is assigned to MFD Cougar 1



Odyssey Consoles (when on foot) display on MFD Cougar 0 and 1



UI Guide

This section covers the settings found of the user interface windows.

Styles

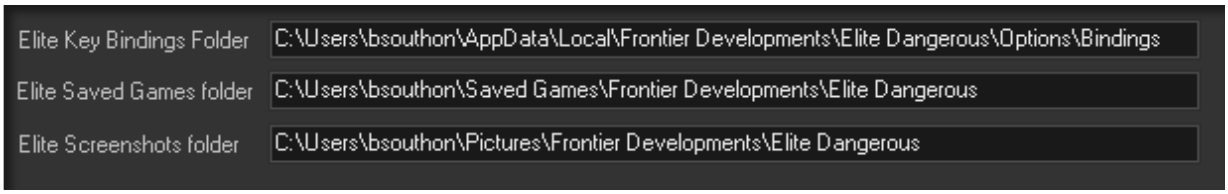
Choose the windows default skin if you find that sometimes the carbon theme is not working correctly (parts of the don't draw correctly) . This option can only be set when the application first loads.

Patreon/PayPal

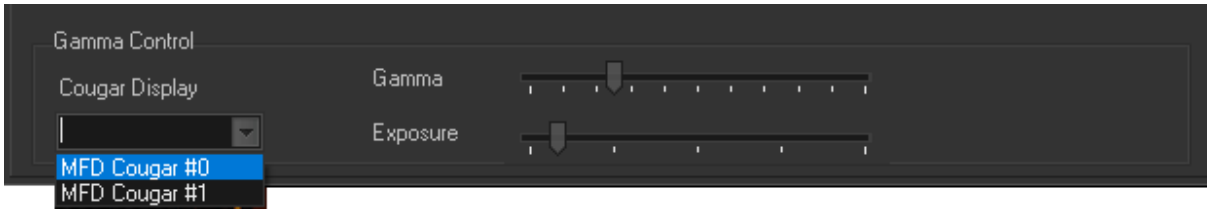
Clicking the Subscribe button opens a web browser at the download page of Cougar Display, where you can purchase a license key or make other donations. The only reason why this option exists is to cover my costs during my year-long development of this project.

Configure Elite's Default Folders

There are three options to specify the folder locations for Key bindings, log files and screen shots. Please ensure these are correct for your system. The application is initialised with default values based on a new installation.



Gamma Control



The Gamma Control tab gives the ability to fine tune the application's tone mapping output to give the best possible display on the USB / additional monitor.

For those with Thrustmaster/WingWing/CubeSim MFD devices, the rocker switches adjust the gamma and exposure levels.

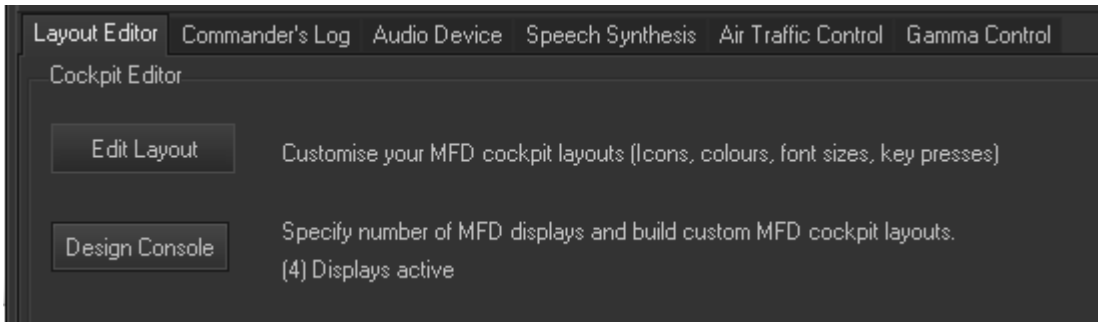
Customisable Cockpit Displays

From the Layout Editor tab, you can change the number of displays, or edit the configuration of the displays.

Customising involves two steps, one to decide on how many displays you want and associate 'roles' to those displays. The other step is to customise the layouts (roles) in terms of font, colours and icons.

Changing the Number of Displays

The number of displays can be switched from 2 to 8 simultaneous displays. A restart of the application is required for the change to take effect. This can be access from the **Design Console** button.



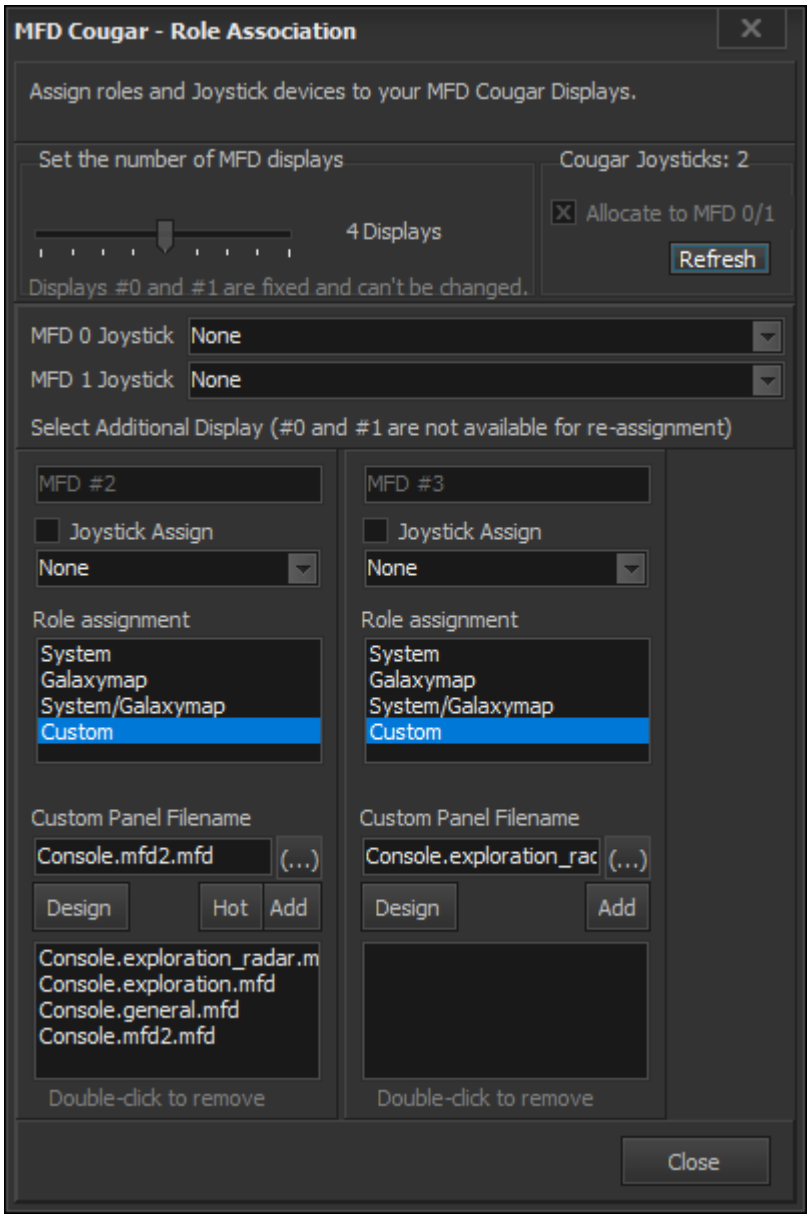
***Note**: for Thrustmaster/WingWing/CubeSim MFD owners, it is assumed that the MFD devices are connected to the displays in linear order (MFD #0 on Cougar Display #0, MFD #1 on display #1, etc.)*

The application makes the basic assumption that there will always be a minimum of 2 displays active at any one time, and the 'layout' role assigned to these displays will be dynamically assigned whilst Elite dangerous is running. This allows for a seamless transition of console layouts from flight to supercruise or switching to and from analysis mode.

The **Design Console** button brings up the **Role Association** screen.

To supplement the fixed (2) displays, you can optionally display the **Orrery** (system role) or **Galaxy Map** on one or more displays, or optionally a selection of **custom** panels in any arrangement you like.

Role Association / Number of Displays

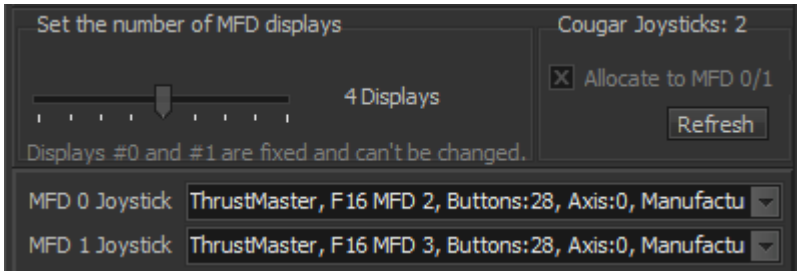


The Role Association dialog is used to link together the following aspects:

- Set the number of display windows
- Associate ‘roles’ to the display windows
- Assign joystick devices to the display windows

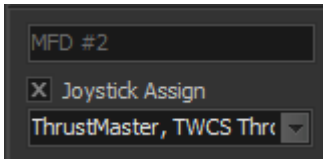
Assigning Joystick Devices to Displays

MFDs 0 and 1 can be assigned joystick devices by selecting the device from the two MFD Joystick lists.
The number of Thrustmaster, CubeSim or WingWing MFD devices is shown, and by default are automatically allocated to MFD 0 and 1 if the lists are left blank or set to **none**.



If your Cougar joysticks are not shown (eg plugged in after application launch), clicking the **Refresh** button will add them to the lists.

For each **Additional Display**, a joystick can be assigned from any available on the list.
This allows the connection of custom made “Cougar” joysticks to be assigned to any display (it’s recommended that any such custom hardware supports at least 20 push buttons).



When the Dialog is closed, the `MFD.INI` file is updated and the manufacturer ID and Device ID of the selected joysticks are written to the `CougarJoystick` setting.
If there are no available Thrustmaster MFD Cougar joysticks, and no other joysticks assigned to MFD consoles 0 and 1, then the `CougarJoystick` setting will contain the value 0 or 1, which has the effect of auto assigning the Thrustmaster/CubeSim or WingWing MFD devices should they become available. This is the default behaviour, which basically means that out-of-the-box, Cougar Display will automatically work with your MFD joysticks (no need to use Role Association to create the binding between controller and display).

Setting the Number of Displays

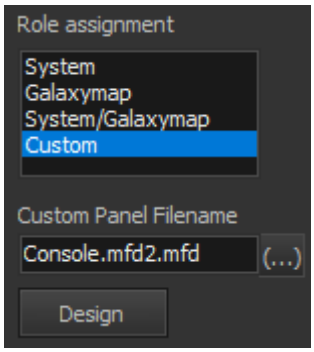
Use the slider to select the number of simultaneous displays required – this will add panels in the **Additional Display** area.

From the **Role Assignment** list, select the role for this display.

When making selections, you will notice that the list of available choices in additional displays will change. This is by design. There can only be 1 display with a Galaxy map, and 1 display with an Orrery (system), thus when these are selected those choices are removed from all other displays.

There will be some commanders who would like dedicated Orrery (system) and Galaxy map displays, whereas others only need a combined system and Galaxy map and thus reduce the number of simultaneous displays from 4 down to 3. The combined **System/Galaxy map** role functions in a similar manner to when there are only 2 displays running – i.e. when Elite is showing the galaxy map, then ED:CD shows the galaxy map, and when Elite is showing the system or orrery, ED:CD displays the orrery. Having a 3rd display with the combined role of **System/Galaxy map** allows a permanent Orrery view and an automatic switch to the Galaxy map when Elite switches over to the game's Galaxy map.

Custom Role Assignment



When the **Role Assignment** is **Custom**, there are additional buttons to access the **Custom Console Designer**. There is no limit on the number of displays created by ED:CD, but for practical reasons it has been limited to 8. Two of these are fixed and can't be changed, leaving up to 6 user-defined uses for the extra displays. Up to 2 of these user-defined roles can be assigned to the Orrery (system) or Galaxy map roles, leaving between of 4 - 6 displays for custom panel designed layouts.

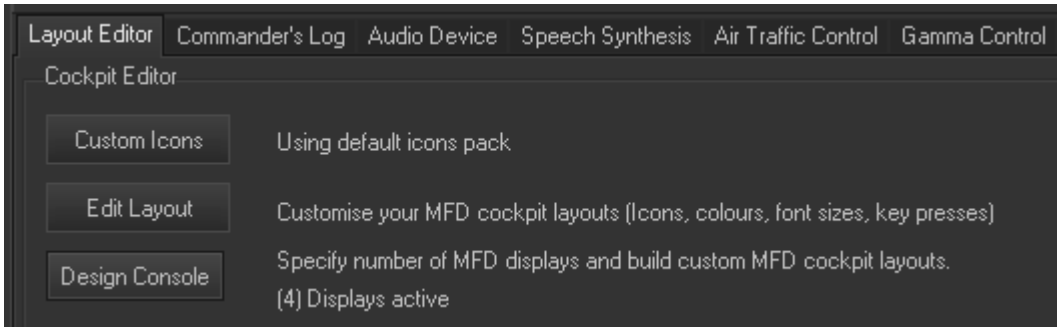
The Custom Panel listed in the **Filename** edit box is the default custom panel for the selected display, and can be edited via the **Design** button. This will become more apparent when you select multiple panels on the same display (refer to the section "Switching between Custom Panels").

Note: When the **Role Association** page is close, all the changes to the number of displays, and the associated roles are saved in the **MFD.INI** file in the application folder.

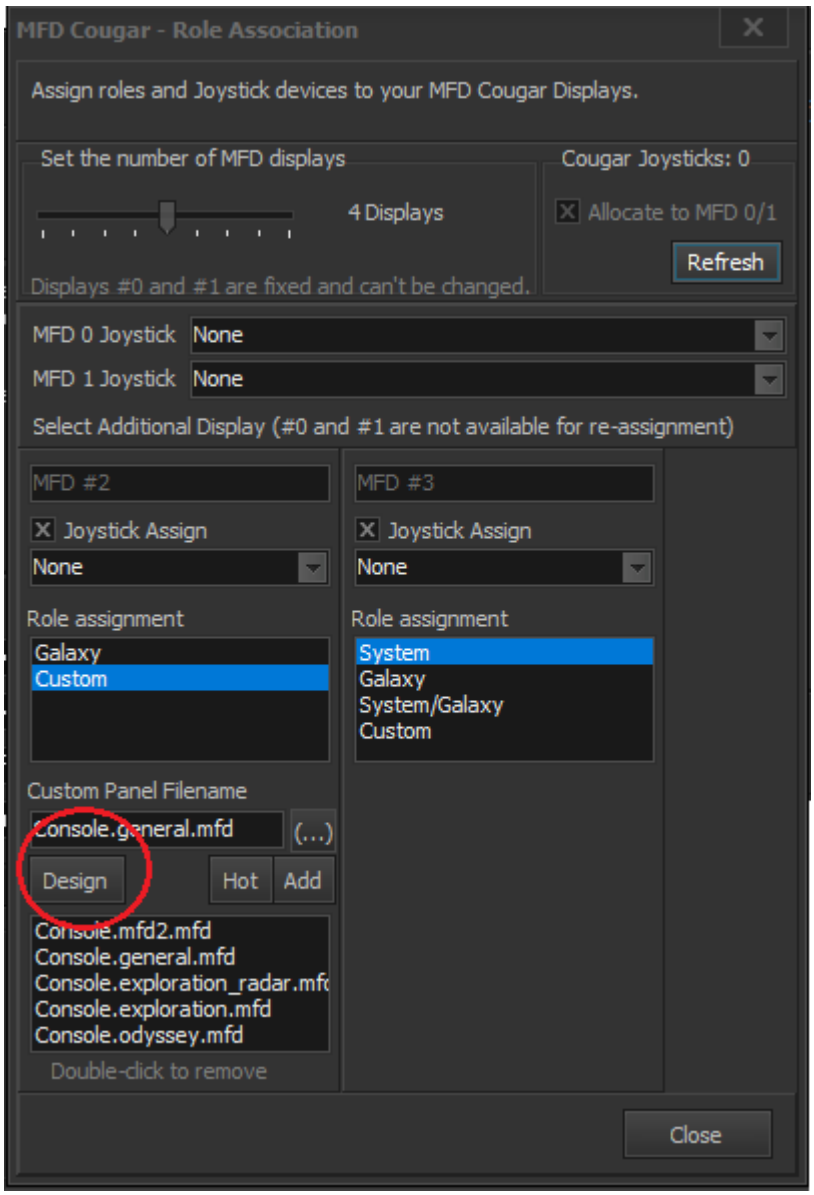
Note 2: Example custom panels are included in the zip package. "Console.mfd2.mfd", "Console.exporation.mfd" and "Console.odyssey.mfd" have been configured and setup for exploration and Odyssey (suits/weapons).

Custom Panels

Custom Panels are user-defined screens which can be built to display specific pieces of information. Custom panels are created and configured from the main application page **/ Layout Editor** tab.



Clicking the **Design Console** button brings up the **Role Association** form where custom panels can then be designed and associated to MFD displays (when the role is set to Custom).



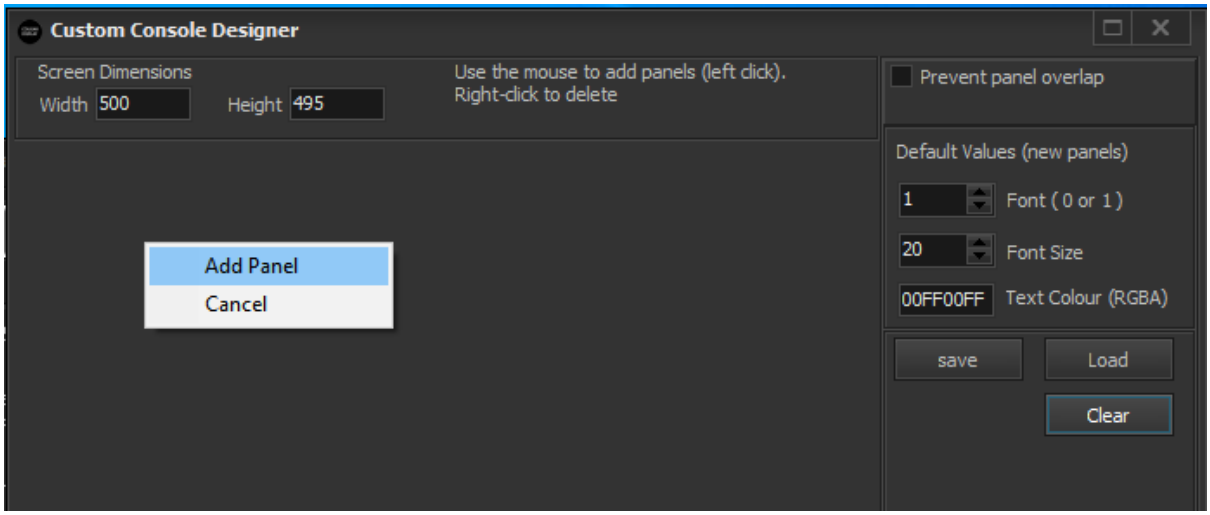
Custom Console Designer

The **Custom Console Designer** is a visual tool to help you build and design your own layouts of useful game information. It's not designed to receive mouse/touch/Thrustmaster/WingWing MFD input.

Think of them as information display panels. You can add as many as you like, in any combination to each display with the **Custom** role assigned. Panels can even be overlapped, so that you can layer text over graphics, if necessary (there is a right-click menu to bring a panel to the 'top').

The initial size of the designer is set to the size of the associated display. The screen size can be changed, however by altering the width and height settings. In reality, you won't need to do this because panels will be scaled to fit the displays at runtime. The only important thing to know is that the **positions** and **relative** sizes of the panels are persisted, and thus scaled to fit the display.

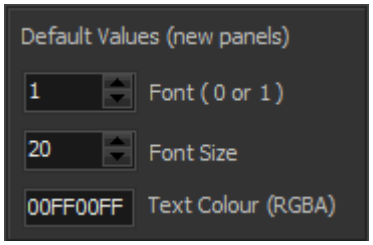
Panels are added via a left-click (pop-up menu), and deleted via a right-click (pop-up menu).



Once a panel is added, it can be **selected** by a **left** mouse **click**

… and **resized** by drag handles, and **repositioned** by holding **left mouse** button whilst **dragging**.
Tip: Holding **SHIFT** key or **CTRL** key and **left mouse** will lock the left or top of the panel whilst dragging.

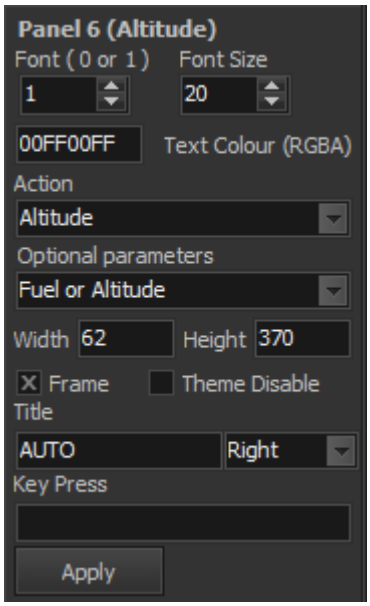
New panels initially get their default values from the **Default Values** section, which can be overridden.



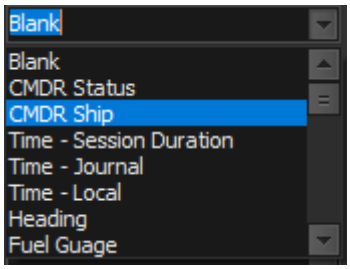
A selected panel can be **cloned**, or sent to the **front/back** relative to other panels.
Tip: Clicking the **Save** button can help resolve drawing panels in the adjusted front to back order.

Selected Panels

When a panel is selected (via left click), its properties are displayed and can be altered.



The most important value to change is the role assignment. This is the kind of information you want drawn to the display (panel).



Some roles will allow the selection of **Optional Parameters**, to fine tune the information in the panel. For example, Fuel Gauge – you can then select either horizontal or vertical.

- **Font** – You have a choice of 2 fonts (0 or 1). Set the **Font Size** to specify the initial size of the text.
- **Text Colour** – Red/Green/Blue/Alpha hexadecimal colour.
- **Width/Height** – Allows you to specify a width or height by hand (rather than drag with mouse).
- **Frame** – This draws a box around the panel.
- **Title** – this allows you to set a custom title for the panel (displayed in the top/right corner by default).
    - Choose the alignment (right, left, centred)
- **KeyPress** – When set, the key sequence is sent to Elite for processing, and the cell flashes. To add modifier keys, type as follows: **key**,**Modifier**, e.g. 1 (F1,LeftShit) e.g. 2 (L,RightCtrl)

Click the **Apply** button to save the changes to the panel.

Click the **Save** button to write the changes to a file (save as). Click the close window button to save as the default filename.

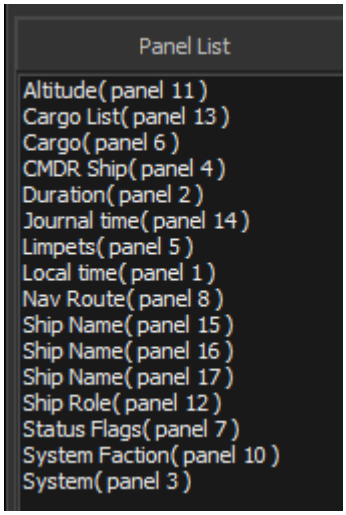
Note: The **Cockpit Editor** can be used to customise some aspects. Select the **Custom** role when in the editor.

Note2: All panels are scaled at runtime so that the contents of the text displayed can fit into the size of the panel area, both horizontally and vertically. Some panels dynamically add extra lines of information when appropriate, i.e. when they contain a value. This is typical of the statistics panels, for example.

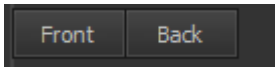
Viewing All Panels – Panel List

Sometimes the design of the custom layout requires the use of overlapping panels so that a panel overlays the background of the panel beneath. One problem this brings is how to select the panel and change its properties if it's fully behind another. Another issue with overlapping panels is that it's not possible to tell how many panels are overlapping in any one area.

The **Panel List** on the left of the designer allows for a full list of panels ordered by their action (role) so that multiple (duplicates) of a particular kind are more obvious to spot and remove.



When a panel is selected, its properties are displayed on the right hand side panel and both the **Front** and **Back** buttons are made visible at the bottom of the **Panel List** area.



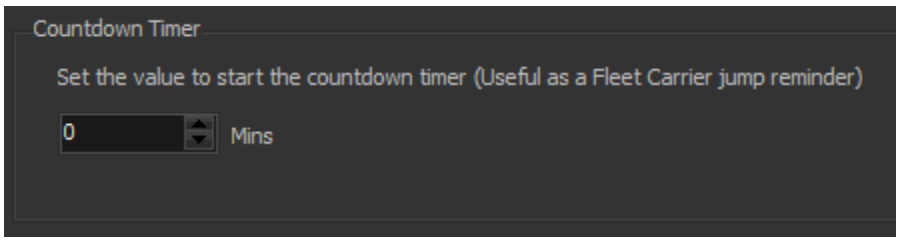
These buttons allow you to move the order of the panel to the front or back (the same way as right-clicking and selecting "Send to …." Menu item).

Notes on Some Useful Panels

Count down timer

This panel is designed to count down from a user-defined value (in seconds) to 0 and then flash. Its intended purpose is for fleet carriers – to remind you when they are about to jump (without you!).

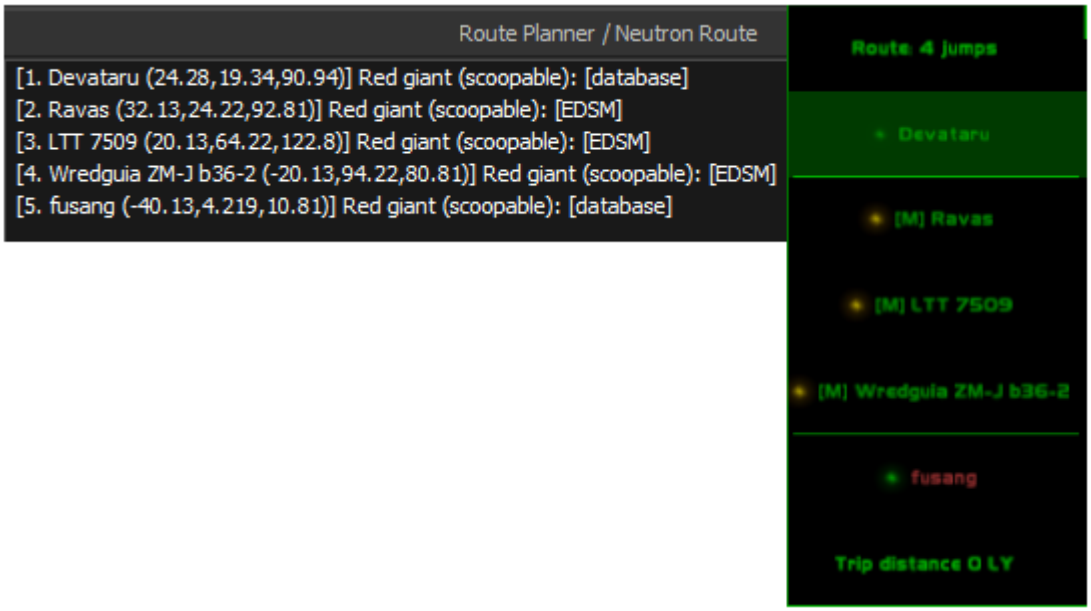
The value for this counter is set on the **Settings** tab of the **Mission Explorer.**



Nav Route

A commander can plot a route in the Elite Galaxy Map between several star systems.

This panel attempts to display this route in the space allocated in the panel. If the panel area is too small, the number of waypoints is reduced, along with the font size. The panel attempts to display as a minimum the initial start location, the destination (coloured in red) and the current position (shown within a green band). Additionally, it will display the previous star system(s) and the next up-and-coming systems – how many it displays depends upon the available screen area.



The source of the known status of the star system is listed as unknown, EDSM or database in the Mission explorer window, or shown as either a red, yellow or green icon in the custom panel.

The class of the star is shown in both the custom panel (as a bracket, eg [M]) and on the mission explorer in plain English, eg Red Giant.

FSS Materials

A useful panel for explorers. This panel displays an easy to read grid of the raw materials available on the landable planet that's just been scanned. The grid is organised in columns relating from grade 1 to grade 4. Additionally, this display will show materials of the planet you are about to orbital descent into.

Tip:

- Whilst the application is running, you can access the Custom Console Designer and make changes to the panels. The application will automatically apply those changes without the need to restart the app. This allows you to switch out panels according to your play style in Elite. It's also possible to have several sets of panels already saved in a folder location, so that you can switch between them rather than having to design the layouts whilst playing Elite.
- Combine panels into a single larger area by first drawing a blank panel with a frame, then adding other panels and dragging over the top of the first. These subsequent panels can be drawn without a frame, and also made to overlap regions of interest, e.g. combine CMDR Ship and CMDR Status.

Blank

Blank panels can be useful to surround groups of other panels, for example to visually enclose them in a single frame.

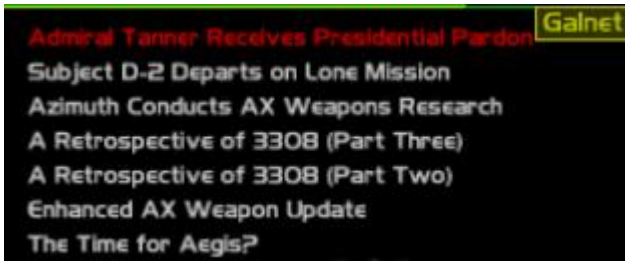
Additionally, blank panels can have a custom background, and you can add external images to Cougar Display. Additionally, Cougar Display has a variety of pre-built background materials (textures and shaders) to select from (widgets). More information is available for these widgets in the section below (Widgets/Materials for Blank Panels).

Background

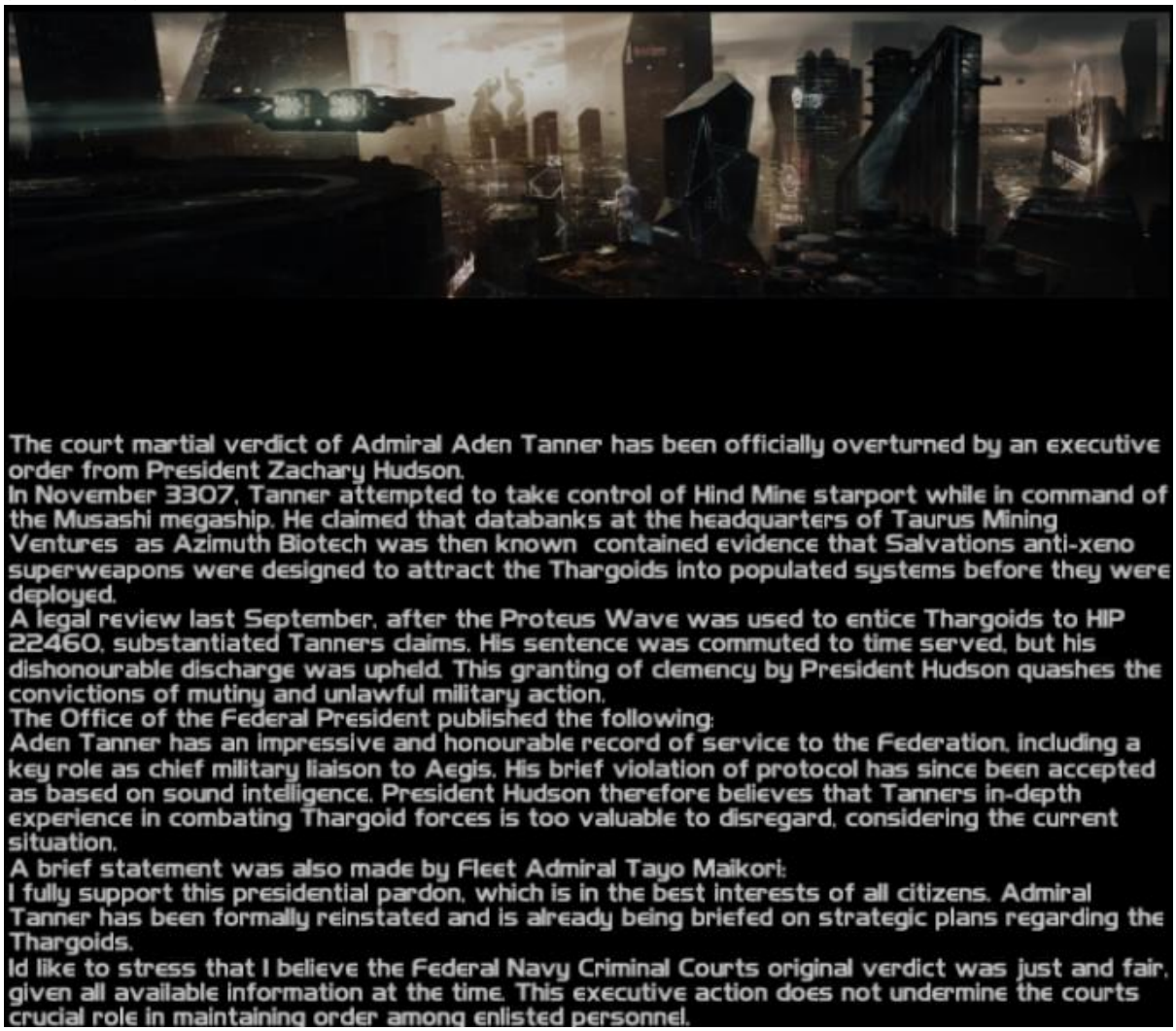
One setting specified in the custom node of the Layout Editor is the background – this value can be set to one of the pre-configured values (eg Tron, Nebula, Swirl or Hexagons). This option is used to draw a background across **all** custom panel consoles.

Galnet Panel

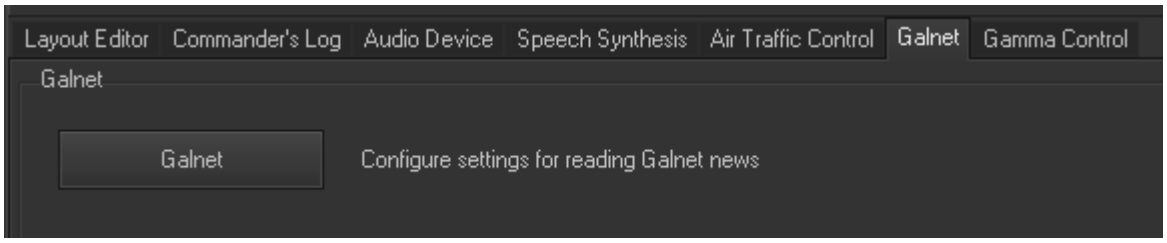
The Galnet Panel is a special custom panel that can receive hot key input to drive the user experience. This panel fetches Elite Dangerous Galnet news articles and displays the list in the panel.



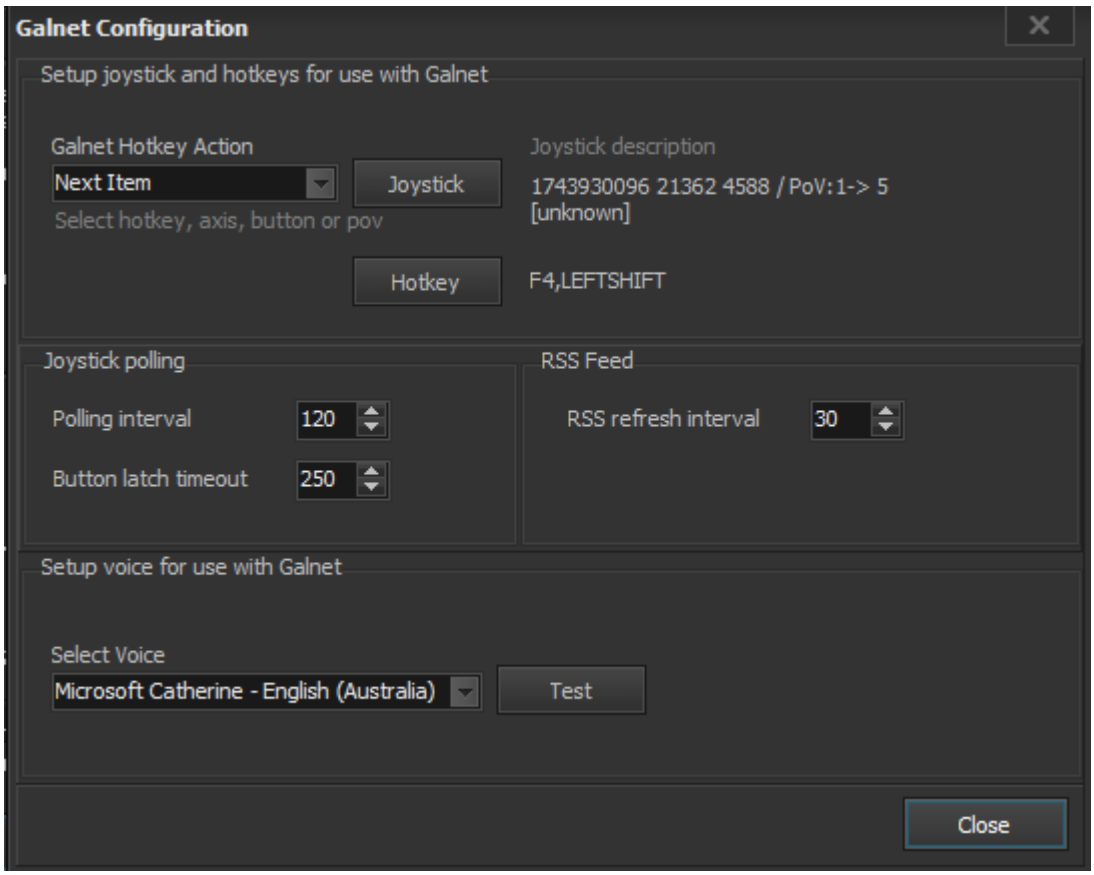
Using assigned hot keys, the Galnet panel can then scroll through the news list and display articles (full screen) and / or read out news.



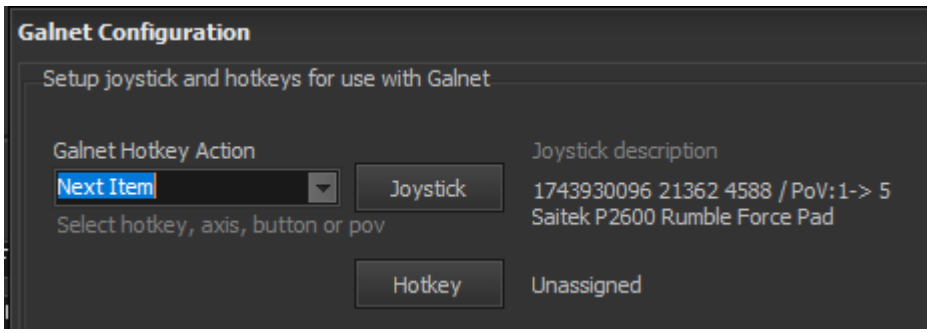
The **Galnet custom panel** has a dedicated user interface, accessible from the **Galnet** tab on the main application.



Galnet Configuration

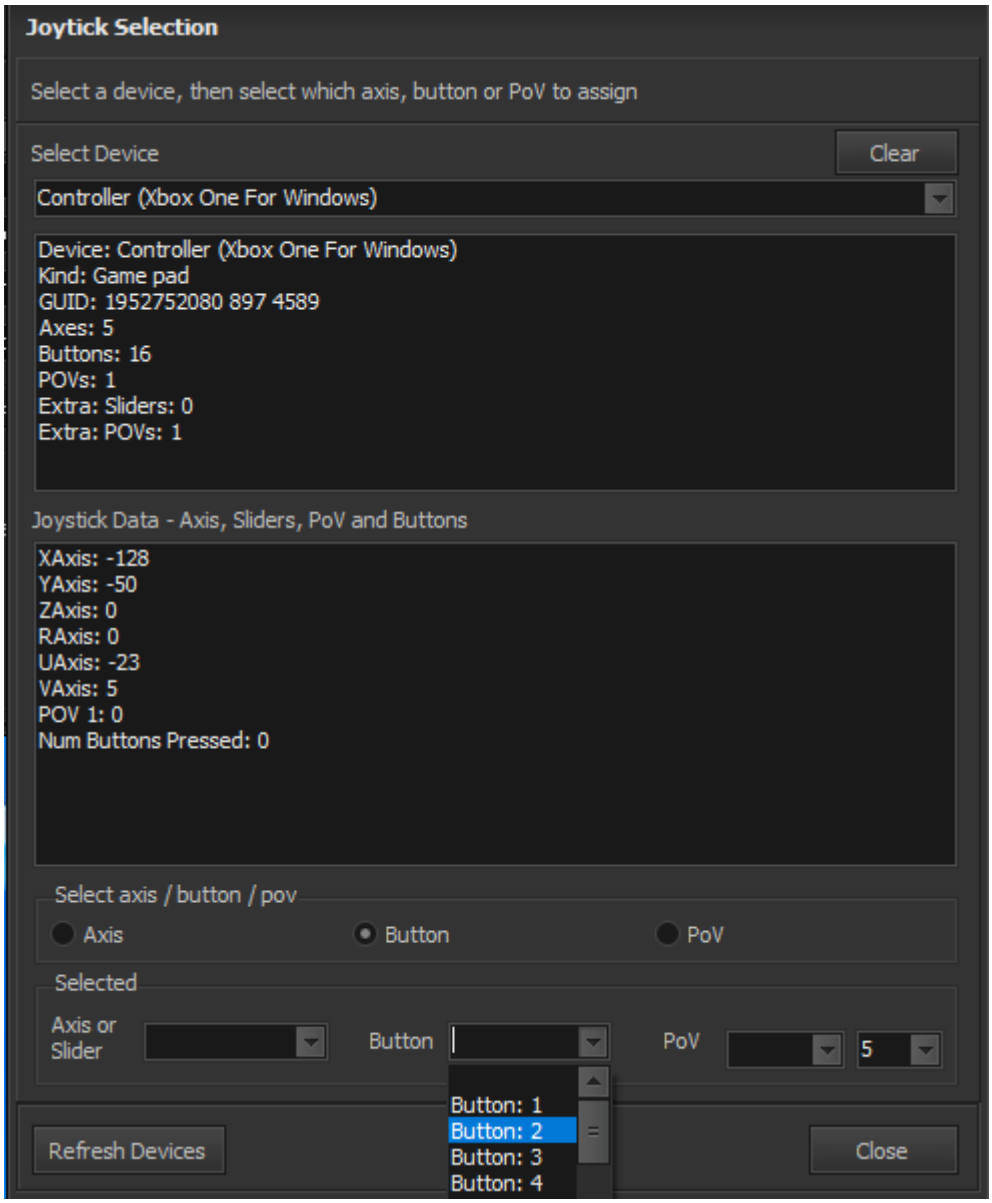


Select the **Galnet Hotkey Action** to configure and press the **Joystick** or **Hotkey** button to assign the device/key press.



Joysticks

Hot key actions can be assigned to any joystick on your system using any button or PoV hat. Axis can be assigned to the **Next Item** or **Previous Item** Galnet actions.



The chosen joystick action is displayed, along with the specific PoV hat value or button.

It is important to test ED:CD's ability to read the device whilst focusing on another application. Use your mouse and click on another application, eg Notepad ++, a browser window, etc and wiggle the joystick device to see if the input is being detected.

Select either an **axis/slider**, a **button** or **PoV** value as the hot key input. Selecting a blank value un-assigns the device from hot key input.

Note: During testing the only joystick which failed to be read as a background task was the **Turtle Beach (Xbox One For Windows) Controller**. This device required MFD Cougar display to be the foreground application when reading the joystick – and thus it's not suitable as a hot key device.

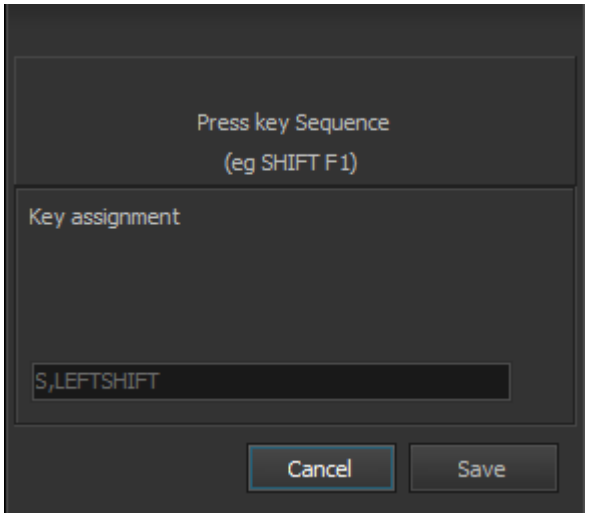
Joystick Polling

The joystick **polling interval** (milliseconds) can be adjusted between 50 – 250 (120 default). The lower the number the more frequent polling and the faster the response to input.

The **Button latch timeout** (milliseconds) can be adjusted between 100 – 500 (250 default). This is the delay between pressing the joystick and being able to press the same button again. It's used to latch the button to prevent too many events firing sequentially.

Hotkey Assignments

Keyboard hot keys can be assigned and mapped to Galnet actions.



The current modifier key is shown on the title bar (Left Shit, Right Ctrl), however with Hot keys there are no distinction between left and right.

Press **Cancel** to abandon the hotkey assignment, or **Save** to keep it.

Note: Certain hot key presses are disallowed by Windows (eg F12), or may already be registered with other applications. See **Elite Key Assignments** (**Additional Notes for HotKey)** in the **Layout Editor** section for more detailed information on hot key limitations.

RSS Feed Interval

The RSS feed interval can be set to a value between 10 – 250 minutes (30 default)

Galnet hot key actions

The list of actions to which hot key input can be assigned is shown in the table below:

Action	Description
Next Item	Assign a joystick input or hotkey to scroll down the list of Galnet news articles
Previous Item	Assign a joystick input or hotkey to scroll up the list of Galnet news articles
Toggle fullscreen	Displays the full Galnet article in the custom panel console. The panel becomes full screen to make reading easier
Read Item	Uses the selected TTS voice to read the news article. **Note** this will disable ATC radio chatter

Galnet Voice

Select the Voice to be used with the Galnet news item reading event (hot key action). This voice will speak without any audio effects (which you normally get with the ATC chatter).

Note: For clarity, **Air Traffic Control** (ATC) chatter is disabled when reading news items. Assign a ATC hotkey from the cockpit **Layout Editor** to toggle the ATC chatter on/off as appropriate after reading Galnet news.

Saved Settings

Galnet configuration settings are stored in the registry - `Computer\HKEY_CURRENT_USER\Software\BionicBytes\EDCD\GalnetRSS`

Custom Panel Themes

The colours used to style specific panels are set via a combination of the **Text Colour** on Custom Panel Designer and various elements from the **Layout Editor** (custom role), such as **Icon Colour On/Icon Colour Off** and **Target Text**. Whilst this scheme offers precise control of individual panels, it's not a quick process or even possible to change all the panels on a custom console, or allow specific custom consoles to have a different set of colours.

Enabling **Theme Colours** solves these problems by defining an overall set of colours (theme) which is used globally by all custom panel console screens (.mfd files). Change the global theme colours in one central place and all custom panels immediately reflect the change.

Theme Hierarchy

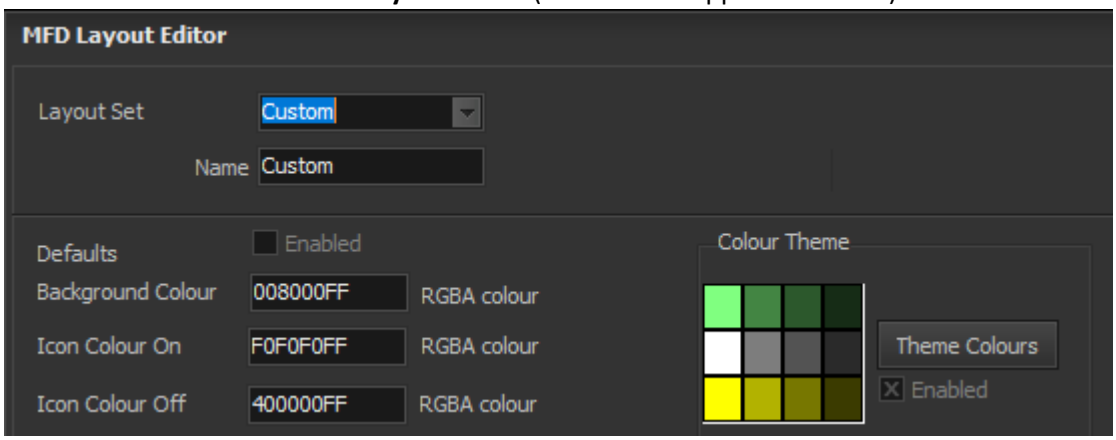
Layout Editor (MFD_layout.xml) → Per Custom Panel Theme → Individual Panels (disable theme option)

The **Layout Editor** sets these colours in the **custom** layout, which is a node stored in the **MFD_Layout.xml** file. When enabled, the theme colours apply to all custom panel console screens and the console title/menu region, but can be overridden by a theme set defined in the custom panel. This hierarchical theme arrangement allows for a quick and easy way to globally set colours for all custom consoles, and yet still provide a way to have specific consoles with different colours.

Themes make customising panels much quicker as it's no longer necessary to edit each individual panel's colour.

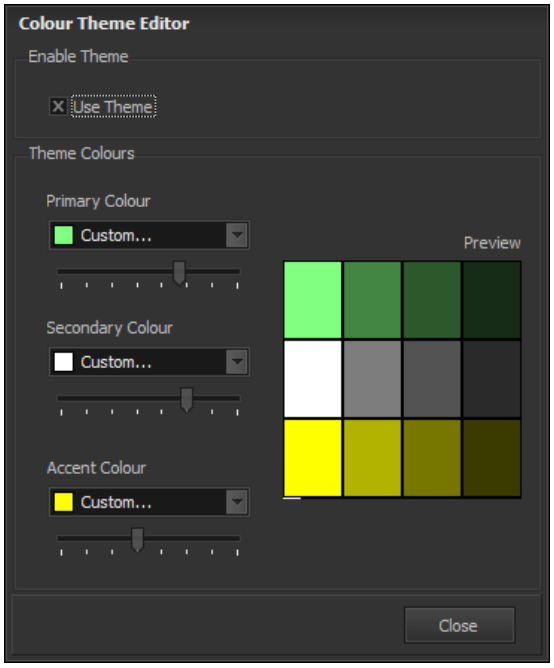
Setting Themes

Themes can be defined in the **Layout Editor** (on the main application form).



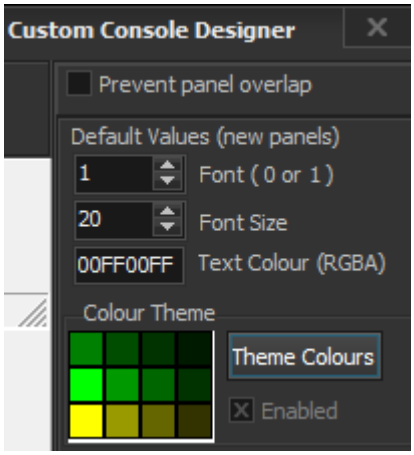
Clicking the **Theme Colours** button will show the **Colour Theme Editor**.

Themes consist of 3 colours (primary, secondary and accent) and 4 shades of each colour. Changing the slider beneath each colour adjusts the shades.



The **Use Theme** checkbox is used to enable/disable the theme.

Themes defined in the **Custom Console Designer** override the theme defined in the **Layout Editor** and allows for per-console themes.



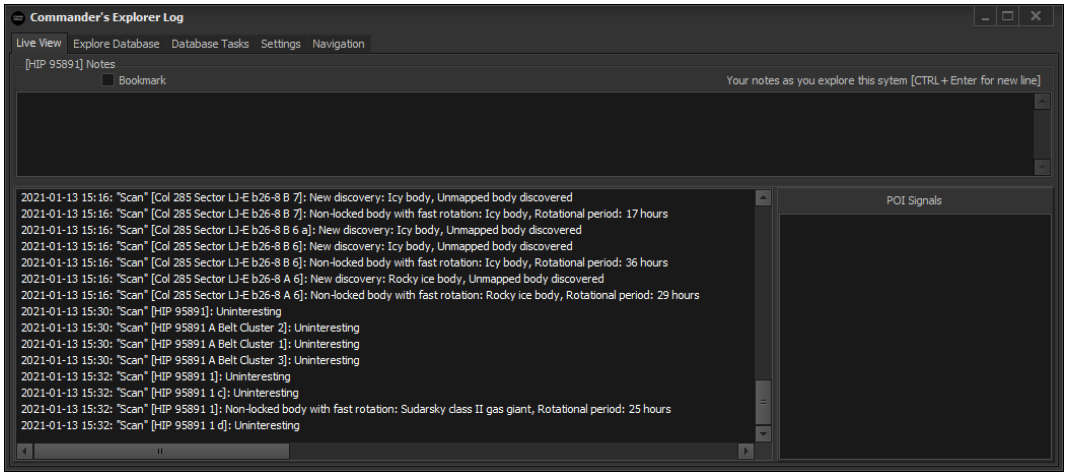
An enabled theme can override the colour specified in the individual panels which make up the custom console. However, each panel has the property **Theme Disable** which is used to disable the theme for the current panel and the **Text Colour** property is used by the shader/material. Note, however, theme colours are always used to draw the panel frame and title text if a theme is enabled (console or Layout editor). The most common reason to set the **Theme Disable** property is for one of the special shaders with the [Blank] action.

Note:
To customise the colour of the Custom Panel Menu/Title area – see “Customising the Custom Panel Menu Colours”.

Example Panel Layouts

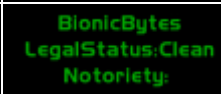
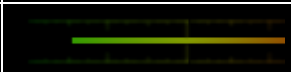
The image below is the example file “console.exploration.mfd” which ships with the application.

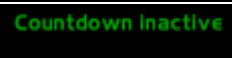
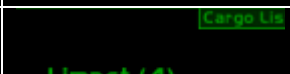
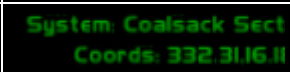
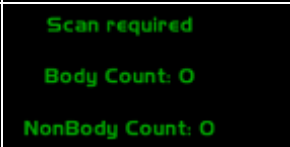
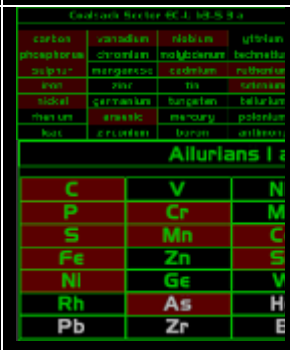
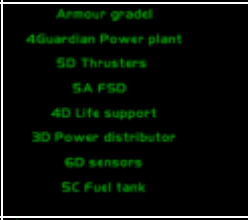
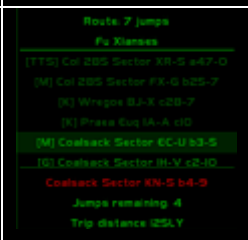
When combined with the Commander’s log, exploration is extremely efficient.



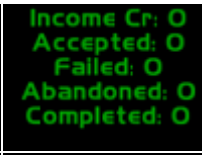
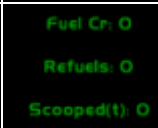
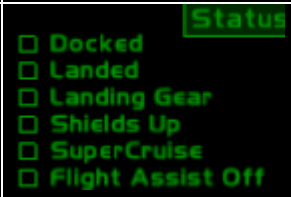
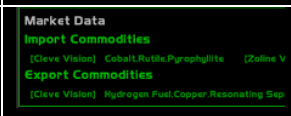
Panel Descriptions

Customising the “Custom” role in the console designer can effect certain panels. These are notes in the description column where appropriate.

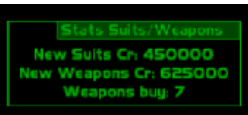
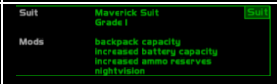
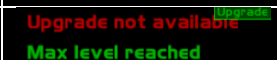
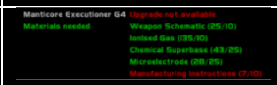
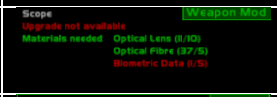
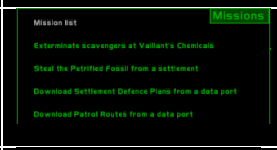
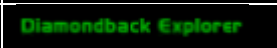
Panel	Example Image	Description
Blank		Useful to create a frame around unrelated panels Also used to draw a background or display one of the panel widgets
CMDR status		Optional Parameter – current balance
CMDR Ship		Image of main ship/SRV/SLF Optional parameter – switch between normal, and blueprint image(s) StatusText attribute sets the ship colour.
Time – session		Duration (mins) of current session
Time – journal		Last time journal event was processed
Time – local		Windows system time
Heading		When approaching a planet – gravity, lat/long, heading, altitude
Fuel gauge		Combined main / reserve. Horizontal needs to be at least 70 pixels in height. Optional parameter – horizontal/vertical
Shields		
Weapons pips		
Engine pips		

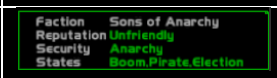
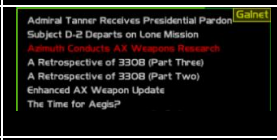
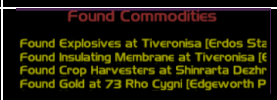
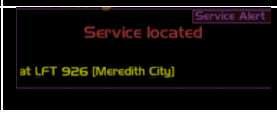
Altitude		Displayed when altitude is activated in Elite. When optional Parameter is set to “Fuel or Alt” the gauge switches to displaying altitude or fuel accordingly. When the title is set to AUTO the title will dynamically change to ”Fuel” or “Alt”
Countdown timer		A timer which counts to zero. Useful for fleet carriers
Limpet count		Number of limpets in cargo
Cargo count		Number of items and capacity
Cargo list		List of all items/quantity in cargo bay
System name		Current star system name
FSS Discovery		After “honk”, number of bodies in system and % discovered
FSS Materials		Raw materials on a planet – either by FSS scan or when approaching planet (orbital descent) Background colour set via IconColour Off attribute Optional Parameter – display element symbol Materials with less than 50% of maximum grade capacity are written in white text to draw your attention they are low
Ship loadout		List of modules in ship Optional parameter – choose between weapons, utility, core, optional modules, basic modules, all
Nav route		List of star systems plotted in Elite. Starting system – top row, destination system – bottom row. Current position highlighted, with next in route below. Red,yellow or green icon indicates unknown, EDSM or known (local database) system [ ] indicates star class, eg [M] main sequence

	[M] Ravas [M] LTT 7509 [M] Wredguia ZM-J b36-2	Attempts to list what's coming next in route and some of the previous systems, space permitting. TargetText attribute sets the destination colour.
Target	Ship: Orca Pilot: Michael Harris Rank: Novice Hull: 100.00 Shield: 100.00 Faction: Green Party of Fu Xian Status: Clean	Current target ship, pilot, rank, shield/hull health, faction, status, enemy ship, human player
Target Ship		Image of targeted ship
Stats Credits	Redeemed Cr: 0	Income/expenditure from: Fines,refuel,repair,rearm,ships,modules,bounty,bonds, missions,trading and exploration. **Only** shows an entry when credits for particular activity have been recorded
Stats Trading	Trading Profit Cr: 0 Sold Items/Cr: 0/0 Bought Items/Cr: 0/0	Credits earned from trade; Sold number items & Credits; Bought number items and Credits
Stats Exploration	Exploration Cr: 0 Matrials Collected: 0 Materials Discovered: 0 FSS Bodies: 0 New Discoveries: 0 Scan New Bodies: 0 Scan Nav Beacon Scans: 0 Scan New Discoveries: 0 Scan UnMapped: 0 FSD Jumps: 2 Jump Distance(ly): 125 Jump Fuel Used(t): 10	Credits earnt, number of materials collected, mats discovered, FSS bodies (scanned), FSS new discoveries; scanned – new bodies, nav beacon scans, new discoveries, unmapped; FSD jumps – total distance, fuel used (tons)
Stats Combat	Combat Cr: 0 Faction Bonds: 0 Faction Cr: 0 Bonds: 0 Bonds Cr: 0 Bounties: 0 Bounties Cr: 0	Credits earnt via combat, faction bonds, bonds and bounties
Stats Mining	Trading Cr: 0 Limpets used: 0 Ores Refined: 0	Credits earnt, number of Limpets used and Ores refined
Stats Docked	Docked: 0 Rearms: 0 Repairs: 0 Refuels: 0 Rearm Cr: 0 Repair Cr: 0 Refuel Cr: 0	Number of times docked, number of rearms, refuels, repairs and associated Cr
Stats Crimes	Fines Cr: 0 Fines: 0	Fines (Cr), number of fines

Stats Missions		Income (Cr), number accepted, failed, abandoned and completed
Stats Fuel		Cr spent, number of refuels, tons scooped
Status Flags		Optional Parameter – grid or vertical arrangement, or Individual status flags. "Iconcolour Off" attribute sets the shaded background colour Panels flash on/off when Elite status is one of the following: Fuel Scooping, Low Fuel, FSD Charging, In Danger, Interdiction, Overheating, Silent Running **MFDCougar.ini** is used to disable flash and set the flash rate. **[StatusBits]** Flashrate=500 Flash=1
Rank Reputation		Optional Parameter – Rank, Rank/Reputation, Reputation. "Iconcolour On" attribute sets the heading text colour
Ident		Ship's ident as specified in the journal
Ship Manufacturer		Ship's manufacturer, eg Lakon, Core Dynamics
Market		Displays Import and Export commodities for the system. Optional Parameters 1-10 increase the message scroll rates. By default, longer lines scroll faster than shorter ones.
Hyperwarp		Displays FSD jump animation and next star information. Optional Parameter used to select which of the two MFD animation screens.

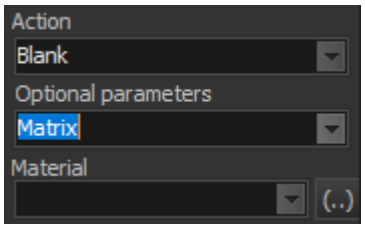
PlayerVitals		Player health/oxygen and environment temperature/gravity
		
		Edit MFDCougar.ini to change the temperature scale between Kelvin, Celsius and Fahrenheit.
		
		`[Odyssey]`
		`Temperature=C`

Stats SuitWeapons		Statistics – number bought/sold, Credits, Upgrades. Suits and Weapons
Status Flags 2		**Odyssey** status flags. Optional Parameter – grid, vertical or Individual, arrangement. “Iconcolour Off” attribute sets the shaded background colour Panels flash on/off when Elite status is one of the following: FSD Hyperdrive, Low Oxygen, Low Health **MFDCougar.ini** is used to disable flash and set the flash rate. **[StatusBits]** Flashrate=500 Flash=1
Suit		Suit name and class. List of equipped suit modules
Suit Upgrade		List of materials needed to allow upgrade
Suit Mods		Table showing Module name and Materials, along with current quantities and required quantities of materials. Red denotes that module isn’t available for upgrade. Optional parameter can be used to display specific module or all modules (current suit)
Weapon Upgrade		Optional Parameter is used to select specific weapon and level. Table shows availability of materials to upgrade to this level.
Weapon Mod		Shows progress towards specific modification, eg scope
Mission List		List of Odyssey on foot missions
Ship Name		Displays ship name (eg BIONIC DBX)
Ship Role		Displays classification depending upon loadout, eg Exploration, Anti Xeno, Mining, Pirate, Passenger
Ship Type		Displays ship type (eg ASP Explorer, Python)

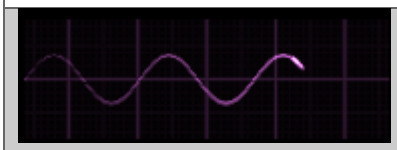
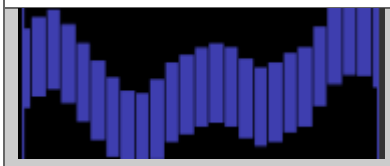
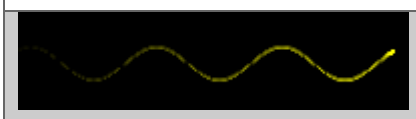
Ship Manufacturer Logo		Draws the logo of the current ship. Optional Parameter – select one of the specific manufacturers
System Faction		Displays information on the current system faction – Name, security level, your standing reputation and active states
SRV		Displays either the SRV or SRV Scorpion (or current ship). Optional parameter can also be used to display the current ship if not in SRV
Galnet		Galnet news. Assign hotkeys/joystick to select article and display or read the item. Galnet expands to full screen when displaying the article.
Texture	Select the material and slice (Optional Parameter)	Allows the custom panel to display any Cougar Display texture. Some textures have multiple layers which can be selected via Optional Parameter.
Commodity Alerts		Upto 20 alerts can be created in the Mission Explorer. Select different display styles
Service Alerts		Service alerts can be created for each of the INARA services on the Mission Exporer (INARA tab)

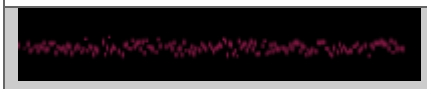
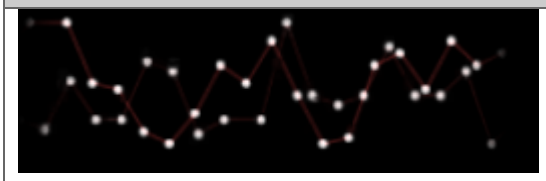
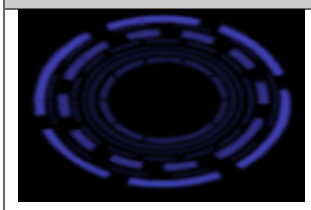
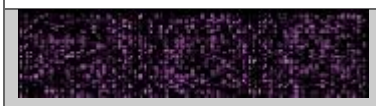
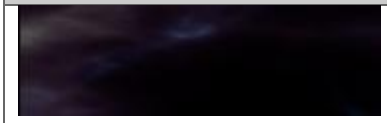
Widgets/Materials for Blank Panels

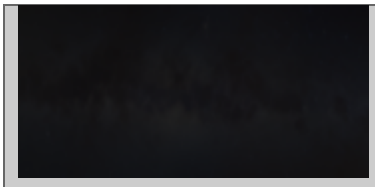
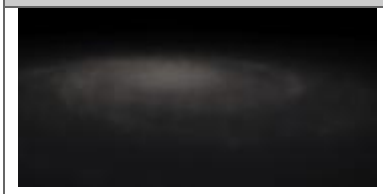
On the **Custom Console Designer**, blank panels (panels where the action is set to 'Blank') have an extra set of configuration properties available which allow a set of pre-defined widgets (shader code and textures) to be drawn.



When the **Action** is 'Blank', and the **Optional Parameter** is NOT set to 'Blank/No Shader', then the **Material** property is shown and populated with custom textures. **Optional Parameter** can be one of the following values:

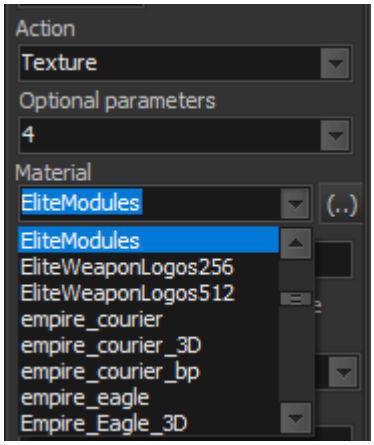
Widget (Material)	Description
	**GalaxyMap** The current position is shown on the map and blinks in panel's font colour. Alternative galaxy images supported via material property
	**ColouredBars** Vertical strips of coloured bands
	**Oscilloscope** Draws background grid and a scope line as a single widget
	**Parallax** Draws 3 sets of graphs, moving at different rates
	**WallWave** A sine wave of vertical blocks
	**GraphGrid** Used as background for other widgets to sit on top, eg Sine Wave, Graph, OscilloscopeLine
	**Oscilloscope Line**

	**Sine Wave**
	**Graph** Scatter graph
	**DualScrollingGraph** Two series graphs superimposed on top of each other
	**ScrollingLines1**
	**ScrollingLines2**
	**ScrollingBars**
	**CircleSpin** Material should be empty for default texture. Alternate material can be specified, and must be a texture loaded in **Custom.TextureFX.txt.** This texture needs 3 channels, each channel is rotated at a different speed / direction and the 3 channels combine to form the image.
	**Matrix** Matrix rain effect
	**GlowPulseWave** Glowing line, rising up the display
	**Pulse wave** Sparks pulsing from side to side
	**Static** White noise static
	**Swirl** Animated background effect
	**Nebula** Animated background effect

	**MilkywayEdge** Use a low alpha value for the Font Colour property to avoid over exposure, eg FFFFFF0A
	**GalaxyBackground** Galaxy image

Materials for Texture Panels

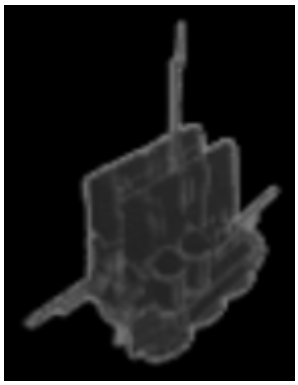
On the **Custom Console Designer**, texture panels (panels where the action is set to 'Texture') have a **Material** and **optional** slice **parameter** to select a specific OpenGL texture/slice.



The **Material** drop down has the entire list of OpenGL textures (see Extending Cougar Display Texture Packs section) available for use within the custom panel. Some of the textures have multiple layers, and when they do the **Optional parameters** shows the slices to choose from.

Elite Icons

Some graphics from the game have been included in the Elite texture pack. These are exposed via the **Materials** "EliteModules", "EliteWeaponLogos256" and "EliteWeaponLogos512".



Material	Available Layers/Slices
EliteWeaponLogos256	Takada, Supratech, Remlock, Manticore, Kinematic Armaments
EliteWeaponLogos512	Takada, Supratech, Remlock, Manticore, Kinematic Armaments
EliteModules	AFMS, Cargo, limpet, Experimental Weapon Stabiliser, FSD Booster, FSD Interdictor, Fighter Hanger, Flight Assist, Hull Reinforcement, Passenger Cabin, Vehicle Hanger, Shield Generator, Shield Cell Bank, Scanner, Refinery, Power Plant, Thrusters, Life Support, Power Distributor, Heat Sink, Point Defence, Chaff, Multi cannon, Beam Laser, Missile Rack, Mining Laser

Note: For the application to pick up the available list of materials/slices for **Texture** panels, the application must have "Started" – ie rendering OpenGL images.

Texture Materials and Custom Materials

Custom Panels with the **Action** set to **Texture**, also work in conjunction with **Custom Materials** (see section **Custom Materials**) for rendering externally supplied images.

As a recap for Custom Materials - you can link various external image files for use as a texture pack and use the **Texture FX Editor** to create the custom texture pack, then each item you added can be selected as a **material** in the material list (the names will be generic, ie “Custom_1”, “Custom_2”, etc). The linked images are then copied into the application’s **bmp\custom** folder.

Custom Panels with the **Action** set to **Blank**, also work in conjunction with **Custom Materials**, just like **Texture** panels. However, **Texture** panels list all available OpenGL textures whereas **Blank** panels list only custom materials.

A More Flexible Way…

if you are looking to add extra images to display in the custom panels and want more control, then follow the procedure below to add extra OpenGL textures into Cougar Display so you can render them onto the custom panel via the **Texture** panel.

Extending Cougar Display Texture Packs

This is an advanced option and involves directing OpenGL to load a texture file. If you incorrectly specify a parameter, or the texture is the wrong file format/unsupported an error will most likely be generated in the LOGS folder – the application should swap to a blank texture but it is possible that you could cause an internal crash – be warned!

Edit the **MFDCougar.ini** file

```
[SceneTextureFX]
NumTextureFX=15
TextureFX1=FX\Scene Files\MFD\Sun_flares.TextureFX.txt
TextureFX2=FX\Scene Files\MFD\MFD.TextureFX.txt
TextureFX3=FX\Scene Files\MFD\Stations.TextureFX.txt
TextureFX4=FX\Scene Files\MFD\Ships3D.TextureFX.txt
TextureFX5=FX\Scene Files\MFD\Ships2D.TextureFX.txt
TextureFX6=FX\Scene Files\MFD\Outpost_pad1.Texture3DFX.txt
TextureFX7=FX\Scene Files\MFD\Outpost_pad2.Texture3DFX.txt
TextureFX8=FX\Scene Files\MFD\Outpost_pad3.Texture3DFX.txt
TextureFX9=FX\Scene Files\MFD\Outpost_pad4.Texture3DFX.txt
TextureFX10=FX\Scene Files\MFD\Outpost_pad5.Texture3DFX.txt
TextureFX11=FX\Scene Files\MFD\Orrery.TextureFX.txt
TextureFX12=FX\Scene Files\MFD\Shipyard.TextureFX.txt
TextureFX13=FX\Scene Files\MFD\ShipyardShips.TextureFX.txt
TextureFX14=FX\Scene Files\MFD\Odyssey.TextureFX.txt
TextureFX15=FX\Scene Files\MFD\Elite.TextureFX.txt
```

Increase the `NumTextureFX`, eg change 15 to **16**

Add an extra line to the texture pack list, replacing *`yourtexturepackname`* with a unique filename.

`TextureFX`**`16`**`=FX\Scene Files\MFD\[`*`yourtexturepackname`*`].TextureFX.txt`

The `[`*`yourtexturepackname`*`].TextureFX.txt` file should then contain the following as a minimum:

```
[Global]
CacheManager=y
Numtextures=1

[Texture1]
Alias=cmdrname_shield
TextureCategory=custom
Description=shield
FileName=custom\Shield.png
Target=GL_TEXTURE_2D
Mipmap=y
MinFilter=GL_LINEAR_MIPMAP_LINEAR
MagFilter=GL_LINEAR
WrapS=GL_CLAMP_TO_EDGE
WrapT=
WrapR=GL_CLAMP_TO_EDGE
Bias=0
Anistropy=1
Scale=1
```

Notes:

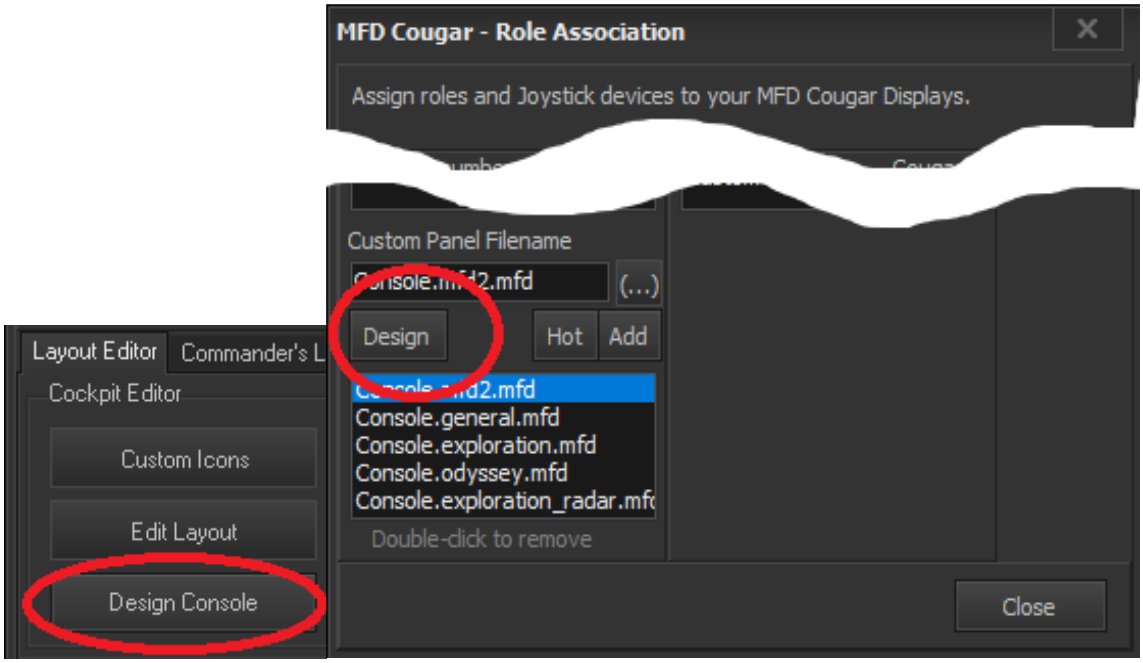
- For each new and additional image, add a new [**Texture**] section.

- **Alias**: Specify a globally unique alias (needs to be unique across all texture packs). It's suggested you prefix with your Elite Commander name
- **FileName**: Specify the relative path of the image file. **Custom** is a suitable location for 3rd party images. Supported images are typically 24/32-bit, bmp/tga or 24-bit png/jpg
- Update the [**Global**] **Numtextures** to match the number of [**Texture**] sections.

Custom Materials

When **Custom.TextureFX.txt** file is detected on application start-up, user-defined textures (materials) are available and added to the **Custom Panel Designer's** list of **materials** (as "custom_1", "custom_2", etc). It a simplified mechanism to add external (aka custom) resources into Cougar Display for use with Custom Panels. For more direct control over the texture creation, follow the Extending Cougar Display Texture Packs section.

Materials can be accessed via the **Console Designer** user interface.



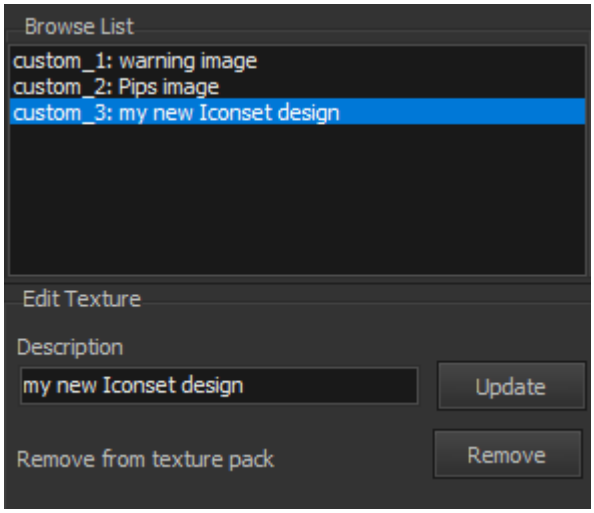
And clicking the **Design** button from the **Role Association** screen.

Use the **Panel Designer**'s browse **Material** button **(..)** to add new images to the custom file.

The custom file **Custom.TextureFX.txt** is automatically created by the user interface and stored in **MFDCougar\FX\Scene Files\MFD** folder; image files used to build this file are copied into **MFDCougar\bmp\Custom**.

Browse for .tga, .bmp, .jpg or .png images and type a description for the new image file. Click **Add** button to add the image to the texture pack.

Existing images in the texture pack can have their descriptions altered or the image can be removed from the texture pack – however some textures may be in use by custom consoles (in which case a warning is displayed).



The **Custom.TextureFX.txt** file is saved to disk when the application exits, but the textures are immediately available to custom panels for use.

When the **TextureFX Editor** is closed, remember to set the **Material** value in the **Console Designer** – as the selection is reset.

The auto-generated alias names added via the editor is very simplistic and limited to “custom_n”, where n is a sequential count. The system can get very confused when you remove textures from the **custom.textureFX.txt** file as the alias numbering sequence may no longer match up to the entries in the texture file. You are free to edit the **alias** values of the **custom.textureFX.txt** file and match those entries against **Console.*.mfd material** values.

Note: Custom files are not backed up by the auto upgrade process, but equally they are not overwritten by an upgrade.

Note2: The **Custom.textureFX.txt** and user interface are only available when the application “starts”, i.e. when the OpenGL MFD display windows are rendering images.

Widget Colour

The colours used in the widgets are set via the **Font Colour** property, unless a colour theme is active in which case the colour is derived from the set of colours set by the theme.

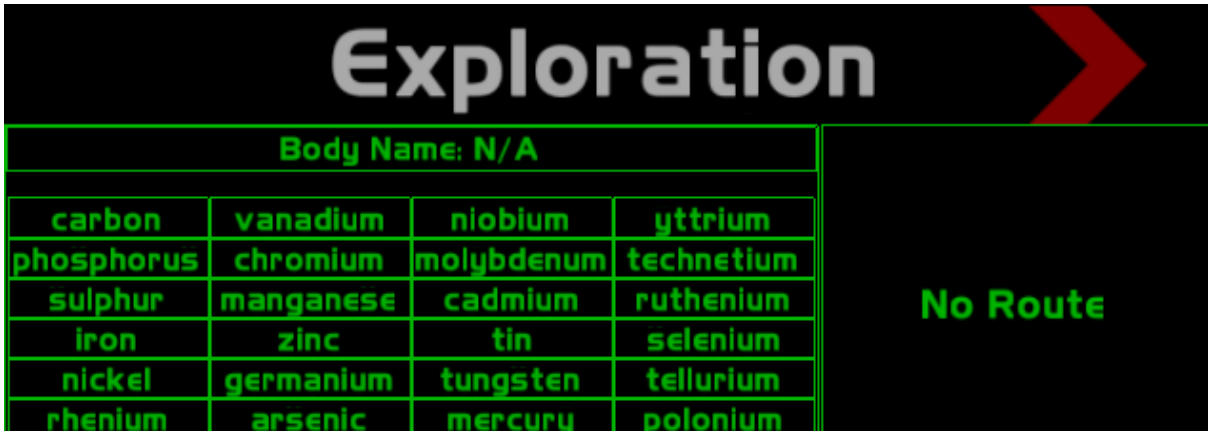
With previous versions of Cougar Display, it was not always consistent or well defined which colours were used to render some of the widgets. Most often, the primary colour was obtained using the panel colour property (Console Designer) with secondary/tertiary colours coming from IconColorOn/Off (Layout Editor) or as a fixed colour, eg white or red. As from version 1.190.0.0 themes are the best way to select the colours used for widgets because the theme colours are always used when the theme is enabled.

Switching Between Custom Panels

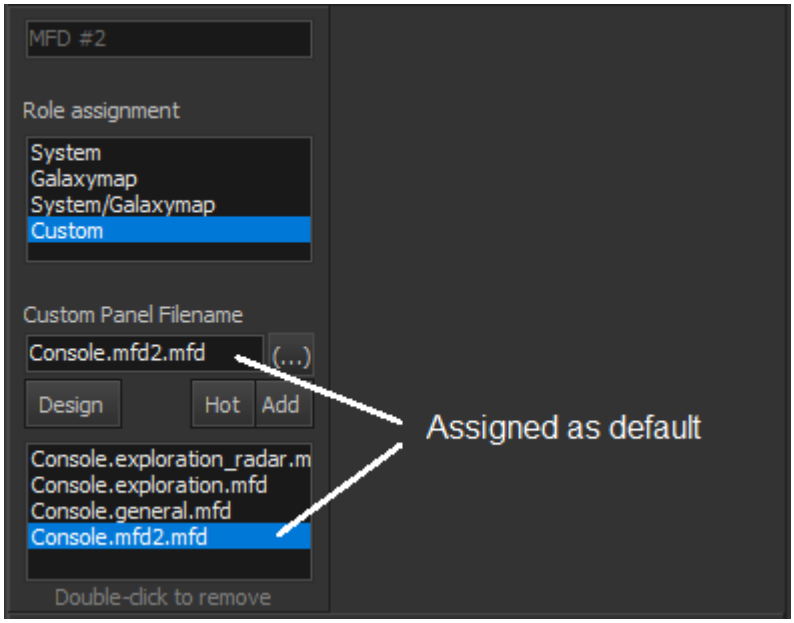
Sometimes, when you have multiple custom panels designed, it would be nice to swap them out whilst still running the application. This can be done by enabling more simultaneous displays and assigning another custom role, or by using the role designer to re-assign the panel's filename.

Adding a Panel Menu to the Custom Panel

A neater solution to switching between multiple custom panels on the same display is to enable the automatic menu feature for the custom panel.



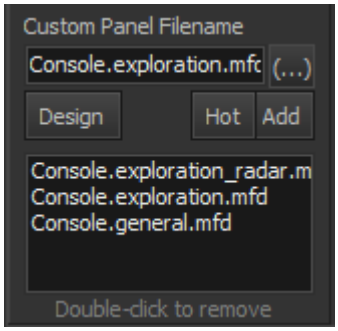
In the Role Association, click **Add** button after selecting a panel filename. Type in a title for the panel (so that you know what the panel contains). Cougar Display will reserve 10% of the vertical screen space for use by the menu system and clicking on the left/right chevrons (**< >** images) will automatically switch to the next panel in the sequence. If you do not wish to display the menu system, do not type a title when clicking the **Add** button.



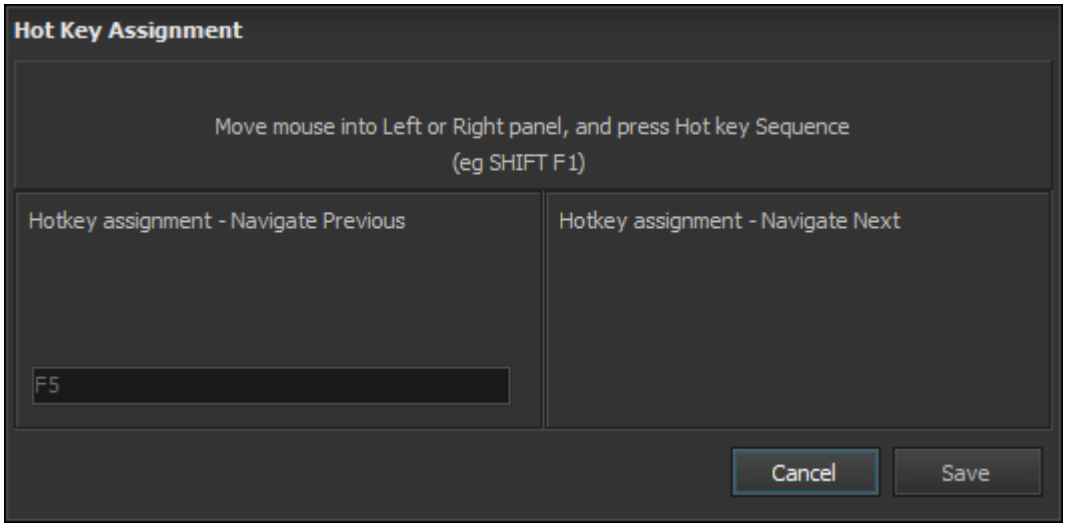
- Double-click an item in the list box to remove a custom menu
- Clicking on an item makes the custom panel the default (and can now be **Designed**)
- The UI allows for 5 custom panels to be assigned per display

HotKey Assignment

When multiple custom panels are assigned to a display, the **Hot** button becomes available for assigning hot key presses.



Move the mouse into either the left or right hand panel to register a key press for either the **Navigate Previous** or **Navigate Next** custom panel.



Note – Not all key presses are available as hot keys (eg F12) and some may be taken by other applications. See **Elite Key Assignments** (**Additional Notes for HotKey)** in the **Layout Editor** section for more detailed information on hot key limitations.

Manual Adjustments to the Custom Panel's Menu

When custom panels with titles have been assigned to a role, or when hot keys have been configured, these settings are saved into the **`MFD.INI`** file in the section for each MFD.

The following table shows the purpose of the INI file setting. You can manually edit these values, for example to re-order the custom panels or change/remove the titles. It may be quicker and easier to edit the file directly rather than use the user interface to perform the same function.

Setting	Purpose
CustomPanels	Default (initially displayed) custom panel filename
CustomPanelsList	Comma separated list of custom panel filenames
CustomPanelsTitles	Comma separated list of titles for the list of custom panels. Can be left empty to remove all titles (and thus remove the menu system across the top row)
CustomPanelsHotkeyPrev	Key press assignment
CustomPanelsHotkeyNext	Key press assignment

Example:

```
[MFD3]
Role=6
CustomPanels=Console.mfd2.mfd
CustomPanelsList=Console.mfd2.mfd,Console.general.mfd,Console.explor
ation_radar.mfd,Console.exploration.mfd
CustomPanelsTitles=Test,General,Radar,Explore
CustomPanelsHotkeyPrev=F3
CustomPanelsHotkeyNext=F4
```

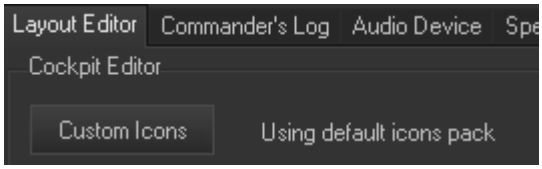
Customising the Custom Panel Menu Colours

Use the **Layout Editor** to change the following attributes on the **Custom** Role.

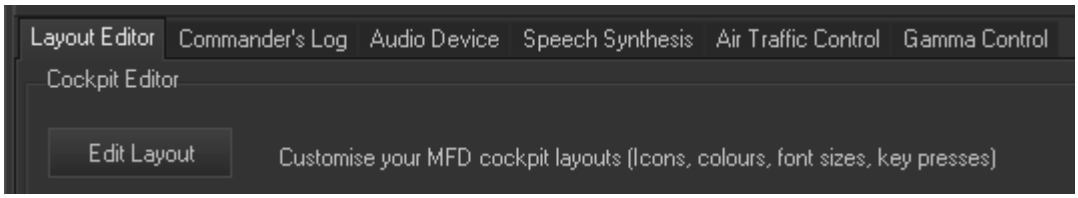
- **NoticeBoard** attribute is used to specify the **title** colour in the menu area.
- **Button 0** is used to specify the **next/prev** menu icon in the menu area (**Colour On** attribute is the highlight colour, **Colour Off** attribute the default colour).
- If **themes** are enabled, then the **title** colour is the **Secondary** colour and **next/prev** are from the **Accent** colour

Customising Layouts

From the main screen, the **Cockpit Editor** can be used to make customisations.



The **Custom Icons** editor will allow you to switch between the default set of icons and a single custom set of icons. The button editor will export the default icons so that you can use these as a template to create a custom set via an appropriate image tool (eg Photoshop, MS Paint).

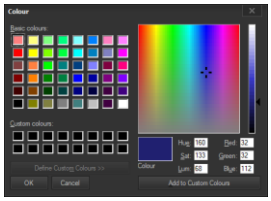


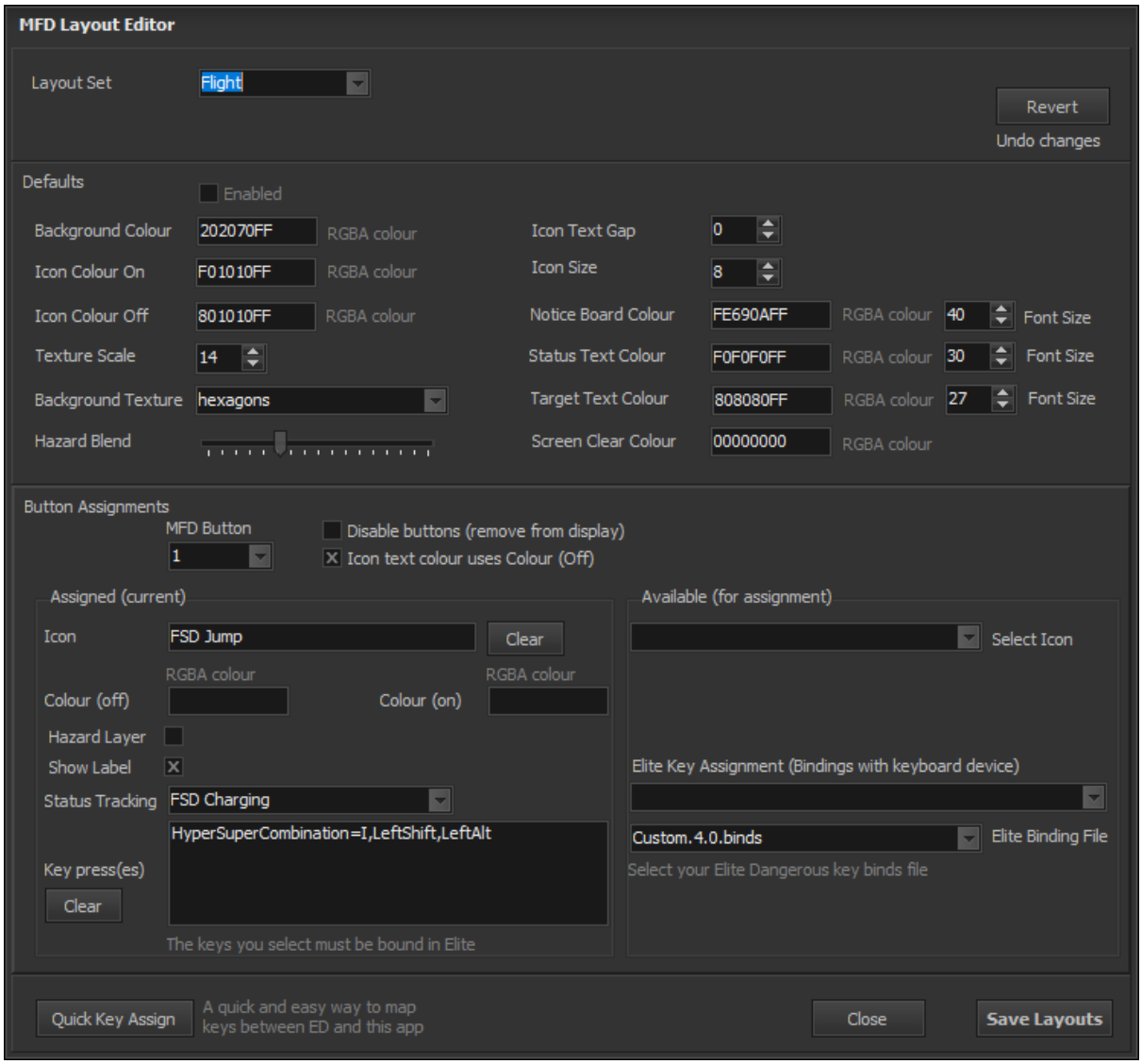
Using the built-in **Layout Editor** tool, the layouts, icons, colours and fonts can be customised, along with key press bindings. These values reside in XML files and are referred to as 'layouts' and they represent the various 'modes' that Elite has, such as flight, supercruise, analysis mode, planetary landing, etc.

Designing Layouts

Use the Layout Editor to make changes to 'layouts' which describe the colours and icons used on a particular console.

Note: Colours are in RGBA format (hexadecimal notation), clicking on the **RGBA colour** label displays a colour picker tool to assist number conversion.





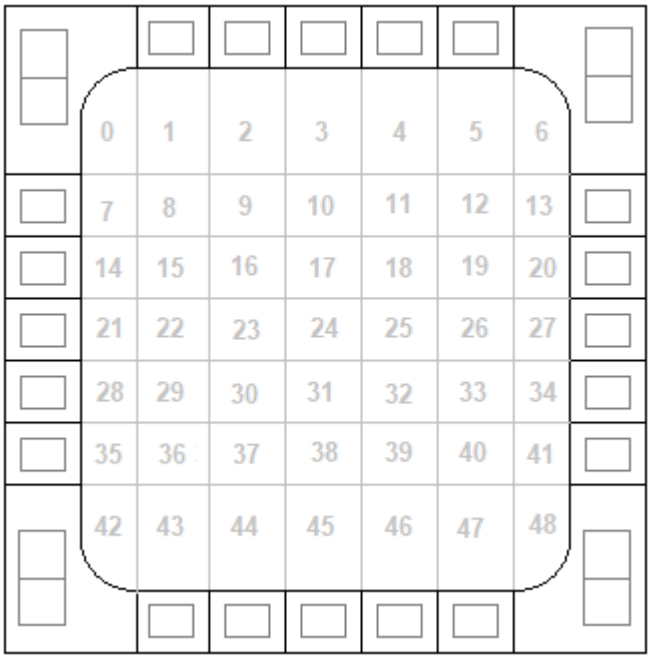
The basic usage of this editor is as follows:

1. Select the layout to edit from the drop-down list, eg Flight
2. Change the default values for the layout if any, eg **Background Colour**
3. Select a specific **Button Assignment** (there are typically 20 buttons corresponding to the 20 available positions around the edge of the display). This will assign **Icon** a value.
4. Assign an Elite action **Available for assignment** to the button, eg Cargo Scoop
5. Select an **Elite Key Assignment**

MFD Icon Arrangement

The basic design of most layout screens is that the icon positions follow the physical button positions on a (Thrustmaster Cougar) MFD joystick.

Pressing the icon on the display should be the equivalent of pressing one of the joystick buttons (and visa-versa).



Since there are 20 buttons placed along the outside of a 5x5 grid, the display is organised into 49 possible grid positions.

Layout Set

This is the layout options for a particular console.
Role option. One of the following roles can be assigned to the layout (Flight, Weapons, Supercruise, Planetary Landing, Exploration, SLF, SRV, Galaxy, System (Orrery), Shipyard or Custom).
Some roles are designed to be on MFD #0, others for MFD #1 and these will be dynamically allocated according to the status of Elite.

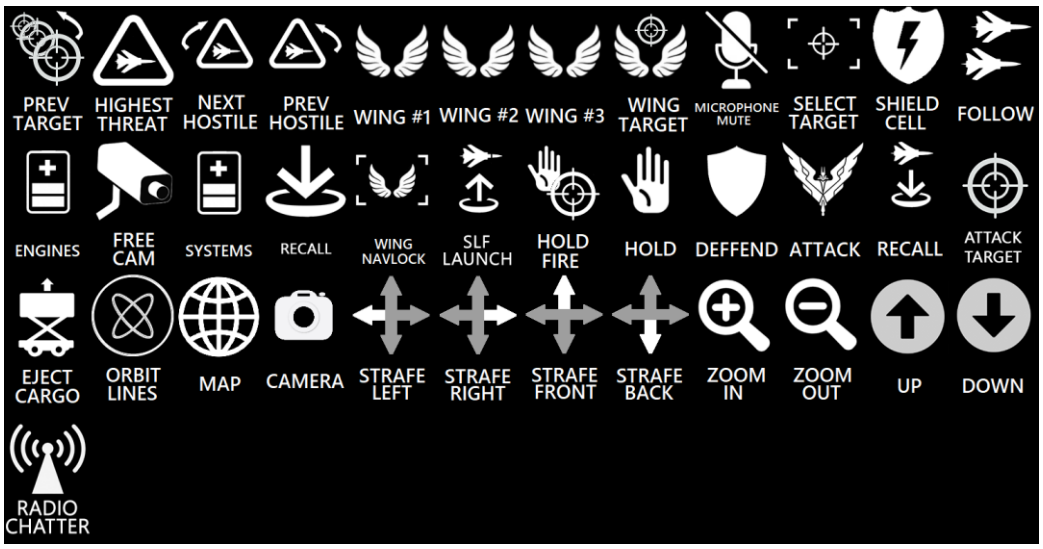
Defaults

The defaults section applies common values to the following sections:

- Colour and texture scaling of the background images
- Gap size between the icons and labels
- Text colour and font size for the notice board display
- Text colour and font size used by the status flags (cargo hatch, landing gear, mass lock, etc)
- Text colour and font size of the selected target text (weapons console)

Icon Palette

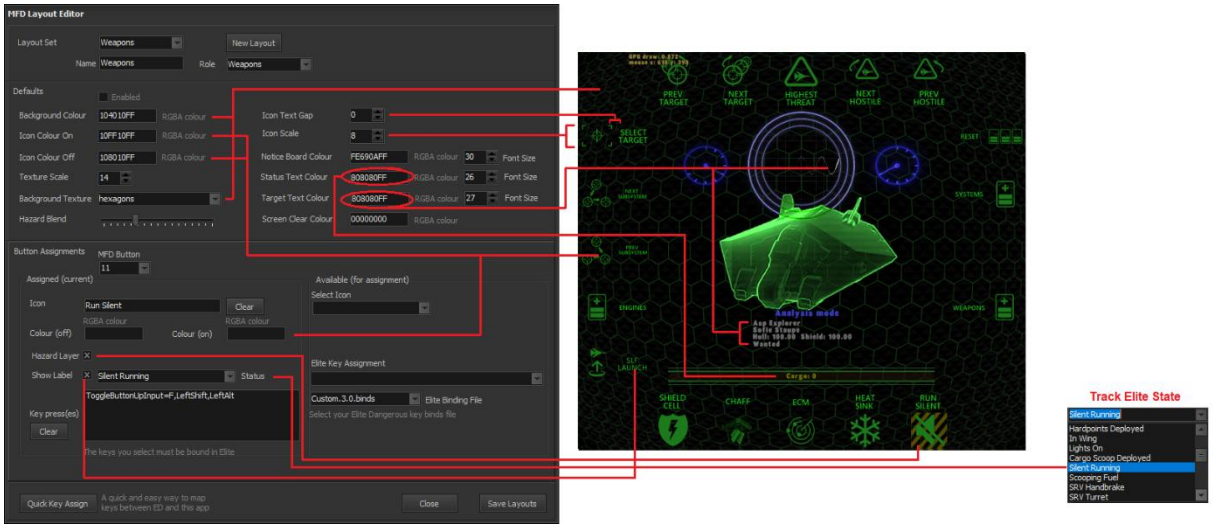
The available default icons to select from is shown below.



Customising Layouts

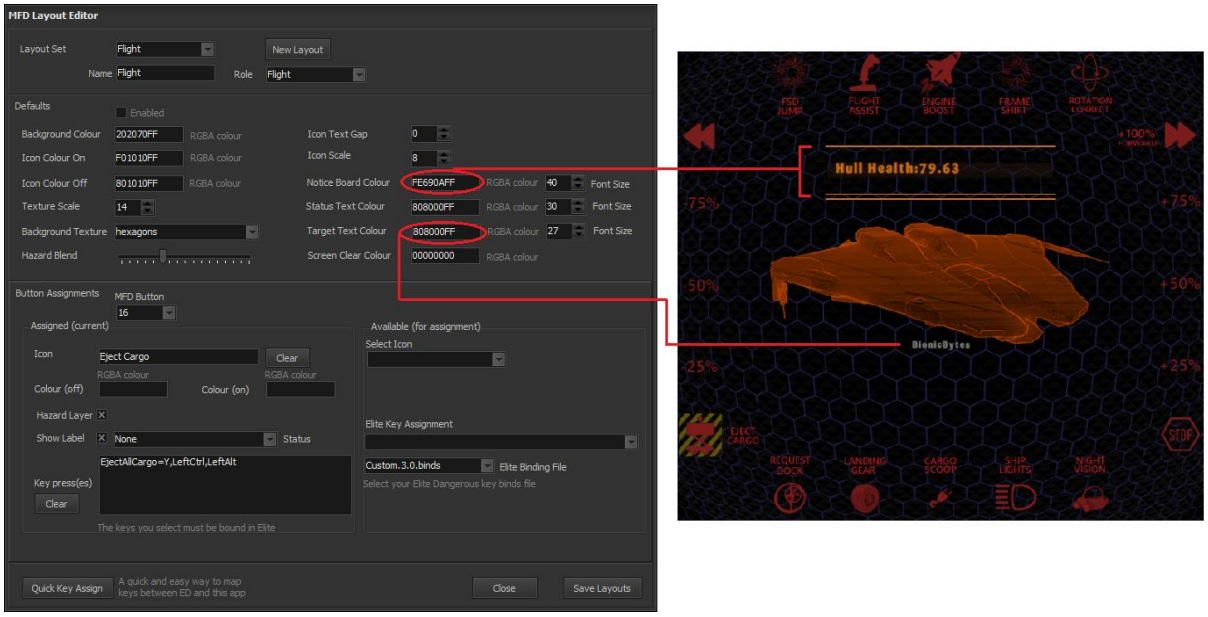
The following images show the relationship between the layout editor (the **MFD_Layout.XML** file) and the elements on the displays.

Weapons Console

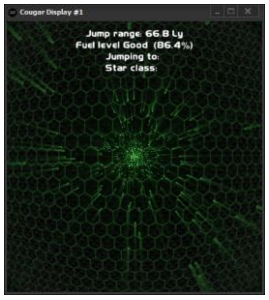


Flight Console

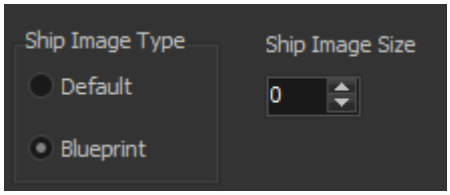
Follows the same basic principles of the Weapons console, plus these specific changes



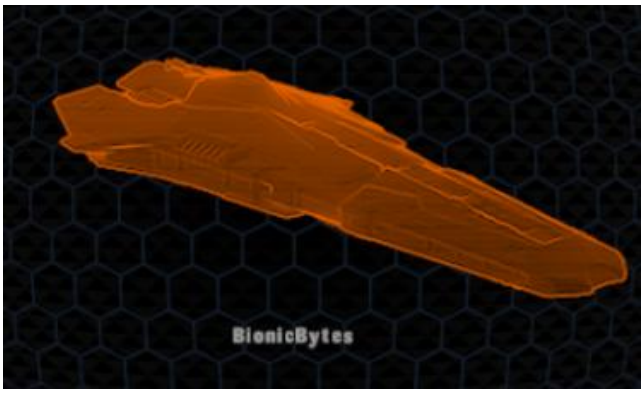
Status Text Colour/Font size – This is used during FSD Jump (displays star class/type).



The Flight/Supercruise Console has an extra set of controls which are used to control the ship image and size.



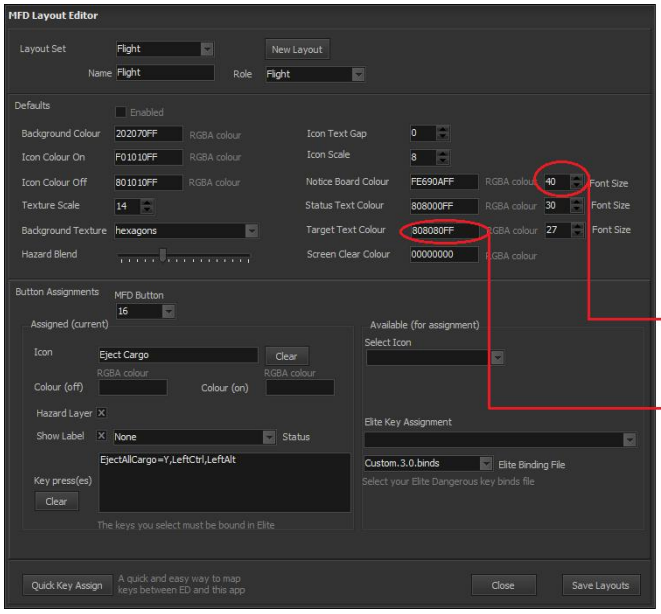
Use the **Ship Image Type** to toggle the main ship display to/from blueprint.

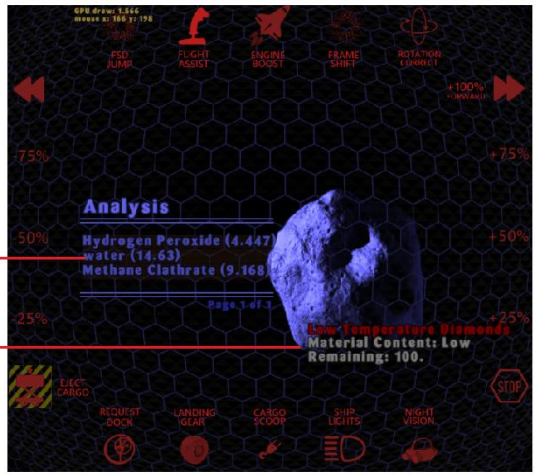


The size of the image can be adjusted using the **Ship Image Size** control (this is the same function as the F16 MFD Cougar rocker switches 21/22)

Flight Console (Mining / Docking)

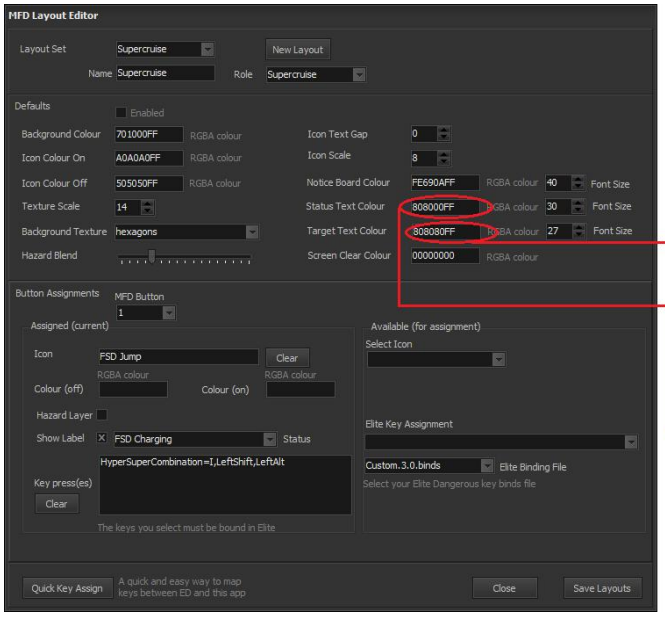
Follows the same basic principles of the Weapons console, plus these specific changes

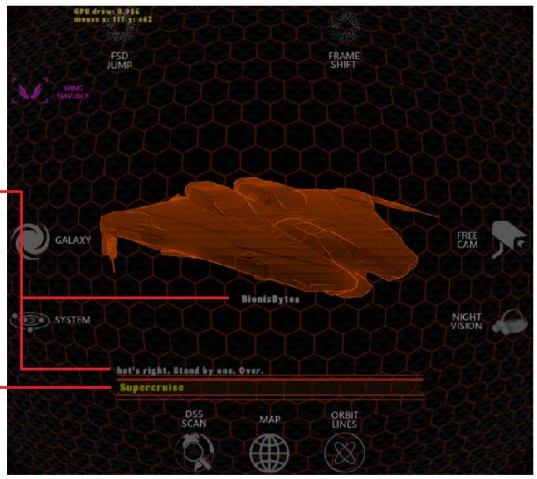




SuperCruise / Exploration (HUD Analysis mode)

Follows the same basic principles of the Weapons console, plus these specific changes



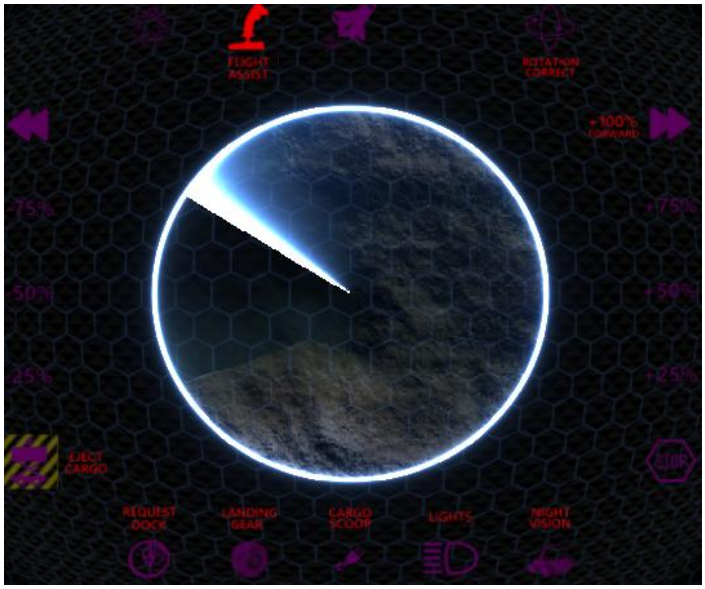


Super cruise also includes a special icon to toggle the radio chatter.

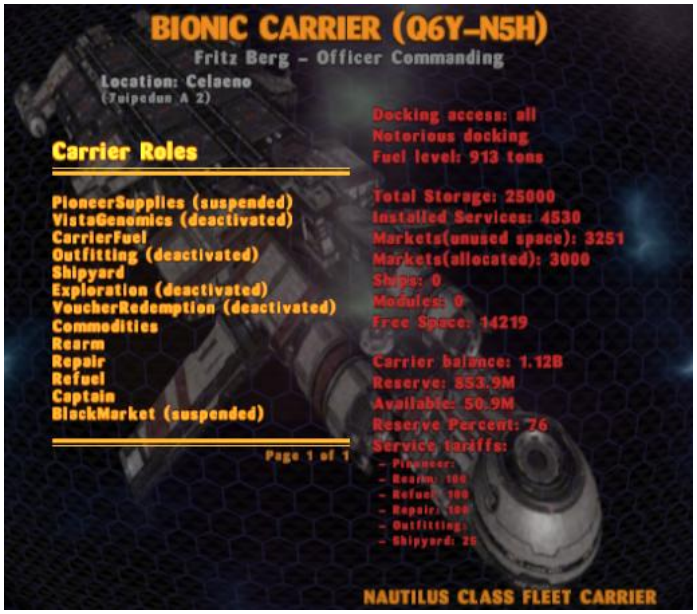
Radio Chatter Icon

The ATC Space radio chatter feature uses an icon to toggle the volume (on/off). The colour of the icon (as specified in the **Colour (on)**/**Colour (Off)**) is automatically determined by the on/off status of ATC Radio Chatter – there is no need to set a **Status** to track it as it's automatic.

Exploration:
When the ship is in analysis mode and in orbit of a planet/moon.



FleetCarrier
Follows the same basic principles of the Weapons console, plus these specific changes:



Notice Board

- Colour and Size of the font for the Services information area

Target Text

- Colour and Font Size – Location (star system name)

Status Text

- Colour and Font Size – Carrier statistics (finances, modules)

Carrier Type

- Select background image of fleet carrier

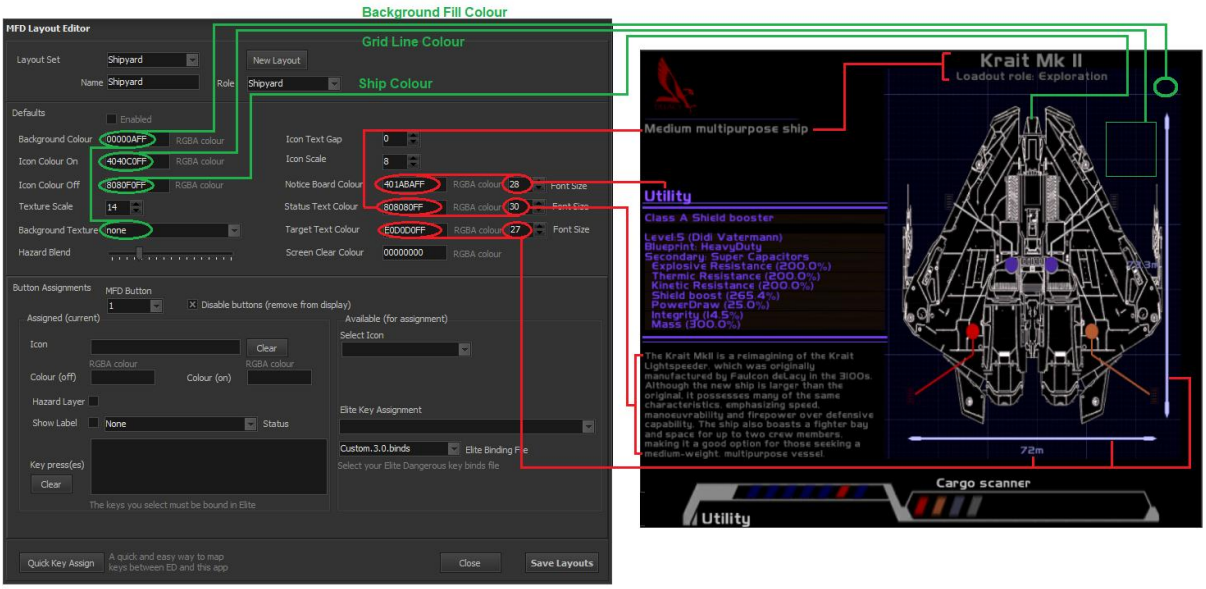
Text Colour/Carrier Text

- Sets the text and colour of the carrier description text

Shipyard

The Shipyard role is a live display of your current ship and allows you to select any of the modules currently installed, see their position within the Ship and inspect their current level of engineering. Use the mouse to select individual modules.

This role is available once your ship is docked at a station with a shipyard. The customisation available in the UI is mapped to the XML file as described by the image below. Note - The UI label names don't really match the target graphic as the UI was designed for the Weapon/Flight roles.

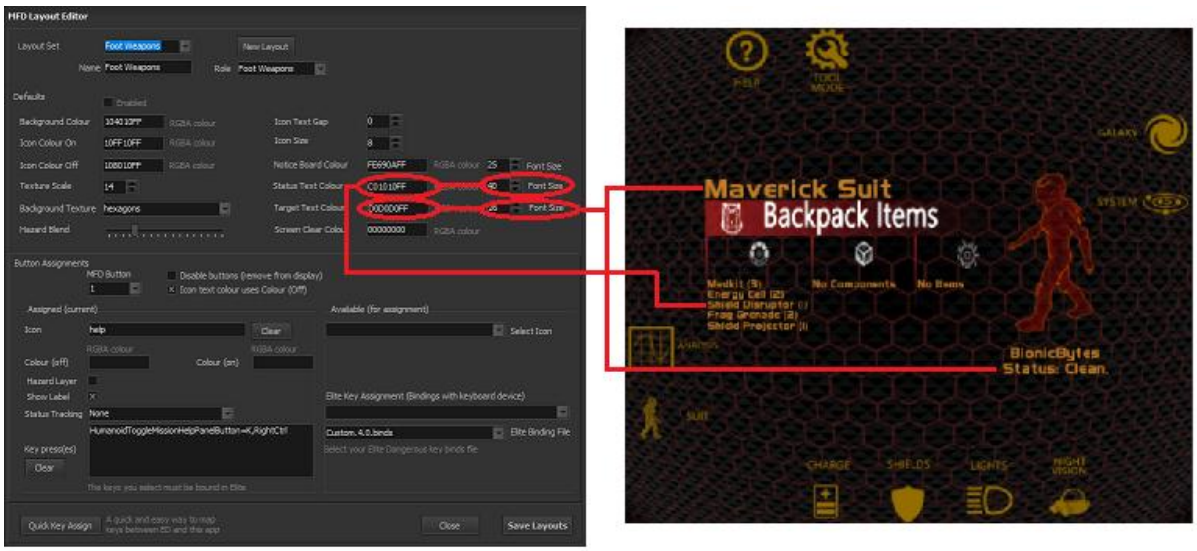


The Background under the grid is set via the **Background Colour** attribute, along with the choice of **background texture** (select from None, Swirl or Nebula).
The background grid colour is set via the **Icon Colour On** attribute.
The module description text is set via the **Notice Board Colour** and **Font Size** attributes.
The Ship type, loadout role and descriptive texts font colour and size is set via **Status Text Colour**.
The Ship colour is set via **Icon Colour Off** attribute.
The grid Axis arrows colour and Axis font are set via the **Target Text Colour** and **Font** attributes.

Custom

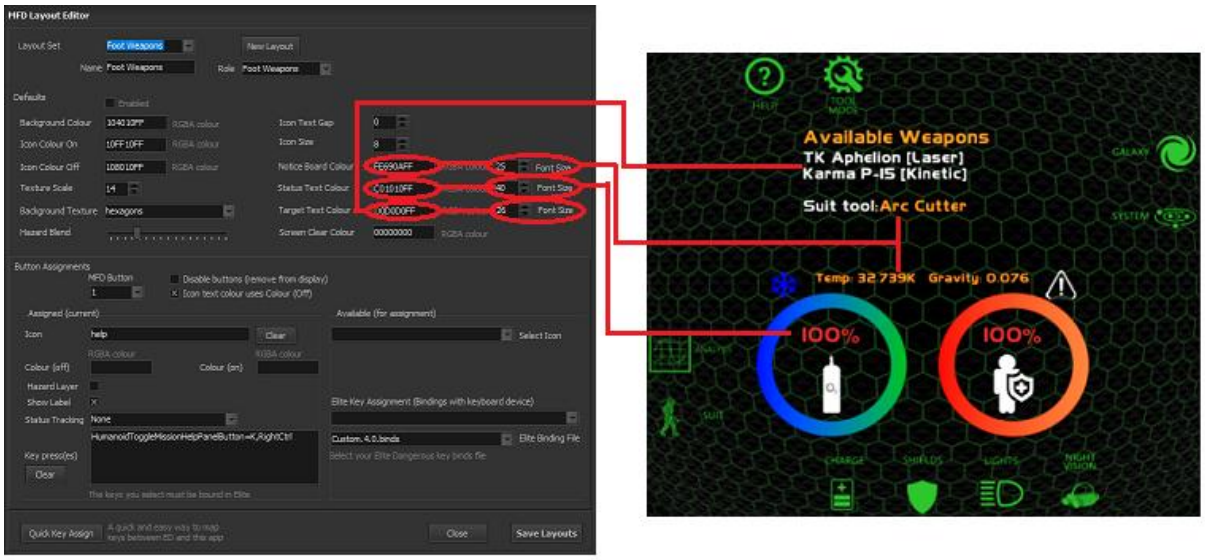
The Custom role is for use with the additional custom panel displays.
Background texture (select from None, Swirl, Nebula or hexagons).
The **Clear Colour** attribute is used to specify the background colour when the background texture is none or hexagons.
Iconcolour Off attribute is used to specify the background for both Materials & Status flags.
Iconcolour On attribute is used to specify the alternate text colour (Rank panel).
Targettext attribute is used to specify the destination in the Nav Route.
Statustext attribute is used to specify the CMDR ship colour.
NoticeBoard attribute is used to specify the Title in the menu area.
Button 0 is used to specify the next/prev menu icon in the menu area (**Colour On** attribute is the highlight colour, **Colour Off** attribute the default colour).

On Foot



Statustext attribute is used to specify font colour and size of backpack items (the font will auto-size to fit the space available).
Targettext attribute is used to specify font colour and size of the suit name and CMDR status.

On Foot Weapons



Noticeboard attribute is used to specify font colour of the “Available Weapons”, Suit tool name and environment temperature/Gravity values.
Noticeboard fontsize attribute is used to specify size of environment temperature/Gravity values.
Targettext attribute is used to specify font colour and size of the weapon modules.
Targettext fontsize attribute is used to specify size of weapon modules and “Available Weapons” title
Statustext attribute is used to specify font colour and size of the oxygen and health values.

The **MFDCougar.ini** file can be edited to switch the temperature scale from Kelvin to Celsius or Fahrenheit

```
[Odyssey]
Temperature=F
```

Key Assignments

Quick Key Assign button – Elite has hundreds of possible key bindings. With one click, Elite can be configured to recognise the Cougars and match the layouts specified in the editor.

Fine Tuning Key Press Responses

If your finding Elite is mixing up key presses or not able to keep up with the number of keys being sent, then increase (or decrease) the delay between each key (in milliseconds) with a value in the **MFDCougar.INI** file.

```
[keyboard]
keydelay=25
```

Touch Screen Tablet/Monitor Support

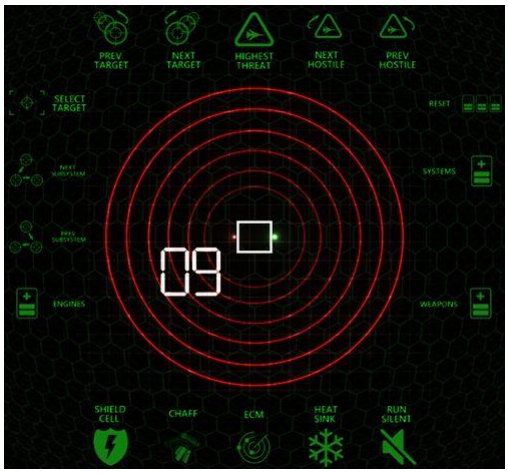
For a better touch screen experience, it may be desirable to keep the keyboard focus on the Elite window after clicking an MFD button icon. This is a personal preference depending upon your hardware configuration. To support this option, the **MFDCougar.INI** file can be modified as follows:

```
[keyboard]
;set the below to 1 to set the active window to Elite after pressing a key
(good for touch screens)
refocusEliteClient=0
```

Landing Pad Assistant

In the standard view, the green ‘blob’ on the circle display indicates the pad location. The outer rings represent the back of the station, whilst the inner rings represent pads next to the entrance. Align your ship so that the station green light is on your ship’s right hand side as you enter the station.

The landing pad assistant shows for Orbis and Ocellus Starports, Coriolis and Asteroid bases.

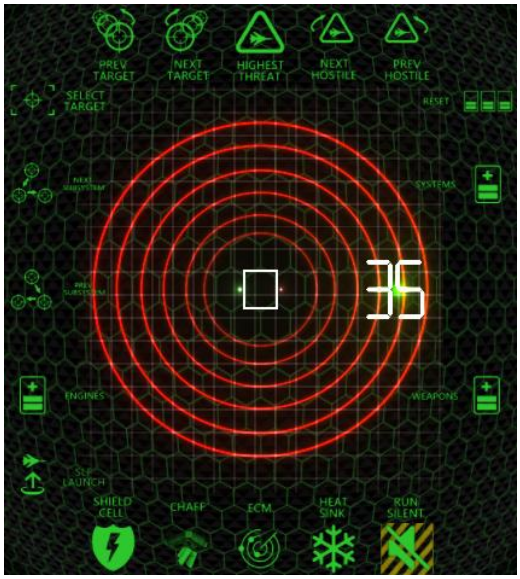


Customisation

The landing pad assistant can be customised to use an alternate view, so that the outer rings represent the entrance of the station, and the inner rings the back of the station.

Add the following into **MFDCougar.INI** file to switch view

```
[Landingpad]
ReverseView=1
```



Weapons Console

The Ship Target display reverts back to full screen size after several seconds (default 18 seconds).

This can be changed to any value (in seconds) in **MFDCougar.INI**

```
[ShipTarget]
Expire=18
```

Status Bar

The status bar (shown on the Flight & Supercruise Consoles) shows the current Elite status flags for game events.
However, due to limited space, only a maximum of 3 or 4 status flag descriptions can be shown at any one time.
By changing the following setting in **`MFDCougar.ini`**, the status flags will be abbreviated and the Status Bar area can now display up to seven status flag abbreviations.


```
[StatusBar]
Abbreviations=1
```

Status Description	Abbreviation
Supercruise	SC
Mass Locked	ML
FSD Charging	C+
FSD Cooldown	CD
Heat Damage	HD
Low Fuel	LF
Scooping Fuel	FS
Silent Running	SR
Cargo Hatch	CH
Landing Gear	LG
Hardpoints	HP

Low Fuel Warning

When the fuel in your main ship drops below a threshold (25% of maximum capacity), a low fuel warning will be displayed during FSD jumps.

Add the following into **MFDCougar.INI** file to customise the default values:

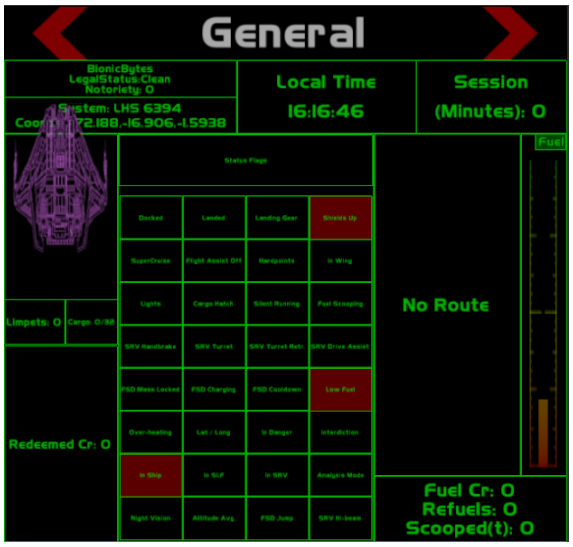
```
[LowFuelWarning]
LowPercentage=0.30
CriticalPercentage=0.1
```

Audio Announcement

When the fuel level drops below the `LowPercentage` value then a text to speech announcement is made, and a further one when the fuel level drops below `CriticalPercentage`.

Custom Panels

Fuel Gauges displayed on custom panels will flash when the fuel level drops below `LowPercentage`.



The rate of flash is determined by the value (in milliseconds)

```
[LowFuelWarning]
GaugeFlashRate=300
```

Set the value to zero to prevent flashing.

MFD Joystick Support

If you possess Thrustmaster Cougar, CubeSim or WingWing MFD joysticks they can be attached to your system and should be positioned in sequence over their respective Cougar Display and monitor. It is assumed that Thrustmaster MFD #0 will be positioned on a monitor with Cougar Display #0 window, and MFD #1 positioned on a monitor with Cougar Display #1 window, etc.

MFD devices can be mounted upside down if necessary (usb wire vertically upwards), but this will require the following change so that the button presses are transposed by Cougar Display. Note: Vertical button translation is not supported for WingWing MFDs.

Locate the **`MFD.INI`** file, and change the [MFD#] section to include **VFlipCougar**=1.
Each MFD Cougar can be mounted upside down, as required.

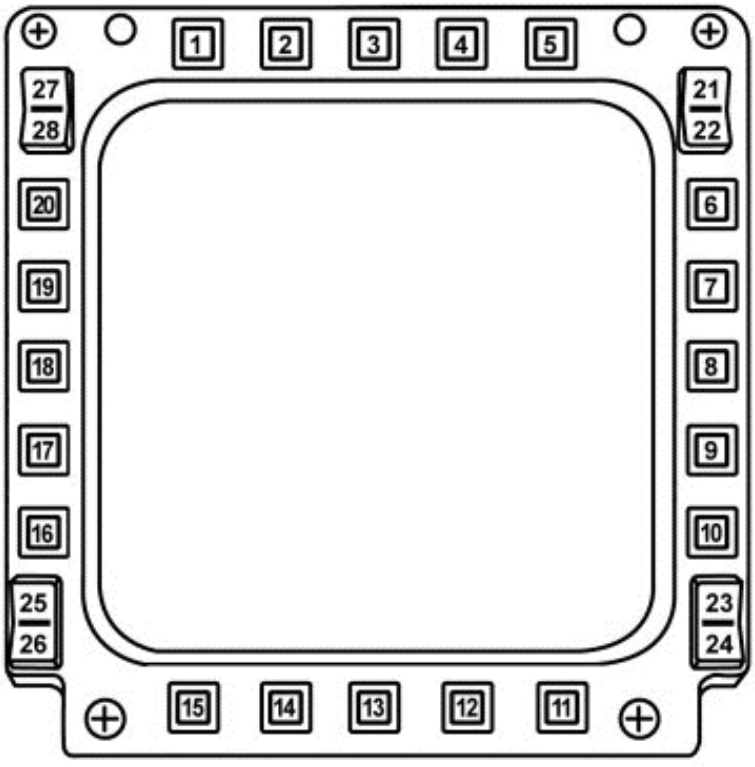
For example to mount MFD #1 upside down:

```
[MFD0]
VFlipCougar=0
[MFD1]
VFlipCougar=1
```

Note:
Using the **Layout Role Editor**, it is possible to assign specific MFD joysticks to consoles.

Thrustmaster Cougar MFD Button Assignments

The icon arrangement of the screens in MFD Cougar Display are based around the button assignments of a Thrustmaster Cougar MFD joystick, and are as follows:



These buttons are mapped to icons/elite actions on the Cougar Display screens via the **MFD_Layout.xml** file.

```
<buttons sizescale="8" enabled="1" IconTextColourOff="1">
    <button bid="0" iconname="FSD Jump" icon="32" keypress="HyperSuperCombination=I,LeftShift,LeftAlt" s
        <colour on="" off=""/>
    </button>
    <button bid="1" iconname="Flight Assist" icon="2" keypress="ToggleFlightAssist=C,LeftShift,LeftAlt"
        <colour on="" off=""/>
    </button>
```

The **MFD Layout Editor** allows you to remap the icons and Elite action associated with the button press.

CubeSim MFD Support

CubeSim make devices whereby an MFD joystick is mounted to an 8” screen – it’s assumed they follow the same button assignment and are supported in the same way as Thrustmaster devices.

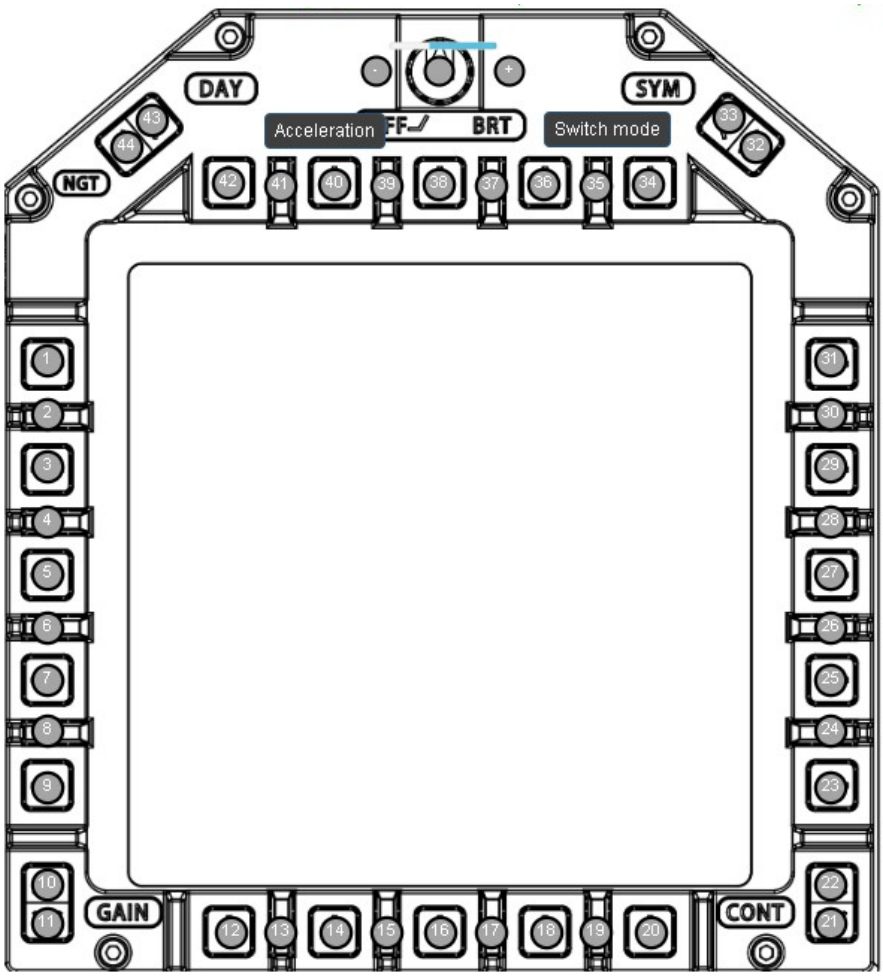
WingWing MFD Support

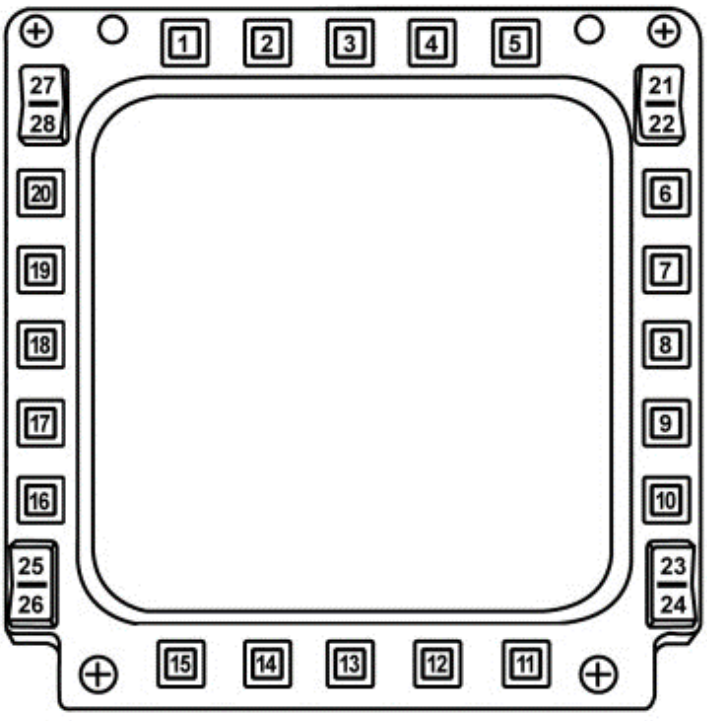
An alternative device with more buttons and flexibility has become available in the market.

This device has 50 buttons, with 4 rocker switches.

Support for these devices limits the buttons to the equivalent Thrustmaster MFD button – ie the thin (vertical) buttons between the square buttons are not used.

The buttons on the device are shown (top), but are mapped to 1-28 (bottom)





Custom Icon Packs

The default set of icons shipped with the application can be customised and extra icons added so that they can be displayed on the main two MFD console screens. Customising existing icons is a fairly simple process where as adding additional icons will involve some extra steps (such as editing XML files by hand).

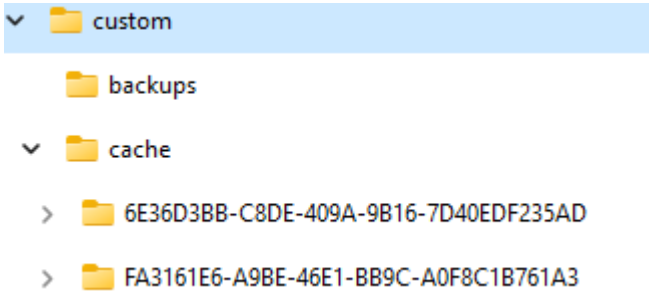
There are two icon set options:

1. Use the default icon pack
2. Use a custom icon pack

There can only be one custom icon pack active in the system, although multiple sets can be installed. A copy of the master file **MFD_Iconset_custom.xml** will be present in the application **root** folder when custom icon packs are active.

Advanced Information

When using published content (icon sets/MFDs) you will swap between downloaded content and your own content; these are all cached in a set of folders **custom\cache**\xxx-yyy-zzz.



One of these cached folders will be your active custom icon set. To know which one, open **Iconset.ini** in the root folder. Upon starting ED:CD, the contents of the cache is copied back into the root folder (to ensure all files necessary for the custom icons are present) if Custom=1 is present in the **Iconset.ini** file.

Inside the folder **custom\cache** is the **contents.ini** file which holds a list of the available icon sets (and also the downloaded content such as custom MFDs and icon sets).

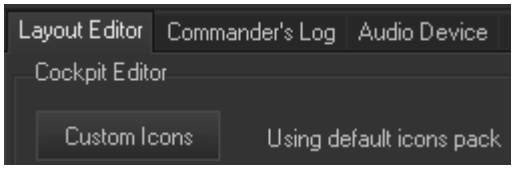
Note: If you manually edit **MFD_Iconset_custom.xml** outside of the **Custom Icons** editor, always change the file in the **custom\cache** folder rather than the application root.

More a detailed explanation of how to manage multiple custom icon sets, see the section **Technical details of publishing / installing** below, specifically the subsection **Installation of Published Icon sets**.

Creating a Custom Icon set

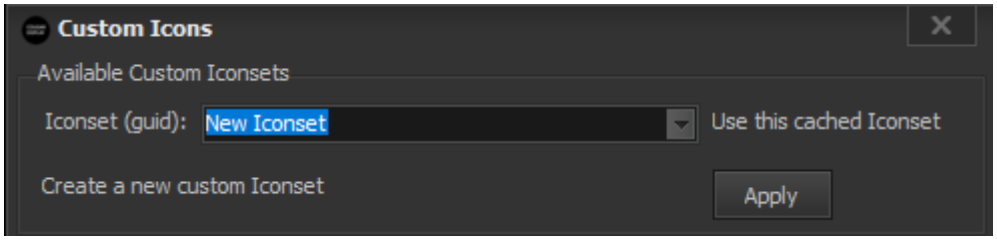
Click the **Custom Icons** button to open the Custom Icons editor.

Note: Once the application is running (rendering to the OpenGL windows) switching the active icon set may require an application restart (if the icons appear as a solid block of colour).



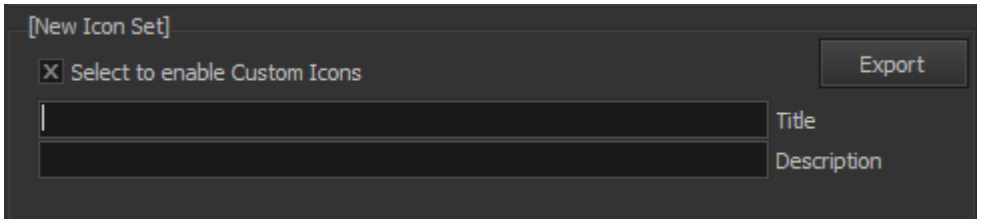
Using the **Custom Icons** editor, a clone of the default **MFD_Iconset.xml** file is automatically created and stored in the **application root** folder (**MFD_Iconset_Custom.xml)**. Note that the button **Custom Icons** will not overwrite an existing custom xml file should one exist.

In the **Custom Icons** editor, use the drop-down list to select a previous custom icon set, or create a new custom icon set. Click the **Apply** button to confirm your selection – this will copy **MFD_Iconset_Custom.xml** from the **custom\cache** folder or create a new one from the application's default **MFD_Iconset.xml** file**.**



A new section of the UI will appear showing the option to enable or disable this custom icon set.

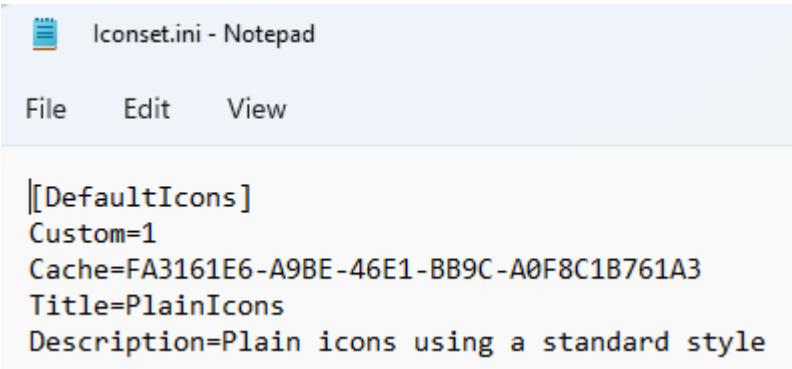
As an optional step, if you are unsure of the contents of the **MFD_Iconset_Custom.xml**, click the **Export** button to force a new export (overwriting the **MFD_Iconset_Custom.xml** and default jpg file(s), eg **Iconset6_custom.jpg**).



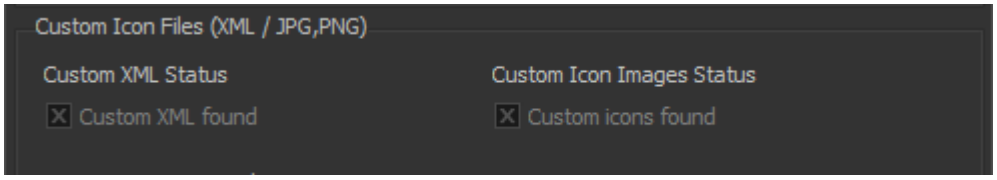
Once the custom xml and custom icon image files are present, the application will allow the use of custom icons, so you will need to use the tick box **Select to enable Custom Icons** to toggle between the default and custom icons. To disable custom icons uncheck this box.

The **Title** and **Description** fields are used to keep track of your custom icon set and when publishing with the ED:CD community. You must ensure they have text values before closing the dialog.

When ED:CD closes, a file **Iconset.ini** is created/updated with the above options.



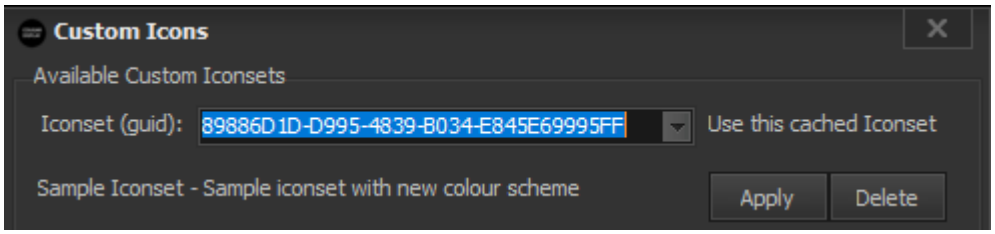
You will not be able to set the checkbox **Select to enable Custom Icons** if either **Custom XML Status** or **Custom Icons Status** are not checked. These check boxes are visual indicators to show whether the required files for custom icons are present in your system or not, and will be disabled (greyed and checked) when the files have been located. These checkboxes should be checked automatically for you because the system will copy the necessary files into the correct locations.



The cloning process will also export the default icon set graphics as a **JPG** file (as of version 1.17.0.0 this will be Iconset6) and will save the image with "_custom" appended. Thus the file **Iconset6_custom.jpg** will be created/overwritten in the **BMP\IconSets** application sub folder.

Managing Multiple Icon Sets

When the **custom\cache** folders contain multiple subfolders for icons etc, the **Iconset (guid)** will contain an entry for each Icon set, and default to the entry with a matching guid in **iconset.ini**.



Click **Apply** to switch to the cached icon set
Click **Delete** to delete the cached icon set from the **custom\cache** folder and remove **MFD_Iconset_Custom.xml** from the application root folder.

Customising Icons

Customising the default icons with a new design is then a matter of using MS Paint, Photoshop, etc to alter the images. If you keep the images in their current order (position) then the default XML id values will match the icon ordering and there is no further work for you to do. If you completely redesign the icons and their ordering, then you must manually edit the **<icons>** section of the XML file so that the icon images match the id values.

Icon Sizing

The default Icon has a dimension of 128 x 200 pixels which includes the image and text region. The actual image is 128 x 128 pixels.
With this arrangement, the icons are packed into "blocks" of 512 x 200 pixels, thus there are 16 icons per block. The icon texture has a maximum (and also the default) height of 800 pixels and thus ensures there are 4 rows of 4 icons per block (16 icons).

Icon Ordering

The icons on the **Iconset_custom.jpg** file are ordered in "blocks" of 512 by 800 pixels. Multiple blocks are then treated as additional slices – thus the default icon set image with dimensions 3072 x 800 contains 6 blocks. The maximum height of the icon set JPG file is limited to 800 pixels (note you can of course make this any multiple of the "icon size" (200 pixels) but the <icon id= > values will all need to be manually recalculated and changed in the **MFD_Iconset_Custom.xml file**).

The arrangement of the icons in a block is shown below.

0	1	2	3
Label	Label	Label	Label
4	5	6	7
Label	Label	Label	Label
8	9	10	11
Label	Label	Label	Label
12	13	14	15
Label	Label	Label	Label

The image above shows that the first 4 icons (id values 0-3) are in the top row, with the next 4 icons (id values 4-7) in the first block, second row. Each block thus contains 16 icons.

When an Icon set contains multiple blocks (such as the default set) the subsequent blocks are arranged horizontally from the previous, thus block 2 and icons 16-32 are arranged as so:

0	1	2	3	16	17	18	19
Label	Label	Label	Label	Label	Label	Label	Label
4	5	6	7	20	21	22	23
Label	Label	Label	Label	Label	Label	Label	Label
8	9	10	11	24	25	26	27
Label	Label	Label	Label	Label	Label	Label	Label
12	13	14	15	28	29	30	31
Label	Label	Label	Label	Label	Label	Label	Label

Special Icons

There are some special cases which need to be configured in your custom/default icon set. An example of this is the Radio Chatter Icon.

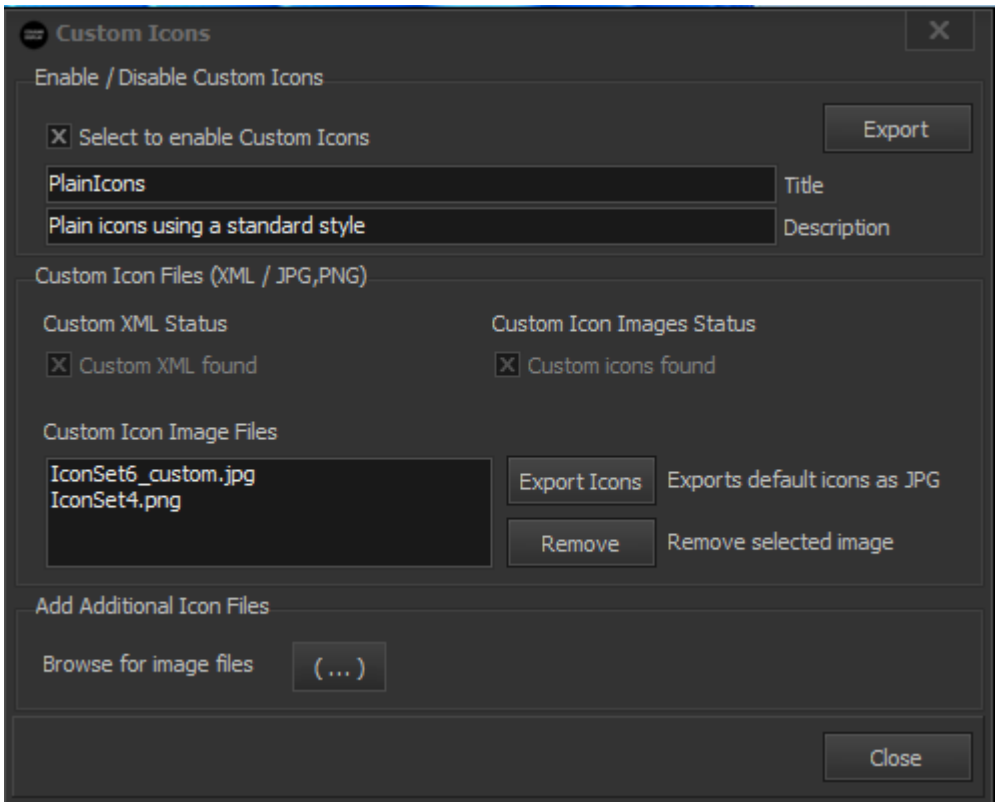
Radio Chatter Icon

The ATC Space radio chatter feature uses an icon to toggle the volume (on/off). To enable this feature the icon set must use assign an icon with the **alias attribute** of “radiochatter” and a **desc attribute** of “Radio Chatter”.


```
<icon id="60" alias="radiochatter" desc="Radio Chatter" Binding="" Key=""/>
```

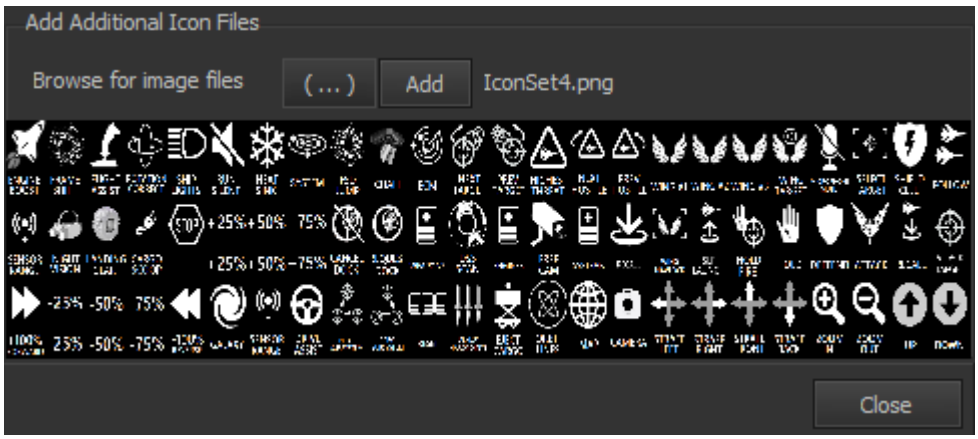
Adding Additional Custom Icons

When the default icon set is exported and the custom variant generated, the xml file (and icon graphic) have 11 unused icon 'slots' available. It is recommended that you do **not** use these as they may get used in future versions.



Use the **Browse for image files** button  to add a second (or subsequent) texture to the icon set (the image above shows an added image file Iconset4.png). Any jpg or png (no transparency) image with minimum width of 512 pixels (and 800 height) can be used as the icon/sprite images. The images should have a consistent icon width and height (same as the previous texture layer) which typically means 128 x 200 pixel icons.

In the **Add Additional Icon Files** section, click the **Add** button to merge the selected image into the icon pack. This will update the custom XML with an extra <texture> node entry and adds new <Icon> nodes for each new icon/sprite image.



To remove an image from the list of images, click the **Remove** button. This will remove the corresponding <texture> node from the XML and additionally any associated <icons> entries. If the list of **Custom Icon Images** is empty, the custom XML file will be considered invalid and the application will automatically switch over to the default icon pack.

Completing Additional Icons – Editing XML

The export/cloning process generates a "custom" version of the default icon set images (see customising icons) and inserts an extra **<texture>** node in the **MFD_Iconset_custom.xml** file for each added icon image file added via the **Custom Icons** editor. The **MFD_Iconset_custom.xml** file will need to be manually edited (eg Notepad ++) so that the **id, alias, desc, binding** and **key** attributes are defined.

You will now need to manually edit the **MFD_Iconset_custom.xml** file in Notepad ++ or other text editor. The **Custom Icon** editor will have inserted extra **<icon>** nodes with an **id** attribute for each jpg/png image added to the **Custom Icon Image Files** list box (16 **id** attributes per block). The number of **id**s added depends upon the dimensions of the image file added.

If the process is completed correctly, the **Layout Editor** should be able to pick up the new icons and add them to the **Available (for assignment)** list box as shown in the image above.

Editing MFD_Iconset_custom.xml

Using a text editor, ensure each **<icon>** entry has values for the attributes as below:

Id Attribute

The **id** attribute is used by the application to number each icon in a block (512x200 pixels). It is possible to add a new <icon> node and specify an existing **id** value, thus aliasing an existing icon as something new. This has been done in the default icon set for some of the Odyssey icons, eg Night Vision has been aliased as Suit Night Vision and both use id=”5”.
Given the default icon image file is 3072 x 800 pixels, and each icon size is 128 x 200 pixels, there are 96 icon images which are assigned id values between 0 – 95. Therefore, the first id value for a custom defined icon should start at 96.

Alias Attribute

The **alias** attribute is used to identify each icon.

Desc Attribute

The **desc** attribute is used by the application to link the **MFD_Iconset_custom.xml** to the **MFD_Layout.xml** entries (**iconname** attribute). It is therefore essential to give each icon a unique **desc** attribute value.

Binding Attribute

The value you type into the **binding** attribute will be one of the many action nodes in Elite’s **Custom.4.0.binds** file, eg < HyperSuperCombination>.

```
<HyperSuperCombination>
    <Primary Device="{NoDevice}" Key="" />
    <Secondary Device="Keyboard" Key="Key_I">
        <Modifier Device="Keyboard" Key="Key_LeftShift" />
        <Modifier Device="Keyboard" Key="Key_LeftAlt" />
    </Secondary>
</HyperSuperCombination>
```

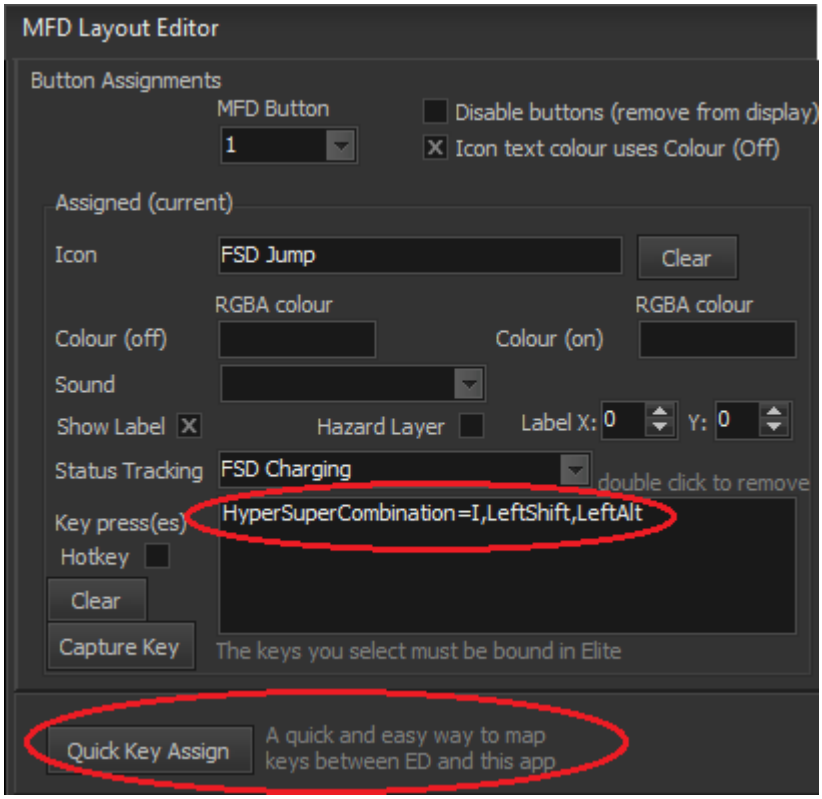
For Odyssey client, the typical location for the **Custom.4.0.binds** file is C:\Users\xxx\AppData\Local\Frontier Developments\Elite Dangerous\Options\Bindings

If you’re playing Horizons instead of Odyssey, then the binding file is typically **Custom.3.0.binds.**

Note: Elite creates a custom.x.binds file when you first modify input controls within the game**.**

Key Attribute

The value you type in the **key** attribute will be the default key press sequence you wish the application to pass onto Elite. This keypress sequence is picked up by the **Layout Editor** and presented as the key press associated to the icon being edited. This will either match the **Key** attribute in **Custom.4.0.binds** file (because you specifically copied the value from there) or you will assign your own key sequence and then use the **Layout Editor** to force an update on **Custom.4.0.binds** file via the **Quick Key Assign** button.



Avoid assigning more than two modifyer keys as most keyboard drivers can't handle too many simultaneous key strokes. Throughly test, or avoid the use of **RightAlt** as this is more than a single key scan code (LeftCtrl + RightAlt). Use some of the testing tools on the web site to debug http://cougardisplay.site/downloadfiles.html or Thrustmaster's Event Tester utility.

Dedicated Alpha Channel (Mask)

When loading the icon set image from file, if an *alpha* file exists then the image will be loaded into the alpha channel of the icon texture automatically. This is an advanced feature which allows the icon author to specify the luminance (per pixel) of each icon. Since JPG files are typically 24-bit and do not contain an alpha channel, this feature allows for custom 32-bit icons.

An alpha file is a jpg file which has "_alpha" appended to it. For example **IconSet6_custom_alpha.jpg**. The alpha file is not exported by this application but can easily be made by copying the xxxx_custom.jpg image file and saving it as xxxx_custom_alpha.jpg.

Alpha files can be:

- 24-bit JPG or 8-bit single-channel JPG
- 24-bit PNG (no transparency)

Note: The red channel is used as the alpha source in 24-bit image files.

Sharing Customisations with the Community

Once you have setup and configured your custom consoles (panels) and/or used custom icons in your cockpit layouts, you might like to share these with the wider ED:CD community.

This is an experimental feature and we'll see how it workout in due course as there's a fair bit of gathering of the right files to distribute to other CMDRs.

Currently, there are two types of content which can be shared: custom consoles or custom icon sets and these can either be *published* or *installed*.

Note: There can only be one custom Icon set installed at any one time, however, multiple custom consoles can be installed at any time.

Publishing

The act of publishing means making the custom files available to the wider community. When you publish content, the system gathers the required files, creates a zip file and uploads it to a private area on the Cougar Display web site.

The title assigned to the published content is used to distinguish between all the published items, so try and make these unique. Publishing will fail if content with the same title exists.

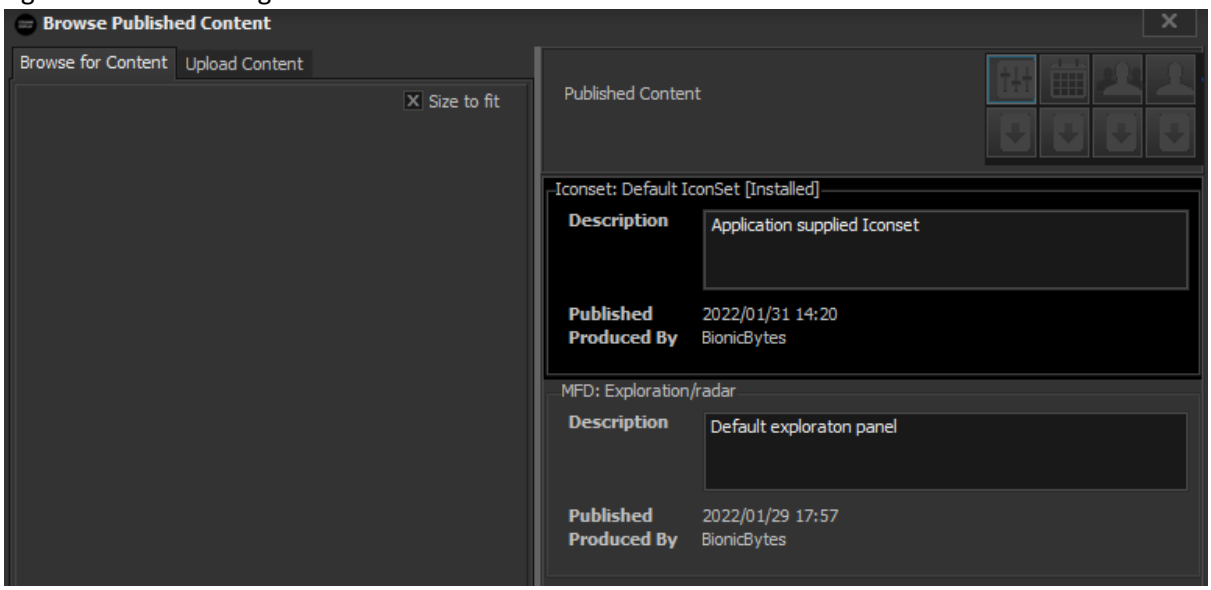
Installing

The act of installing means the selected (public) content is downloaded from Cougar Display website and the required files from the zip package are extracted to the correct locations (according to content type). Previous files (your existing custom content) may get overwritten (custom icon sets).

Note: Installing content may erase your own custom content! Either publish your custom content, or back it up first.

Browsing Published Content

When the **Browse Published Content** dialog is opened from the **Published Content** menu (on the main application display), a list of the available icon sets and custom consoles are displayed on the right side of the dialog.

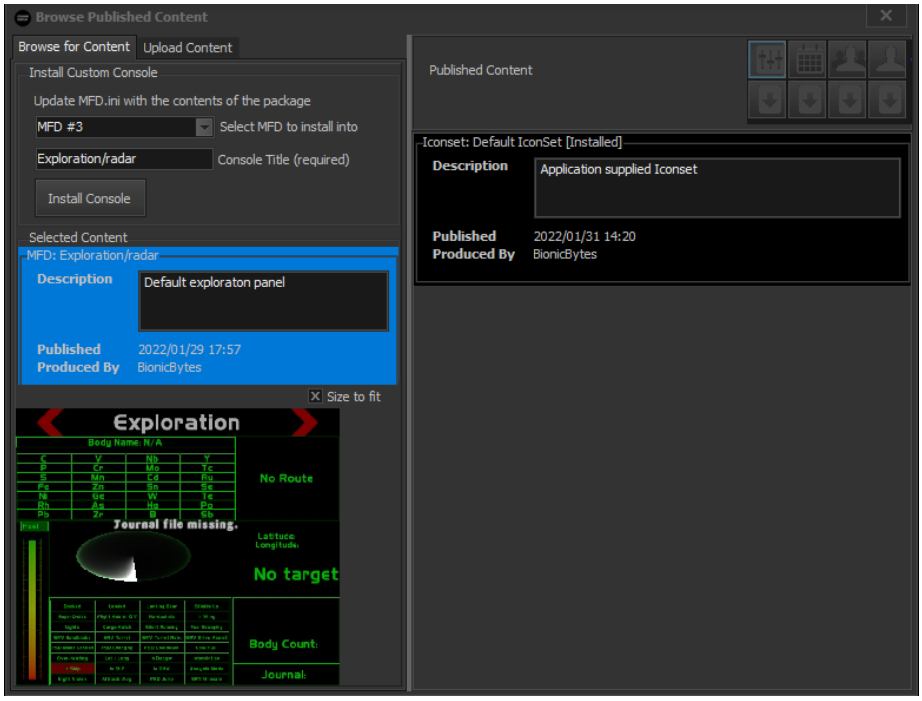


Previously installed content is indicated in two ways:

- The content title displays **[Installed]**
- The content panel background is drawn in black

Uninstalled content is indicated with a grey background.

The currently selected content is highlighted in light blue (unless it's been installed) and moves across to the left hand side panel so that it can be installed. Selecting another item from the list of available content, deselects the previous item and returns it to the main list of **published content**.



Content Sorting / Filtering

Published content is categorised according to content type (consoles/icon sets), publisher (CMDR name) and title.

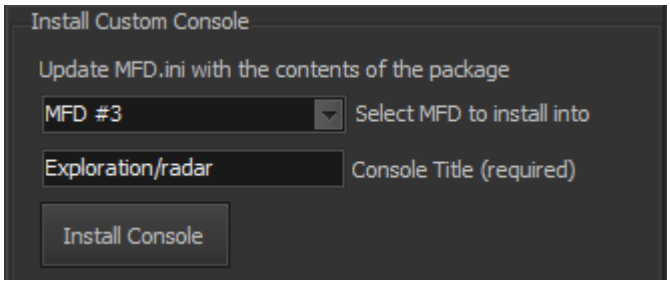
Using the buttons, the published content can be filtered and sorted by clicking on the 4 available buttons and the sort direction reversed by clicking on the direction arrow buttons below.

The default sorting is by content type. The other options are date, CMDR name and finally a filter to view only the content you have published yourself.

Installing Content

Once the content is on the left hand side, simply click the **Install** Icon set/Console **button** to download and install the content.

Custom Consoles have the additional configuration options as follows:



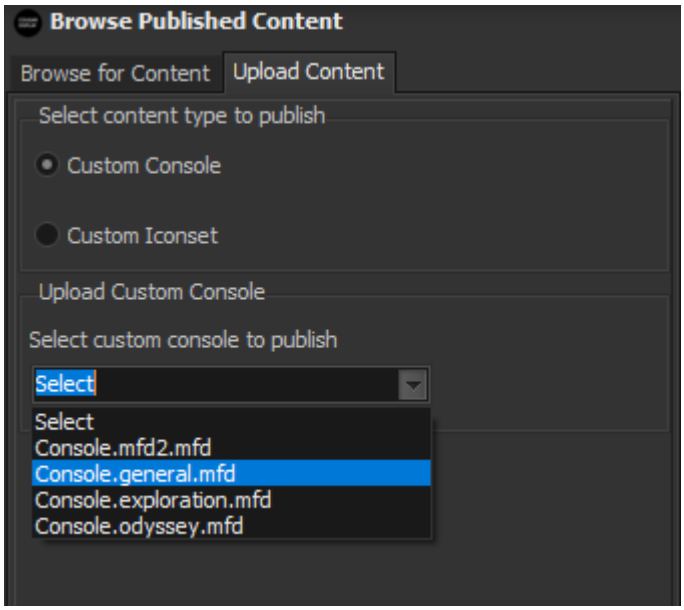
Select the MFD you wish the customisation to apply. This will add an entry into the MFD.ini in the exact same way as using the **Design Console** user interface.

If **Console Title (required)** is displayed, then the custom consoles on the MFD are all using titles, so you must also choose a title for this console.

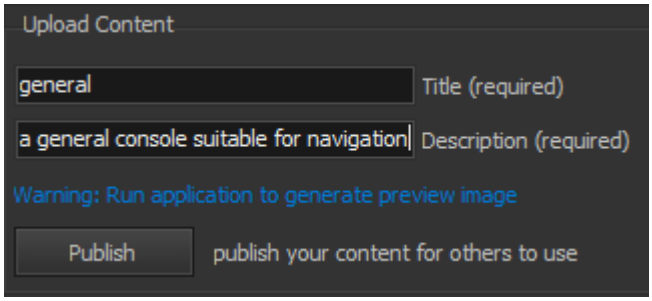
The selected content will now have **[Installed]** appended to the title.

Publishing Content

Switching to the **Upload Content** tab will allow you to publish your custom content.



For custom consoles, select which of available consoles to publish in the **Upload Custom Console** panel.



In the **Upload Content** panel, type a description and give a title which will identify this console from others you have published.
Click the **Publish** button to upload the content.

Note: An image of the content is generated automatically, as long as the application is started (the MFD OpenGL context windows are rendering images). A warning message is displayed otherwise.

For Icon Sets, there is no other information required other than choosing a unique title for the publication, and a description. The **MFD_Iconset_custom.xml** in the application root folder is then published.

Once published, the content is visible on the right hand side of the Published Content window.

Technical details of publishing / installing

This section gives some detail about what is happening when using the user interface to publish or install content. For most users, the details in this section may not be relevant and is for information only.

Custom Consoles

Custom console are the **.mfd** files in the **root** of the application folder; these files are used to display a group of panels to display information, eg Date/time, CMDR name, ship status, etc.

Custom consoles (from version 1.19.0.0) can also display custom backgrounds via the "blank" panel. To support this the console designer allows you to select any texture defined in the system – the textures are exposed as a list to the designer and is automatically extended by the system finding and loading a **Custom.TextureFX.txt** file on start-up. If you are publishing custom consoles which rely upon these extra images, you'll need to include the Custom.textureFX.txt file along with the console file. The publishing user interface will do this for you.

To publish/install custom consoles the following files are copied/overwritten:

Console.xxxx.mfd
FX\Scene Files\MFD**Custom.TextureFX.txt** (if the file exists)

Installation will then copy images from the zip to **bmp\custom** folder and add the console filename (Console.xxxx.mfd) to the **MFD.ini** file in the **CustomPanelsList** entry.

Note: It is recommended that you enable a colour theme for your console in the **Console Designer** – so that other CMDRs can view your console using the colours you intended to be used.

Custom Icon Sets

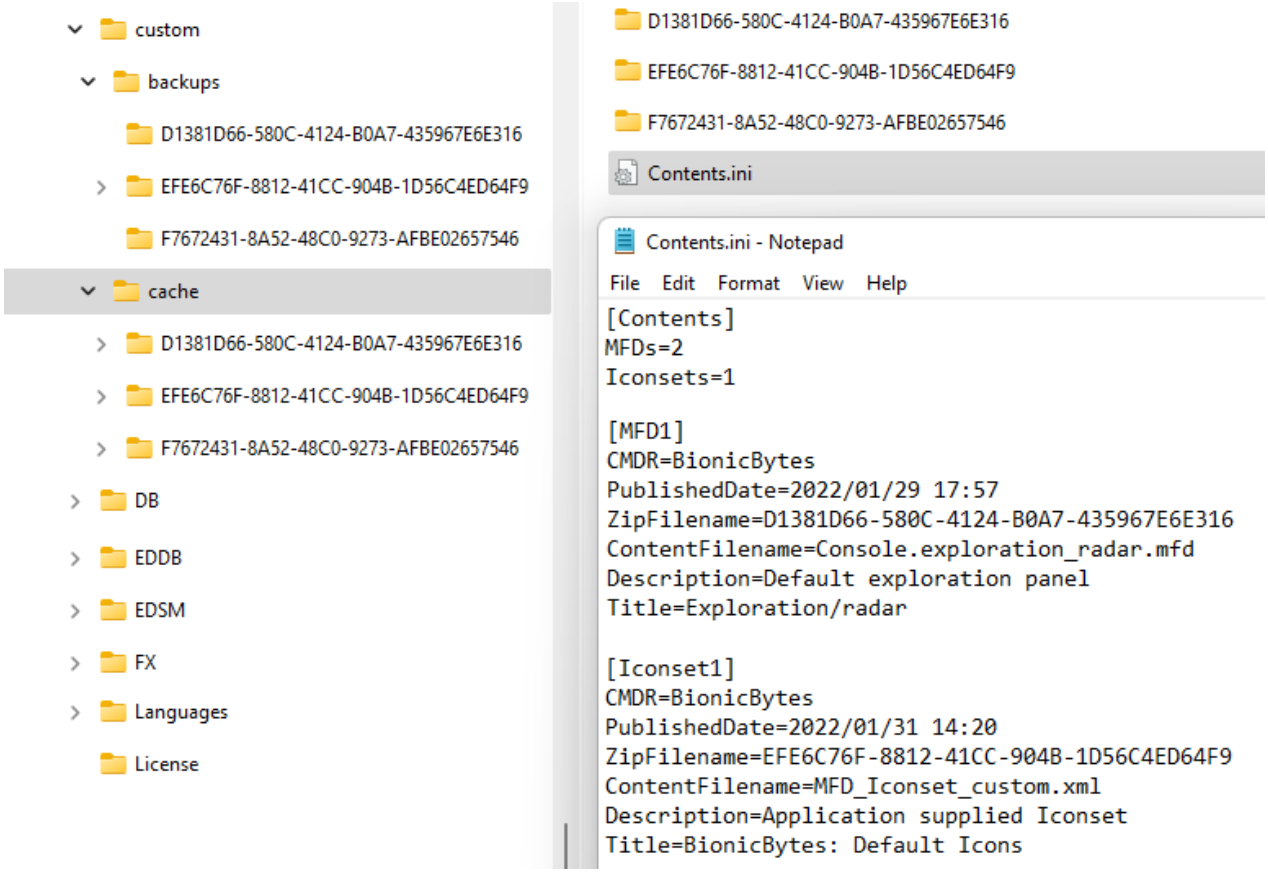
Custom Icon Sets consist of image(s) and a XML file and are used to extend the number of icons displayed in the MFD layout screens. These files are located in the following folders:

Bmp\IconSets folder
MFD_iconset_custom.xml application root folder

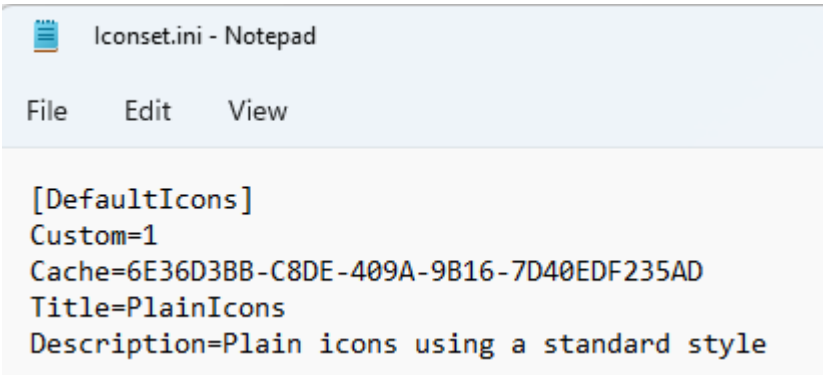
The XML's **<texture>** node describes the exact file name and attributes for the image files used in the icon set.

Installation of Published Icon sets

Creating your own custom icons or installing content will create a cached folder of the website's zip file in the **custom\cache** folder. This folder contains **Contents.ini** and a set of folders with GUID directory names corresponding to the entries in **Contents.ini**. This file and folders are used to allow the system to track which public icon set content has been installed in the current system. Before any content is installed, a backup is created in the **custom\backup** folder using the same GUID directory name as the newly installed content – this is an easy way to (manually) recover from the published content overwriting the currently installed custom content.

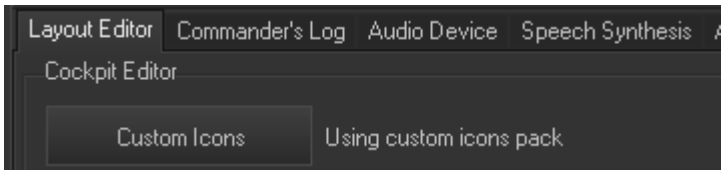


The system tracks the currently installed custom icon set via the **Iconset.ini** file in the **root** folder. This file specifies the GUID of the downloaded (or locally created) cached content.



Each time ED:CD starts up, the **Iconset.ini** file is parsed and if **Custom**=1 and **Cache**= has a valid GUID matching a folder in the **custom\cache**\xxxx folders, then the contents of the cached folder are

copied into the ED:CD folders. This means that whilst custom icons sets are enabled, the **MFD_Iconset_custom.xml** will be copied into the root folder (possibly overwriting any local edits to this file). If you have made edits to **MFD_Iconset_custom.xml** in the **root** folder, then copy the file into the cache folder or make the edits there instead.



The main application screen indicates that custom icons have been enabled and can be altered by clicking the **Custom Icons** button, and deselecting the **Select to enable Custom Icons** check box, or by editing **Iconset.ini** and setting **Custom**=0 and then starting ED:CD.

Key Bindings

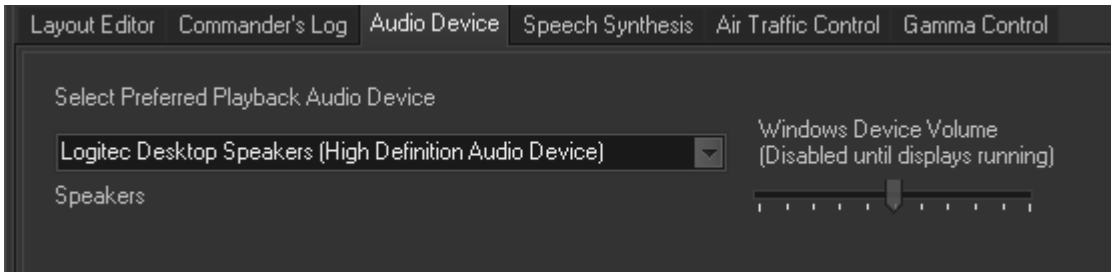
There is one important and easily overlooked aspect to installing someone else's custom Icon set files - Key Bindings. The file being installed is **MFD_Iconset_custom.xml** and this file describes all the icons used on the Layout screens (via the Cockpit Layout Editor). However, it also contains *default* key bindings which *could* get used when using the **Layout Editor**'s **Quick Key Assignment** button. To prevent key assignments from the published icon set appearing in your layout files (**MFD_Layout.xml**) the system will attempt to locate and swap out key bindings and use Elite's key binding file for the actions contained in the **MFD_Iconset_custom.xml** or **MFD_Iconset.xml** if the custom xml file doesn't exist. In this way your assigned key bindings are preserved.

Text to Speech Synthesis (TTS)

Brings Elite Dangerous to life by making NPCs 'talk'. Multiple NPC types (Police/Station Authority/Pirates) are given different (user defined) voices for automatic speech synthesis – assuming your Windows system has suitable voice packs available.

Select Audio Device

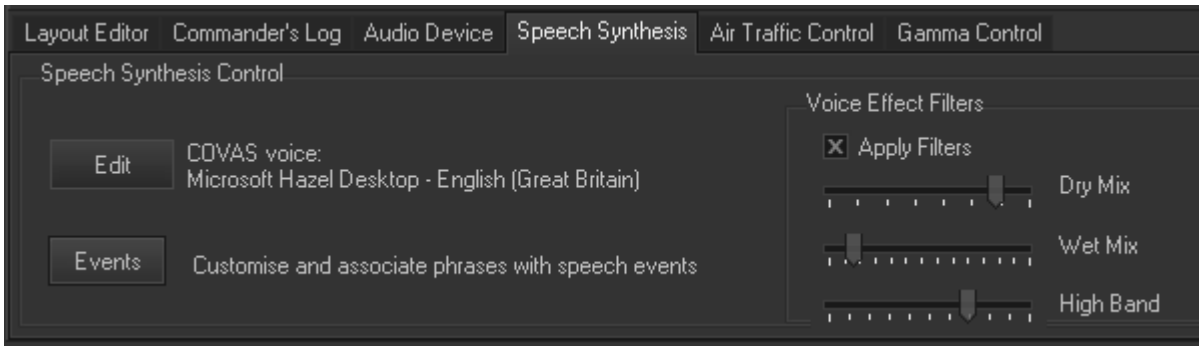
Text to speech synthesis can be routed to a specific audio device on the **Audio Device** tab.



For your convenience, a slider to adjust the master Windows sound volume has been included (however this is only available once the application has fully started).

Select Voices for TTS

To assign voices, locate the **Speech Synthesis** tab**,** and click the **Edit Voices** button.



You can assign a unique synthetic voice as an assistant to your Elite COVAS voice, as well as unique sets of synthetic voices to use used by system authorities / police, station controllers and pirates.

Voice Filtering

The filters used to distort the text to speech synthesis can be adjusted to give it a more authentic radio sound. **Voice Effect Filters** can be applied to all voices (except the COVAS assistant assigned voice).

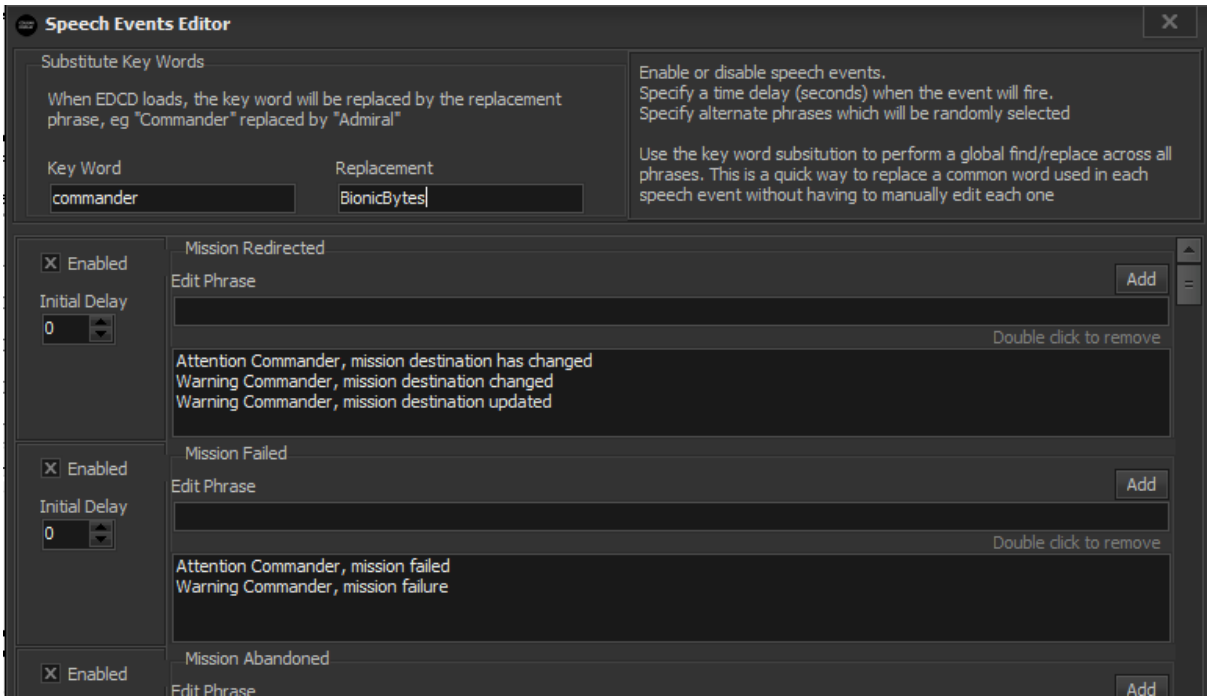
When the Voice Filter sliders are set as show above (default values) they give an acceptable level of distortion to most of the built-in Windows 10 voices.

Dry Mix: min 10; max 80; increments of 10; default 60
Wet Mix: min 0; max 80; increments of 10; default 5
High Band: min 500, max 1500, increments of 100; default 1200

When dry mix values are low, and wet mix values high, distortion is added to the voice – typical of long distance radio transmission. To make voices clearer, use the default values (high dry mix and low wet mix).

Speech Events Editor

The phrases used during Elite game play can be customised in any language your TTS voices support.



Each game event/Cougar Display event exposes the following properties:

1. Enabled – Unset to disable the speech event
2. Initial Delay – specify a delay (in seconds) when the event should trigger the event
3. Alternate Phrases – Add or edit sets of phrases to be spoken. The system selects a phrase at random from those defined. Some events support a **%s** parameter – this is provided by ED:CD at run time and is context sensitive to the specific event.

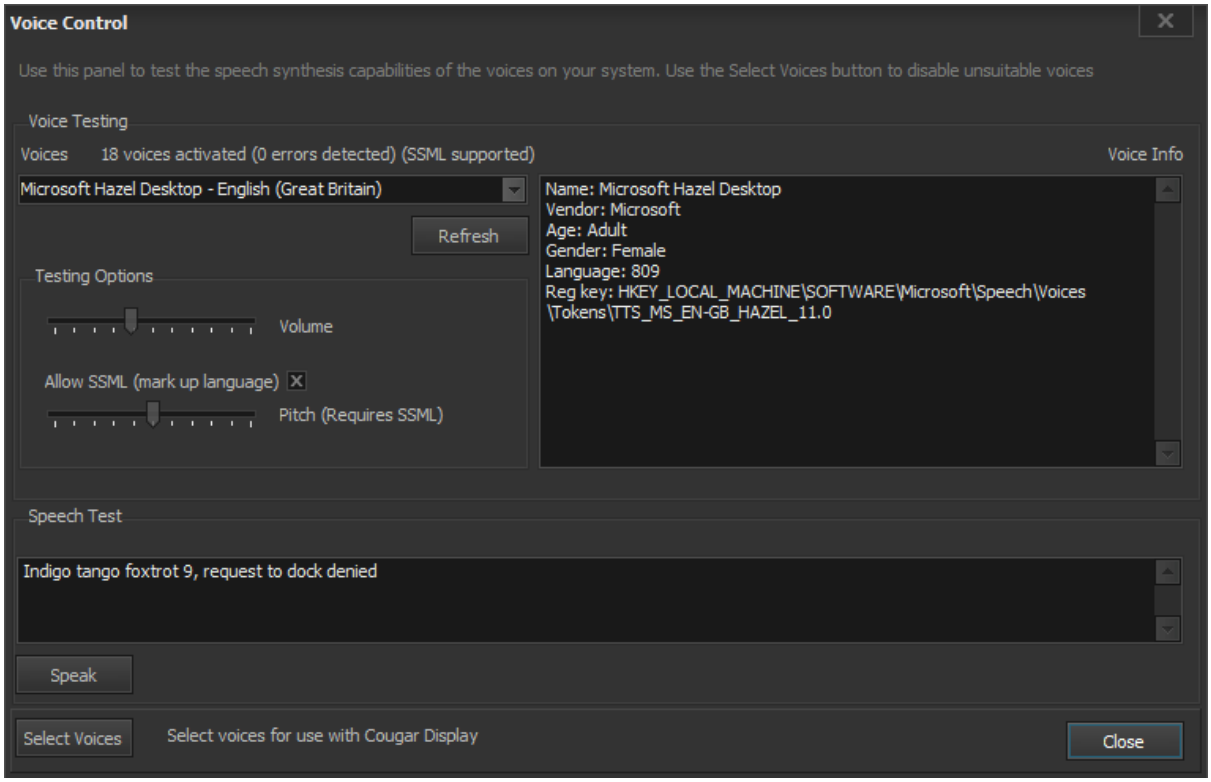
Additionally, there is a key word replacement feature available which is used to perform a kind of global find/replace across all phrases changing the key word for the replacement phrase. This is useful if you don't want your COVAS assistant to say "Commander" and instead say something else.

Events With available parameters

Event	Parameter example
ShipTargeted	Wanted target sighted, Enemy target sighted
NPC Attack	Hit man attack, Powerplay attack, Bounty hunter attack
NPC Attack Speech	Text spoken by the NPC, eg “Big haul like that…”
NPC Interdiction	Pirate Inderdiction, Bounty Hunter Interdiction
NPC Interdiction Speech	Text spoken by the NPC, eg “Surprised you made it this far…”
USSDrop Mission Target	USS drop type
USSDrop Mission Target Threat level	Threat level 1..9
Notoriety	Notoriety level
Commit Crime	Crime type, eg NPC murder
JetCone Damage	Module name
Docking Denied	Reason, eg Fighter launched, distance
Docking Granted	Pad number
Material Discovered	Material name
Material Collected	Material name
Promotion	Rank and level
Scan Unmapped	Planet class type, eg Ammonia world
npcPassengerLiner	Text spoken by the NPC
npcPolice	Text spoken by the NPC
npcMilitary	Text spoken by the NPC
npcRescue	Text spoken by the NPC
SupercruiseDestinationDrop	Threat level 1..9
SupercruiseDestinationDropHighThreat	Threat level 1..9. Used when the threat is greater than level 4

Voice Control

The Voice Control screen is used to get a list of all the supported SAPI/TTS voices on your system. Each voice can be selected and tested with a sample phrase.

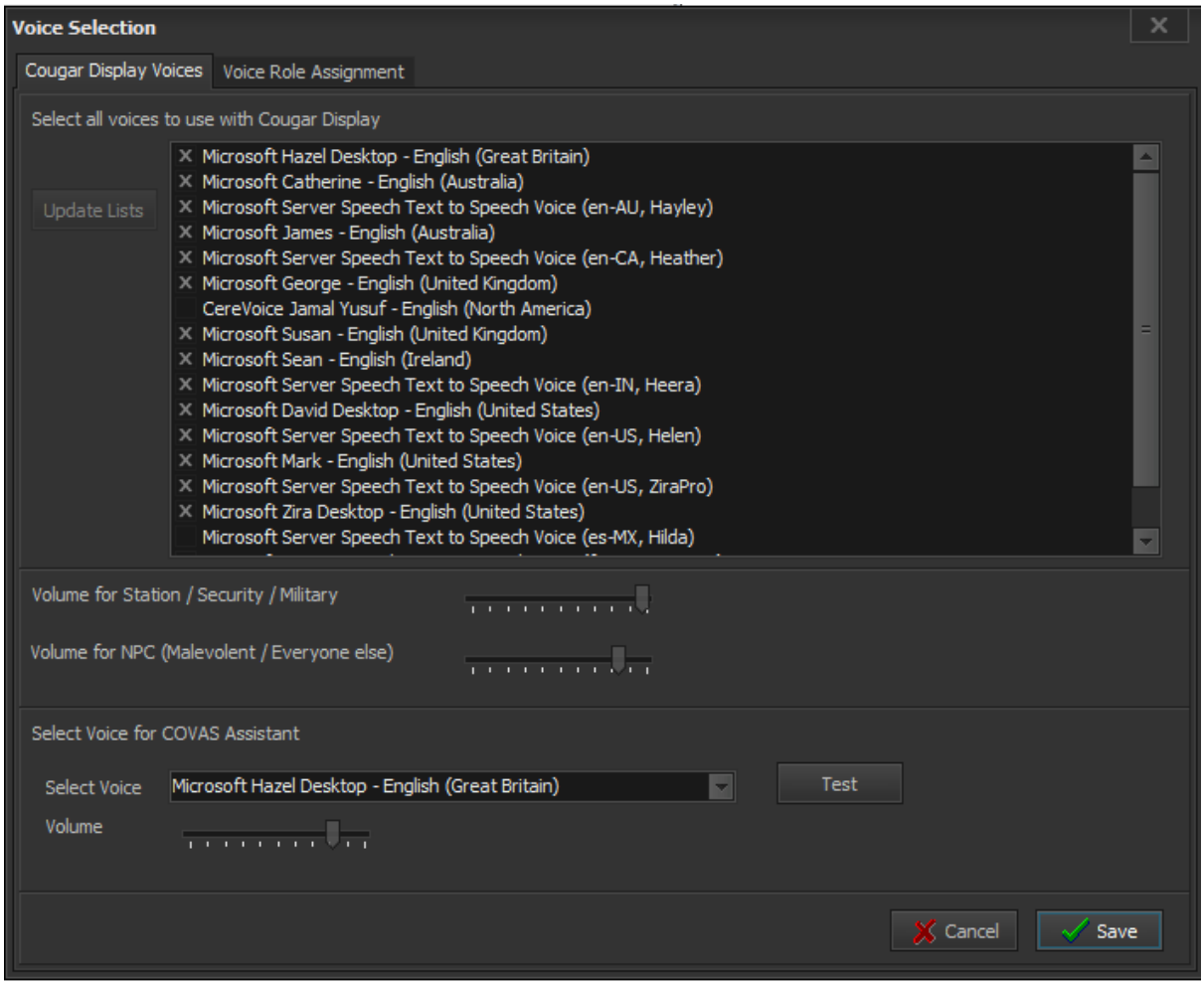


Allow SSML is checked only if all voices on the system successfully render a simple SSML text message. Cougar Display uses SSML to reduce the volume of multiple simultaneous NPC communications and to add time offsets so they don't all speak at once.

Click the **Select Voices** button to assign TTS voices to roles within Cougar Display, and to allow / disallow TTS voices from being used.

Voice Selection

The **Cougar Display Voices tab is** used to assign TTS voices for use within the application.



The enumeration of the text to speech (TTS) system voices, along with the selected Covas assistant voice and volume levels are recorded in the “SAPI.ini” file in the application root folder.
The list shown in this screen are all the voices which have passed the enumeration and SAPI tests, so all of them should work in the application. Place a check mark against each voice you wish to use and click **Update Lists** button to save the changes into the SAP.INI file.

The SAPI voices are enumerated when the application is first executed. If this list is out-of-date (following a Windows update), then delete the **SAPI.ini** file and the application will re-create the file on the next run; alternatively click **refresh** on the **Voice Control** dialog.

Note: Voices that are causing errors on speech rendering are marked with -1 in the **SAPI.INI** file.

Accessing More Voices

There’s a bit of a trick to get access to all of the voices installed on your system. Some voices are hidden from 32-bit applications but they can be exposed and made to work with a bit of registry editing in this key:
`HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\Microsoft\SPEECH\Voices\Tokens`
My system, for example, has 18 voices by a combination of installing Windows feature packs and editing the registry keys. See **EDCD_Tecnical_guide.pdf** for more information.

Space Radio Chatter / Air Traffic Control (ATC) Simulation

Note: Version 1.7 of Cougar Display upgrades the NASA space radio chatter system to Air Traffic Control (ATC). The previously included script has been re-issued as an ATC script.

Fully customisable scripts can be set to play in the background during **supercruise** or **system space.** Using a set of audio filters, synthesised voices are rendered along with ambient noises. Morse code beeps can be heard panning from speaker to speaker, and a full blown Apollo 11 Moon landing conversation is acted out to break up the silence of supercruise voyages.

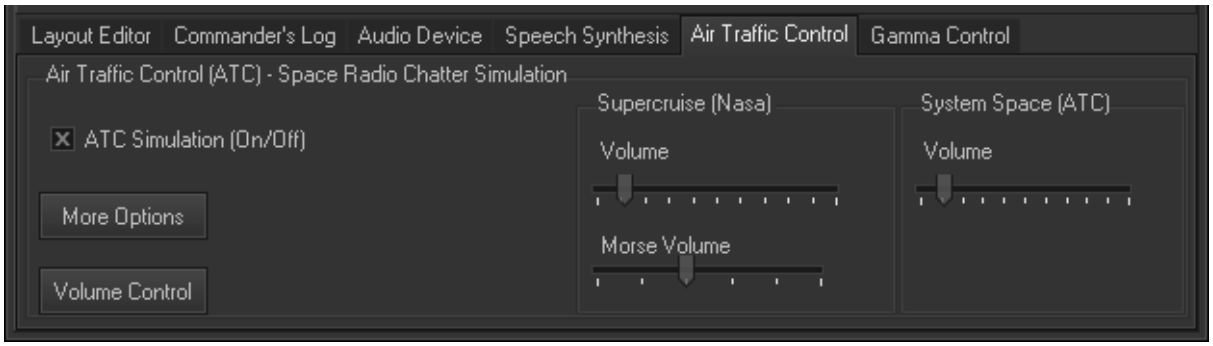
The scripts can be edited (notepad) and actors assigned to each line, along with randomising the voices and lines spoken. A simple help text file is included in the **Audio** folder to explain the details.

Supercruise / system space scripts

Two distinct sets of scripts can be assigned to the ATC system – one for use in Supercruise and the other during system space. The original NASA space radio chatter script was designed for use in supercruise, as it was a single, lengthy conversation. The script can be edited or replaced with any text of your choice. However, this text is displayed on console #1 so the language must be English.

System space scripts are designed around the concept of ships docking at stations. The text is not displayed on any console and thus it could be replaced with any supported language that Windows has a TTS voice for. The spoken script is designed to fade in volume when important events or speech is presented in game, for example during combat, interdiction or NPC chatter. By reducing the volume it is intended that the chatter system does not detract with the more important in-game events.

Volume Control



Morse volume is clamped between 0 and 10, with 0 being no sound at all. This setting is for exclusive use during Supercruise.

Separate volume controls exist for the **supercruise** and **system space** ATC scripts when using the UI sliders.

Whilst playing Elite, you can use a joystick axis as a quick way to change the volume of ATC chatter. Click the **Volume Control** button to assign a joystick to this action.

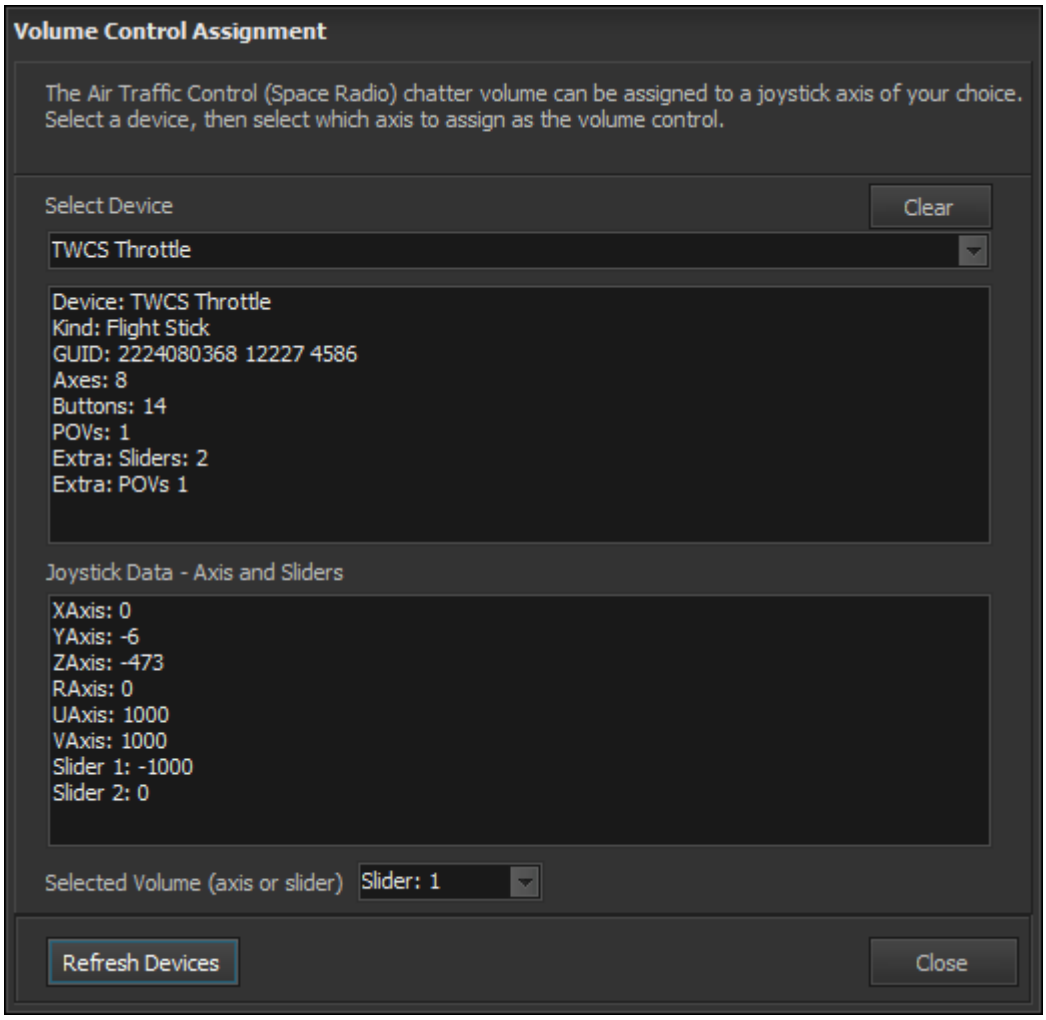
Dynamic Volume Control

The volume of the Space Radio chatter can be adjusted while Elite is being played in one of two different ways:

1. Assign a joystick axis as the volume control
2. Click on one of the MFD icons to toggle ATC Radio chatter on / off

Joystick Assignment

Clicking on the **Volume Control** button brings up a panel so you can assign a joystick axis to dynamically adjust the volume whilst you play Elite.



Using the Volume Control Assignment, select the joystick and the axis you with to use.

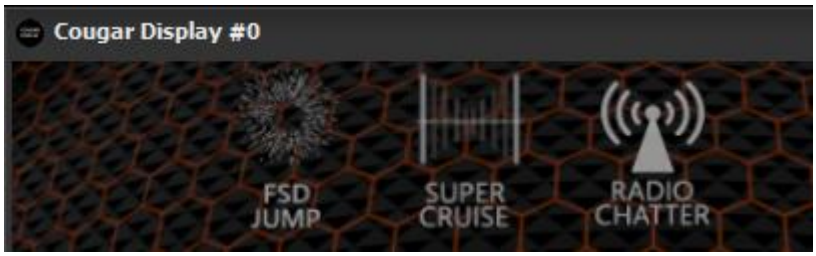
The Joystick polling interval can be adjusted (in milliseconds) in the `MFDCougar.ini` file

```
[Nasa]
JoystickPollInterval=500
```

When using the joystick to alter volume, both the supercruise and system space radio chatter volume will be set to the same value.

ATC Radio Chatter Toggle

The other method to quickly mute ATC radio chatter is to press the **Radio Chatter** icon on the MFD screens (**Supercruise** Layout has the icon by default).



Pressing this button will suspend the processing of new supercruise and system space ATC radio chatter events. The icon will track the current state of the event processing and will be lit (on) when queue processing is enabled.

This icon can be added into the **flight** layout if necessary by adding or replacing an existing icon with icon 60 (Radio Chatter). Use the **Layout Editor** to make this change to update the `MFDLayout.xml` file and also optionally add a hot key press.

Notes

ATC scripts are disabled when any of the following events occur:

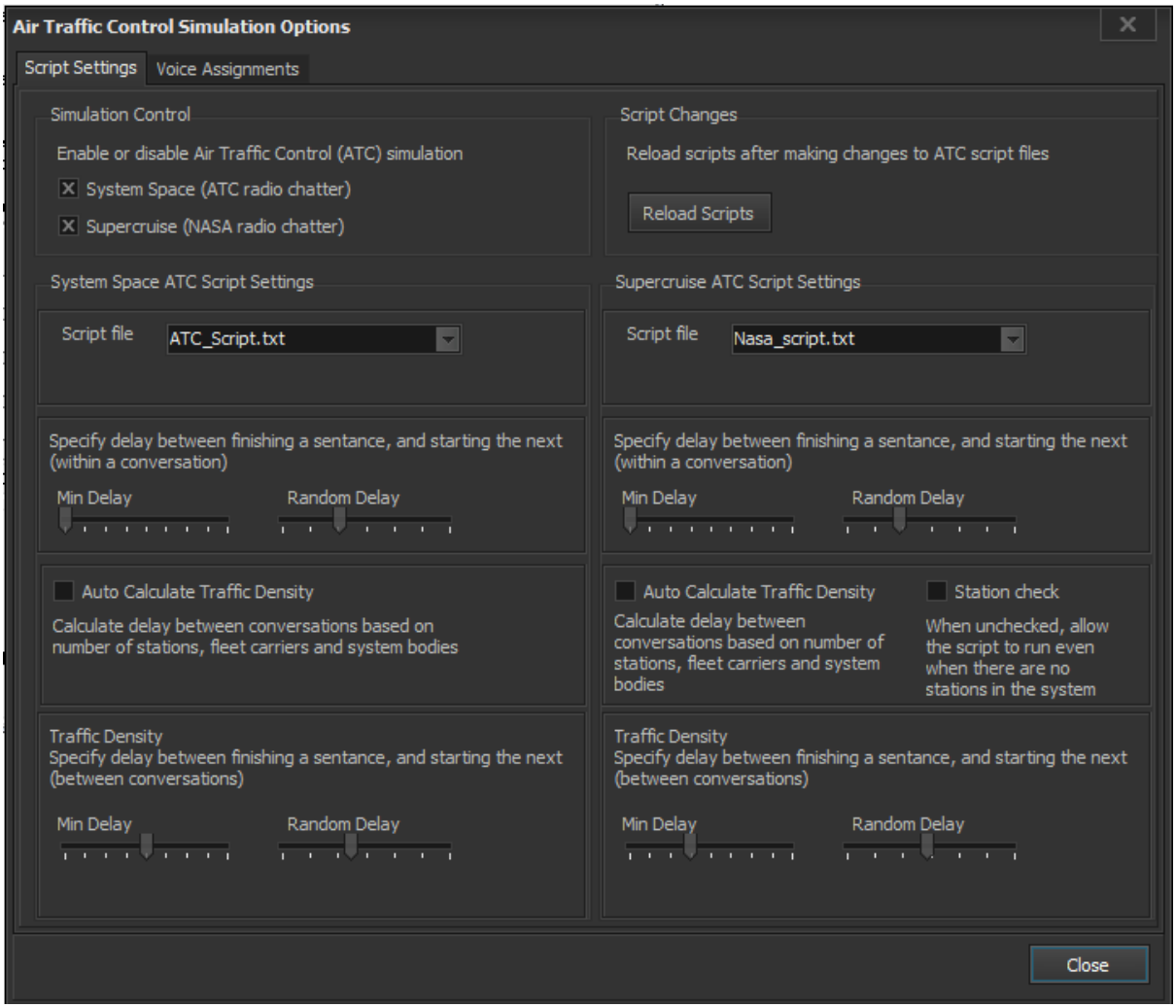
1. Landed or docked
2. Entered a star port mail slot
3. Not in the main ship
4. Approaching planet (to descend/orbital cruise)
5. On a Fleet Carrier
6. Main Menu
7. Galnet custom panel reading news item

Conversely, ATC scripts won't run until a FSD jump has been made to a system – so that the number of stations, fleet carriers and system bodies can be obtained. A file "stations.txt" file is written to the **cache** folder. If there are no stations or fleet carriers, then the ATC script will not process any conversations – clear the **station check** override (a supercruise ATC script option) to force the script to execute regardless of the station count.

Please report any instances where system space ATC scripts are running and you feel they shouldn't be.

More ATC Options

The **Air Traffic Control Simulation Options** dialog allows fine tuning of the ATC scripts.



Simulation Control

Supercruise or system space ATC scripts can be enabled (default) or disabled.

Script Changes

When scripts are edited whist Cougar Display is running, they can be re-loaded.

ATC Script Settings (Supercruise/System space)

Script file

The choice of script to execute is available from the drop-down list. Supercruise and system space scripts can be any of the available. The **MFDCougar.ini** file has a list of the script files which populate the drop-down list. You can add your own, custom script files to this list.

Auto Calculate Traffic Density

The **Auto Calculate Traffic Density** is used to determine the delay timings *between* different conversations (interactions). When this setting is enabled, the **Traffic Density** sliders are disabled.

The delay between sentences read out by the 'actors' in the radio chatter is controlled via the `ScriptMinDelay` and `ScriptRndDelay` values (these are in seconds). There is a minimum time interval between any two sentences governed by `ScriptMinDelay` and a random amount added via `ScriptRndDelay`. These values affect the timings of the sentences *within* a conversation.

Traffic Density (delay between conversations) can be manually specified by unchecking **Auto Calculate Traffic Density**. The Min Delay and Random Delay are as described above, except they apply to the time between conversation blocks (interaction as denoted by <I> in the script files).

The **Auto Calculate Traffic Density** is initially disabled for supercruise chatter, and enabled for system space chatter.

Station Check

When checked, the ATC script system checks to see if there are any stations in the system, and if not disables radio chatter. However, when running the NASA version of the script (in supercruise) the station check may not be desirable, and thus it is recommended to disable station check.

Traffic Density Calculated Value

Traffic density is based on 3 variables: # stations, # FleetCarriers and # Bodies.

Stations and # FCs are prime candidates in the amount of communications between pilots.
The combination of the two in the same system likely increase chatter as there would be extra traffic going back and forth between the two (ie resupply, refuel, etc.)

The number of bodies is important for 2 reasons:
Firstly, each body in a system can have a maximum of 16 fleet carriers - indicating a maximum density within the system.
Secondly, the # bodies is that it implies size of the system which helps to simulate range of communications - larger system will reduce traffic density.

If system contain either FCs or Stations, but not both, then:

$$\text{Time between transmissions (in seconds) variable} = \frac{1}{/\ \frac{\#\text{ of stations or FCs}}{\#\text{ of bodies x 16}}\ \backslash}$$

If system contains both stations and FCs, then:

$$\text{Time between transmissions (in seconds) variable} = \frac{1}{/\ \frac{(\#\text{ of stations X 2}) + \#\text{ of FCs}}{\#\text{ of bodies x 16}}\ \backslash}$$

ATC Script Editing – Conventions

Using the following conventions, you can create or customise your own ATC scripts.

Start and end interactions

<I> Start a new interaction. Randomises actor voices, call signs, station name, super power and ship type
</I> Ends interaction

Conventions within an interaction:

<A> Use voice 1 - (station) authority
<O> Use voice 2 - Pilot
<Q> Use voice 3 - security voices (Q for Quick Reaction Force or QRF)
<T> Use voice 4 - Pirate (T is from Tango as P is used for a pause - described below)

<QS> QRF call sign
<CS> App to generate a random call sign for the pilot
<S> Random Station name from the current system for this interaction
<SP> Super power noun (Empire/Federation/Alliance)
<SS> Super Power adjective (Imperial/Federal/Allied)
<ST> Ship type - one of:Federal Corvette,Imperial cutter,Imperial eagle,Diamond back explorer,beluga liner,Annaconda,Cobra
<SM> Ship manufacturer - Lakon, core dynamic, etc

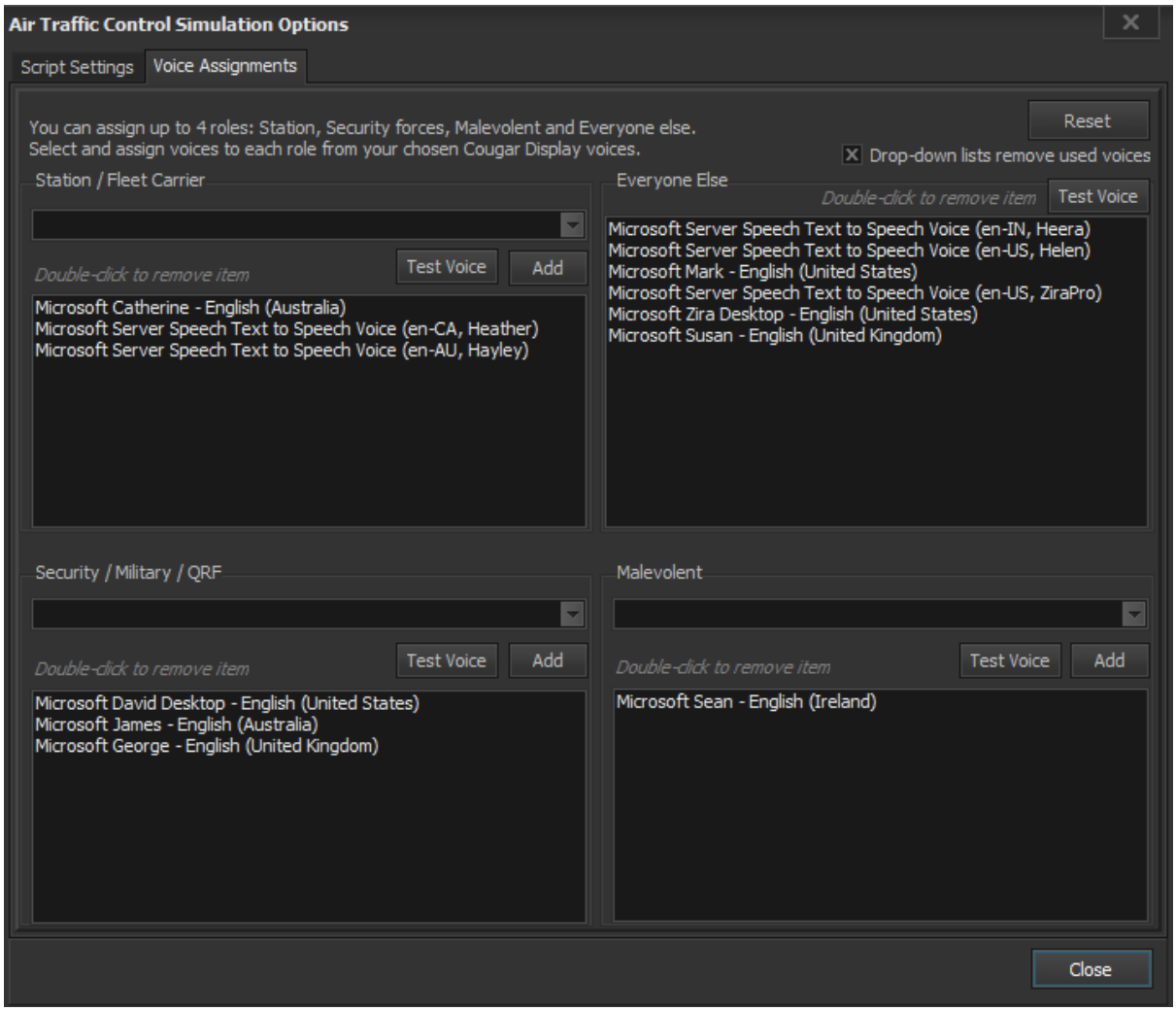
<#9> Random number up to the specified number (1 - 9 are allowed)
<##> Use previous random number
<P10> Pause for specific amount of time (10 sec)
<P*10> Pause for a random amount of time up to 10 seconds

=== denotes a comment line
XXX Substituted for commander ship ident
YYY Substituted for commander name

Tip: To add slight pauses in a sentence, use punctuation such as comma (,) or full stop (.)

Voice Assignment

A maximum of four (4) voices can be in any one ATC conversation. To facilitate a random voice selection, the Voice Assignments tab allows the selection of specific voices for use with each of the actor roles (station, security, malevolent or everyone else).



If any list is empty, the actor's voice will be chosen from any of the Elite enabled voices which are defined and selected in the **Voice Selection** dialog (see Speech Synthesis).

Click the **Select Voices** button to assign TTS voices to roles within Cougar Display, and to allow / disallow TTS voices from being used.

Use the Click the **Select Voices** button to assign TTS voices to roles within Cougar Display, and to allow / disallow TTS voices from being used.

Voice Selection button on the **Voice Control** dialog to assign TTS voices to roles within Cougar Display, and to allow / disallow TTS voices from being used.

Uncheck the **Drop-down lists remove used voices** to keep assigned voices in the drop down lists, so that the same voice can be assigned to more than one role. By default, assigned voices are removed from the available list of voices so they can only be assigned to one role.

Voice Selection

Double-click an entry to remove the voice from the role. When the voice is removed it's added back in to the **Everyone Else** role.

Click the **Test Voice** button to sample the speech from the selected voice.

Refreshing the lists

From time to time new voices may be added to Windows and not reflected in the lists (or Windows update removes any manually added entries via the registry). Additionally, changes to the Speech Synthesis tab (changing the voices enabled for Elite) will require a Reset.

Use the **Reset** button to clear all the lists and read all the Cougar Display enabled voices from the TTS/SAPI **Voice Selection** configuration on the Speech Synthesis tab of the main application (Edit Voices) button.

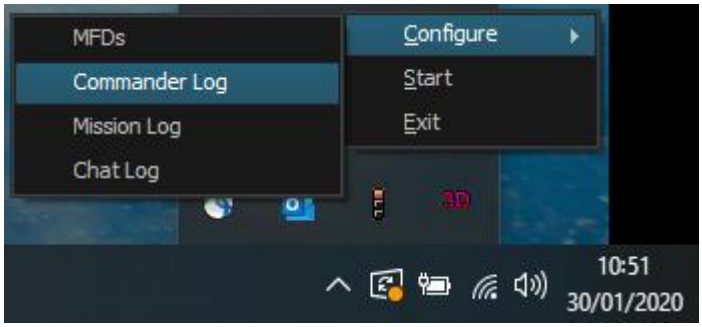
TTS Voices – Memory Consumption

Loading TTS voices into memory consumes large amounts of memory, and the more voices used by the application the more memory consumed. Cougar Display is a 32-bit application (3GB enabled) and as such the overall memory consumption must be below the memory limit for 32-bit processes.

Unlike Microsoft, CereProc voices take an additional 60+ MB of memory when they are first loaded. It's entirely possible when the application is running that playing a new voice will cause an out-of-memory error (and indeed in versions of Cougar Display below 1.12.0.0 this was happening). CereProc are currently working on fixing a memory allocation error in their TTS engine to handle situations like this (at the moment this error causes catastrophic problems for Cougar Display).

Viewing Additional Windows

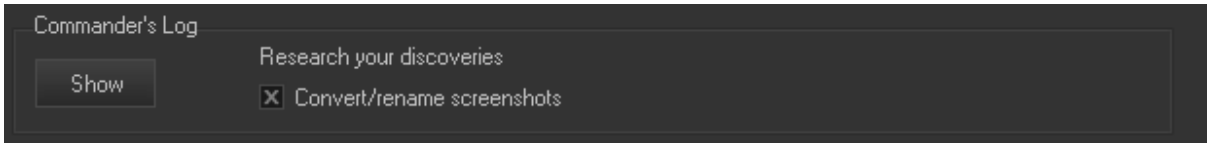
By default, the Commander's Log, Mission Explorer and Chat Log windows are displayed on start up. These can be closed or minimised at any time. To reactivate these windows, use the system tray applet:



Commander's Log

A complete application built-in.
The **Commander's Log** is a comprehensive set of hooks and rules designed to capture the names and bodies in the Elite universe where strange and interesting stellar peculiarities exist.

From the main screen, locate the **Commander's Log** panel and click **Show**.



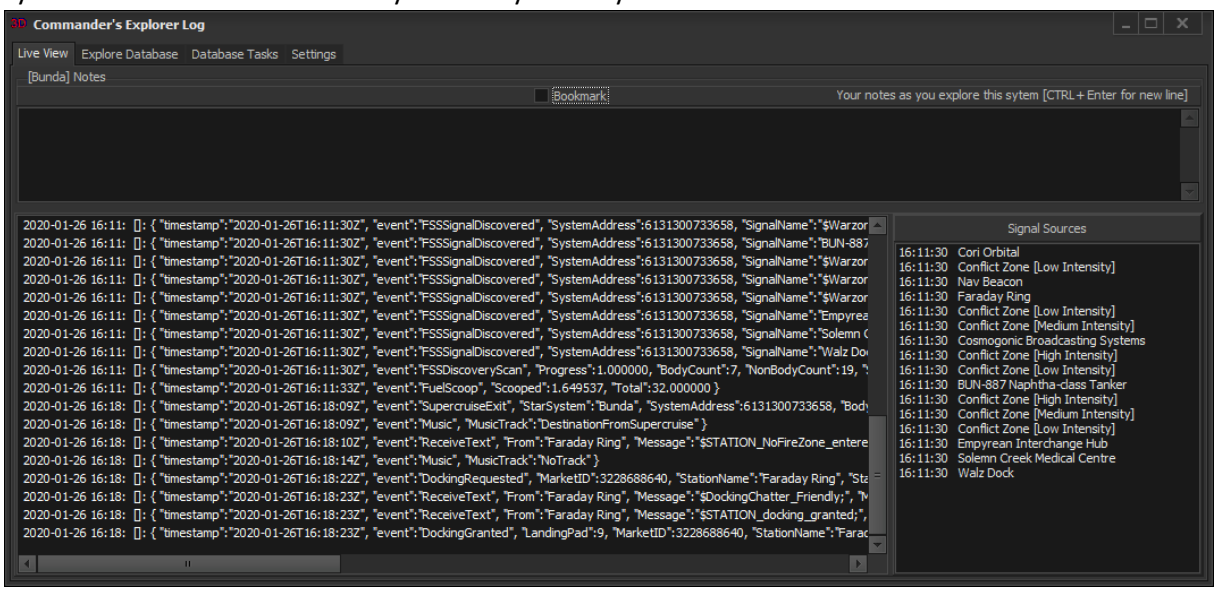
Selecting the option **Convert/rename screenshots** activates the feature to convert bitmap images to png images in the folder Elite uses to capture images. Additionally, the file is renamed to something more useful which includes the date/time, system and body name. A bookmark is added into the database so you can quickly identify systems of interest.

Once open, the **Commander's Log** has four tabs across the top to perform the following functions:

- Live view of the journal logs
- Live view of signal sources
- Explore the database of previous events
- Import journal logs
- Configure which events are written to the database

Live View

Live view is linked to the current system in Elite and shows any notes you have recorded for this system. You can bookmark this system for your easy reference at a later date.



When a commander jumps to a new system, the **Commander's log** module will analyse the results of a DSS scan and report back any features of interest which are categorised and stored in a local database for later analysis.

Stellar features of interest include:

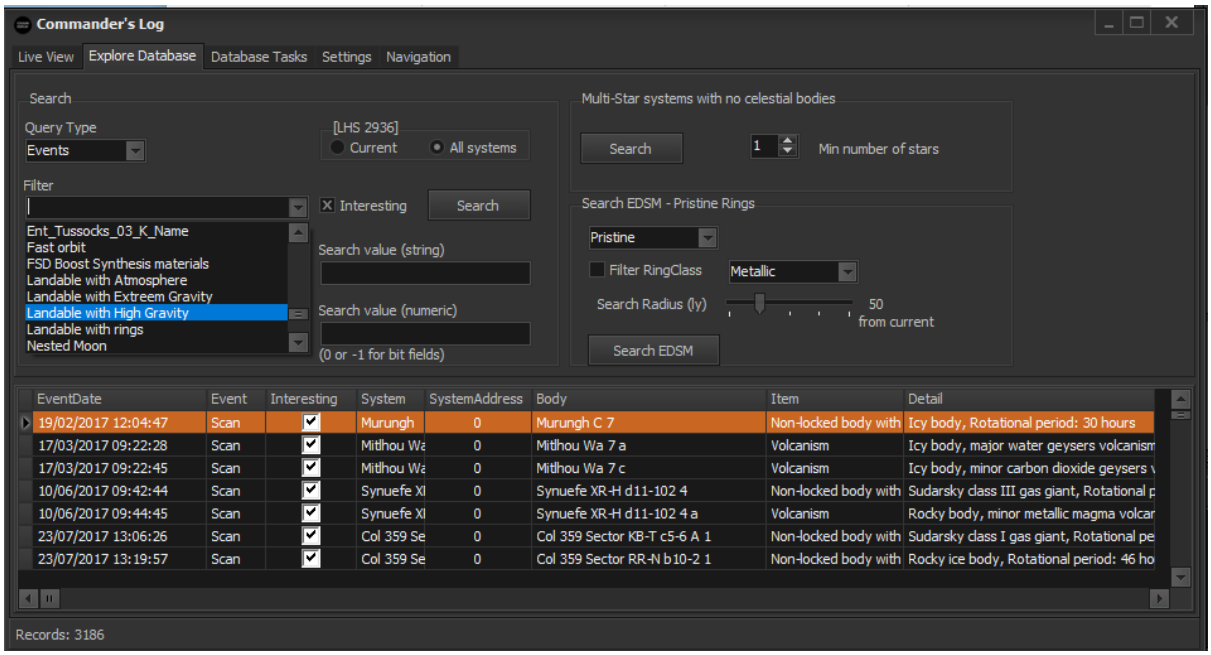
- Systems containing 5 or more of the materials needed for FSD jumps ("jumponium")
- Landable planets with a terraform state
- Landable planets with an atmosphere (Odyssey)
- Landable planets with rings
- Landable high gravity planets
- Landable large planets
- Planets with wide rings
- Planets in close orbit
- Planets with moons in the rings
- Moons with moons (nested)
- Really small bodies
- Bodies with fast rotation
- Bodies with fast orbits
- Bodies with high eccentricity
- Stars - close binary pairs
- Stars - colliding
- Stars with rings
- Systems where a codex discovery was made

Additional Features

Display of the signal sources in the system – see **Settings** tab
Add your own notes and bookmark the system.

Exploring the Commander's Database

The database contains a rich set of information which can be searched.



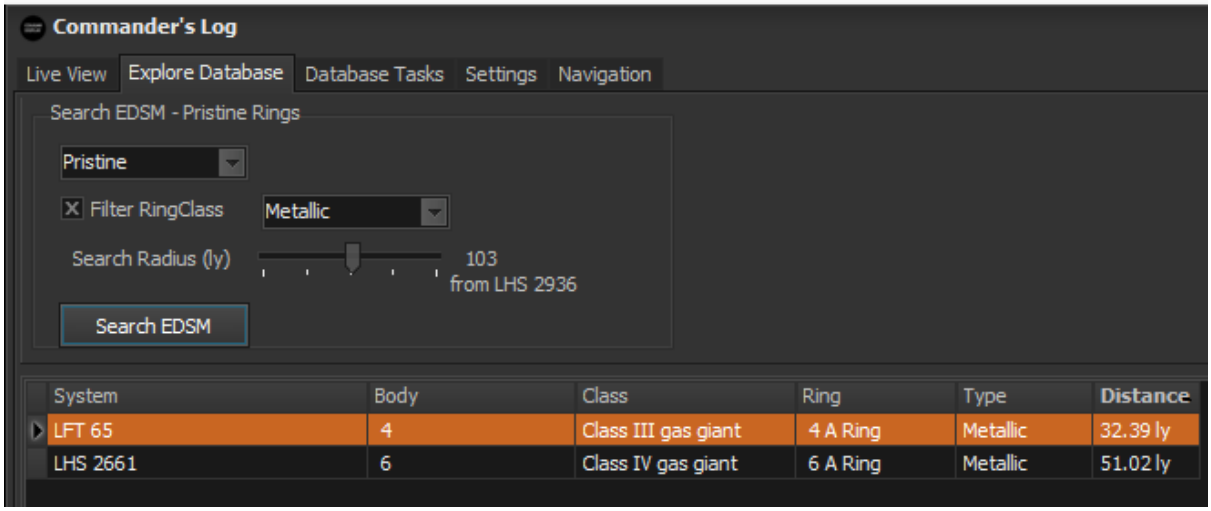
The Commander can:

- Look for any of the "items of interest" features (nested moons, colliding binaries, etc.)
- Look for codex entries
- Search in the scan events for Stars, planets and rings of interest (radius, atmosphere composition, etc.)
- Search for rings with hotspots / geological POIs
- Search for a multi-star system with zero bodies which is something I know from the forums that some Commanders would like to find
- Search for Pristine Metallic Ringed systems nearby

Mining Assistant - Searching for Pristine Metallic Ringed systems nearby

When laser mining, in particular, it's often best to locate a pristine system with metallic ring because selling Panite is very profitable.

To help locate suitable systems, use the **Search EDSM – Pristine Rings** section to set up what **Reserve level** and **RingClass** you're interested in. Selecting the **Filter RingClass** limits the results to only the chosen class, otherwise all types of ring are returned (Icy, Rocky, Metal Rich and Metallic).



Use the search radius to increase or decrease the number of systems searched relative to the current system.

EDSM is used as the source of data because it's most likely that your local explorer database won't contain the information on the surrounding systems.

Database tasks

Clear Database

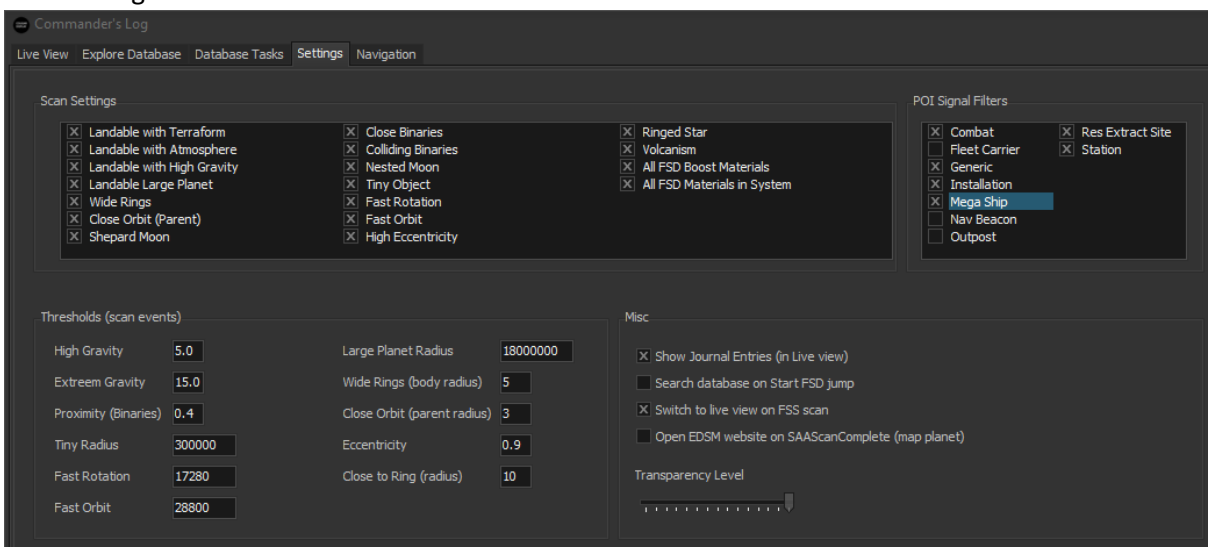
Database tasks tab allows the commander to clear out the portion of the database relating to journal scan events. This is useful so that a re-read/refresh of the journal logs can be performed. This is a safe operation and bookmarks, system notes are not affected.

Import Elite Dangerous Journal Logs

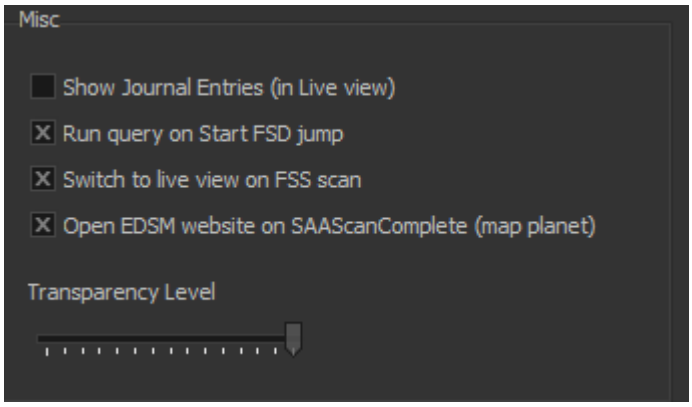
This option populates a local database with all the codex and scan events found in the journal logs. There is a wealth of information contained in these logs and some of it is very useful to collect – i.e. for mission route optimisation.

Commander Log Settings

The Commander can also tweak the **settings** of the scanning module so that he can define what an "interesting" item means.



Misc Options



Show Journal Entries (in Live View)

This option acts like a debug view – it shows the journal entries as they are written by Elite.

Run query on Start FSD Jump

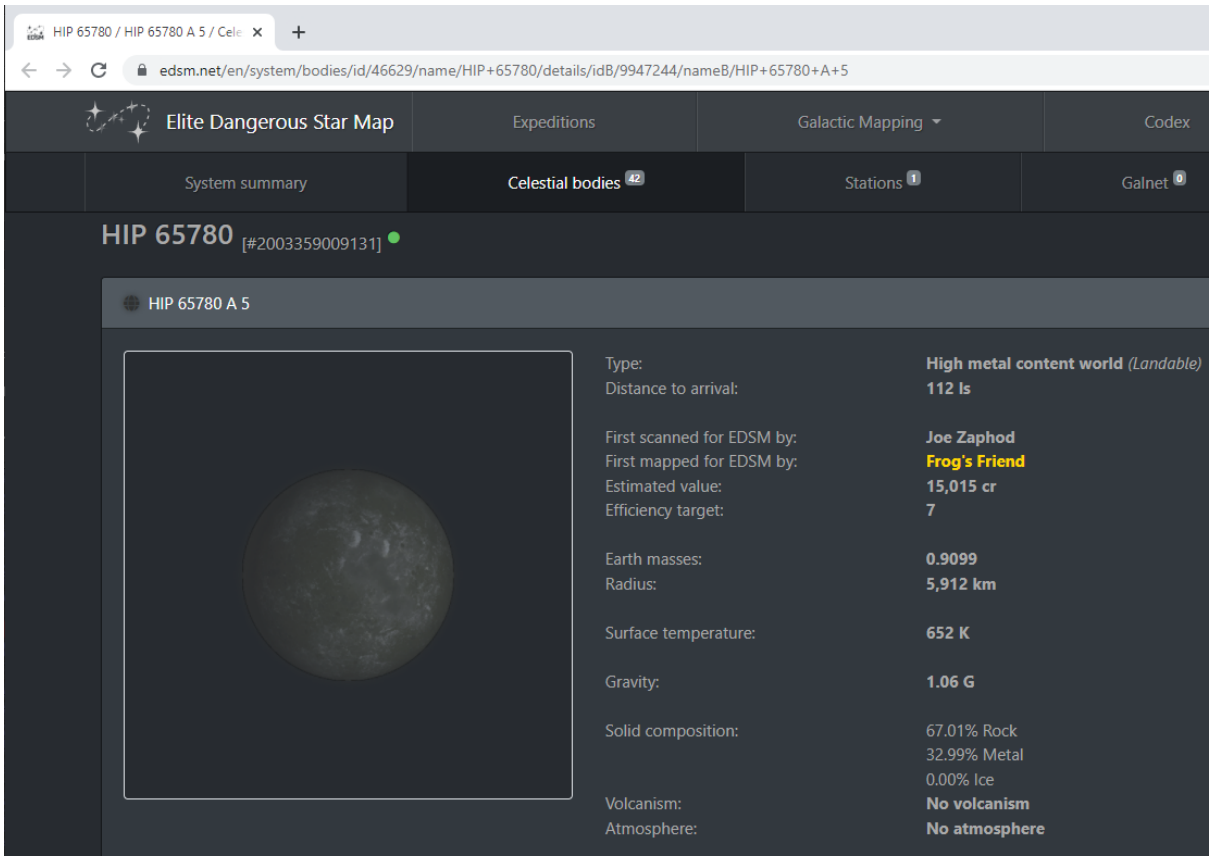
This option runs a database query and shows a summary of the new system you are about to jump to. The **Explore Database** tab is activated.

Switch to live on FSS scan

This option switches and activates the **Live View** tab so that signal sources can be seen, as well as the results of the scan analysis looking for "interesting" items.

Open EDSM website on SAAScanComplete (map planet)

This option opens EDSM website when mapping a planet. It displays body information that EDSM holds about that planet. If ED:CD has not used EDSM to obtain system or body data, then no webpage is opened.

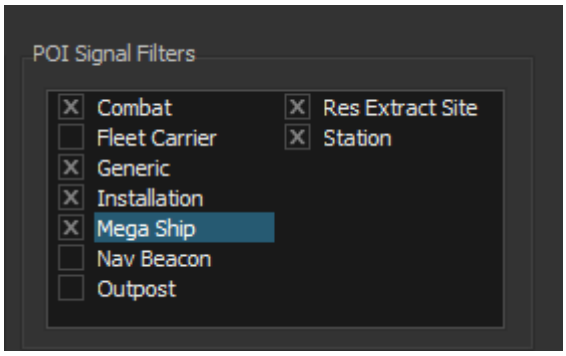


Transparency Level

Sets the level of transparency for the window so that background windows show underneath. Good for using a window as an overlay.

POI Signals

The **settings** tab contains a **POI Signal Filters** area to select individual control over which journal signal type gets displayed in the **Live View's POI Signals** display.



Unchecked items are removed from the **POI signals**.

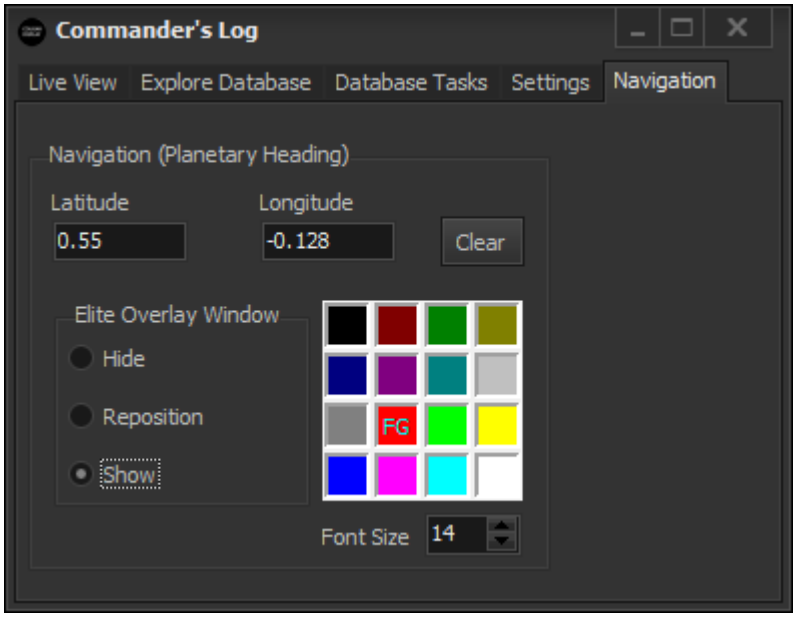
Notes:
Fleet Carrier – is unchecked by default (removed from the POI Signals).
Outpost - refers to planetary outposts/settlements with landing pads and is unchecked by default (removed from the POI Signals).
Mega Ship - also includes stronghold carriers.

The **Live View** tab contains a **POI Signals** area with all the (unique) signal source entries from the FSSSignalDiscovered events in the journal (subject to the above filters).

Stations are prefixed with #
Planetary Outposts/Settlements are prefixed with ##
Mega ships/Stronghold Carriers are prefixed with [Mega]
The distance from the main star to the station is shown (xx LS), and the body it's orbiting/positioned.

POI / Navigation Heading Assistant

The commander's log features a navigation tab where longitude and latitude values can be set for a specific point of interest. The system will automatically calculate a heading when approaching the target body (planet).



Elite Overlay Window

Additionally, the heading text can be displayed as an overlay on top of Elite via an invisible window that you can place in your desired location. The windows will maintain its windows Z-order when the heading window is receiving latitude and longitude information from the journals (typically on planet descent/landing).

Enter your target **Latitude/Longitude**, or click **Clear** to remove (stop the feature)

Set the **Elite Overlay Window** to:

- **Hide** to close the window (effectively disabling the feature)
- **Reposition** to move the window into the desired location (on top of Elite's borderless window)
- **Show** to dynamically show the window when Elite is generating body/lat/lon data

Customisation

The overlay font size and colour can be set using the colour grid and font size controls.

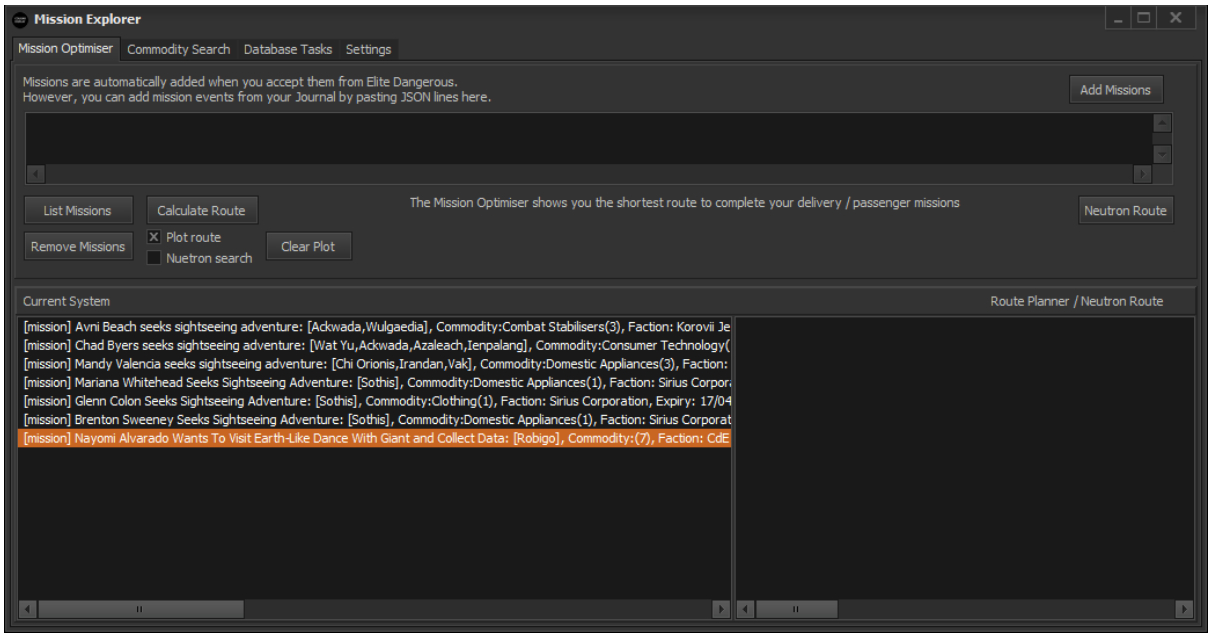
Mission Explorer

The Mission Explorer module consists of:

- Mission Route Optimiser
- Commodity Search
- Commodity Search Alerts
- System/Station facility searches
- Route Plotting / Neutron star route plotting

Mission Optimiser

This module can help a Commander plan his routes from station to station. On each new mission accepted, the system builds a list of known missions and calculates the most efficient route between star systems. The computed route is based only on the systems in the missions and performs distance/jump calculations based on your current ship and its cargo load. As part of the supply/demand missions, the system will add stations you must visit to buy goods (using the commodity search module functionality). The current cargo load/capacity and mission cargo are taken into account when calculating the route.



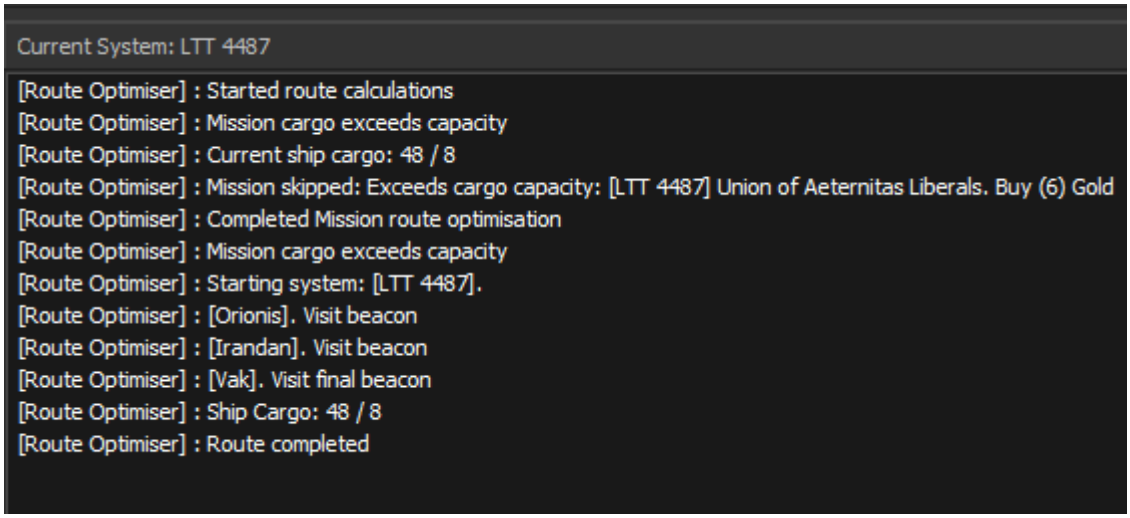
List Missions

The **List Missions** button lists the known, active, missions in the database. Usually, these are identical to Elite's mission list – but there is an option (see **Add Missions** below) to add historical missions.

Each mission entry includes the system name, station name (where appropriate) and the overall quantity of commodities required.
Missions with a secret change of destination are also shown and highlighted.

Calculate Route

The **Calculate Route** button calculates the best possible route through the star systems listed as part of the mission list.
The system automatically recalculates the route on every FSD Jump, or when a mission is failed, completed, abandoned or re-directed (to a different system).



If the **Plot route** option is selected, the route is plotted on the built-in galaxy map.

Remove Missions

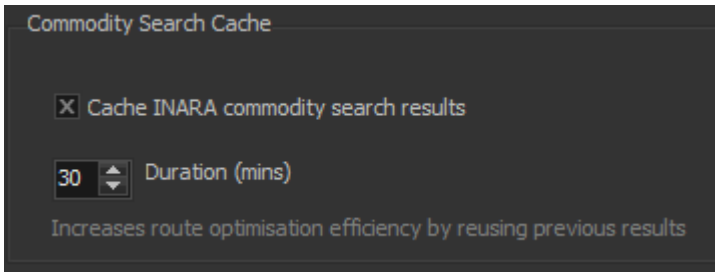
This means that the missions table is cleared.

Add Missions

This is an advanced user feature available so that Commanders can cut/paste an entry from the journal file and have the mission optimiser process that entry to add/delete/update an available mission.

INARA Commodity Search Cache

To aid in website traffic efficiency, the results of commodities search requests are cached for 30 minutes. This is particularly important for the route optimiser which may run many times requesting the same information. The settings are available on the **Settings** tab of the **Mission Explorer** module.



Route Plotting

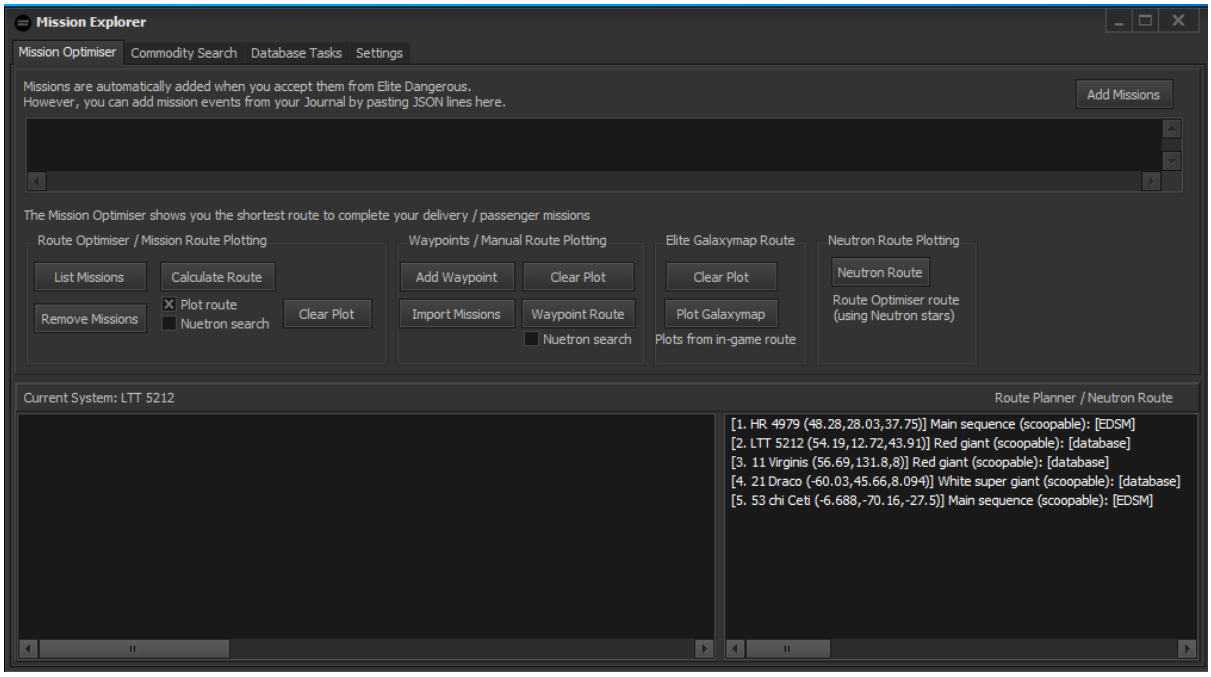
Various routes from different sources can be plotted in the Galaxy Map and a visual representation of the route shown.
Routes can be generated via Elite missions (cargo/passenger), manually created by adding waypoints, or automatically added to waypoints via plotting a route in Elite Dangerous.

Mission Optimiser Route Plotting

Mission (passenger/delivery/data/scan) routes can be plotted on an in-built galaxy map. This tool allows the commander to visualise the mission route and see the results of the mission optimiser. Using the left, right and middle mouse buttons, the galaxy map can be rotated, strafed and zoomed. The keys WSAD also perform the same function.

The nodes on the plotted route can be selected (left mouse button) and the camera will focus and zoom in on the selection. Zooming in a little further will then display the system Orrery.

Route plotting is useful for long missions and can identify loops and inefficient waypoints.



Neutron Route Plotting

The **Neutron Route** option allows the route to be altered to include the use of neutron star (boosts). This option is especially useful for planning long range journeys where the number of jumps can be significantly reduced.

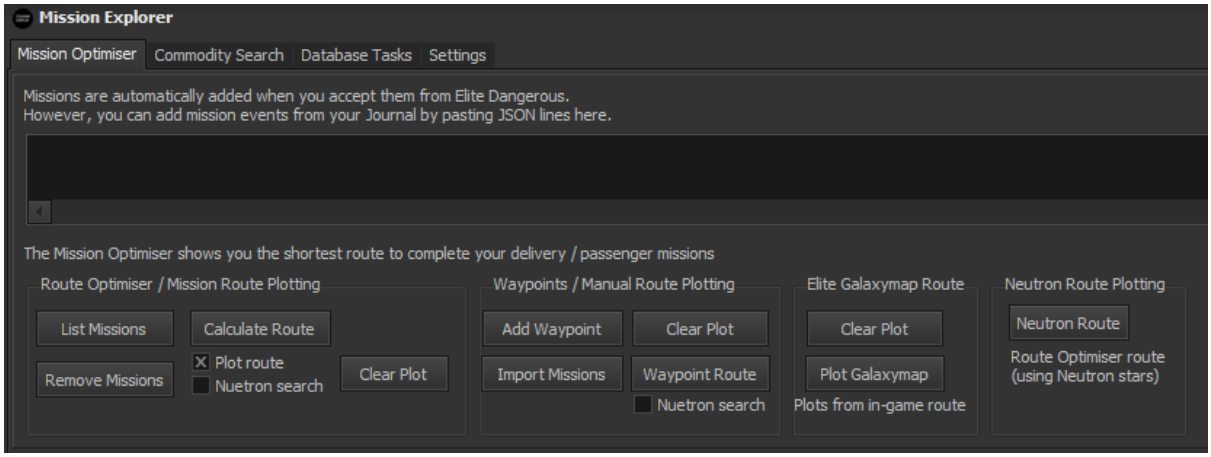
The system already knows about the current jump range of the player's ship and takes this into account when calculating the route. Different results could be obtained by having an empty cargo hold/less fuel when initiating the plot.

Clicking on the star systems listed in the **Route Planer / Neutron Route** output copies the system name to the **clipboard** for easy use within Elite.

WayPoints / Manual Route Plotting

In addition to the mission route plotting, manual **waypoints** can be plotted on the galaxy map.

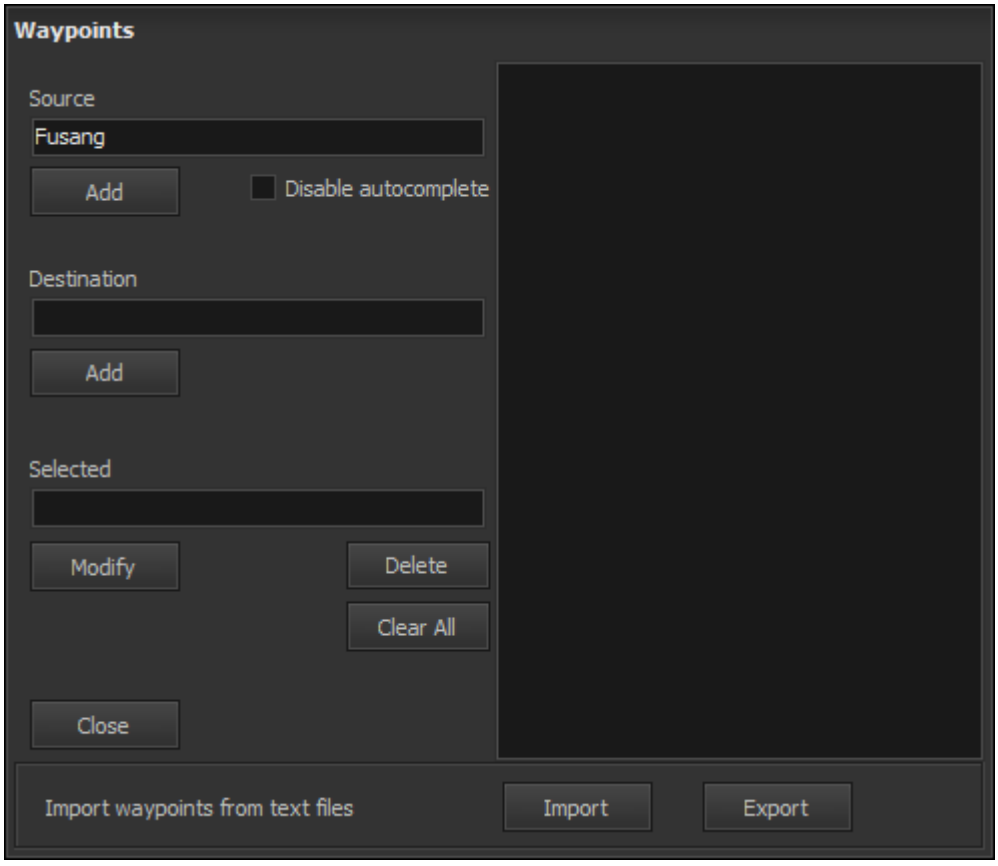
Note: Waypoints are automatically added when a route is plotted via Elite Dangerous.



Use the **Add Waypoint** button to add system names to use during the route. Click the **Waypoint Route** button to plot the route plot and optionally check the **Neutron Search** option to plot the route using neutron star FSD boosts.

Adding Waypoints

The current system will automatically be added and displayed as the **Source.** This field can't be changed.



Add additional destination systems to the **Destination** edit field. As you type a list of possible system names are shown. Click from the list (using the mouse) to select a system name, then click Add. The

list of system names is sourced from the Mission Explorer EDDB database. If your list is empty, import from EDDB (**Database tasks** tab → **Import EDDB** (systems_populated)).

To edit a waypoint, select it from the list – it will now appear in the **Selected** edit field. Click the **Modify** button to correct a spelling, or **Delete** button to remove this system from the **Waypoints** list.

Click the **Clear All** button to remove all waypoints.

Note: System names that can't be located (in your explorer database or via EDSM) will be automatically removed from the plotted galaxy map.

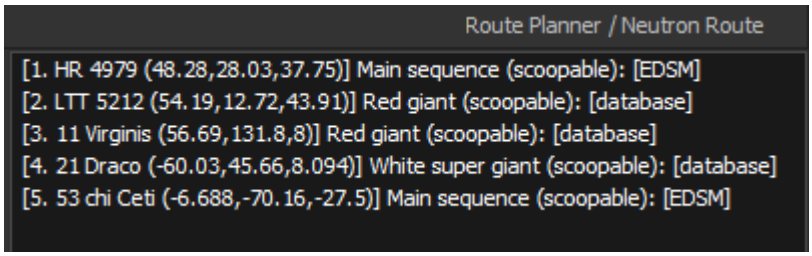
Import/Export

If you find yourself wanting to repeat a multi star system trip, it is useful to save the list of waypoints. To do this, click the **Export** button – the waypoints are saved in comma separated (CSV) format and can be easily re-imported via the **Import** button, or via Microsoft Excel.

Additionally, simple lists of system names can be imported (where each line ends in carriage return/line feed).

Elite Galaxymap Route

When a route is plotted in-game, the star systems are added to the waypoints list. Clicking the **Plot Galaxymap** button plots the route onto the galaxymap MFD and also gives a route summary in the Route Planner Display.

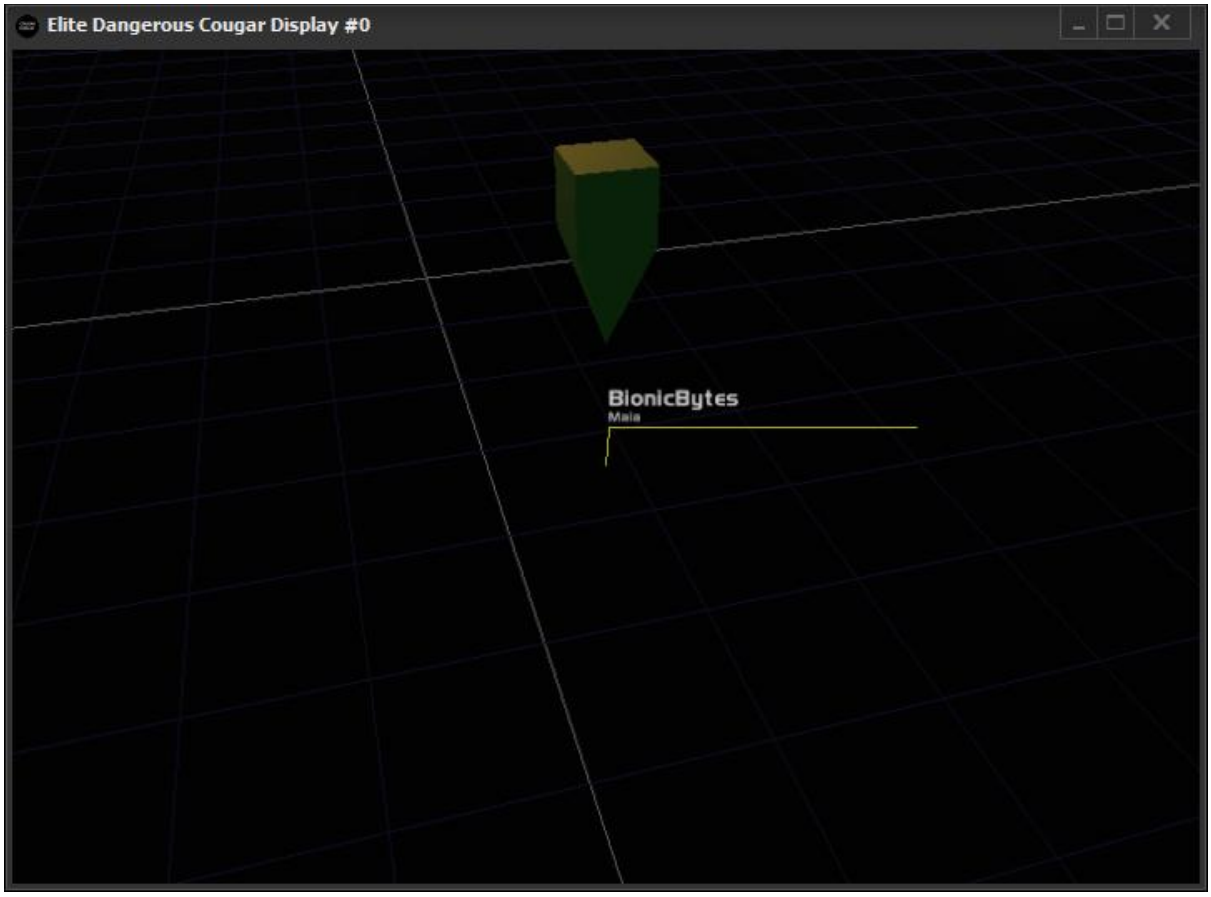


This display gives an indication of the main sequence star, if it's 'scoopable' and if the star system is already known – ie making it easy to spot an undiscovered system in the route.

The indication of [EDSM] and [database] show where the system is known – database means you have discovered the system and it's contained in ED:CD's local database; EDSM means that the system is unknown to you but has been reported by other Commanders to EDSM.

Galaxy Map / Orrery

The galaxy map shows the route plotted by the Mission Explorer module – either mission routes or manual waypoints. If neither of these have been populated, then the galaxy map centres on your Commander's last known star position and is indicated by a large (green) pin drop icon.



The display can be rotated, translated (left/right) and zoomed (forward/backward) by the use of left, right and middle mouse buttons.
Additionally, the WSAD keys are assigned to the forward, reverse and rotate (left/right) functions.

Mouse Operations

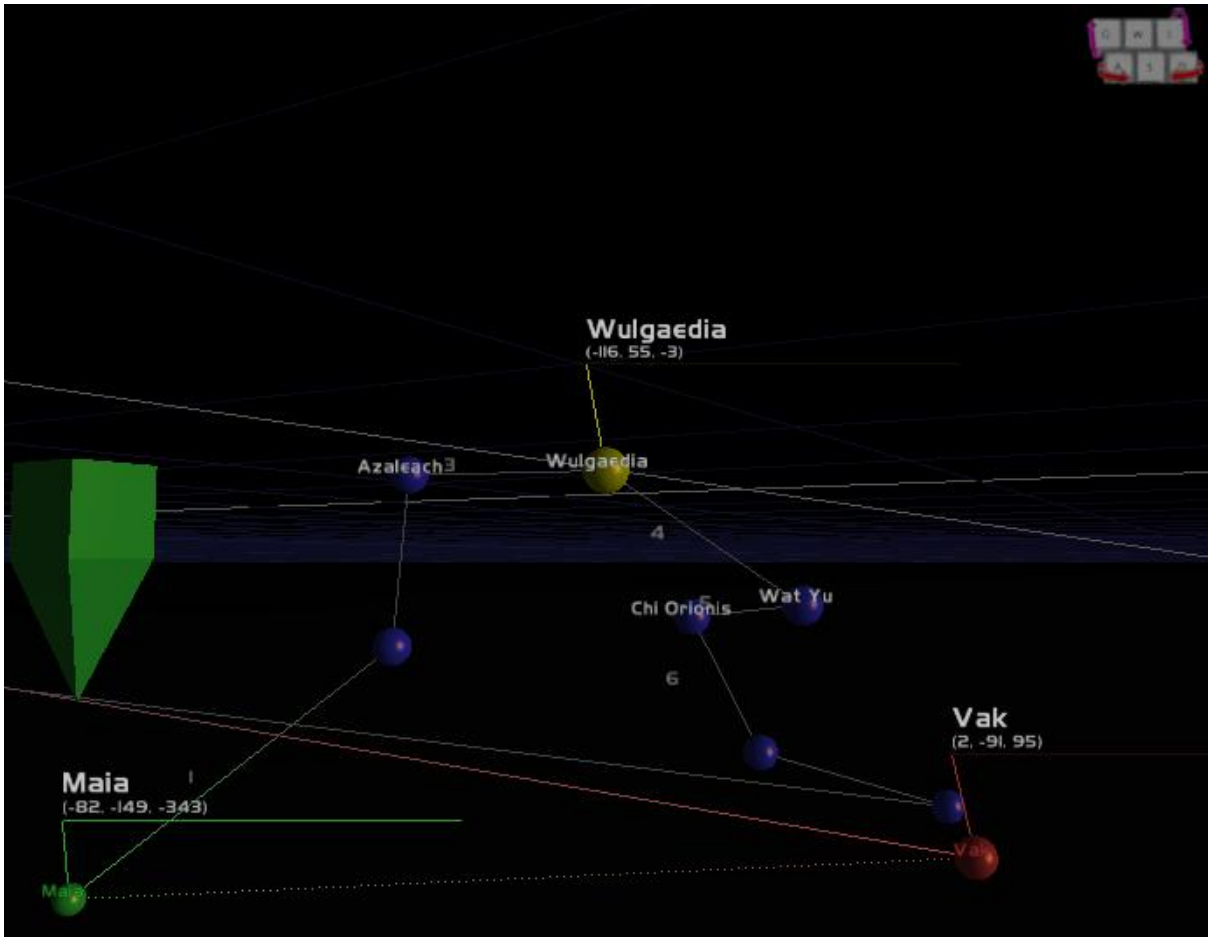
Hold the mouse button down (left/right/middle) for the control to operate.
The mouse wheel is equivalent to holding the middle mouse button and moving the mouse forwards/backwards (the zoom function).

- During rotation, the view is always focused on the currently selected system (node).
- Zoom (forwards/backwards) is always focused on the currently selected node. The closer the camera is to the node, the less responsive it becomes (logarithmic zoom).
- Translate (right-mouse click) is a free form movement that allows you to move the camera position up/down and left/right whilst holding a view lock onto the currently selected node.
- Clicking a node will select it (displays in yellow) and the camera will automatically track to the location.
- Clicking an empty space will start the rotation function.
- Clicking a system name in the **Mission Explorer's Route Planner** will select the node.

Route Node Link Information

Each of the connected nodes can display some extra Meta data.

- The order of the connected nodes is displayed midway between any two nodes.
- The starting node, end node, mid-way nodes display the system name at the centre of the node. Every route graph displays these 3 node's Meta data.
- For larger graphs, the nodes immediately prior and the two after the selected node have their system names printed at the centre of the node.
- The number of nodes with printed connection metadata depends upon how many nodes are in the connected graph. Nodes with less than 10 members will display less node meta data.



Customisations

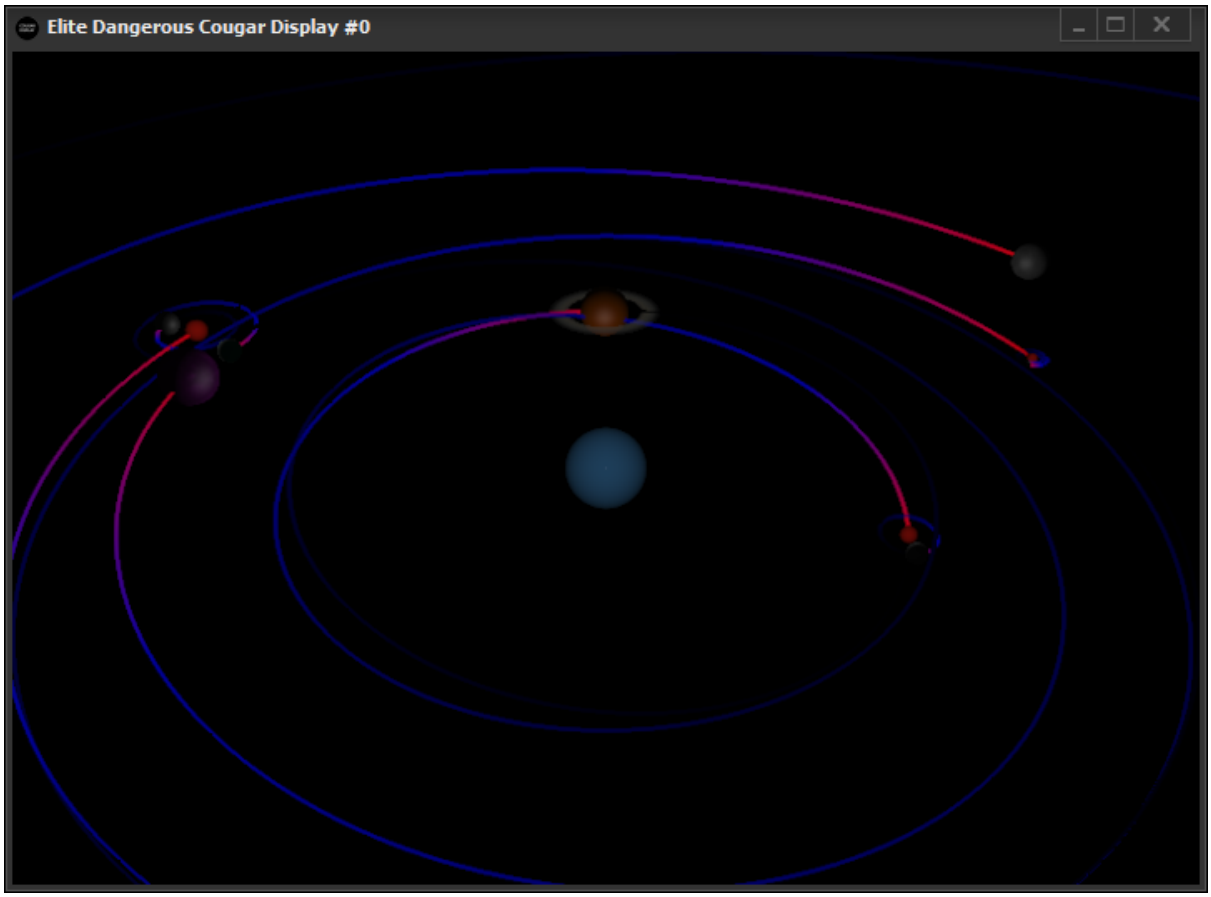
There are a few settings which can be tweaked in the **MFDCougar.ini** file. These values are explained in the file and affect the look of the galaxy map stars.

Mouse sensitivity can be changed in the INI file with values ranging from 0.01 to 10.0

Galaxy Map Built-in Orrery

When a node has been selected in the galaxy map, a zooming in towards the node will automatically switch to the orrery when the camera is very close up.

When in the orrery, moving the camera far away from the central star will switch back to the galaxy map.



In this mode, the orrery does not contain any user interface.
To see the full orrery, complete with user interface, switch to the Orrery/System map within Elite, or enable 3 or more Cougar Displays (with the System role).

The orrery user input (mouse/keyboard) is identical to the galaxy map.
For example, press and hold the [S] key to pan back the camera and exit into the galaxy map.

Orrery

The Orrery is displayed when Elite's HUD focuses on the System map or the Orrery view.
The planet bodies displayed in the Orrery are obtained from EDSM and cached locally in an EDSM folder. If no data is returned from EDSM, then the Orrery is built from the data contained in the local database (from the Journal scan events). Using EDSM as the primary source of data means that the Orrery is most likely to display system bodies even when you have not visited a system before.

Configuration

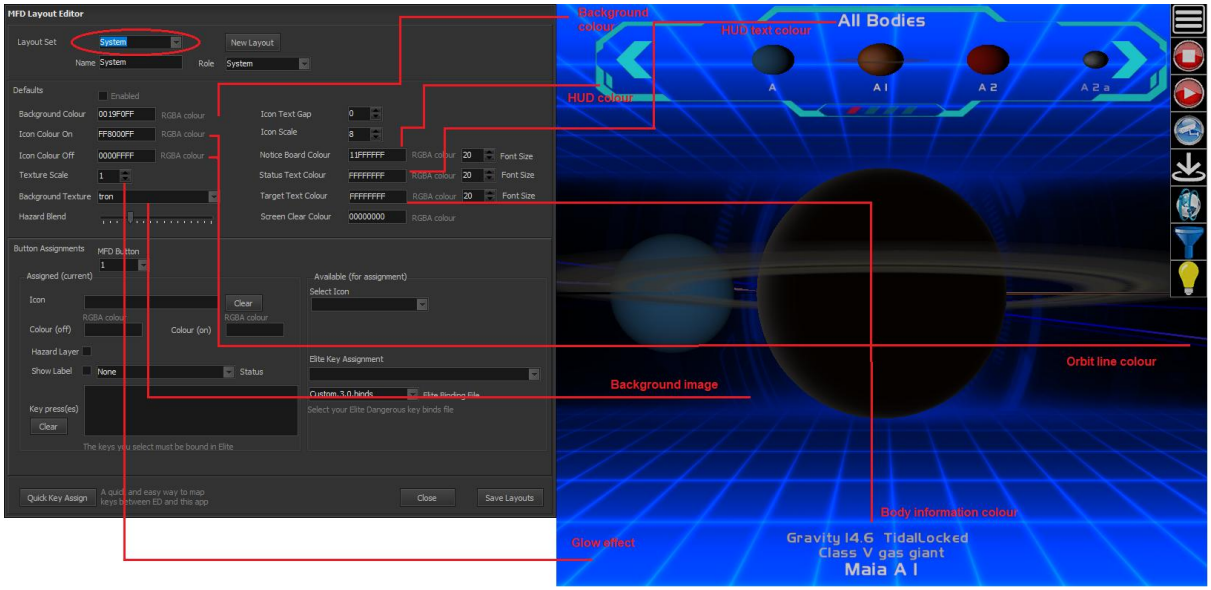
The **MFDCougar.INI** file contains the following tweakable settings:

- **IdleTimeOut** (default 20 seconds) – Number of seconds with no user input before switching to demo mode. A value of 0 (zero) will disable demo mode.
- **MouseWheelDir** (default 1) – When switched to -1, the direction of the mouse wheel and HUD arrows are reversed.

Layout Editor

The Layout Editor (available from the main form of the application) is used to customise the appearance of Cougar Display screens. The **System** role is used to customise the Orrery.
Icon Colour On/Off attributes are used to set the orbit line colours.

The following image depicts the relationship between the layout editor user interface and the Orrery rendering.



It's worth noting that there are several supplied background images for use with the orrery.

1. "Tron" – the default background, uses both an image and a pixel shader to generate the background. The value for the background colour shown above is 0019F0FF.
2. Selecting "Milkyway_edge" has no pixel shader associated. A suggested background colour would be 0808080B. A low alpha (transparency) value of 0B is recommended for this image.
3. Selecting "none" clears the screen to the configured background colour.
4. Selecting "nebula" selects the nebula uber shader.
5. Selecting "swirl" selects the swirl uber shader and is coloured via the background colour attribute.

Demo Mode

When the Orrery detects no user input for a configurable amount of time (20 seconds), it will switch the camera into “demo mode”. This mode cycles the camera between random views of the system.

User Interface (Menu)

The menu area of the Orrery displays the HUD, controls the time scale, camera, orbit lines, system body colour scheme and lighting effects.

Toggle the menu (and HUD) visibility

Pause time

Cycle between different time modes

Reset the camera to default position

Display the Orrery with a bird’s eye view

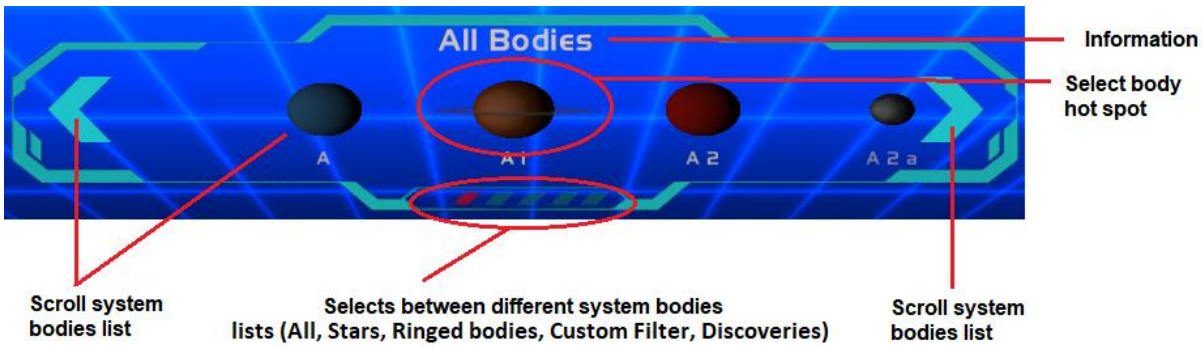
Toggle orbit lines on/off

Display custom filtering menu

Toggle lighting effects

User Interface (HUD)

The HUD area of the Orrery controls the list of system bodies displayed. When clicking the body hot spot, the Orrery will focus on the selected body.



Position the mouse over the HUD to activate menus (left/right scroll arrows, list buttons). Use the mouse wheel to cycle between the listed system bodies.

HUB Buttons

The buttons switch the information display between five modes: All, Stars, Ringed bodies, Custom Filter and Discoveries.

These ‘filters’ allow for the quick selection of stars or bodies and is especially useful in systems with a large number of system bodies.
The **Discoveries** filter allows you to quickly identify those bodies which have just been discovered (in this session) and Elite is reporting as never been discovered. Once you change system, these discoveries will no longer show as ‘new’ and will be included amongst all the regular filters.

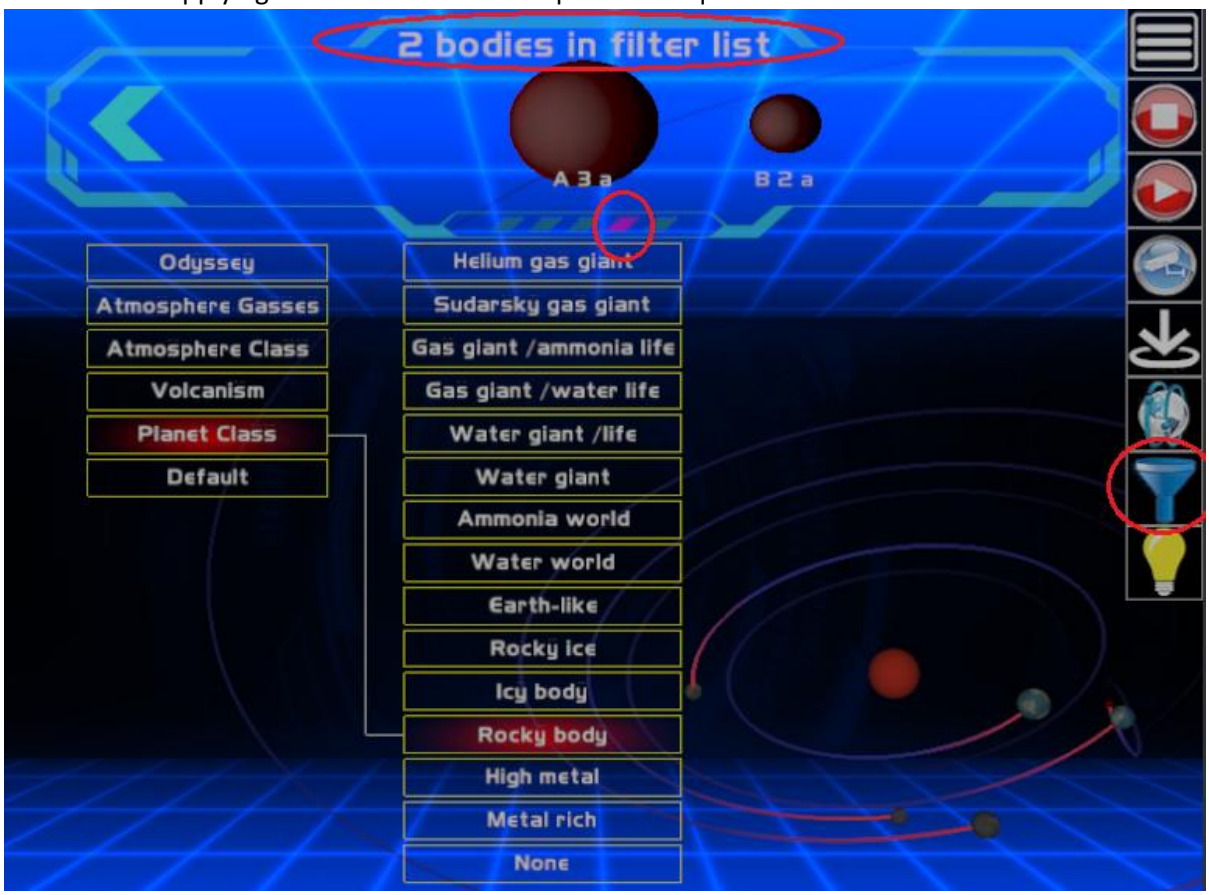
Newly discovered bodies are now dynamically added to the Orrery display and are highlighted with a blue diamond above them.

EDSM

If the Orrery displays “(via EDSM)” then the source of data to build the Orrery originated from EDSM; this means that EDSM contains more system body information than your own local database. You should explore the current system more to discover these planets for yourself. Each time you visit a system, a copy of the EDSM json is saved locally (to prevent spamming EDSM). If you discover new bodies not previously found in EDSM, then the locally cached EDSM file will be updated with the discovery and the cache entry will not expire until EDSM contains more system bodies than the cache (at which point the downloaded json file will replace your local copy).

Custom Filter

The custom filter is selected from the filter menu item on the right side of the Orrery screen.
This filter is a method to control which system bodies are available in the HUD for selection, as well as method of applying a colour scheme to the planets for quick visual identification.



The custom filter has 6 options of which four can be simultaneously active (Planet Class, Volcanism, Atmosphere Class and Atmosphere Gasses). The Odyssey filter selects landable planets with a thin atmosphere.
All planet bodies default to a grey colour, unless the body meets the filter criteria and either red, blue or green component colour is then added to the final colour output.

When the menu is linked (active) Atmosphere Gasses draws the planets in a blue colour
When the menu is linked (active) Atmosphere Classes draws the planets in a blue colour
When the menu is linked (active) Volcanism draws the planets in a green colour
When the menu is linked (active) Planet Classes draws the planets in a red colour
When default is active, a fixed colour scheme is used

In the custom filter image above, the Planet Class menu is active (not linked to the bottom item "None") and thus "Rocky Body" is the active filter. This filter will apply to all planet bodies which have the rocky ice attribute in the journal's scan data, and thus will be drawn in red.

Note: When multiple menus are active at the same time, the colours will combine (red and blue make purple, for example). When a menu item denotes major or minor, then the intensity of the added colour will vary between the two options (by 0.5).

Note: Stars are always drawn with a colour corresponding to their star class.

When **default** is activated, the planet colours are computed as follows:
For the atmospheric composition, the amount of the gas (percentage) is converted into an intensity (value from 0.0 – 1.0).

Gas	Red	Green	Blue
Sulphur		Intensity	0.5 * intensity
Carbon		Intensity	0.3 * intensity
Methane	0.4 * intensity	Intensity	0.3 * intensity
Hydrogen	Intensity		
Helium	Intensity	Intensity	
Nitrogen			Intensity
Oxygen		Intensity	
Water			Intensity
Argon	intensity		0.7 * intensity
Neon	Intensity		0.7 * intensity
Iron	Intensity	0.7 * intensity	
Silicates	0.5 * intensity		0.5 * intensity

Note: Earth-like and Water worlds are drawn with a fixed texture. Ammonia worlds have a green bias, and water worlds have a blue bias to the colours.

Texturing

When rendering planets, the textures drawn on the body sphere are an accumulation of several textures representing gas giants, rocky, icy or metallic surfaces.

To add variation, there are several variations of the base planet type:

Base type	Number of variations
Metal	1
Icy body	3
Rocky Icy body	2
Rocky body	3
Helium	1
Gas Giants	5

Controls

Mouse

Hold the **left** mouse button and move the mouse to change the camera view.
Left click on a system body, or the body hot spot in the HUD to focus the camera.
Hold the **right** mouse button and drag the mouse to rotate the camera around the focus.
Scroll the mouse **wheel** to zoom, or select a new hot spot HUD body.

Tip: Hold the W key whilst holding the left mouse button to ‘fly’ around the system bodies.

Keyboard

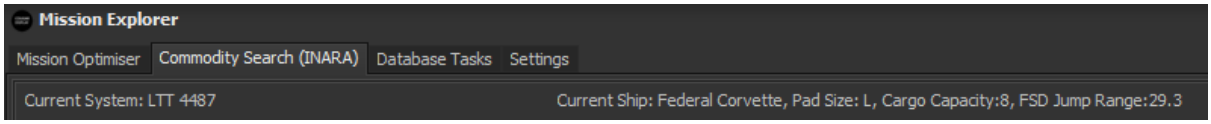
W/S Zoom the camera long the direction of the current view
Q/E Rotate the camera in the Y plane around the focus.
A/D Rotate the camera in the XZ plane around the focus.

Note: Q/E and A/D perform the same actions as the right mouse button with drag.

Tip: Use the UI menu to **reset** the camera after using the **bird’s eye view**.

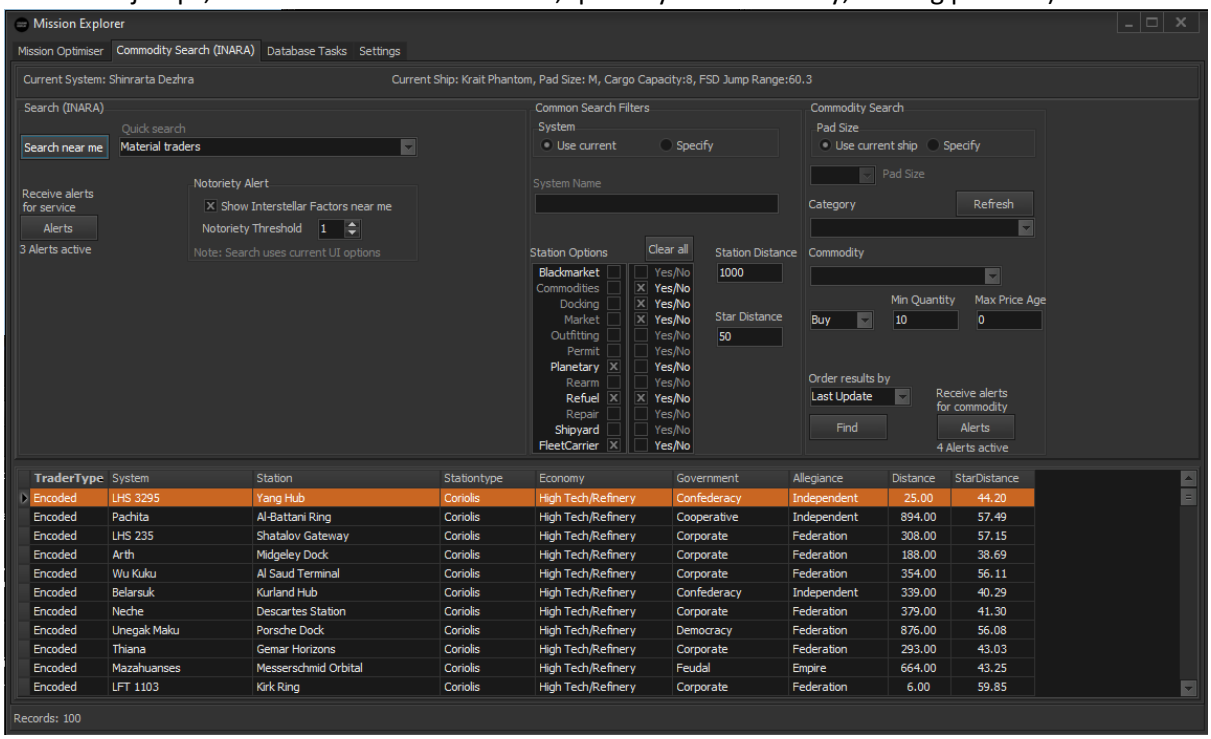
Mission Explorer: Commodity Searching / BGS Search / Station facility Search

The **Commodity Search** tab contains live in-game information for the current system, your ship and its FSD jump range.



The **Mission Explorer** allows a commander to:

- Commodity search - results are ranked for the most efficient use of time (considers number of jumps, distance of station from star, quantity of commodity, landing pad size)



Screen organisation

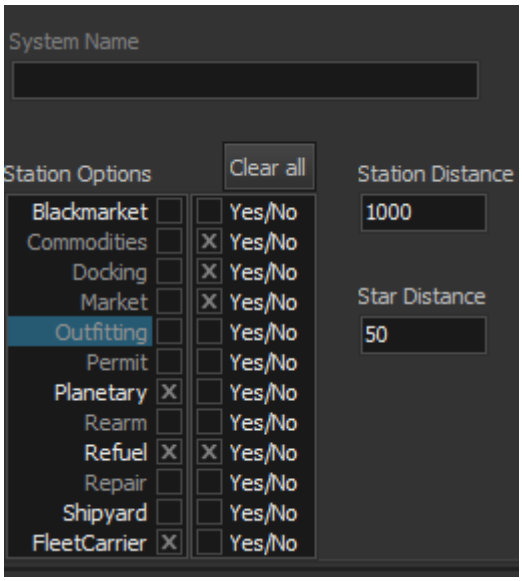
The Commodity Search screen is vertically split into two sections. On the right are the options for commodity searching and on the left are options applicable to general system searching.

Common Search Options

These options are additional criteria to reduce (filter out) the number of stations returned in a given search.

Of particular interest is the reference system and planetary requirements. **Common Search Options** and **Landing Pad size** are always applied to every search. You should always check that these filters contain the values you need.

Note: The current UI values (common search options) are also used by the mission route optimiser when it runs in the background.



Station Distance

This value is the distance from the main sequence star. By using this option, you are reducing the travel time needed to visit stations in the list.

Star Distance

This value is the distance from your current location (or reference system). By using this option, you are reducing the jump of jumps needed (and reducing travel time).

Station Options

There are two sets of vertical check boxes.

The right set toggles the option value to yes or no; the left set (Blackmarket, Shipyard, etc) is a toggle to include or exclude the option from the search -

- If unticked, then the **Yes/No** option is disabled as the option is not used.
- If ticked, then the **Yes/No** option on the right becomes available. **Yes** means that the option must be available at the station, **No** means the option must not be available (or required).

The default S**tation Options** are setup so that INARA search results perform the following filters: exclude planetary bases, exclude fleet carriers, and include refuel (blackmarkets and shipyards are not considered and ignored).

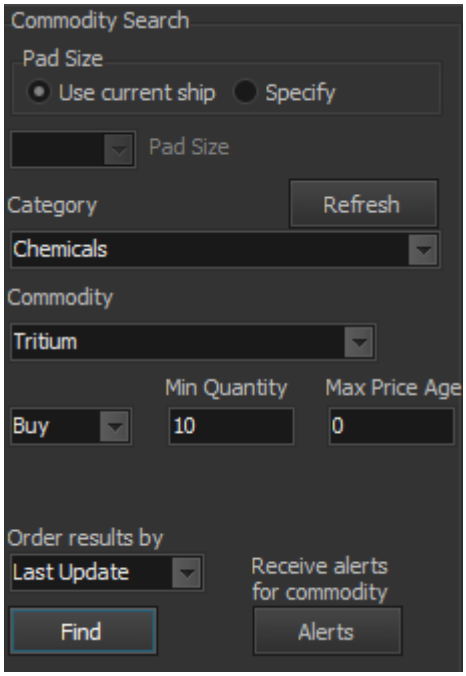
Note: Due to the closure of EDDB.io, only the following **Station Options** are implemented in the INARA replacement commodity searching: Blackmarket, Refuel, Planetary and Shipyard.

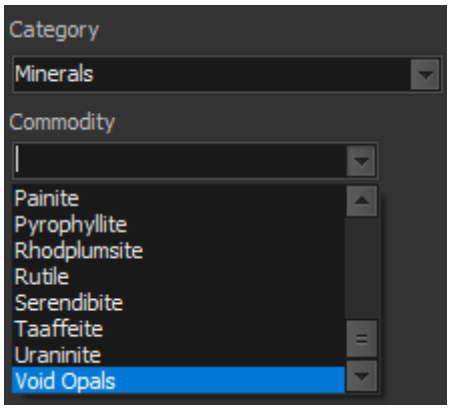
Commodity Search

The Commodity Search options are heavily dependent upon the **common search** options.

- Select the category (or all) and the corresponding commodity item from the lists.
- Select to either specify the landing pad size, or use the current ship size.
- Select the minimum quantity to search for.

Clicking **Find** will execute a query using INARA.cz with the results ranked according to number of jumps, distance from arrival star (ie supercruise travel time), refuel available, Shipyard, pad size, etc.





Click the **Find** button to execute commodity searches. This search is always a “search near me” variety and is also subject to the **Common Search Filters** such as landing pad size. Results from this search are ranked in order of convenience. For example, a system several more jumps away could be ranked lower (more favourable) than a closer system because the system has a short super cruise, or perhaps an outpost rather than a Coriolis station.

Rank	SystemDistance	Supply	UpdatedMins	Updated	SystemName	StationName	distance	StationType	pad	planetary	refuel	shipyard	blackmarket
-4.00	12.51	276	8	8 minutes ago	Heveri	Wiberg Hub	8.00	Ocellus	L	☐	☑	☑	☑
-3.00	30.99	664	43	43 minutes ago	Momoirent	Jahn Dock	34.00	Coriolis	L	☐	☑	☑	☑
-3.00	24.44	190370	44	44 minutes ago	V902 Centauri	Berners-Lee Hub	29.00	Coriolis	L	☐	☑	☑	☐
-2.00	45.96	249	35	35 minutes ago	Huangais	Cooper Orbital	51.00	Ocellus	L	☐	☑	☑	☑
-2.00	52.66	342963	60	1 hour ago	Beta Trianguli Aus	Lunan Hub	228.00	Orbis	L	☐	☑	☑	☐
-1.00	58.20	12246	30	30 minutes ago	Geb	Bryusov Orbital	458.00	Coriolis	L	☐	☑	☑	☐
-1.00	52.66	262232	53	53 minutes ago	Beta Trianguli Aus	Hutchinson City	305.00	Orbis	L	☐	☑	☑	☐
-1.00	25.28	410	60	1 hour ago	V886 Centauri	Tanner City	958.00	Ocellus	L	☐	☑	☑	☑
-1.00	53.10	2171	60	1 hour ago	Sirsir	Mitchell Hangar	274.00	Coriolis	L	☐	☑	☑	☐
0.00	22.94	509	120	2 hours ago	Fong Wang	Schachner Dock	58.00	Coriolis	L	☐	☑	☑	☐
0.00	18.83	1566	480	8 hours ago	Komovoy	Silves Dock	6.00	Coriolis	L	☐	☑	☑	☐
0.00	24.44	214389	1440	1 day ago	V902 Centauri	Olsen Hub	39.00	Ocellus	L	☐	☑	☑	☐
0.00	21.96	318	120	2 hours ago	LHS 317	Northrop Port	215.00	Coriolis	L	☐	☑	☑	☐
0.00	38.05	2444	50	50 minutes ago	Nagnatae	Howard City	817.00	Ocellus	L	☐	☑	☑	☐
1.00	34.06	256085	840	14 hours ago	PW Hydrae	Freycinet Station	29.00	Orbis	L	☐	☑	☑	☐

When a rank is negative, it means that the algorithm has boosted the result in the rankings (good availability of the commodity when buying) and when the rank is above zero, then the algorithm has penalised the result as it’s a sub-optimal result.

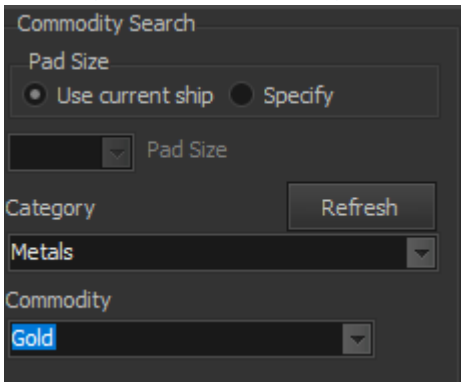
The weighting factor is adjusted as follows:

- For each 500 ls of supercruise, +1
- Planetary bases, +5
- Medium pads not available at the station, +1
- No refuel facility, +1
- Supply > 2 * required quantity, -1
    - However, if the last update is less than 1 hour old, -3
- Last update less than 30 mins, -3
- Last update less than 1 hour, -2
- Last update less than 24 hours, -1
- Last update, +1 for every day above 24 hours

The goal of the weighing factor is to either boost or reduce the rank of the station in the search results list. Planetary bases take a long time to land at, so they are penalised the most. Commodities most recently updated are boosted in the ranks the most and conversely penalised for stale updates.

For more in-depth content and help with searching using the Mission Explorer, visit http://cougardisplay.site/missionlog_tutorial.html

Ensuring Commodities are Up-to-date



Click the **Refresh** button to download from INARA and populate the **Category** and **Commodity** list boxes on the **Mission Explorer**'s **Commodity Search** panel. The **Mission Optimiser** makes use of the commodities cache for supply missions. The refresh can be automated via **Scheduler Settings** on the **Settings** tab.

INARA Commodity Search Logging

When testing/debugging commodity prices, it's often useful to check the application results against the INARA web site to see how the ranked results compare.
To log API calls to LOGS folder, add the following to the **MFDCougar.ini** file:

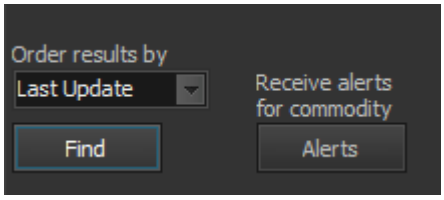
```
[INARA]
DebugQuery=1
```

Commodity Search Alerts

When exploring the Elite galaxy, sometimes it would be nice to be alerted when jumping into a system that a particular commodity is nearby. With **Commodity Alerts** you can set up to 20 alerts that will visually show you which commodity is available at a particular system/station.

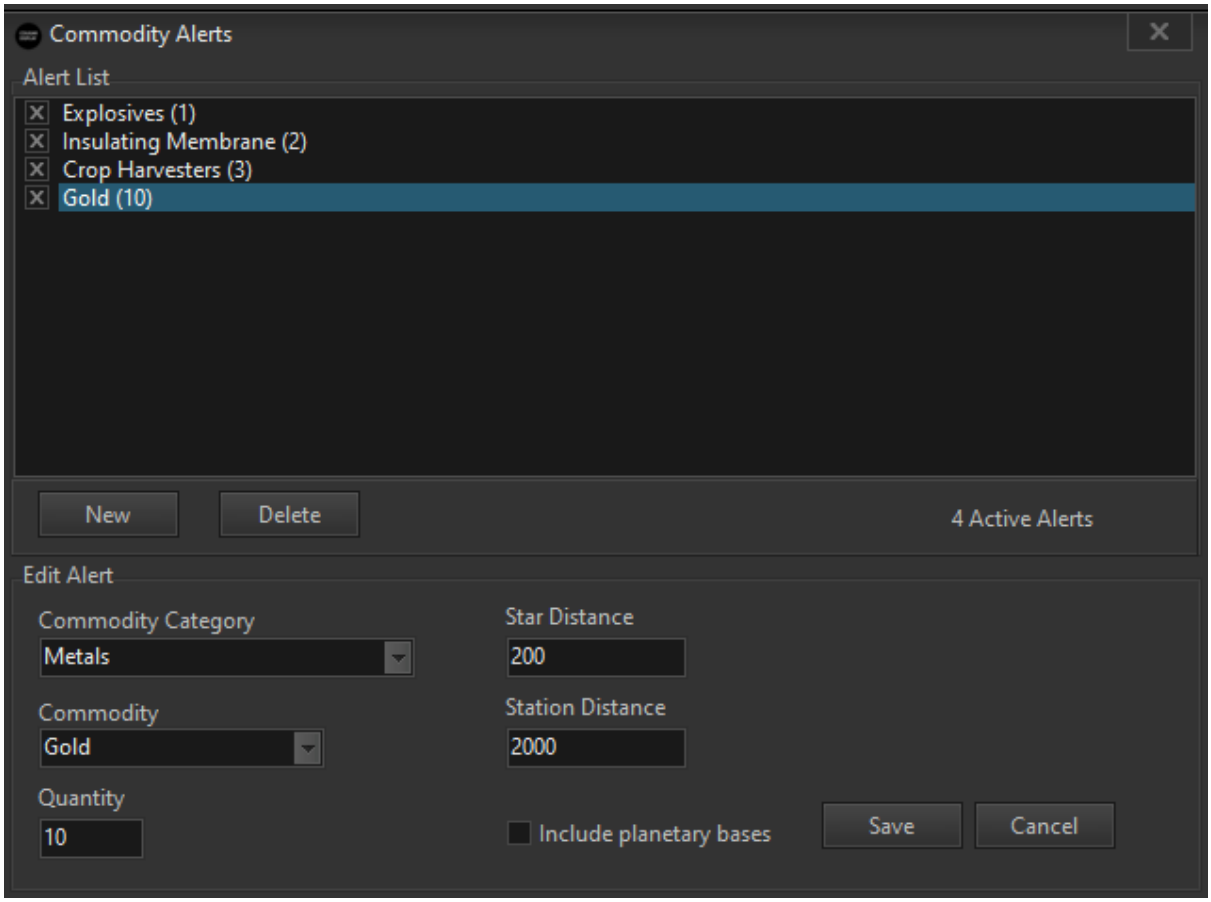
There are different styles of display/notification, so that you can see at a glance how many "jumps" away the commodity is, or have a custom panel flash to grab your attention.

Click the **Alerts** button on the **Commodity Search (INARA)** tab to setup **Commodity Alerts**.



Setting Up Alerts

Click the **New** button to add a new alert.



Complete the details on the **Edit Alert** panel and click **Save**.
The list of currently active alerts is shown in the **Alert List**.
Select an alert from the **Alert List**, and click **Delete** to remove the alert.

Alerts with a check mark on the **Alert List** are "active" and will trigger an INARA commodity lookup on the next FSD jump. Deselect an alert using the left mouse click.

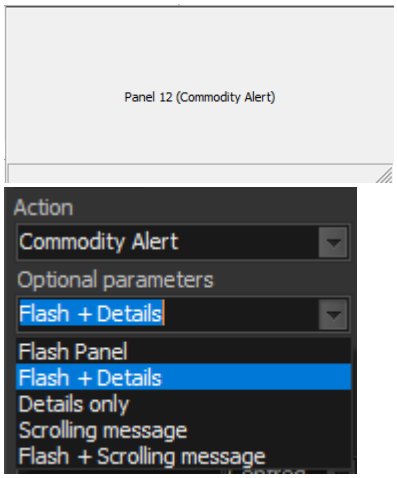
When the **Commodity Alerts** page is closed, a **CommodityAlerts.json** file is written to the application root folder. This file is processed when a FSD jump is completed (either by the CMDR or Fleet Carrier).

Triggered alerts are notified in the Custom Panels, and also in the **Flight Console Noticeboard** area.

Note: Text in the noticeboard area is limited to 30 characters, so not all of the message will be visible (particularly when the system or station names are long).

Custom Panels

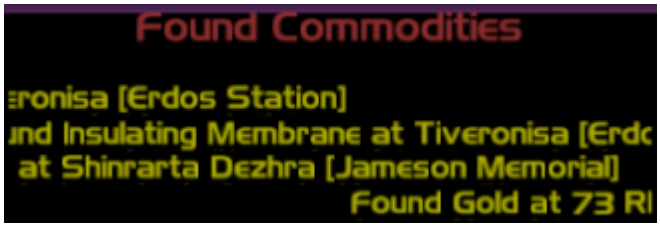
Commodity Alerts are shown in the **Custom Console Designer** as “Commodity Alert” panel.



Using the **Optional Parameters**, select the “display style” that suits you best and display one or several alert panels.

The **Details** style is a table showing the commodity, number of jumps to the system and the station to go to.

The **Scrolling message** style is a ticker-tape display style which wraps the text to fit the display area.

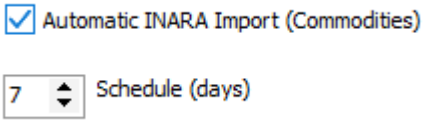


When **Details** and **Scrolling message** styles are combined with **Flash** style, then the title area will flash/blink.

Troubleshooting

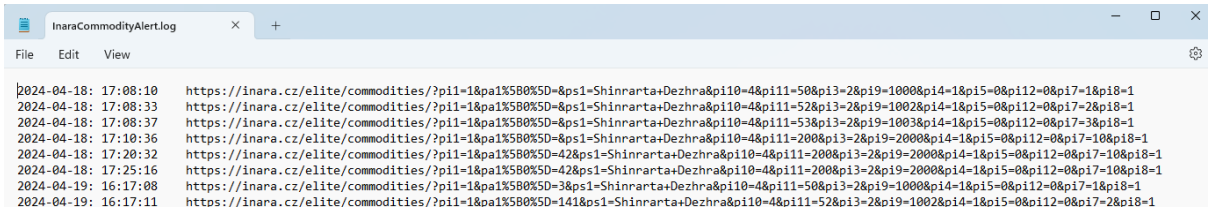
Alerts are only triggered when a jump is made into a system.

For the alerts to work, there must be a **Commodities.html** file in the application/INARA folder. This file is usually (automatically) downloaded from INARA website on a scheduled basis (see **Mission Explorer**, **Settings** tab).



If the file is missing, the alerts mechanism will attempt a fresh download.

To view the INARA queries generated, view the **InaraCommodityAlert.log** file in the application/**Logs** folder.

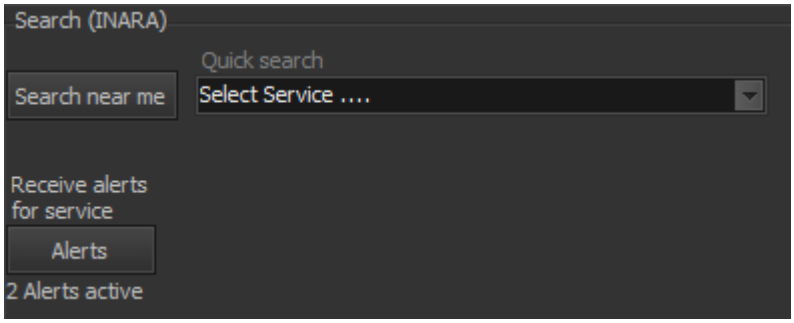


Logging is enabled by the following entry in the **MFDCougar.ini** file:

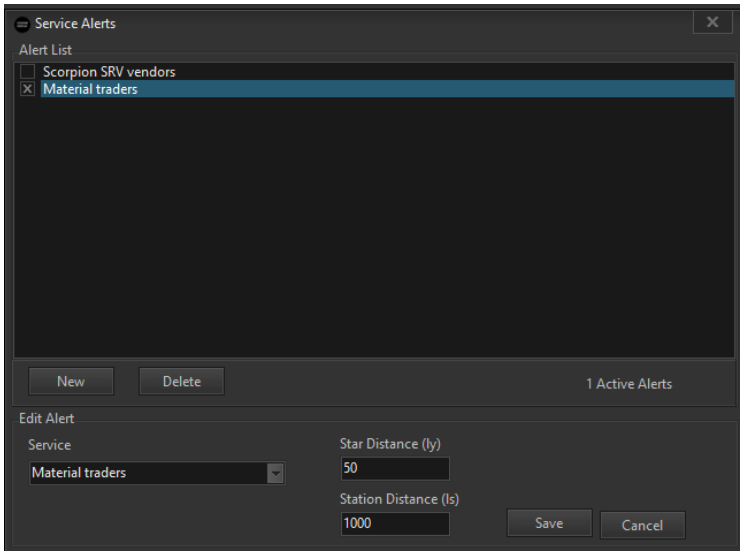
```
[INARA]
LogQuery=1
```

Service Alerts

INARA Service Alerts are special searches executed against the INARA website to locate services of interest to the CMDR.
The **Mission Explorer** (Commodity Search (INARA)) tab has the **Search near me** button to locate services of interest. The search performed using this button has the benefit of caching the results to a grid for easy sorting and viewing – but it can't alert the CMDR in real time whilst playing Elite.



The INARA service search can be extended into alerts – much like the **Commodity Alerts**. When configured, the alert will display a scrolling message in a Custom Panel when a system is within the jump range as specified in the alert panel.

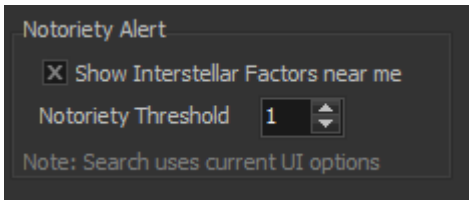


The available list of INARA service alerts are:

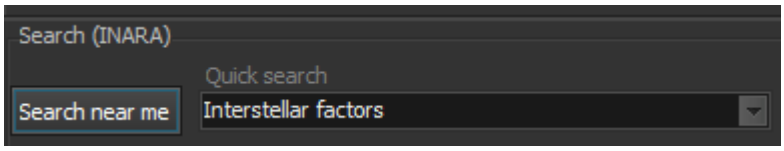
- Material traders
- Technology brokers
- Interstellar factors
- Fleet carrier outfitting
- Scorpion SRV vendors
- Anarchy systems (outbreak)
- Damaged stations
- Compromised NAV beacons

Notoriety – Interstellar Factors Search

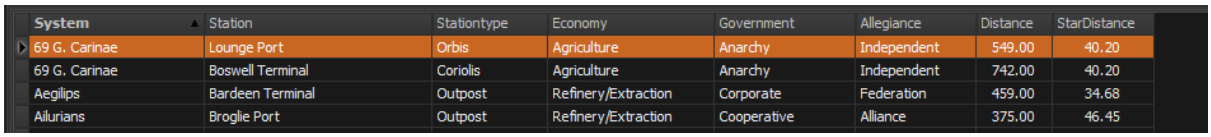
When the CMDR's notoriety level is on or exceeds the **Notoriety Threshold** trigger, if the **Show Interstellar Factors near me** is checked, then the **Mission Explorer** user interface automatically executes an INARA search for interstellar factors near the current position.



Searching uses the current UI search filters (**star distance**, **station distance**, etc) and is the same set of results as if the **Quick Search** had been changed to Interstellar factors.



The Results grid is updated to show the top ranked results from the INARA search.

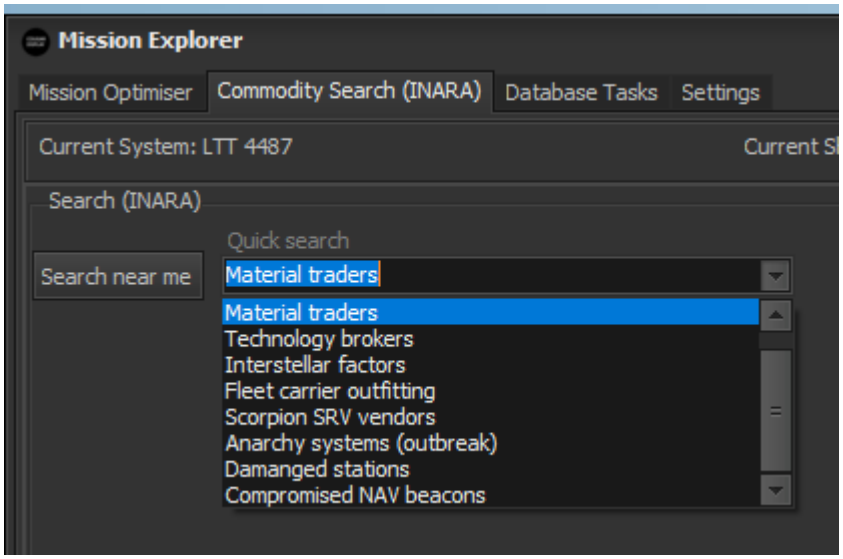


System	Station	Stationtype	Economy	Government	Allegiance	Distance	StarDistance
69 G. Carinae	Lounge Port	Orbis	Agriculture	Anarchy	Independent	549.00	40.20
69 G. Carinae	Boswell Terminal	Coriolis	Agriculture	Anarchy	Independent	742.00	40.20
Aegilips	Bardeen Terminal	Outpost	Refinery/Extraction	Corporate	Federation	459.00	34.68
Ailurians	Broglie Port	Outpost	Refinery/Extraction	Cooperative	Alliance	375.00	46.45

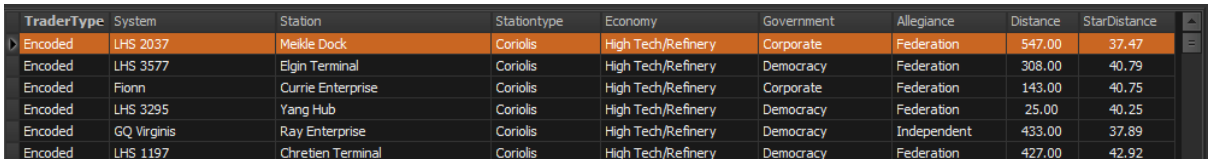
Note: To clear a bounty, you often need to travel to an Interstellar Factor greater than 20 Ly away.

System / Station / BGS Searching

The left hand side of the Mission Explorer module integrates with INARA web site and returns the results of various system wide searches near the commander's current location (or specific location as specified in the **Common Search Filters**).



Depending upon the chosen search, the results are shown in the table below (which can be sorted on any column). Common search filters (system/station distance) are applied in most cases.

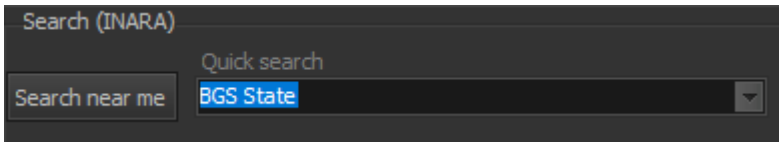


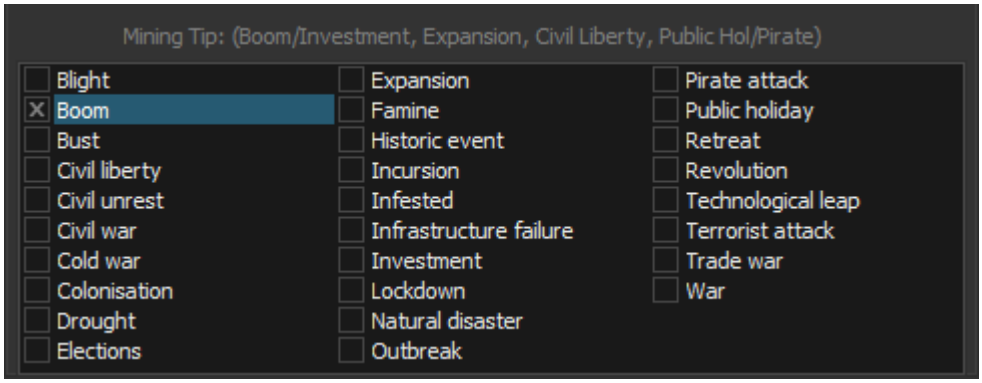
TraderType	System	Station	Stationtype	Economy	Government	Allegiance	Distance	StarDistance
Encoded	LHS 2037	Meikle Dock	Coriolis	High Tech/Refinery	Corporate	Federation	547.00	37.47
Encoded	LHS 3577	Elgin Terminal	Coriolis	High Tech/Refinery	Democracy	Federation	308.00	40.79
Encoded	Fionn	Currie Enterprise	Coriolis	High Tech/Refinery	Corporate	Federation	143.00	40.75
Encoded	LHS 3295	Yang Hub	Coriolis	High Tech/Refinery	Democracy	Federation	25.00	40.25
Encoded	GQ Virginis	Ray Enterprise	Coriolis	High Tech/Refinery	Democracy	Independent	433.00	37.89
Encoded	LHS 1197	Chretien Terminal	Coriolis	High Tech/Refinery	Democracy	Federation	427.00	42.92

BGS State Search

It's quite often useful to search for particular systems in a combination of BGS state (good when selling Void Opals, etc).

Select BGS state from the **Quick Search**



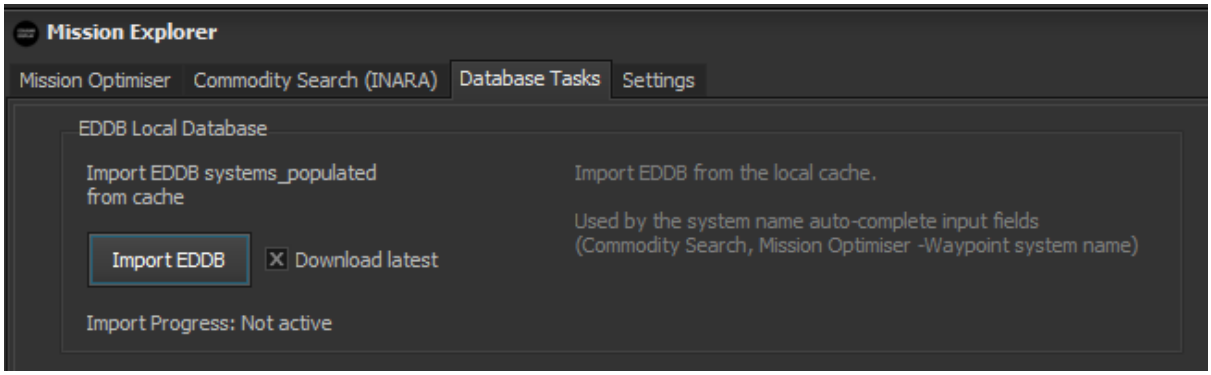


Select a combination to search (INARA supports a max of 4 or 5 states).

Database Tasks

Import EDDB

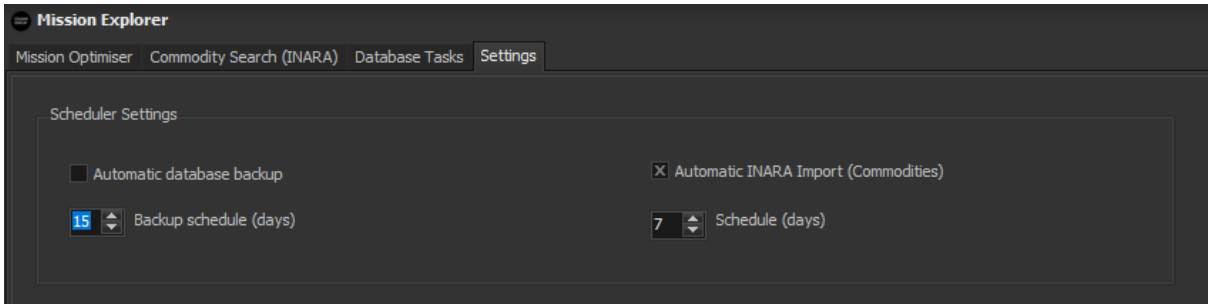
This section downloads the last available EDDB systems_populated JSON file from CougarDisplay and imports them into the database.
This database is used by the **Commodity Search** tab's **system name** edit box to auto-complete typed system names.



If the *EDDB\systems_populated.json* file exists it will be overwritten if the **Download Latest** check box is ticked; otherwise the current file will be imported.

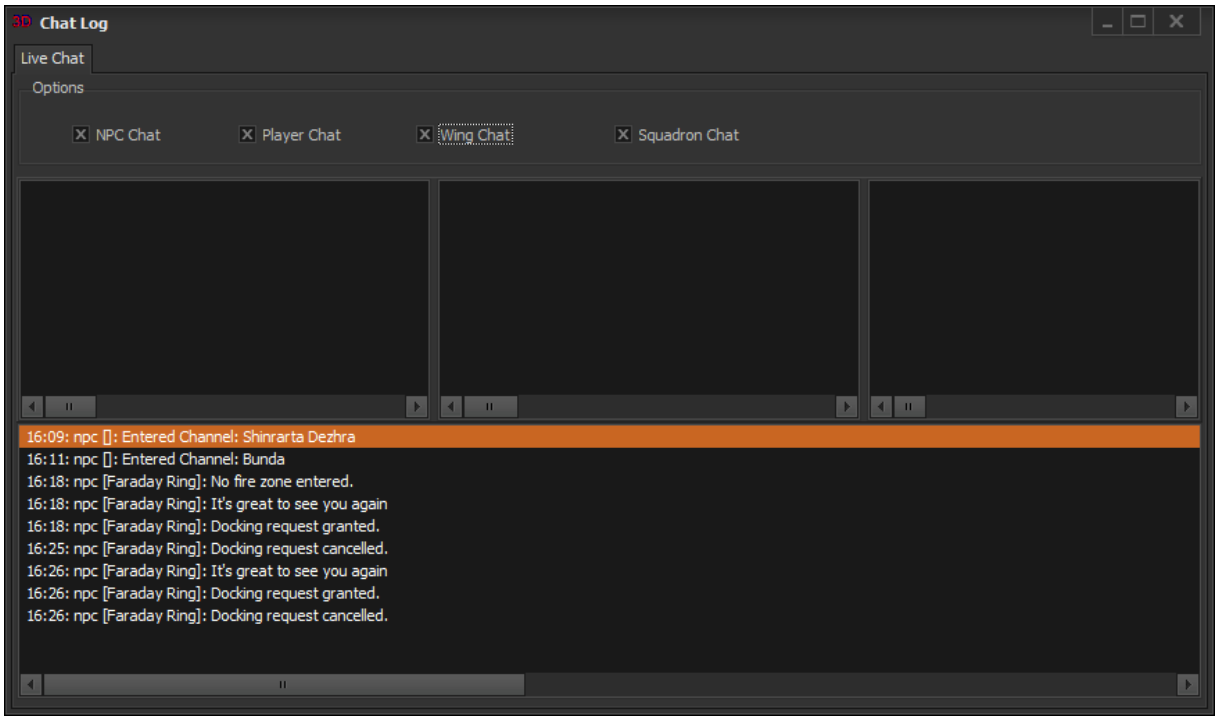
Settings

Settings allows you to specify the frequency for automatic backups and INARA commodity updates.



Chat Viewer

A window with views to separate and display NPC chat, player chat, wings and squadron messages. Simplifies the display of messages be resizing the space for viewing them to make them more readable.

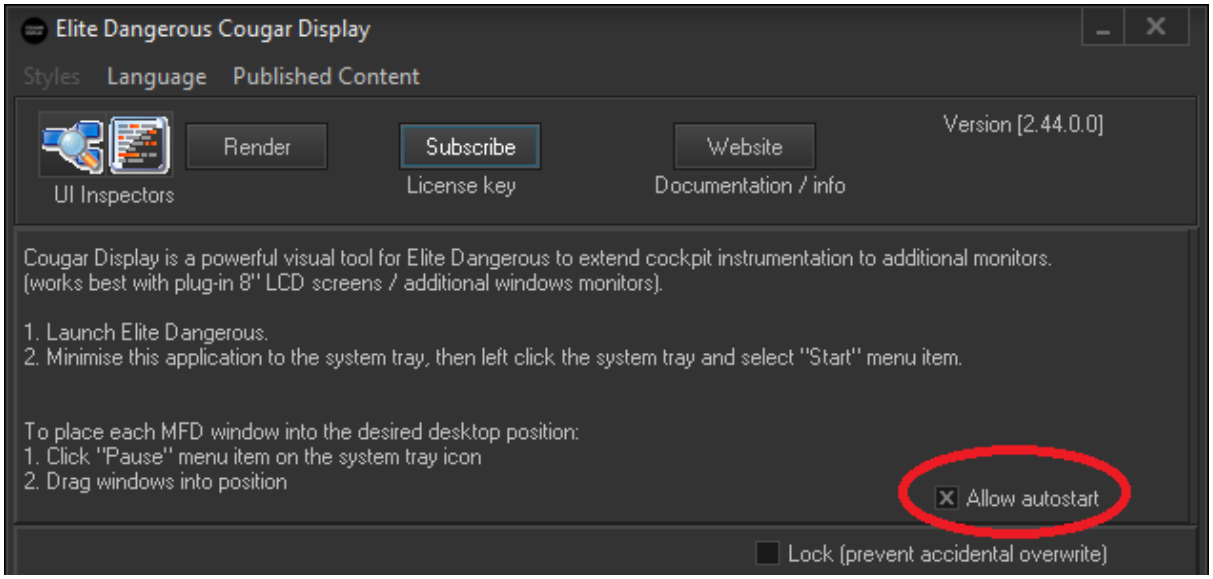


Each of the panels can be resized, and using the check boxes, panels can be hidden. If the window is not needed it can be closed or minimised to the task bar.

Auto Starting Cougar Display

The usual start-up sequence for Cougar Display is to run the application, wait for the download files to complete (INARA), minimise to task bar and then click START from the system tray icon.
After several successful launches, it's expected that the MFD window positions and sizes will be where you want them and won't be changed – the application considers the configuration "complete". To record this fact, a text file (autostart) is written to the application folder containing the count of unchanged launches. When the count is greater than 2, the application will auto start the MFD displays (and minimise to system tray).

To enable this feature, set the **Allow auto-start** checkbox.



If the MFD windows are repositioned, their dimensions altered, or the autostart file deleted – then the autostart file is reset (the count is reverted to 0).
The autostart file resides in the MFDCougar root folder.

NOTE: Auto start is disabled when Cube Sim Support is enabled (via MFDCougar.ini)

MFD Joysticks

Cougar Display was originally designed around the use of Thrustmaster MFD Cougar joysticks; these devices have a grid of 20 buttons and 4 rocker switch buttons in the corners giving a total of 28 buttons (each rocker can be in two switch positions).

Later versions will allow for custom MFD solutions with a different sized array of buttons.

Polling Interval

Joystick polling can be controlled via a configuration change in the **MFDCougar.ini** file

```
[Joysticks]
PollInterval=100
```

Enable Joystick Button Press (whilst docked)

By default, Elite Dangerous doesn't allow joystick input when docked and thus Cougar Display also disables MFD Cougar button presses (assigned to the two main display windows) when docked. This can be overridden with the following change:

```
[Joysticks]
ButtonPressWhileDocked=1
```

Disable the Rocker Buttons

By default, Cougar Display uses the 4 Thrustmaster MFD rocker buttons for specific purposes (such as gamma/exposure). These buttons are not currently programmable or re-assignable. These buttons can be disabled from within the main Cougar Display windows with the following change:

```
[Joysticks]
CougarRockerSwitches=0
```

Language Support

The main application now includes a Language menu which is dynamically built when the application starts. Switching language will refresh the open dialogs and force the visual user interface elements to redisplay their text in the chosen language.

Supported Languages

English, French and German are supported and a **language.ini** file is shipped in the **\Languages** folder. To support each language a subfolder correlating to the ISO 639 country needs to be present with the language.ini file in it.

Adding a new language

Additional languages can be supported by looking up the ISO 639 language code from the supplied **ISO 639-1 codes.csv** file and create a sub folder for it, eg French ISO 639 code is fr; German ISO 639 code is de; Italian ISO 639 code is it.

Once the application starts, the sub folders are translated into a descriptive language name (via the **ISO 639-1 codes.csv** file) and the language menu items are built.

Limitations

Due to the controls taking a fixed amount of space, some languages may struggle to fit within the allocated space for the caption within the control.

Elite Legacy Mode

Since Odyssey update #11 (April 2022) Elite's journal files have been created with a new file naming convention.
The **MFDCougar.ini** file contains a **[Journal]** section to control the naming convention used to decode player journal files. For Odyssey/Horizons 4.0, use **PreFilterJournalFormat**=1 and legacy Elite Clients (version 3.8) should use **PreFilterJournalFormat**=0.

Use the **Elite Client** option to select which journal file format to use.

Note: ensure **Lock (prevent accidental overwrite)** is unchecked.

When Elite 3.8 is detected in the journal file the application will append **(legacy)** to the commander's name to distinguish legacy mode from the regular Elite 4.0 commander.

Troubleshooting

This section deals with some common messages which the application may display from time to time. For more information on other common troubleshooting areas, such as Key Binding process see http://cougardisplay.site/keybinds.html and http://cougardisplay.site/troubleshoot.html#bindings

Difficulty positioning/Resizing Windows

Some users have reported that using pause/start to bring up the MFD Cougar windows can sometimes crash the application. The exact cause of this is unknown but it's suspected that it's to do with the OpenGL window creation process at the driver level. A seemingly simple thing such as displaying/hiding a window title is actually a heavy weight operation. When you use the pause/resume menu items you are destroying and recreating the window handle upon which the OpenGL contexts rely.

The very first time Cougar Display is executed the MFD Cougar windows are displayed with their windows titles showing – allowing you to move, size and place the windows on your desktop.

Upon subsequent launches, a file called "autostart" will have been created and the presence of this file prevents the window titles from showing.

Thus, to force the MFD Cougar windows to show the title area delete the "autostart" file and restart the application.
Alternatively, single threaded mode can be enabled to achieve the same thing – edit **MFDCougar.ini** and add the following into the [**debug**] section:
SingleThread=1
Note: It is a limitation of single threaded mode that minimising the application main form to the task bar will also minimise the MFD Cougar windows (i.e. minimising the application main form should not be done).

Message

Journal file missing

Since Odyssey update #11 (April 2022) Elite's journal files have been created with a new file naming convention.
The **MFDCougar.ini** file contains a **[Journal]** section to control the naming convention used to decode player journal files. For Odyssey/Horizons 4.0, use **PreFilterJournalFormat**=1 and legacy Elite clients (version 3.8) should use **PreFilterJournalFormat**=0.
In case another naming convention is introduced, disable pre-filtering with **EnablePreFilterJournal**=0 as this may help the application pickup journal files.

Default values are shown here:

```
[Journal]
PollInterval=1000
PreFilterJournalFormat=1
EnablePreFilterJournal=1
```

Message

Status.json file missing

Please check (in the registry) that the path specified for Elite is valid.
`Computer\HKEY_CURRENT_USER\Software\BionicBytes\EDCD`
The **`SaveGamesFolder`** key should contain a value such as `C:\Users\xxxx\Saved Games\Frontier Developments\Elite Dangerous`

Ensure Elite Dangerous is running and you have logged into the game and loaded your CMDR.

Message

Check date/time. System clock out of sync

Please synchronise windows time (to the internet) which you can do from the system tray clock on the task bar. Your PC's BIOS battery may be weak and the computer's internal clock is out of sync with the internet – resulting in either a slow or fast PC date/time. This message is displayed if a journal event is detected with +/- 30 second difference between the journal time stamp and the current Windows time.

Upgrading Cougar Display

Periodically there are updates for Cougar Display. You can upgrade automatically by following the prompt during start-up (which launches **CougarAutoUpdate.exe**), or download the various packages from the webpage.

Advantages of the Auto Update

Cougar display persists quite a bit of information and these could get deleted/lost if a manual upgrade is performed.

The main areas to preserve are:

- Database (Explorer_DB.mdb) – containing:
  - Missions – information about the current missions within Elite
  - Fleet Carriers – information about the current status / crew / services and finance
  - Bookmark Notes – information you may have entered about each system (Commander's log)
  - EDSM – Locations cached from various FSD jumps
- EDSM\bodies – a cached copy of the systems visited and looked up via EDSM
- Cache\Loadout.json – The current ship build and configuration
- Cache\<cmdrname>_shiplocker.json – Odyssey materials
- Cache\<cmdrname>_backpack.json – Odyssey materials
- speechEvents.json – customised COVAS text
- SAPI.ini – Speech synthesis and COVAS/NPC/Police voices

By running the **Auto Update** utility in the **current folder**, the above files and database tables are preserved.

Files not preserved during auto update

It's also equally important to know that upon every update, some files may get replaced. Whilst some of these are regenerated, or can be imported, others are simply replaced and any custom values will be overwritten.

Custom Files which may get overwritten:

- MFD layout customisations with key bindings and colours (**MFD_Layout.xml**)
- MFD icon layout templates (**MFD_Iconset.xml**)
- Layout of the (two) MFD displays (**MFD.ini**)
- Global application values (**MFDCougar.ini**)
- Audio – (**Nasa_script.txt** and **ATC_script.txt**)
- Console panels (*.mfd)
- SpeechEvents.json

Auto update will make a backup of the above files before replacing them.

Manual Updates

It is possible, of course, to extract files from any downloaded MFD Cougar package. The application can run from any location because all the files are self-contained. However, it is recommended to always update/run from the same original location to preserve as many of the customised files as possible (see above for the main areas to preserve).

A manual upgrade involves the following steps (as a guide line)

1. Download new package and extract into a temp folder
2. If a new database file (DB\Explorer_DB.mdb) is in the ZIP package – follow the Database upgrade procedure (**before** replacing any files)
3. Copy/paste 'core' folders and files from the temp location into the master folder
4. Open and inspect **INI** files, and merge changes
5. Open and inspect **XML** files and merge your current values/attributes into the new files, then copy the new files into the master folder
6. Ensure you have either merged or copied all files in the package

Note: 'Core' files are considered to be binary files (.wav, .exe, .dll, .xga, .glsl) and would be the contents of the following folders (root folder, bmp, FX, Shaders)

Database Upgrade

To preserve as much as possible from the database, it's recommended to export data before replacing the database with the upgraded file.

Manual Database Export

Launch the application with a command-line parameter. The easiest way is from a command prompt, or create a shortcut and then modify the properties.

```
MFDCougar.exe /E
```

The application will start up, export database files as XML into an export folder and then exit.

Manual Database Import

To import the exported database files, simply place them into an Import folder (or rename the export folder). When the application starts up, it detects the presence of the files and imports them (deleting the import files as it does so).

After Upgrading

After performing either a manual or auto upgrade, you should consider performing the following tasks:

1. Use the Commander's Log and re-import the journal files
2. Use the Mission Explorer and re-import the files from INARA/EDDB
3. Re-run the **Quick key bindings** in the **Layout Editor** on the main application form (see page 5 – Key Bindings).

Note: The re-importing tasks are only needed if the database file (**Explorer_DB.mdb**) was replaced during upgrade.

Beta Testing Elite

From time to time, Frontier release alpha or beta versions of Elite for the public to test. In order for Cougar Display to read the journals of alpha/beta Elite Dangerous, it's necessary to change the Beta flag in the **MFDCougar.ini** file so that "JournalBeta.xxxxx.json" or "JournalAlpha.xxxx.json" files are read.

The typical location for journal files from Elite is:

```
C:\Users\xxxxx\Saved Games\Frontier Developments\Elite Dangerous
```

Change the **Beta** flag to 1 for an Alpha or Beta Elite Dangerous.

```
[EDClient]
Title=Elite - Dangerous (CLIENT)
Beta=1
BetaJournalSuffix=alpha
```

Ensure the title of the Elite window (Options → Graphics → Display → Fullscreen → Windowed mode) match, and that the Suffix is set to either "alpha" or "beta" accordingly.

Note: New versions of Elite typically have new key bindings files. Be sure to setup ED:CD to map key strokes using custom.**X**.0.binds, where X would be 3 (Horizons) or 4 (Odyssey), etc.

The typical location of the Elite key bind files is:

```
C:\Users\xxxxx\AppData\Local\Frontier Developments\Elite Dangerous\Options\Bindings
```

Uninstalling

To completely remove Cougar Display from your system two actions are required:

1. Delete Cougar Display folder
2. Remove registry key

Registry

The registry is used to store user preference values such as the number of MFD displays, license key, Elite folder locations, etc.

It is not necessary to remove the registry keys to troubleshoot/fix application issues. Default values are added to the registry by the application.

The entire key can be deleted and is located in the following location:

`Computer\HKEY_CURRENT_USER\Software\BionicBytes\EDCD`

Use **regedit** to edit the registry and remove the **BionicBytes** key.

What's New

This section lists the main new features and changes.

Version 2.47.0.0

Improvements

- New Stronghold carrier animation on Docking Requested event
- Custom Panels – System Market imports and exports
- Layout Editor:
    - New icons for additional status2 bits (supercruise over charge, supercruise assist, npc crew active)
    - Ability to adjust button/icon text positions

Version 2.46.0.0

Improvements

- New fleet carrier animation on Docking Requested event
- Updated Odyssey suit and weapon engineer recipes
- Workaround issue in journal misidentifying station types (for colony shipyards in deployment)
- Commander's Log:
    - fix: Commodity search using Specified System name
    - Stronghold Carrier option added
    - POI Signal source display and filters

Fixes:

- Docking animation - Bernal stations types incorrectly drawn as mega ship
- Custom panels - altitude gauge when negative

Version 2.45.0.0

Improvements

- Colonisation support
    - Layout Editor - new Colonisation screen showing Cargo to transfer
    - New docking images for Colonisation Ship
    - Workaround issue in journal misidentifying station types (for colony shipyards in deployment)
- FSD Jump - Hyperwarp layout displays warning (in yellow) for neutron cone emitting stars (white dwarf/neutron)

Fixes:

- NASA speech SAPI voices
- MissionAccepted mission journal event processing (could sometimes cause a crash)
- Fixed star class description for some star class types
- Air Traffic Control
    - fix to audio missing/stuttering
    - fix to render scrolling ATC text

Version 2.44.0.0

Improvements

- Autostart

Version 2.43.0.0

Improvements

- New ship - Cobra Mk V
- Languages - selected language stored in registry

Shipyard Fixes:

- Mandalay shipyard callouts
- Mandalay shipyard textures
- Cobra mkiii shipyard image clean up
- Orca shipyard callouts
- Background axis line positioning

Version 2.42.0.0

Improvements

Odyssey Ascendancy Support:

- PowerPlay 2.0 support
- Holoscreen hacking journal entry

Fixes

- Application is now Windows 11 DPI aware
- Shaders recompiled for compatibility with Intel Iris GPU
- Minor updates to journal events

Version 2.41.0.0

Improvements

- Type-8 Transporter added

Fixes

- Fixed ship images for Viper Mk IV

Version 2.40.0.0

Improvements

- Python Mk II added

Fixes

- Fixed an issue processing journal json whilst adding scanned systems into database

Version 2.39.0.0

Improvements

- Service Alerts
  - Configure INARA service alerts in Mission Explorer
  - Alerts added to Flight Console notice board
- Custom Console for Service Alerts

Version 2.38.0.0

Improvements

- Commodity Alerts
  - Configure up to 20 alerts in Mission Explorer
  - Alerts added to Flight Console notice board
- Custom Console for Commodity Alerts
  - Select display alert styles: Flashing, Details, Scrolling message

Version 2.37.0.0

Fixes

- F16 MFD Cougar/Cubesim. Fixed issues detecting joysticks on some systems
- Minor fixes to journal state logic when starting application

Improvements

- Fleet Carrier
  - Can specify which carrier type to be displayed
  - Shows flight officer (captain role) crew member name
- Layout Editor

  Flight console - can select image type (blueprint/non-blueprint)

  Fleet carrier console:
  - select carrier type to be displayed
  - specify fleet carrier type text
  - select fleet carrier type text colour

Version 2.36.0.0

- More ship blueprint images - for use with custom consoles – ship panel
- Fix for Cubesim/F16 MFD Cougar joystick detection

Version 2.35.0.0

- Faster text to speech (ATC Chatter) particularly with CereProc voices which are CPU intensive. Rendering a voice to memory is now asynchronously performed in another thread.
- SpeechEvents - fix to docking denied event
- Fixed OpenGL Core porting issue - Supercruise, printing system name whilst also rendering ship target text in analysis mode
- Docking - Station landing pad fix for megaships/FleetCarriers

Version 2.34.0.0

- 64-bit MFD Cougar Display
  - Core application compiled for 64-bit.
- Fixed WingWing button mapping
- Custom Consoles - Hyperwarp console could cause some panels to not print text

Version 2.33.0.0

- Support for WingWing hardware (similar to Thrustmaster Cougar MFD)
- SpeechEvents added - SupercruiseDestinationDrop and SupercruiseDestinationDropHighThreat

Version 2.32.0.0

- OpenGL 4.5/4.6 Core Profile
    - Removal of all OpenGL “Compatibility profile” and depreciated functions.
    - New OpenGL rendering engine

Version 1.32.0.0

- Odyssey update #15 support
    - Journal changes - SupercruiseDestinationDrop event added, some others added extra information
    - Additional events added to the cusomisable file "SpeechEvents.json" / Speech Editor which adds "ThargoidInterdicted" and "SupercruiseDestinationDrop" (threat alerts)
- New Features
    - Can now select any OpenGL texture and display on a custom panel.
    - New Elite game icons added to available textures
        - Logos of Takada, Supratech, Remlock, Manticore, Kinematic Armaments
        - Ship modules and weapons, eg AFMS, Cargo, limpet, Experimental Weapon Stabiliser, FSD Booster, FSD Interdictor, Fighter Hanger, Flight Assist
        - Hull Reinforcement, Passenger Cabin, Vehicle Hanger, Shield Generator, Shield Cell Bank, Scanner, Refinery
        - Power Plant, Thrusters, Life Support, Power Distributor, Heat Sink, Point Defence, Chaff, Multicannon, Beam Laser, Missile Rack, Mining Laser

Version 1.31.0.0

New Features - Mission Explorer: Search for the nearest

- Material Traders
- Technology Brokers
- Interstellar Factors
- Fleet Carrier Outfitting
- Scorpion SRV Vendors
- Anarchy Systems in outbreak
- Damaged Stations
- Compromised Nav Beacons
- Systems with specific BGS states

Version 1.30.0.0

Depreciated functionality

- EDDB.io site is closed. All EDDB dependencies are removed.

New Features - Mission Explorer: **Route Optimisation**

- now uses INARA for source/supply missions
- now takes cargo capacity into account (as well as jump range)

Version 1.29.0.0

Depreciated functionality

- EDDB.io site is closed

New Features - Mission Explorer:

- EDDB - download and import the last available systems_populated.json from EDDB.
- INARA commodity search integration
- "Commodity Search" tab renamed as "Commodity Search INARA" to show its reliance on INARA.cz providing data
- Debugging option added to MFDCougar.ini

Changes:

- Remove legacy code due to EDDB integration
- Improved application start-up times (EDDB database/import no longer required)
- Reduced memory footprint due to EDDB tables

Version 1.28.0.0

Depreciated functionality

- EDDB.io site announces closure, which affect the following:

  - custom panel - system imports/exports on FSD jump
  - Mission Explorer
    - commodity searching
    - Route optimisation builder - commodity searching
    - EDDB database import - station/state searching UI
    - Auto-complete text prompt when typing in system names (Mission Explorer: commodities search; Waypoint system names)

Fixes:

- LayoutEditor:
  - KeyBindings - detects for the presence of StartPreset.4.start file as well as
    Main application:
- Main Application:
  - Commander's log tab - "Show" button now restores previously minimised commander's log window
  - Air Traffic Control tab - joystick axis selection
    - volume level corrected for axis values between -1.0 and 1.0.
    - UI improved so that joysticks with the same GUID are further differentiated by name. This fixes issue that selected joystick is not found/used for volume control.
- Mission Explorer:
  - EDDB - no longer runs the schedule to download EDDB databases. For now, the functionality still exists to manually download (assuming the site is still running).
  - "Commodity Search" tab renamed as "Commodity Search EDDB" to show its reliance on EDDB.io providing data

Version 1.27.0.0

Additions

- Custom Panel Designer

        - Panels are list on the left hand side for easy selection (particularly hidden/overlapped panels)
        - Panel position/dimensions rounded on save - to prevent frame misalignment due to excessive floating point precision

- Custom Panels - Elite Status

  Panels flash when Elite has one of the following states set: Fuel Scooping, Low Fuel, FSD Charging, FSD Hyperdrive, In Danger, Interdiction, Overheating, Silent Running, Low Oxygen, Low Health

  Control the flash rate and whether to flash or not from the **MFDCougar.ini** file:

  [StatusBits]

  Flashrate=500

  Flash=1

Fixes:

- Cargo list - exception decoding cargo manifest
- Custom panels
    - altitude - AUTO title area
    - theme alpha transparency
    - engine/weapon/shields pip graphics drawing in wrong vertical position

Version 1.26.0.0

- Fixes
    - Fixed the re-sizing OpenGL windows when clicking pause/resume system tray menu
    - Mission Explorer - Commodity search - incorrect SQL statement built when primary query is not set to "combined"
    - Console Designer - TextureFX Editor - added missing bmp\custom folder when creating custom textures
    - Joystick slider\Axis enumeration
    - Custom console theme colour
        - incorrectly applying to multiple consoles
        - prevent the theme alpha (transparency) from overriding the panel alpha (allows each panel to set the alpha)
    - Custom Panels - fix the drawing position of Weapons, Engine, shield custom panels when the custom console has a menu title present
- Additions
    - Custom Panels - Titles can be right aligned (default), left or centred. Text is trimmed for white space
    - Galnet RSS feeder panel
        - switches off ATC chatter when reading news
        - Refresh Interval
        - Display Images/full article/summary
        - Assign hotkeys - display next/prev article/ read article / toggle full screen
        - Text to Speech

Version 1.25.0.0

- Support for Odyssey update #14 and Legacy Mode
    - (legacy) added to commander name when Elite 3.8 detected
- Additional status flags (custom panels) for Odyssey update #14

- Odyssey status flags can be tracked in the Layout Editor and assigned to icons
- FSD Hyperdrive Charging state flag added to MFD Layout Editor (allows tracking both supercruise & Hyperjump FSD charging)
- Bug fix – custom icon editor – fixed issues creating a clone of the default icon set

Version 1.24.0.0

- Improvement to minimize application to system tray with multiple active OpenGL context windows
- Custom Icon Editor
  - fixes (Correctly allows multiple textures in the texture sets and adds <icons> xml nodes)
  - Adds ability to switch between icon sets and delete icon sets
- Sample icon press 'beep' sounds added to audio folder
- Layout Editor changes:
  - Radio Chatter - fixed icon assignment bug
  - Hotkey assignments to buttons
  - Play sound when MFD button pressed
  - Latch button presses

Version 1.23.0.0

- Commander's Log
  - Added ability to search EDSM for pristine metallic ringed systems (useful mining tool)
- Fix to Callouts.json - Lakon spelling mistakes corrected
- Journal Events - improved journal reading (prevent skipped events)
- Added support for the following events: FuelReservoir
- Better cargo limpet tracking following changes to Odyssey updates

Version 1.22.0.0

- Odyssey update 13 support
  - Added new journal event NavRouteClear
  - Added DestinationSettlement as a new optional parameter for the MissionAccepted event
  - Added Genuses parameters in SAASignalsFound event

Version 1.21.0.0

- FleetCarrier
  - fix to navigation custom panel
  - Dedicated countdown timer messages
- Odyssey update #12 support
- Fixed SAASignalsScan journal event
- Added FSSBodySignal journal event
- Countdown timer custom panel – displays time in mins/sec

Version 1.20.0.0

- FleetCarrier – various fixes including:
  - Corrected location updates
  - Odyssey updates to finances (rearm,refuel,repair,pioneer tax rates)
- Orrery
  - Odyssey planet filter

- Application launching
    - MFD Cougar – display window titles on first start
    - Single thread – debug option
- Fix - Weapons console
    - Target pilot name and rank
- Improvement – Sending Key presses to Elite
    - Added delay between activating Elite Dangerous Client window and sending key press
    - Key Press option added to Layout Editor button assignments

Version 1.19.0.0

- Custom Icons Pack
    - Share and install custom consoles and icon sets with the ED:CD community
- Themed Colour schemes
    - Quickly apply a set of colours to your consoles
- New Custom panels:
    - Dynamic widgets, graphs and bars, moving things
    - Draw Galaxy, with position marker
    - Custom Background
    - Spinning texture UI materials
- Bulk Key Assign
    - Merging keybinds will use MFD_Iconset_custom.xml (rather than MFD_Iconset.xml) if the custom file exists
- MFDLayout Editor
    - Can select RGB colours from a colour picker
- Odyssey update #11 support
    - Launch SRV – now detects SVR Scorpion
    - Telepresence/Physical crew status flags
    - INI file setting to toggle new Journal file naming convention
- Fleet Carrier fixes
    - fix to track the location when not on board after it jumps
    - added faction tracking to jump event
    - fixed displaying fc roles in conjunction with fc shipyard
- Radar Custom Panel
    - Fix to shader – now draws correctly with panel area
- NASA Script
    - Fixed some wrong actor assignments, plus some de-personalisation of the script.
- Journal Imports (Commander's Log)
    - - Fixed some journal data issues causing exceptions during import (invalid star types)
    - - Fixed some database field lengths too small to contain body scan parent IDs
    - - Fixed an array size limitation of loadout event - max engineer modules too small

Version 1.18.0.0

- Multiple language support (UI static text)
    - English, French, German
- Surface Scanning a planet – optionally opens website to display info
- Improvements to system clock out-of-sync checks
- Improvement to Nav Route Custom Panel

    - Now shows coloured icon next to each system name
- New Custom panels:
    - Ship manufacturer logo (Lakon, Core Dynamics, etc)
    - System faction info (name, reputation, security level, active states)
    - SRV/Scorpion or dynamically switch to/from Ship

Version 1.17.0.0

- Orrery improvements:
    - When approaching planet, displays planet information automatically (no longer runs demo mode)
    - Better planet lighting when selecting to view a specific planet
- Chat Log - Friend chat directed to Wing chat region
- Mission Explorer
    - Mission Optimiser tab now shows active states in the current system alongside the current system name
- FSD Jump - Speech event for hostile factions (warns you if there are hostile factions in the system)
    - hostile factions added to notice board
- Support for custom icons
    - Clones existing icon set for easy customisation
    - Support for separate alpha channel
    - Allows additional icons to be added
- Layout Editor
    - Odyssey (OnFoot and OnFootWeapons) layouts corrected (fixes key assignments not working)
    - MFD_Iconset.xml/MFD_Layout.xml fixed to link IconName attribute with Description attributes

Version 1.16.0.0

- Support for decimal separator in regional settings
- Layout editor max font size increased to 100 pixels
- Odyssey patch 9 support:
    - SRV Scorpion
    - Multi-Limpet controller
- Improvements to Custom Nav Panel:
    - Fixes occasional 'random' background fill
    - Better font resizing between nav plotting
    - Better drawing of the navigation route
- Mission Explorer:
    - Plot Elite Galaxymap now actually performs EDSM lookup (reduces spurious 'undiscovered')

Version 1.15.0.0

- Orrery improvements:
    - New discoveries filter
    - New discoveries tagged with blue diamond icon
    - “Via EDSM” if sourced from EDSM
    - New discoveries added to Orrery

Version 1.14.0.0

- New UI skins and skin persistence

- Odyssey Engineering – inventory update following suit/weapon upgrade

Version 1.13.0.0

- Custom panels for suits/suit upgrades/suit modules/Weapon upgrades and modules
- Odyssey mission list custom panel
- Ship Name, Type and Role custom panels
- Altitude custom panel now either displays fuel or altitude (if orbiting a planet). Select optional parameter to “Fuel or Altitude” for new behaviour.

Version 1.12.0.2

- Bug fixes
- Joysticks – disable rocker switches override
- Joysticks – allow the use of buttons when docked

Version 1.12.0.0

- Reduced memory footprint – deferred EDDB commodities database loading
- 3GB application support – load more TTS voices into memory
- Very hot/cold indicator (Odyssey)
- Suit Grade (Odyssey)
- Application skin – revert back to windows default
- Odyssey update 5 support
- Speech events – 4 new events added (Passenger liner,Police/security,Military,Search & rescue)
- Auto shutdown – controlled via setting
- Bug fixes:
    - Galaxy map divide by zero
    - Galaxy map – incorrectly displays orrery whilst panning the camera (route optimiser)
    - Orrery – now picks the orbit colours from the system layout
    - SAPI.ini – initial voice detection

Version 1.11.0.0

- Flexible joystick assignments to consoles
    - allows assignments of custom made "Cougar" joysticks
    - Move code base to Directx Input
- Odyssey – In Station – now shows list of active missions
- Fuel gauge - better rendering
- Health/Oxygen (On Foot) - truncates health to max of 5 digits
- Suit - now shows shield “on” graphic
- Custom Panel - Odyssey status flags
- bug fixes:
    - Prevent too many voices overloading the audio mixer
    - Status bits - key press - prevent multiple mouse clicks whilst first is still active
    - Speech Events - MissionFailed now retrieves correct phrases
    - Rank (custom panel) - no longer displays Odyssey ranks when in Horizons
    - Rank/progression events are always read from the journal

Version 1.10.0.0

- Speech Event Editor
- Speech synthesis for NPCs now use dedicated voices for Station controllers, security (Police/Military), pirates and everyone else
- Odyssey - updated journal events and statuses following game release

- Simplified and combined Air Traffic Control and Speech Synthesis
- Speech synthesis incorrectly applying pitch and filter effects to COVAS assistant voice.
- Count down timer facility is auto started for Fleet Carrier Jumps and Odyssey book taxi/Drop ship
- Bug fixes:
    - Layout editor (button options – enabled/IconTextColourOff)
    - Altitude (custom panel) wasn't always working as intended
    - Altitude - improved texture rendering (especially with long alt meters)
    - Ship (custom panel) was drawn shifted due to panel title/menu area
- Autoupdate – backs up potentially customisable files

Version 1.900

- Elite Odyssey support
- Can customise the weapons console's active target animation delay (time to return PiPs and shields back to normal)
- Dynamic volume assignment – ATC Radio chatter (via joystick / button press)
- Dynamic Market custom panel (Import/Export commodities)
- Odyssey Player Vitals custom panel (health/oxygen/temperature/gravity)
- Odyssey suits/weapons statistics custom panel (credits,bought,sold,upgrades)
- Hyperwarp custom panel – new animation sequence of starting FSD jump
- FSSMaterials custom panel can display elements by name or symbol. Also materials with less than 50% of max capacity are written in white text
- Mission Log – can now import waypoints from text files (CRLF or CSV)
- Improvements to Hyperspace jump animation – Status text colour uses Flight layout console
- Key Bindings – quick key assignment has been improved (and bugs fixed)
- Layout Editor bug fixed
- In theory – now support custom MFD hardware (other than Thrustmaster) and with different number of buttons.
- MFD Cougar Joysticks can be assigned to displays other than 0 or 1 (future development planned for this feature)
- CubeSim compatibility support
- Autoshutdown when Elite closes

Version 1.800

- Custom panels – automatic menu system (to switch between custom panels) with hot keys
- Custom panels – can assign keypress and background flashes on press
- Landing pad assistant – can now reverse image
- Low fuel warning (hyperspace jump, custom panels fuel gauge)
- Fleetcarrier – Carrier statistics (finance) – human readable currency
- FleetCarrier – uses TargetText font and colour from XML file
- Windows Monitor scaling – no longer affects window size and positions
- FSS / DSS scans – better noticeboard descriptions
- Fix to COVAS assistant (wrong voice)

Version 1.700

- Custom panel – Elite rank/ reputation (combat/trade/exploration and super power progress)
- Supercruise POI no longer lists Fleet Carriers (too many) and uses ## to denote planetary stations, [Mega] for mega ships and # for stations
- Air Traffic Control (Radio chatter simulation) – replaces and enhances NASA radio chatter

- Additional icons to track State (Mass lock/FSD charging/cooldown, Analysis mode)
- Better NPC voice caching (remembers NPC voice for upto 3 minutes)
- Can now choose audio device
- NPC text to speech now uses audio filters
- XML Layout editor – IcontextColourOff for icon text colour
- Heading overlay fixed – can now customise font size and colour
- Docking assistant improvement for fleet carriers/megaship
- Fixes to Commander's Log (volcanism, eccentricity)

Version 1.611

- Flight Console (#0) adds "FSD Jump ready" red/green status lights to assist commanders and let them know that they can now hyper jump (lights for the status flags hardpoints, landing gear, cargo hatch, FSD cooldown and FSD Mass Locked).
- Exploration Panel pre-built and included in the zip package
- Changes to custom panels – MFD_Layout.xml is used to control the colour of some elements on some panels. Nav route panel add (ly) to information.

Version 1.610

- Custom panel designer - better screen user interface handling width/height input
- Custom panel - Materials - now shows materials of body when approaching orbital descent
- FSS Discovery - now automatically updates when all bodies found event occurs
- High gravity warning - planet approach
- Heading overlay window - now maintains Z order when active
- Better multi-monitor support - detecting and repositioning of display windows out of bounds from any monitor

Version 1.600

- DB upgrade - to version 110 - Cmdr name column added to mission table to support multiple commanders.
- Ship targeting - Weapons console: Engines / Weapon pips graphics now retain the correct width:height ratio regardless of the screen size.
- Target information text centred rather than left-aligned
- Right and Left mouse click to bring up menu from taskbar shell icon
- Shipyard, now supports module sell/buy/swap and sorate store/retrive journal events
- (it's not perfect, but the modules are now correctly referenced)
- Up to 8 display windows
- New Feature: Can assign roles to specific MFDs (displays) --> system,galaxymap,custom
- Custom Panels for the display of information (which would otherwise be hard to fit on the main flight/weapons consoles 0 and 1).

The custom panel shipped include:

1. Countdown timer (useful for FleetCarrier owners as a jump timer). Set timer from the Settings tab on the Mission Explorer module.
2. Session time
3. Journal time
4. Limpet count
5. Target Info
6. Fuel Gauge
7. Altitude

8. Heading Info
9. Cargo list
10. Current system name/co-ords
11. Credits / stats:
12. Trading
13. Exploration
14. Combat
15. Mining
16. Docked
17. Crimes
18. Fuel

Version 1.501

- Role assignment change/fix.
    - With 4 MFD Cougar displays active, console #2 is the System Orrery, console #3 is the Galaxy map (previously the otherway round)

Version 1.500

- New Feature: Shipyard display – displays modules with engineering levels when docked.

Version 1.400

- New Feature – up to 4 simultaneous displays
- Master Volume control – added to Speech synthesis tab
- Joysticks: increased max joysticks from 6 to 10
- Solar system - demo mode added and improvements to the camera tracking, system name at the bottom of the screen.

Version 1.310

- POI / Signal Sources / Station services -font size is derived from NoticeBoard attribute on the Exploration Layout
- Journal Reader - bug fix Loadout cache fix
- License file reading - decryption overflow
- Mission Explorer - fixed Current ship display - cargo/pad size/fuel

Version 1.300

- GUI controls for NASA simulation (distortion effect, Morse volume and script timings)
- Support for tablet and touch screen and how Elite gains window focus
- Bug fixes to graphics, EDDB import.
- Refactoring of MFDCougar.INI
- All logging redirected to the Logs folder

Version 1.200

- Orrery added. Galaxy map with Orrery integration.

Version 1.160

- NASA space radio chatter.
- Heading assistant