Gary
Administrator
Hero Member

Posts: 2712

bindED - Copy Elite: Dangerous key bind files (.binds) to VoiceAttack variables
« **on:** November 06, 2016, 03:57:16 PM »

About the bindED plugin

UPDATE - v1.0.0.1 - Fixes file sort issue.

UPDATE - March 7, 2018 - v1.0.0.1 - Licensing file now included.

Edit - By request, source now supplied as a download. I'm sorry, but I cannot provide any type of support or help for the source. Use at your own peril ;)

The bindED plugin was created to copy the keyboard key bind information for Elite: Dangerous directly into VoiceAttack keypress variables. The variables can then be used in conjunction with key press actions within VoiceAttack (the latest betas of VoiceAttack allow for keypresses based on variables).

The idea is to be able to change your key bindings from within Elite: Dangerous and have each change be reflected (semi-automatically) in VoiceAttack, possibly saving some time and/or hair loss.

Thanks to Lavaeolus for starting up the EDMap.txt layout, as well as all those that participate in making plugins all they can be. Also thanks to Fiery Toad (https://www.twitch.tv/fireytoad) for testing, sharing .binds files and moral support.

The bindED plugin is attached, but can also be downloaded here: http://www.voiceattack.com/bindED

Setting up the bindED plugin

Copy the entire bindED folder from the zip file to the VoiceAttack Apps folder (the whole folder, not just the contents). The folder is usually located in C:\Program Files (x86)\VoiceAttack (An installer may be created later to make this easier... for now it's just a zip)

For most, this is all you'll need to do. For others having trouble or want to explore, read on:

Inside the bindED folder, you will find the plugin file itself (bindED.dll), which uses VoiceAttack's Plugin v4 interface... that just means that you will need VoiceAttack beta version v1.5.12.32 or later to use the plugin.

You will also find a file called, 'EDMap.txt'. If you open this text file, you will see that it contains each of the Elite: Dangerous key bind types associated with a numeric code. The code is simply the keyboard key code expressed as a number. For instance, the, 'A' key is 65, 'Z' is 90, 'Escape' is 27 and so on.

This is for English-US keyboards. Your keyboard layout may not line up exactly with this mapping, so adjustments may need to be made (you can edit this file as you see fit, just as long as it follows the KeyName;Code structure that you see inside the file).

If you happen to create a layout that is different or find a mistake/omission in the English-US layout (this is new stuff, after all), please feel free to upload your update here for others to use.

To see the list of key codes, go here: https://msdn.microsoft.com/en-us/library/windows/desktop/dd375731(v=vs.85).aspx
Note that the key codes are expressed in hexadecimal and will need to be converted to integer values.

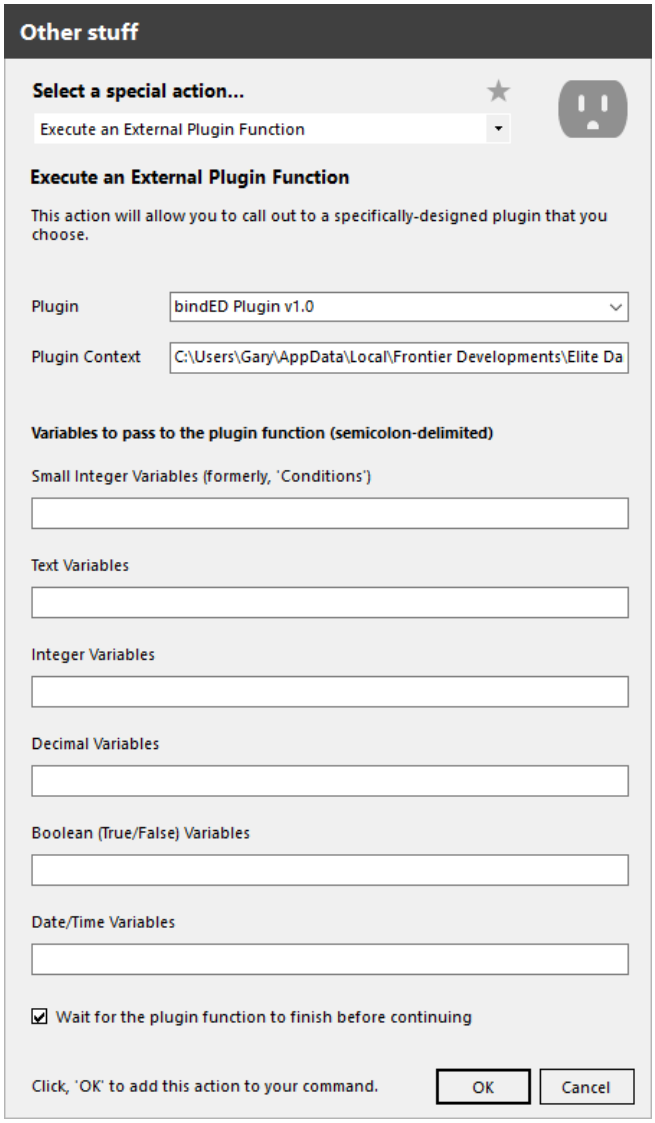
Using the bindED plugin

To use the plugin, you will need to do a few things.

1. First, you will need to turn on plugin support in VoiceAttack. This is done from the Options screen.
2. Next, you will need to create a command in VoiceAttack to call the bindED plugin which (optionally) includes the Elite: Dangerous binds file that you want to use.
   To do this, add a new command and then click on Other -> Advanced -> Execute an external plugin function.
3. In the new plugin action screen, select the, 'bindED' plugin from the Plugin list (bindED will be available if you had turned on plugin support and had installed the bindED plugin correctly).
4. There are two routes you can take here. If you only have ONE binds file, and you want the bindED plugin to try to access your .binds file that is located in C:\Users\YOUR_USER_NAME\AppData\Local\Frontier Developments\Elite Dangerous\Options\Bindings, simply leave the, 'Plugin Context' box blank.
   The plugin will simply grab the NEWEST .binds file it finds in that directory and use it.

   If you have MULTIPLE binds files for different situations, OR, the default behavior indicated above does not meet your need, you can specify which binds file the bindED plugin is to access by typing or pasting the full path to the binds file that you want to use in the 'Plugin Context' box.

   Selecting the, 'Wait for the plugin to finish...' option is up to you. Generally this should run very quickly, but if you have a situation where things need to occur in a very specific order, select that option.



The full path is cut off in the image, but you get the idea ;)

5. Click OK, Done, etc.

The next time that this new command is executed (whether by itself to, 'reload' bindings or from a profile change) the VoiceAttack variables will be updated with what is indicated in the bind file.

How to use the keypress variables once they are updated by the bindED plugin

There are a lot of words here, but as you'll see, there's not a whole lot going on.

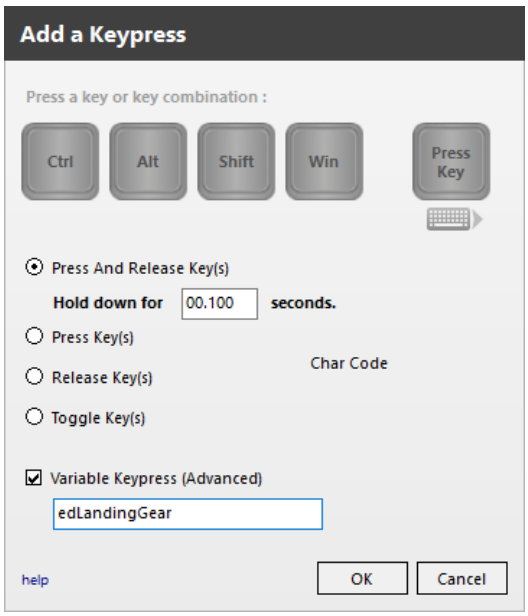
First, you will need to be familiar with the Elite: Dangerous key binds in-game, as well as where those binds are stored in the configuration (.binds) files.
For this example, the landing gear toggle will be used. To toggle the landing gear, let's say the 'L' key is used in-game.
The configuration file (.binds) will store that information in an xml element called, 'LandingGearToggle'. Inside the binds file, you will see the entry looks something like this:
Code: [Select]

```
<LandingGearToggle>
	<Primary Device="Keyboard" Key="Key_L" />
	<Secondary Device="{NoDevice}" Key="" />
</LandingGearToggle>
```

When VoiceAttack encounters this element, it creates a TEXT variable called, 'edLandingGearToggle'. It's just simply the element name with, 'ed' prefixed to it. Then, VoiceAttack notices that the key used is 'Key_L'. VoiceAttack gets the value indicated in the EDMap.txt file and puts it in the, 'edLandingGearToggle' variable (you don't necessarily have to know this... I threw it in there in case you were wondering ;))

At the bottom of this post, you will find a (my) current list of the variables that can be set by the bindED plugin, according to the .binds file that I have.

So, in VoiceAttack (later betas), you can have a key (or keys) pressed based on a variable. At the bottom of the keypress screen, you will see an input box that will allow you to specify what variable to use. Before, you would have, 'L' selected at the top of the screen. Now, you would just type in, 'edLandingGearToggle' (no quotes) in the variable input box:



Your command probably looks like this, currently:

Code: [Select]

```
When I say... 'Landing Gear'
Press L key and hold for 0.1 seconds and release
```

It would now look like this:

Code: [Select]

```
When I say... 'Landing Gear'
Press variable key(s) [edLandingGearToggle] and hold for 0.1 seconds and release
```

If anybody out there wants to create an Elite: Dangerous variable mapping to what is seen in-game, that would be really cool 8)

Advanced stuff and things to consider for those who want even more confusion.

If you are going to be creating profiles for others, it might be a good idea to put some extra steps in your commands in case your users do not have the bindED plugin (either not installed, not configured properly or not initialized). Going to expand on the whole landing gear thing again in this example...

Code: [Select]

```
Begin Text Compare : [edLandingGearToggle] Has Been Set
    Press variable key(s) [edLandingGearToggle] and hold for 0.1 seconds and release
Else
    Press L key and hold for 0.1 seconds and release
    Write '[Orange] Variable keypress not set.  Using default keypress.' to log
End Condition
```

Note that if the variable HAS BEEN SET (not null), the keypress uses the variable. If the variable has not been set, you can have a fall-back keypress, or, just show a warning.

You can have multiple plugin actions to load several (possibly custom) bind files at the same time if you need to. Each subsequently loaded bind file will override the previous ones. You can also do this with one action by separating the bind paths with a semicolon. You may never need to do this, but it's there if you want to do something with it.

Way off the chart... Something else to try that is left over from testing (and left in for anybody that wants to use it) is if you want to load a binds file on plugin initialization (that is, when the plugin first loads (when VA starts up)... not when the plugin is invoked), create shortcuts to the binds files and place them in the same folder as the plugin .dll. Each shortcut is processed in alphabetical order. This is one way to have an automatic, global initialization of variables that do not require the plugin to be invoked. If this doesn't make any sense to you, do not worry ;)

One more thing... if you happen to have a '.binds' file for Star Citizen that has a good variety of key/key combinations (with modifiers... ctrl/alt/shift/win), please let me know and I will create a bindSC plugin.

Variable List Reference

Below is a current list of the variables that bindED can update (and you can use in your keypress actions). Remember, the element within the .binds file is whatever the variable name is below, minus the, 'ed' prefix. So, 'edAutoBreakBuggyButton' below correlates to the 'AutoBreakBuggyButton' element in the .binds file. Note that if no keys are set for the particular element, the variable value will be null (Not

Set). This list can change at any time depending on Frontier, and is only here to give you an idea of what's currently available without having to dig into your .binds file. Good luck, Captain!

edAutoBreakBuggyButton
edBackwardKey
edBackwardThrustButton
edBackwardThrustButton_Landing
edBuggyPitchDownButton
edBuggyPitchUpButton
edBuggyPrimaryFireButton
edBuggyRollLeftButton
edBuggyRollRightButton
edBuggySecondaryFireButton
edBuggyToggleReverseThrottleInput
edBuggyTurretPitchDownButton
edBuggyTurretPitchUpButton
edBuggyTurretYawLeftButton
edBuggyTurretYawRightButton
edCamPitchDown
edCamPitchUp
edCamTranslateBackward
edCamTranslateDown
edCamTranslateForward
edCamTranslateLeft
edCamTranslateRight
edCamTranslateUp
edCamTranslateZHold
edCamYawLeft
edCamYawRight
edCamZoomIn
edCamZoomOut
edChargeECM
edCycleFireGroupNext
edCycleFireGroupPrevious
edCycleNextHostileTarget
edCycleNextPanel
edCycleNextSubsystem
edCycleNextTarget
edCyclePreviousHostileTarget
edCyclePreviousPanel
edCyclePreviousSubsystem
edCyclePreviousTarget
edDecreaseSpeedButtonMax
edDeployHardpointToggle
edDeployHeatSink
edDisableRotationCorrectToggle
edDownThrustButton
edDownThrustButton_Landing
edEjectAllCargo
edEjectAllCargo_Buggy
edEngineColourToggle
edFireChaffLauncher
edFocusCommsPanel
edFocusCommsPanel_Buggy
edFocusLeftPanel
edFocusLeftPanel_Buggy
edFocusRadarPanel
edFocusRadarPanel_Buggy
edFocusRightPanel
edFocusRightPanel_Buggy
edForwardKey
edForwardThrustButton
edForwardThrustButton_Landing
edGalaxyMapOpen
edGalaxyMapOpen_Buggy
edHeadlightsBuggyButton
edHeadLookPitchDown
edHeadLookPitchUp
edHeadLookReset
edHeadLookToggle
edHeadLookToggle_Buggy
edHeadLookYawLeft
edHeadLookYawRight
edHMDReset
edHyperspace
edHyperSuperCombination
edIncreaseEnginesPower
edIncreaseEnginesPower_Buggy
edIncreaseSpeedButtonMax
edIncreaseSystemsPower
edIncreaseSystemsPower_Buggy
edIncreaseWeaponsPower
edIncreaseWeaponsPower_Buggy
edLandingGearToggle
edLeftThrustButton
edLeftThrustButton_Landing

edMicrophoneMute
edMouseReset
edOpenOrders
edOrbitLinesToggle
edOrderAggressiveBehaviour
edOrderDefensiveBehaviour
edOrderFocusTarget
edOrderFollow
edOrderHoldFire
edOrderHoldPosition
edOrderRequestDock
edPause
edPhotoCameraToggle
edPhotoCameraToggle_Buggy
edPitchDownButton
edPitchDownButton_Landing
edPitchUpButton
edPitchUpButton_Landing
edPrimaryFire
edQuickCommsPanel
edQuickCommsPanel_Buggy
edRadarDecreaseRange
edRadarIncreaseRange
edRecallDismissShip
edResetPowerDistribution
edResetPowerDistribution_Buggy
edRightThrustButton
edRightThrustButton_Landing
edRollLeftButton
edRollLeftButton_Landing
edRollRightButton
edRollRightButton_Landing
edSecondaryFire
edSelectHighestThreat
edSelectTarget
edSelectTarget_Buggy
edSelectTargetsTarget
edSetSpeed100
edSetSpeed25
edSetSpeed50
edSetSpeed75
edSetSpeedMinus100
edSetSpeedMinus25
edSetSpeedMinus50
edSetSpeedMinus75
edSetSpeedZero
edShipSpotLightToggle
edShowPGScoreSummaryInput
edSteerLeftButton
edSteerRightButton
edSupercruise
edSystemMapOpen
edSystemMapOpen_Buggy
edTargetNextRouteSystem
edTargetWingman0
edTargetWingman1
edTargetWingman2
edToggleBuggyTurretButton
edToggleButtonUpInput
edToggleCargoScoop
edToggleCargoScoop_Buggy
edToggleDriveAssist
edToggleFlightAssist
edToggleReverseThrottleInput
edUI_Back
edUI_Down
edUI_Left
edUI_Right
edUI_Select
edUI_Toggle
edUI_Up
edUIFocus
edUIFocus_Buggy
edUpThrustButton
edUpThrustButton_Landing
edUseAlternateFlightValuesToggle
edUseBoostJuice
edUseShieldCell
edVerticalThrustersButton
edWeaponColourToggle
edWingNavLock
edYawLeftButton
edYawLeftButton_Landing
edYawRightButton
edYawRightButton_Landing
edYawToRollButton