

USER GUIDE

The Ultimate Voice Assistant



SELECT A TOPIC

[Getting Started](#)

[Create a Plugin](#)

[Design a Plugin](#)

[Share a Plugin](#)

[Command Sets](#)

[Commands](#)

[Command Options](#)

[Command Lock](#)

[Simultaneous Events](#)

[Change to another Command Set](#)

[Actions & Action List](#)

[Key Action](#)

[Mouse Action](#)

[Move Mouse Relative from Center Action](#)

[Move Mouse and Recenter Action](#)

[Move Mouse Start to End Action](#)

[Determine the Mouse Scaling Factor](#)

[Play Audio Action](#)

[Wait Action](#)

[Launch Website/App Actions](#)

[Send Text Action](#)

[Change To Action](#)

[Execute Command Action](#)

[Load Plugin Action](#)

Emily Interface

The Emily Interface is used to execute spoken commands and to access the Plugin Designer and Plugin Manager.

A button is provided to manually toggle between Activate or Standby Modes. Changing modes can also be accomplished by saying the Activate/Standby Commands: “[Get Ready Emily](#)” and “[Standby Emily](#)”.

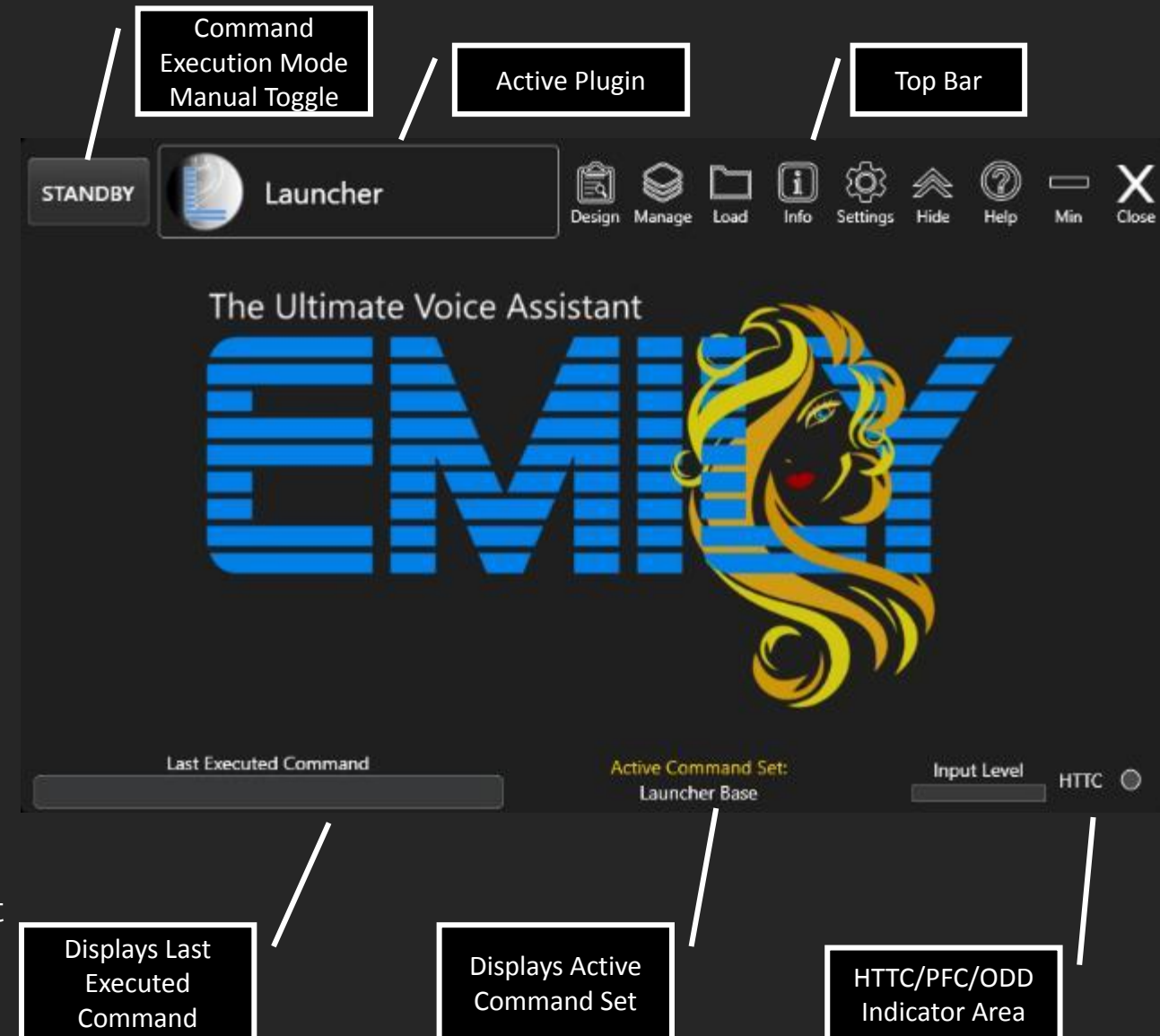
In Active mode, Emily will execute any Command in the **Active Command Set**, including **Global Emily Commands**. In Standby mode, all Commands are ignored, except the Activate Command.

The current Active Plugin and it's icon are next to the Command Execution Button. The Current Active Command Set is shown on the bottom.

Each time Emily is launched, the Launcher Plugin is loaded by default.

The Launcher Plugin is used to:

- Load Games or Apps and their associated Primary Command Set of the Plugin
- Launch websites
- Perform Desktop Tasks



Emily Interface

The Top bar offers quick actions:

Select **Design** to launch the Plugin Designer. The Active Command Set will be loaded for editing.

Select **Manage** to open the Plugin Manager to perform tasks such as duplicating, importing/exporting, or creating a new Plugin. If the **Manage** icon has changed to the **Plugin Update** icon, then select it to open the Manager. The Plugin that requires updating will be marked as such.

Select **Load** to quickly load a Plugin. Loading Plugins can also be accomplished with a spoken command via the Launcher Plugin.

Select **Info** to display all Commands available in each Command Set of the Active Plugin. Keep this up on a separate screen to quickly reference commands.

Select **Settings** to adjust volumes, Global Commands, HTTC/PFC toggles, Confidence Level, and the toggle the On-Demand Design (ODD) feature.

Select **Collapse** to reduce the size of the Emily Interface to minimize screen real estate. Expand to restore the size.



The HTTC/PFC/ODD Indicator Area shows if either of these functions are active. These will be discussed next.

HTTC/PFC/ODD Indicator Area

Emily Interface – HTTC/PFC

Using Hold to Talk for Comms (HTTC) or Push for Commands (PFC) is highly recommended when talking with others.

Select the Settings icon, then select the HTTC/PFC icon.

Select either Keyboard or Controller.

Select a Mode: HTTC or PFC.

Record the Key or Controller Button.

Follow the Setup Instructions to work in conjunction with Discord® or in-game Comms.



Emily Interface – On-Demand Designer

The On Demand Designer (ODD) feature is used when designing a plugin.

When enabled, you can swap between the Game/App and the Plugin Designer with two Commands: “[Open Plugin Designer](#)” and “[Close Plugin Designer](#)”.

This removes the manual effort of pressing Alt+Tab, then clicking the Design Button, then clicking the Save button in the Plugin Designer, then pressing Alt+Tab.

Turn off ODD when not editing a Game Plugin.



Install an Official Emily Plugin or a Community Created Plugin

There are Official Emily Plugins and Community designed plugins available at not cost.

To install, select the Manage button on the Emily Interface to open the Plugin Manager.

Select the **Get More Plugins** button.

Select the Plugin you want and select the **Install** button.

Official Emily Plugins can not be edited, however, they can be duplicated and then edited by selecting the **Duplicate** button.

Once installed, select either **Load** to begin issuing Commands in the Plugin or select **Design** to see how the Plugin is structured.

Commands can be reviewed on the Emily Interface also by selecting the Info button.

Community Plugins have been reviewed and approved by the Emily Software Team.



Setup an Official Emily Plugin

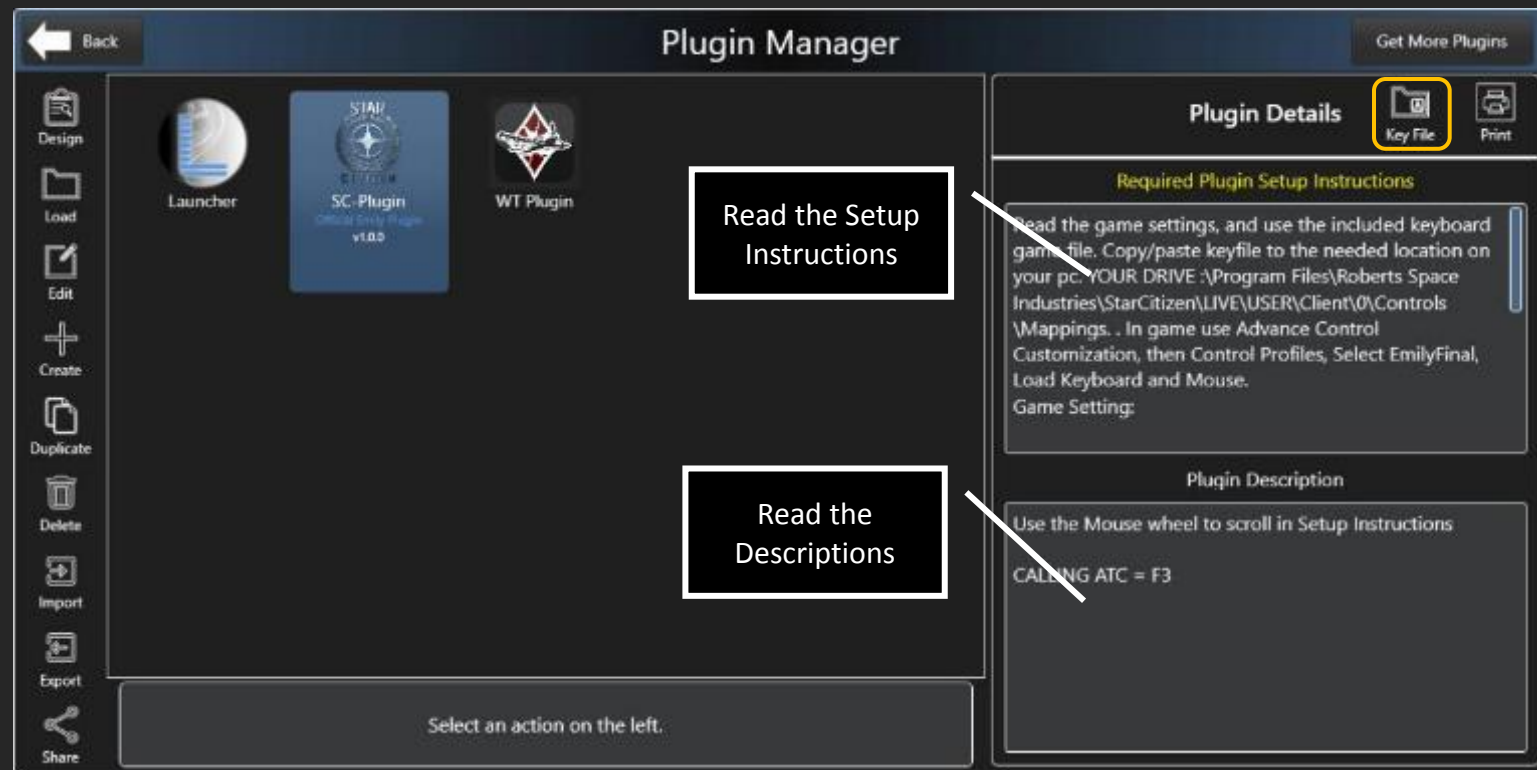
In this example, the “SC-Plugin” includes Required Plugin Setup Instructions.

Adhering to these instructions ensures successful Plugin functionality.

Be sure to read the Plugin Description for important information.

The “SC-Plugin” also includes a Key Binding File, indicated by the **Key File** button appearing on the bottom of Left Bar.

Select the **Key File** button to COPY/PASTE the file into the respective game folder, as instructed in the Setup Instructions.



Import a Plugin

If a friend has created a Plugin, it can be imported. Select the **Import** button.

Likewise, any Plugin you create can be exported.

If you wish to create a Plugin, see the **Create a Plugin** guide.

Plugins are exported as .zip files. The .zip files are not password protected. Do not password protect the .zip file, as this will prevent your friend from being able to import the .zip file.

Select the **Import** button to select the .zip file and import the Plugin.



Learn Commands via Quick Reference

Once a Plugin is loaded into the Emily Interface, you can begin issuing Commands for the active Command Set.

You can also learn the Commands using the Quick Reference.

Select the Info button on the Emily Interface.

Select a Command Set

Quickly filter Commands by Category.

Select a Command and the Description and Critical Notes appear.

Two Print options are available:

- PDF – Detailed List of Commands
- Text – Only Commands by Category

Emily's Listening Mode is presented at the top left and the Last Executed Command on the bottom left. If in standby, the latter is hidden.

Global Commands are also presented



Create a Plugin

Plugins are created in the Plugin Manager.

Select the Manage button.



Create a Plugin

Select the Create button.

Enter and select information about the Plugin

Important notes:

The Primary Command Set is the first Command Set that loads when the Plugin is loaded.

Select the appropriate type of Command Set based on the Game or App:

- Select Character if the Game begins in with controlling a character.
- Select Vehicle if the Game begins with controlling a vehicle.
- Select Interface for Desktop Apps
- Select Misc for miscellaneous desktop tasks.



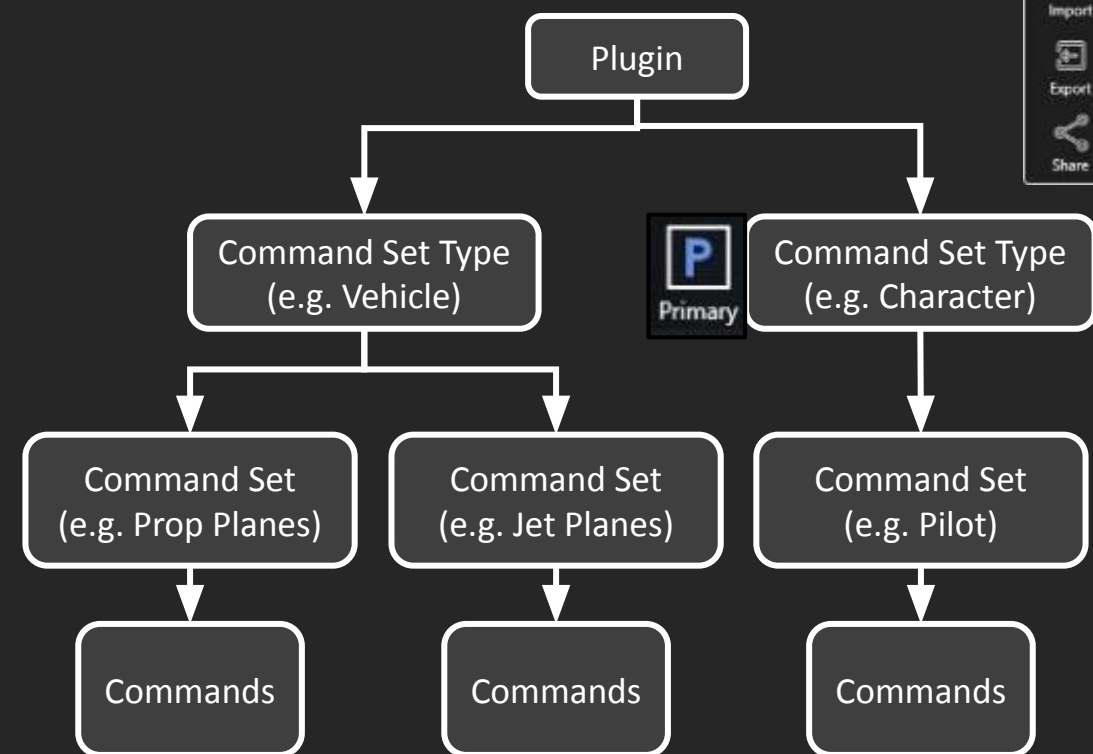
Create a Plugin

Select the Design button to begin designing the Plugin's Primary Command Set.

Important Note:

The Plugin Designer will always open the Primary Command Set.

Example Plugin Structure



Design a Plugin

Plugins are designed in the Plugin Designer.

Designing a Plugin focuses on creating Command Sets for the Active Plugin (Green), which contain Commands specific to the type and purpose of the Command Set (e.g. Vehicle | Prop Planes).

Select the Design button to open the Plugin Designer.



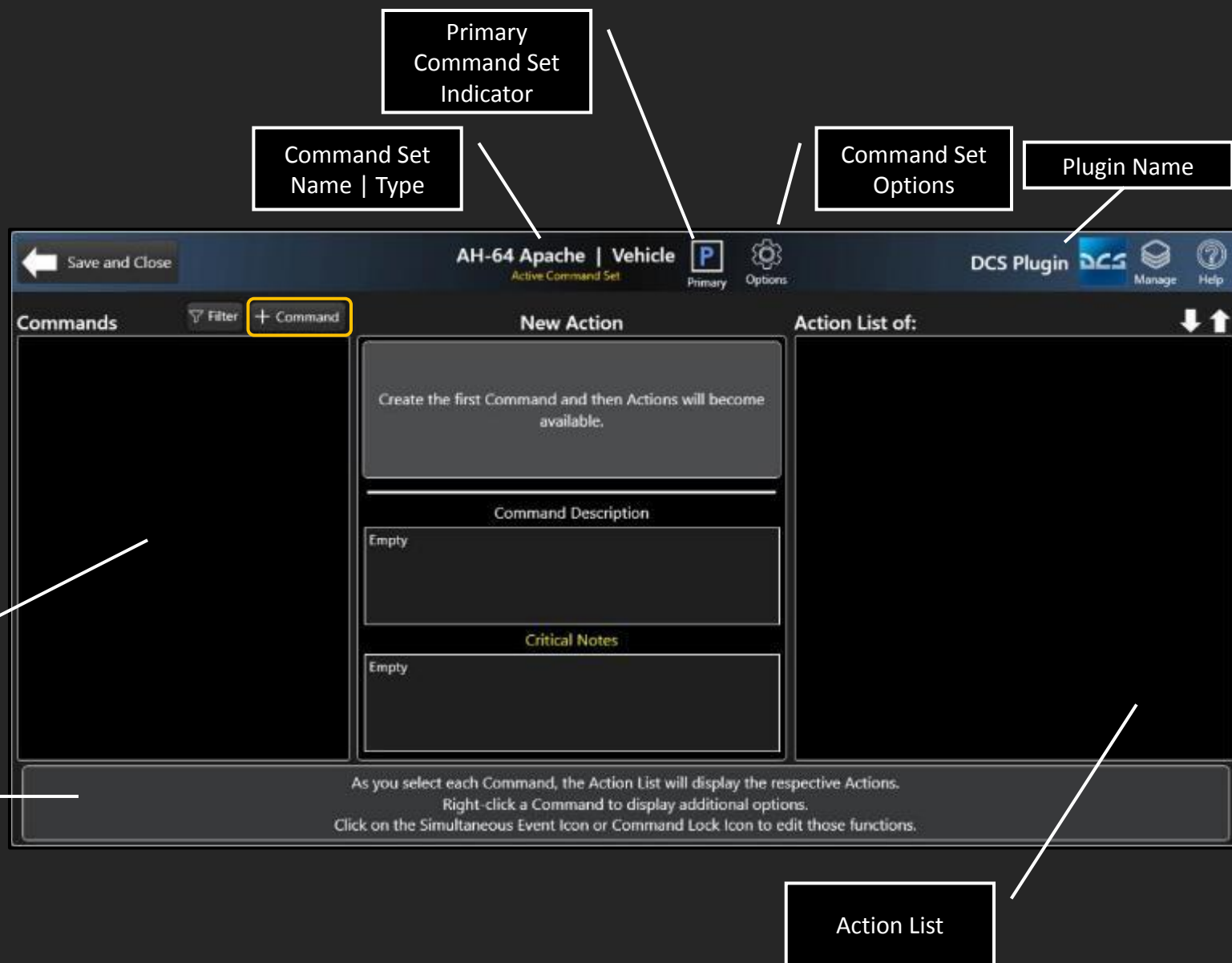
Design a Plugin

When the Plugin Designer is opened, the last edited Command Set is loaded, unless, in this case, where only one Command Set exist, the Primary Command Set will load.

The Plugin Designer is organized in 3 major areas:

- Top Bar
- Commands and Action List
- Instructions Bar

Select the Create Command button to create the first Command.



Design a Plugin

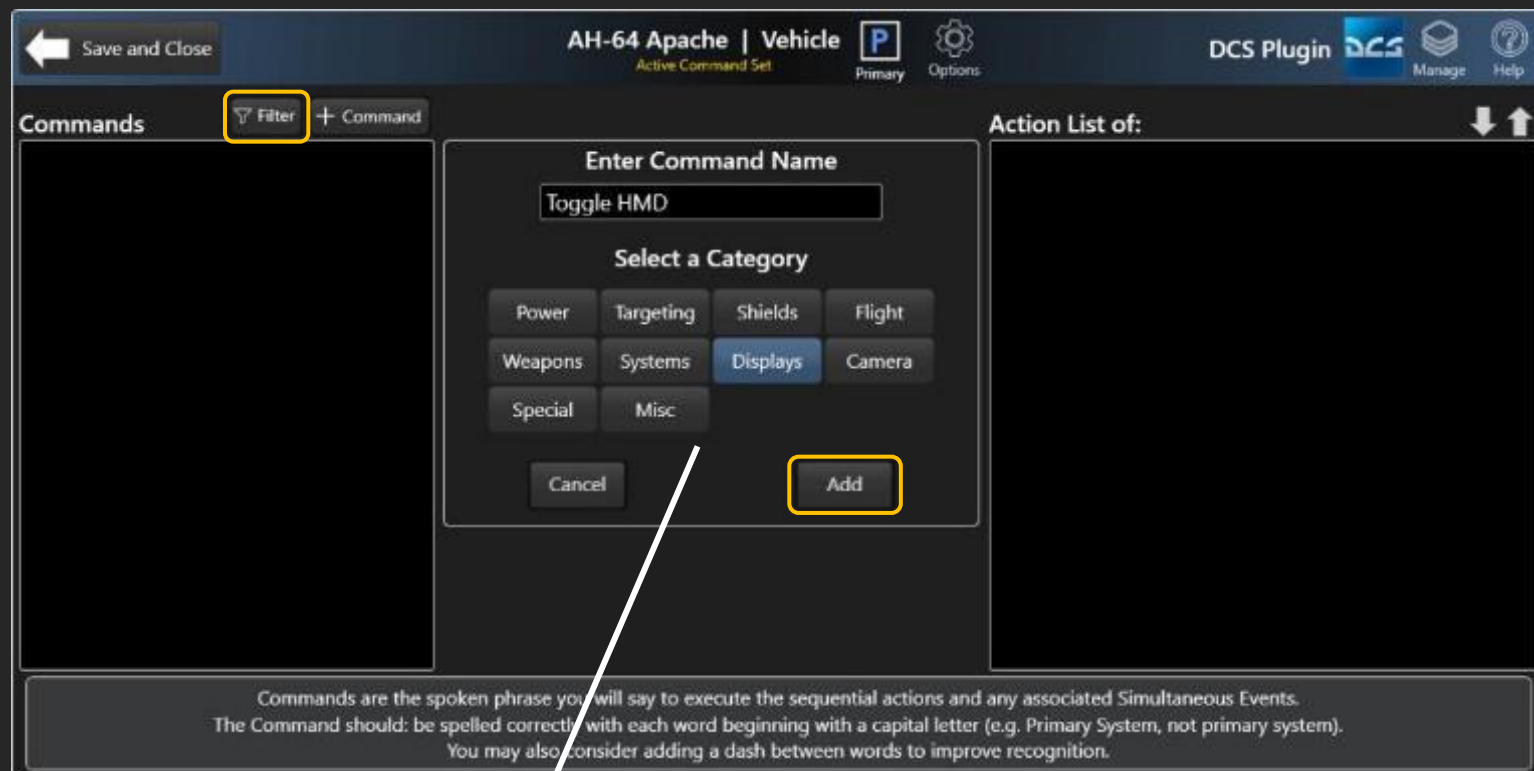
Commands must be assigned to a category.

The Command List can be filtered by selecting the Filter button, which can make locating a Command much easier.

To show all commands after filtering by Category, select the Reset button.

Both the Command List and Action List include scrolling.

Select the Add button in the Edit Area to add the Command to the Command List.



Edit Area

Design a Plugin

Commands require at least one Action.

In this example, the Key Action is added to the Action List.

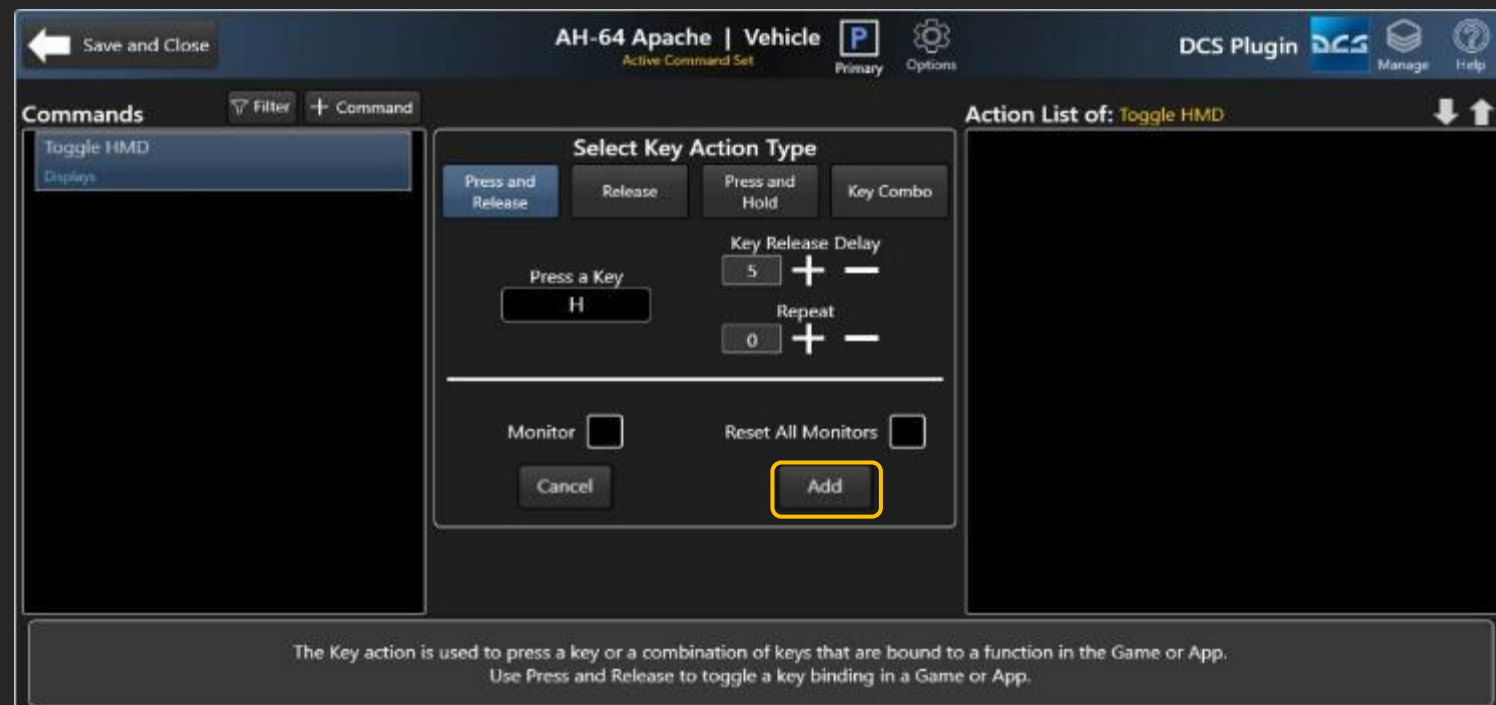
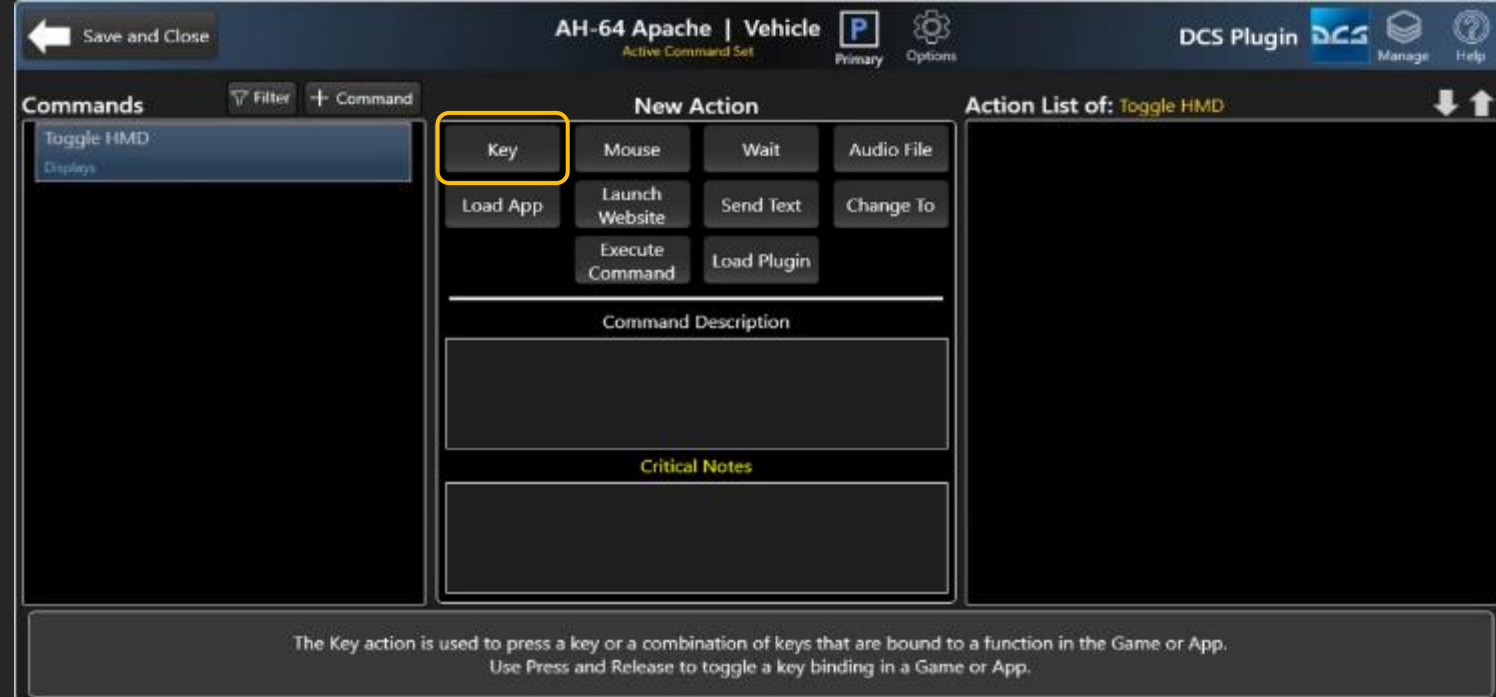
Select the Key button.

The Key Action has four (4) types. Select the appropriate type (Press and Release in this example).

Adjust parameters as required.

Press a Key, which is the key binding in the Game or App.

Select the Add button to add the Key Action to the Action List of the selected Command.



Design a Plugin

Now the first Action is available in the Action List.

Actions are performed sequentially, from top to bottom.

As you select each Command, the Action List will display the respective Actions.

Use the UP/DOWN arrows to change the position of the Action

Right-click the Action to access additional options.



Add more actions to the Actions list, as required.



Add Description of the
Command + Critical
Notes (as required)

Design a Plugin

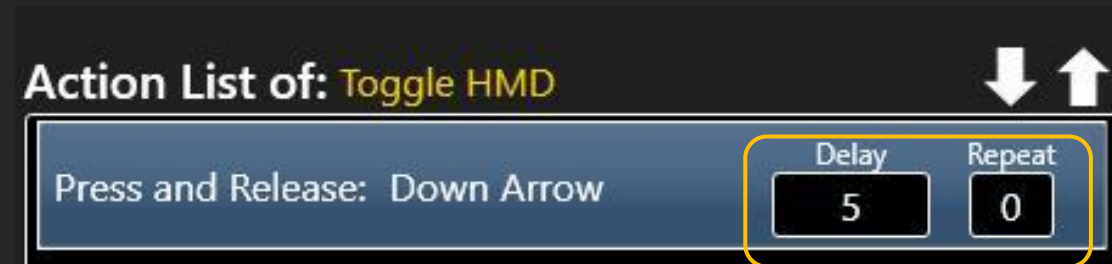
Some Actions have in-list adjustable parameters. Select the respective box and type in the new value:

Delay – For Press and Release Actions, this is the duration of time from pressing the Key down and releasing the Key. If the Action is not performing as intended, adjust this to compensate for server lag, game animations, etc.

The exception to this rule is the Execute Command Action, where Delay is the sum of all Actions within the Command.

Only increase the Delay value to ensure that Execute Commands Actions occur sequentially.

Repeat – This value indicates the number of times the Action should repeat. Example, a value of 2 would process the Action two times.



Design a Plugin

One of the best features is the ability to play an Audio File to confirm that a Command or series of Actions has processed.

There are two options:

- Any supported Audio File
- Emily Voice Pack Audio File

Select the option to Play Randomly when there are two (2) or more desired responses.

The Repeat option is also available, if for example, a tone must be repeated twice.

Supported Audio File types:

`.mp3`, `.m4a`, `.wma`, `.wav`



The Audio File action is used to provide feedback after a command is executed.
Note that long duration Audio Files require additional time to process.

Share a Plugin

Another excellent feature is the ability to share a Plugin you have developed with the Community.

Select the Share button and note the instructions.

Once we review and approve your Plugin, it will be available in the Plugin Manager for others to install.



Command Sets

Command Sets make up a Plugin.

A Command Set is made up of Spoken Commands and their respective actions.

There is no limit to the number Command Sets in a Plugin.

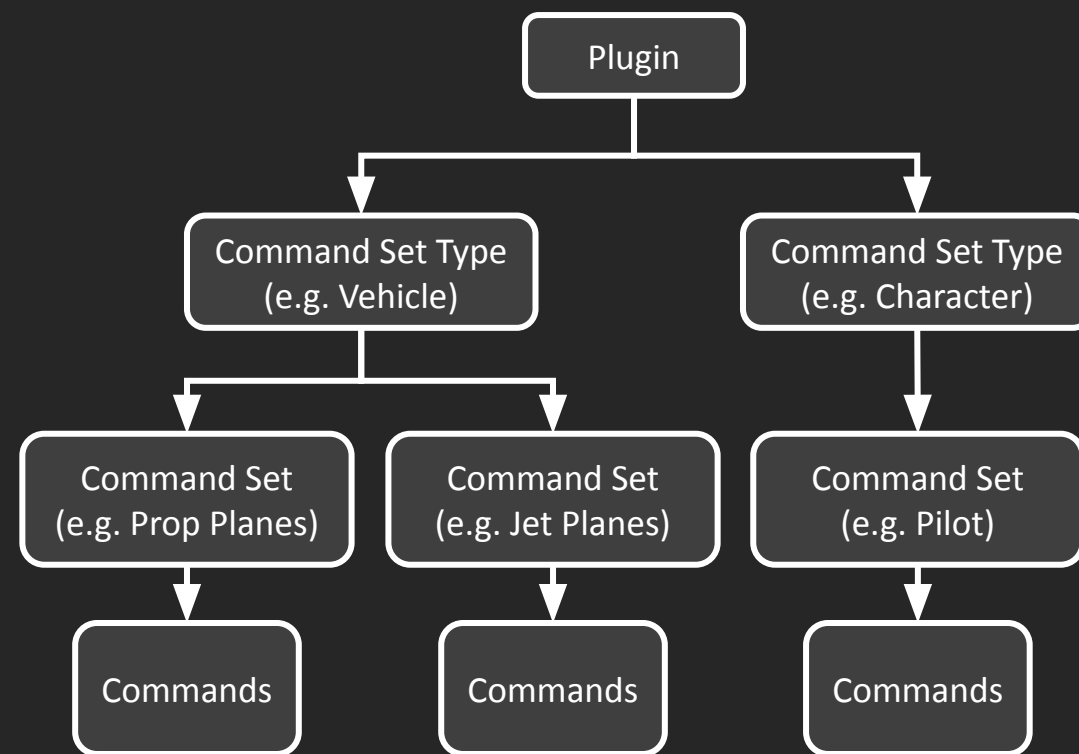
Use multiple Command Sets to keep your Commands organized and to optimize Emily's performance. There are four (4) Types of Command Sets that help logically organize Spoken Commands:

- Character - Used to control character actions.
- Vehicle - Used to control vehicle functions.
- Interface - Used for interacting with Inventory, Map, MFDs, etc.
- Misc - Used for Desktop Apps or anything else.

Each Command Set Type has categories to help organize Spoken Commands.

See the Change To Action for more info on seamless transition between Command Sets.

Plugin Structure



Example:

In the WoWP Plugin, there exist aircraft with propeller engines and aircraft with jet engines. Both share similar commands, but also have distinct commands to control systems. We can create two Command Sets and quickly switch between them via a Command using the Change To Action (discussed latter). Both are Vehicle Type Command Sets. We can also switch to a "Pilot" Command Set which is used to control a Character.

Command Set Options

There are several options for managing Command Sets in the Plugin Designer.

Select the Options icon to reveal the menu.



Select **Switch** to switch to a different Command Set for editing.

Select **Create** to create a new Command Set.

Select **Duplicate** to create a duplicate copy of the Active Command Set.

Select **Edit** to change the name of the Command Set or set the Active Command Set as the Primary Command Set.

Select **Import** to add a Command Set from another Plugin or from another Emily user. Select **Export** to the Active Command Set with others.

Select **Delete** to permanently delete the Active Command Set.



Commands

Commands make up a Command Set.

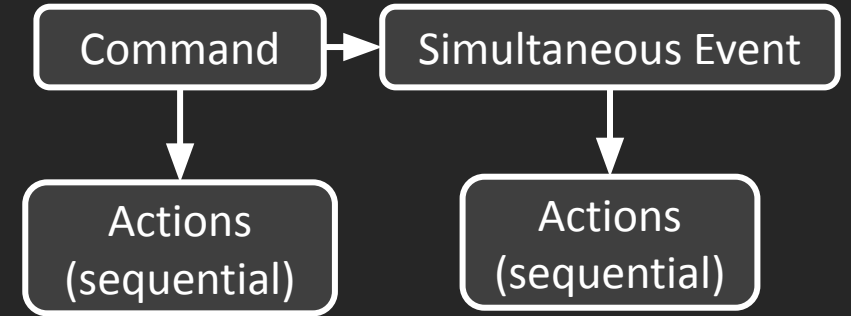
There is no limit to the number Commands in a Command Set, however, the recommended maximum number of Commands is 250 to maintain Emily's response performance and to help you keep Commands organized.

Commands require Actions. There is no limit to the number of Actions a Command can execute, however, you may wish to execute other actions simultaneously, in which case, a Simultaneous Event is recommended.

Actions execute sequentially, one after the other.

See Simultaneous Events for more info. The possibilities are endless!

Command Structure



Command Options

Right-click on a Command to reveal several options:



Duplicate the Command.

- Helpful for creating multiple Commands with the same task, but slightly different Action variations, for example, selecting menu items.

Edit the Name and Category assigned to the Command.

Copy the Command, Simultaneous Events, and Actions to a different Command Set.

- Helpful when there are two or more Command Sets with similar commands.

Create a Simultaneous Event that executes at the same time as the Command.

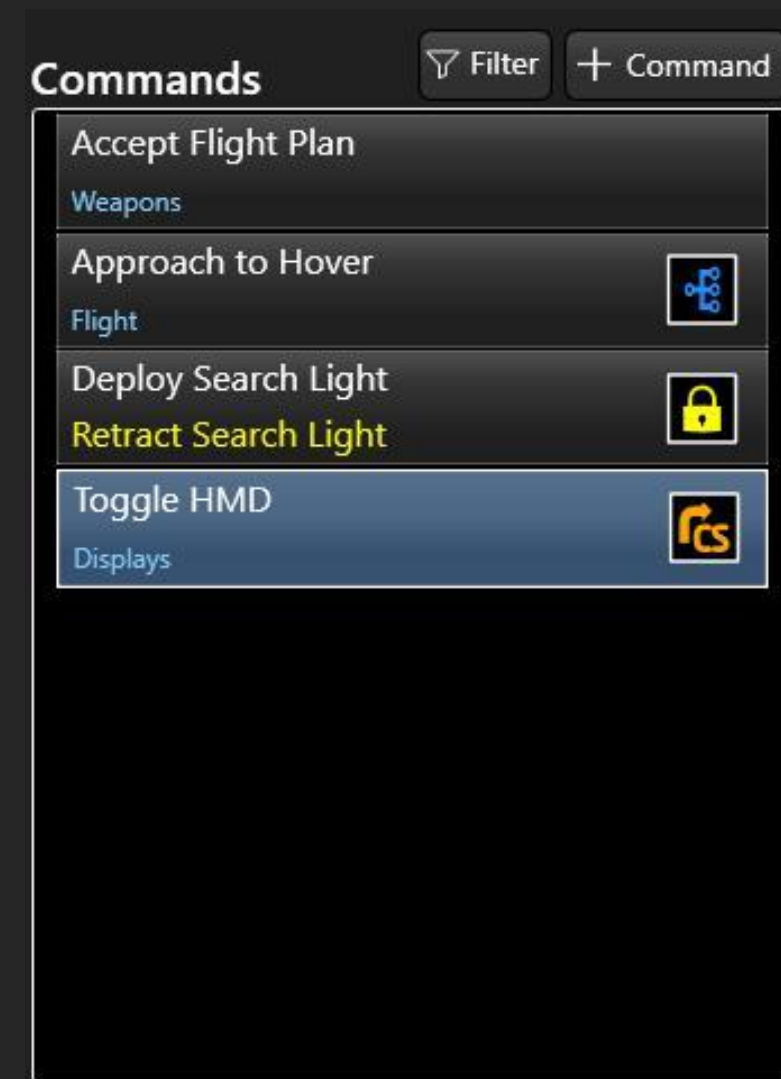
- Click this icon to create additional events.
- Click this icon on the Command to view all events.
- An incredibly useful feature for complex tasks.

Pair the selected Command with an opposite Command to prevent Command execution if you have already said the command.

- Assign reminder Audio Cues that the Command has already been executed.

Delete the Command, it's Simultaneous Events (if present), and it's Actions.

The Create Simultaneous Event and Command Lock options will add an icon to the Command.



Command Lock

Prevent a Command from executing if you already issued it with Command Lock.

Create two Commands that toggle the state of something.

Example use of Command Lock:

Create a Landing Gear Deployed Command, then create a Landing Gear Retracted Command

Right-click one of these two Commands, and select Lock.



Select the Opposite toggle Command from the list.

Select both reminder audio files.

Reminder audio files will play if the respective state has already been toggled.

Important Notes:

The lock states are cleared when switching between Command Sets or Plugins.

The lock state can be manually reset by saying the Global Command “[Reset All Command Locks](#)”.

Example



Simultaneous Events

Achieve complex tasks concurrently and quickly with Simultaneous Events.

Example use of Simultaneous Event

Create a Primary Command called: Approach to Hover. Add an Audio File and a Key Action.

Select Create from the right-click menu to create each Simultaneous Event.



Name the first Simultaneous Event: Adjust Cyclic. Add a Key Action, then adjust the Delay to equal the duration to hold the Key down.

Select Save when all Actions are added. Several Actions can be added, just like the Primary Command.

Repeat this process for all Simultaneous Events. Each must be added separately.

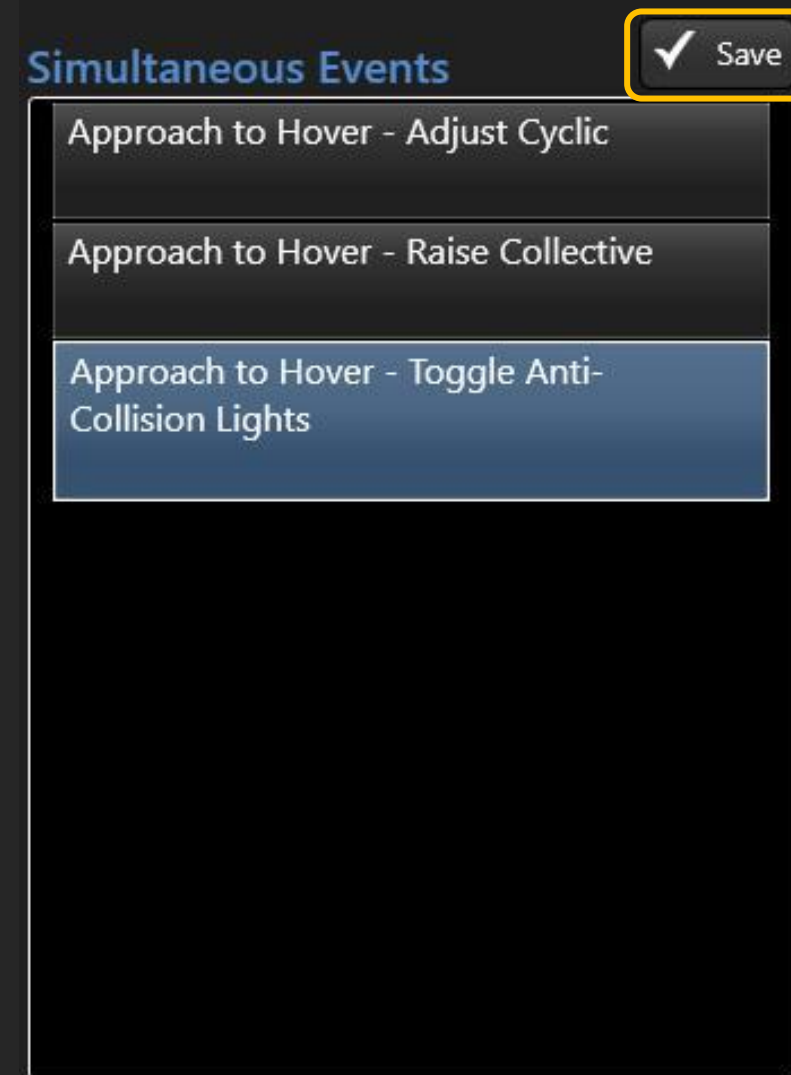
Important Notes:

All Simultaneous Events execute at the same time when the Primary Command is spoken.

Add Wait Actions to delay their execution, if required.

Actions in each Simultaneous Event Action List occur sequentially, just like the Primary Command, from top to bottom.

Example



Commands used to Change To Another Command Set

Easily load another Command Set in the Active Plugin using a Command.

Example use of Change To Action:

Create a Command to load the HMD Interface Command Set.

Select the HMD Interface Command Set from the list.

Now the Change To icon appears next to the Command.

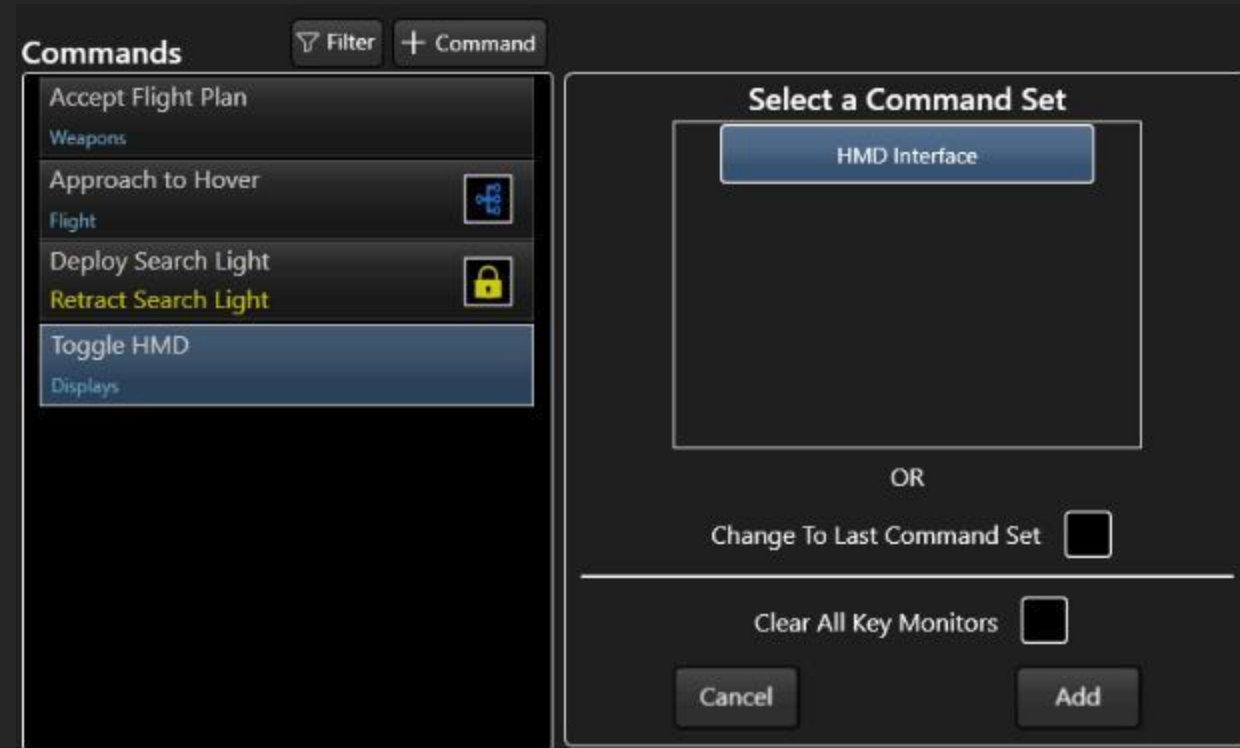


Important Notes:

Add a similar Command to the other Command Set to change back and forth between the two.

Select the “Change To Last Command Set” , which will load the Last loaded Command Set. This is useful for switching to/from another Command Set to briefly issue Commands, for example, using an Inventory in a game.

Select Clear All Key Monitors option if the Command Set being loaded has Key Actions that are being monitored in the current Command Set. See Actions – Key for more details.



Actions

Each Command performs Actions.

There is no limit to the number Actions in a Command.

Most Commands perform a single action, such as Pressing a key or launching a website.

Some Commands perform multiple actions to accomplish repetitive and complex tasks. See Example.

Each will be described in this guide.

Remember to look for helpful information in the Instructions Bar at the bottom of the Plugin Designer.

Each Action is performed sequentially in the order you place it in the Action List.

Action List of: Perform Preflight Checks



Play Audio File: Start Up Protcols Initiated.m4a	Repeat
	<input type="text" value="0"/>
Press and Release: E	Delay
	<input type="text" value="5"/>
	Repeat
	<input type="text" value="10"/>
Press and Release: Y	Delay
	<input type="text" value="5"/>
	Repeat
	<input type="text" value="0"/>
Wait: 2000	
Play Audio File: Power is Normalized.m4a	Repeat
	<input type="text" value="0"/>
Modifier Combo: Left <CTRL> + K	Repeat
	<input type="text" value="0"/>
Wait: 1000	
Play Audio File: Ready Ding.m4a	Repeat
	<input type="text" value="0"/>

Key Action

Perform various types of Keyboard actions, including Key Combos.

There are four types:

- Press and Release – Includes Down/Up Delay, Repeat, Monitor, and Reset Monitors options
- Press and Hold – Only holds the Key down
- Release – Only releases the Key
- Key Combo – Includes up to 3 modifier keys and a Main Key, and three distinct delay adjustments to obtain the perfect execution.

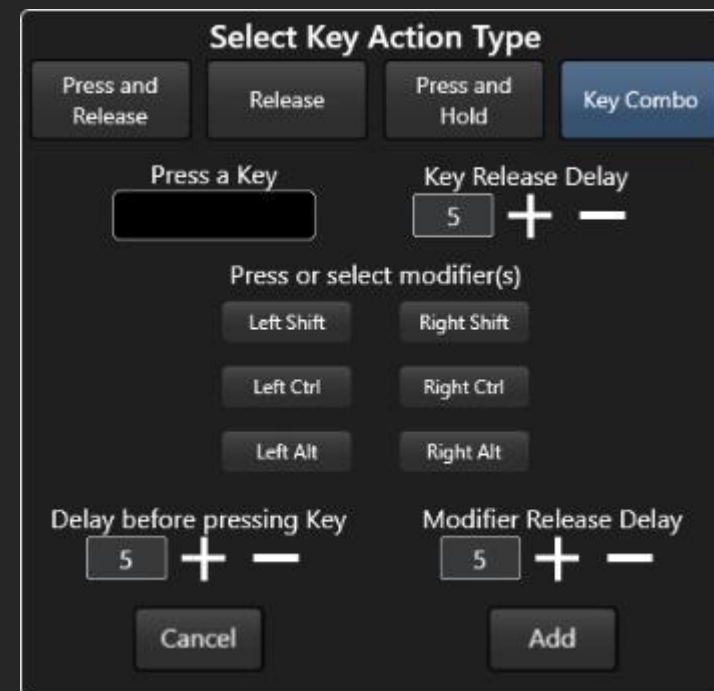
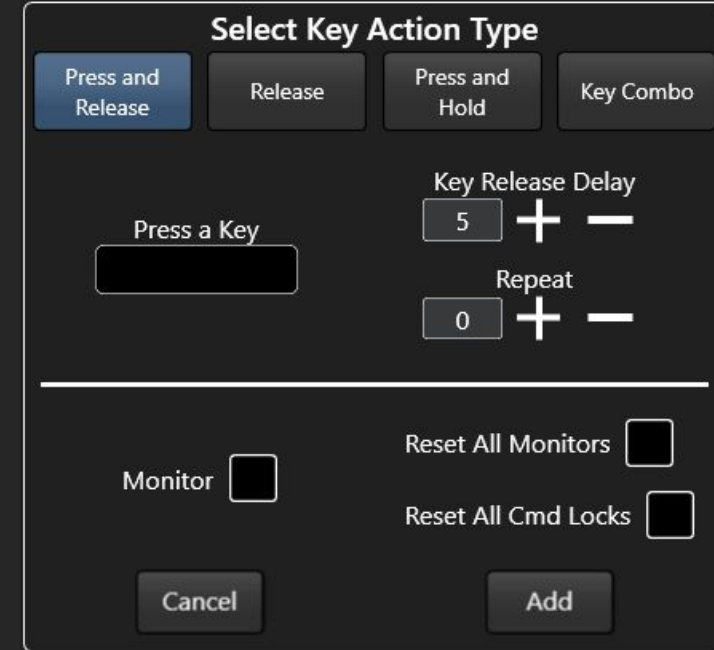
Select the Type and then press a Key on the keyboard. Select Options as required.

Select Monitor to instruct Emily to not press this Key if it has already been manually pressed or pressed via a Command (see Key Monitoring section for details).

Select Reset All Monitors to create an Action that clear the monitoring memory when the Action executes.

Select Reset All Command Locks to create an Action that clear all active Command Locks.

Emily only supports US QWERTY keyboards.



Key Monitoring

Use the Monitor Action to check if the key has already been pressed.

If so, then it is skipped. Else, the key is pressed and released.

Say the command "RESET ALL KEY MONITORS" to reset the Monitor until the key is pressed again either manually or via Command.

If testing the command, you must reset any game states, then Say the command "RESET ALL KEY MONITORS".

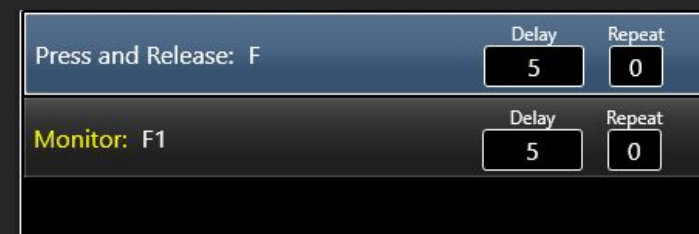
Example use of the Monitor option:

The "F1" Key is used to open/close the Interface. The Interface has specific sections, which are accessed by selecting a button on the Interface. There exist a Command called "Open Comms" that opens the Interface and clicks on a button that opens the "Communication" tab. A command that closes the Interface should press and release the Key (not Monitor the Key) to "Reset" its state.

When this command is spoken, two possibilities exist:

If the Interface is closed, the Interface will open (pressing the F1 Key) and the Communication tab is selected.

If the Interface is open, then the F1 Key will not be pressed because it is Monitored the Communication tab is selected.



Mouse Action

Perform various types of button actions, scroll, and move actions.

There are nine (9) types of Mouse Actions:

- Press and Release – Includes Down/Up Delay and Repeat option
- Press and Hold – Only holds the Mouse button down
- Release – Only releases the Mouse button
- Scroll – Includes Repeat option
- Move Absolute – Moves the mouse to a position on the screen
- Move Relative from Center – Moves the mouse to a position relative from the Center of the screen
- Move and Recenter – Moves the mouse to a position relative from the Center of the screen and then returns to the Center of the screen
- Move Start to End – Moves the mouse from the center of the screen to a start position and back, then to an end position and back to the center of the screen
- Move Relative from Last Point

Select the Type and follow the instructions.

Select Options as required.

Important Note:

Manually moving the mouse when issuing a Command that contains Mouse Move Actions will override the Action. If override is required, say the Global Command “Cancel Command”.



Mouse Move Relative from Center Action

Moves the mouse to a position relative from the Center of the screen.

Use exclusively in Games to move the mouse to one position on the screen and then perform another mouse action, such as a button press and release.

Record the Mouse Position, then test that the mouse moves from the center screen to the recorded position.

Wait Actions should be added after the Mouse Move to account for game animations, so that subsequent actions execute correctly.

Example use of Move Relative:

The Cockpit in your spacecraft has an Multi-Function Display (MFD) that you can interact with using the mouse, but in order to do so, you must zoom into the MFD. In the example, the F Key is held down to enable free-look, a Wait Action accounts for Game animation, the A key is pressed to snap the characters view left, and the Middle Mouse Button is pressed to zoom in to the MFD. After accounting for Game animation, the mouse is moved to the Power button on the MFD, Clicked by the Left Mouse Button, and then the F Key is released.

Important Notes:

In some Games, there may be boundaries, where mouse acceleration is different than other parts on the screen.

If required, right-click the Move Relative Action to manually fine tune the position using the directional arrows.

Example

Action List of: Select Power ↓ ↑

Press and Hold: F		
Wait: 800		
Press and Release: A	Delay: 5	Repeat: 0
Press and Release: Middle Mouse Button	Delay: 5	Repeat: 0
Wait: 800		
Move Relative from Center: -644,-416		
Wait: 5		
Press and Release: Left Mouse Button	Delay: 5	Repeat: 0

Mouse Move and Recenter Action

Moves the mouse to a position relative from the Center of the screen and then back to the center of the screen.

Use exclusively in Gamed to move the mouse from center to several different positions.

Important Notes:

After recording the mouse position, four (4) Actions are added to the Action List:

- A Move to Position, which moves the mouse to the position you recorded
- A 20 ms Wait Action (adjustable)
- A Return to Center Position, which returns the mouse to the center of the screen
- A 20 ms Wait Action (adjustable); provides time between multiple Move and Recenter Actions

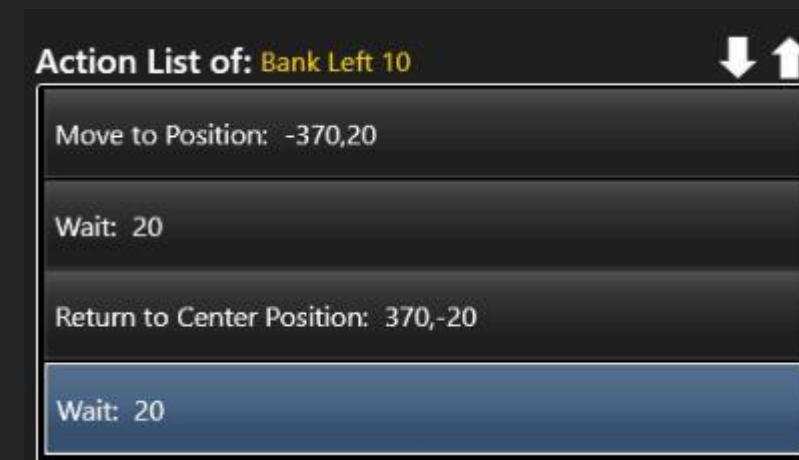
The Wait Actions account for Game Lag, server Lag, etc. Do not delete these Actions. Test the Mouse move in the game and adjust the Wait Actions as required.

If required, right-click the Move to Position Action to manually fine tune the position using the directional arrows. The Return to Center Position will automatically update.

Another option to optimizing the accuracy of the Mouse Move is to apply a Scaling Factor. If using an [Official Emily Plugin](#), look at the Plugin Details in the Plugin Manager to see if there are recommended Scaling Factors. If so, be sure to enter them for each Move and Recenter Action. **Otherwise, Scaling Factor is not likely required.**

How to Determine Scaling Factor is covered later in this guide.

Example



Mouse Move Start to End Action Example

Let's say that the goal is to move things in an inventory in the Game using a Command that moves an item in the inventory to the character's backpack.

Create a Move Start to End Action. Make sure to set the scaling factor, if one is provided (for Official Emily Plugins). Record the Start Position and End Position. Return to Center Actions and Wait Actions are automatically created to allow for Game animations.

Create a Mouse Press and Hold Action (Left Button). This action will grab the item at the Start Position, before moving to the End Position.

Create a Mouse Release Action (Left Button) to drop the item in the backpack.

Add a Wait action of 20ms between the Press and Hold Action and the move Actions, and between the Release Action and the Move Actions.

Test the Command in the Game. If the mouse moves to the correct positions, but the item is not moved, adjust the Wait Actions. If required, right-click the Mouse Start Position or Mouse End Position to manually fine tune the position using the directional arrows. The respective Return to Position will automatically update.

Important Notes:

After recording the mouse position, eight (8) Actions are added to the Action List. Do not delete these Actions.

Action List of: Hellfire Setting Bravo

Mouse Start Position: -212,-95

Wait: 20

Press and Release: Left Mouse Button Delay: 5 Repeat: 0

Wait: 5

Return to Center Position: 212,95

Wait: 20

Mouse End Position: -69,39

Wait: 20

Do not delete the last Wait Action in the list, which is crucial if you wish to repeat this Command several times (see Execute Command Action).

Mouse Move Relative from Last Point

Moves the mouse to a position relative from the last mouse position, with the first being relative from the Center of the screen.

Use exclusively in Games to move the mouse to one position on the screen and then perform another mouse action, such as a button press and release and then move the to another point to perform additional actions and so on.

This action automatically inserts a “centering” mouse move of -1,1. Do not delete this action.

Wait Actions should be added after the Mouse Move to account for game animations, so that subsequent actions execute correctly.

Say the command Stop Recording when all mouse positions have been recorded.

It is often easier to record the mouse positions and then add in the other actions.

Important Notes:

In some Games, there may be boundaries, where mouse acceleration is different than other parts on the screen.

If required, right-click the Move Relative Action to manually fine tune the position using the directional arrows.

Example

The screenshot shows an 'Action List' for 'Select Map Button'. The list contains the following actions in order:

- Press and Release: F1 (Delay: 5, Repeat: 0)
- Wait: 800
- Wait: 20
- Move to Absolute Position: -1,1
- Wait: 20
- Move Relative from Last: 11,-202 (H Scale: 0, V Scale: 0)
- Wait: 20
- Move Relative from Last: -408,153 (H Scale: 0, V Scale: 0)

How to Determine the Scaling Factor

Determining the Scaling Factor will save time as it can be applied to each subsequent Action. Scaling Factors are only used for Games, not Desktop Apps.

Make sure the Game Resolution is the same as the screen resolution. Horizontal and Vertical Scaling Factors may be different.

Scaling Factor can be determined by:

1. Create a Command called **TEST HORIZONTAL MOVE**
2. Using a Move and Recenter Action, capture a precise position on the Game Screen, such as the graphic that does not change.
3. Once captured, increase the Wait Action between the Move to Position and Return to Center Position from 20 to 2000. This allows ample time to see where the mouse is moving to before it returns to center.
4. Close the Plugin Designer and return to the Game. Make sure that the mouse is dead center of the screen.
5. Test the Command in the Game.
6. If not correct, create a new Move and Recenter Action and adjust only the Horizontal Scaling Factor from 0 to 0.5.
 - Do not adjust the Vertical Scaling Factor yet
7. Record the same position in the Game. Close the Plugin Designer.
8. Test the Command in the Game. The mouse will move to the first incorrect position and then to the position just recorded. Note the difference in Horizontal movement.
 - If the mouse moved to far horizontally, then decrease the Horizontal Scaling Factor from 0.5 to 0.4.
 - If the Mouse did not move far enough horizontally, then increase the Horizontal Scaling Factor from 0.5 to 0.6.
9. Repeat until the mouse is moving to the same horizontal plane as the desired position.
10. Once the Horizontal Scaling Factor is determined, write it down.
1. Create a Command called **TEST VERTICAL MOVE**.
2. Repeat Steps 2–8, noting the vertical position of the mouse and adjusting the Vertical Scaling Factor only.
 - Note: the Horizontal Scaling Factor can be entered, but it is not required.
3. Write down the Vertical Scaling Factor. Delete the **both** Commands.

Play Audio Action

One of the best features is the ability to play an Audio File to confirm that a Command or Action has processed.

There are two options:

- Any supported Audio File
- Emily Voice Pack Audio File

There are four (4) recommended ways to add an Audio File.

1. Add the Audio file at the beginning of the list to get confirmation before other Actions occur (e.g. Command = Deploy Gear, Audio = Deploying)
2. Add the Audio file somewhere between other Actions to get confirmation of progress.
3. Add the Audio file at the end of the Action List to get confirmation of all actions complete. (e.g. Command = Deploy Gear, Audio = Deployed)
4. If time is critical, and immediate confirmation is required while other Actions are occurring, add the Audio File to a Simultaneous Event.

Select the option to Play Randomly when there are two (2) or more desired responses.

1. If the Audio Files must play immediately, then create a Simultaneous Event with the audio files. Right click on the Command to create the Simultaneous event.
2. If the Audio must play to get confirmation of progress, then create a Simultaneous Event with the audio files and then add a Wait Action at the top, which has a value that matches to approximate position in the Command's Action List.
3. Add all of the Audio to the end of Action List.

The Repeat option is also available, if for example, a tone must be repeated twice, set Repeat to 2.



Wait Action

Use the Wait Action to allow for Game animations, server lag, or to simply perform actions after the specified duration of time.

The example shows two methods of adding wait durations for a Pressing and Releasing a key.

- The first Action is a Press and Release Key Action with the delay of 2000ms. This means that the F Key will be pressed and held down for 2000ms and then released.
- Use the Press and Hold + Wait + Release combo if you need to add other Actions before or after the Wait Action.

Also use the Wait Action at the beginning of a Simultaneous Event to delay its execution slightly, as required.

Enter Time before next Action

Milliseconds

Start typing numbers.
Tap the Enter Key when finished.

Cancel Add

Example: Emote

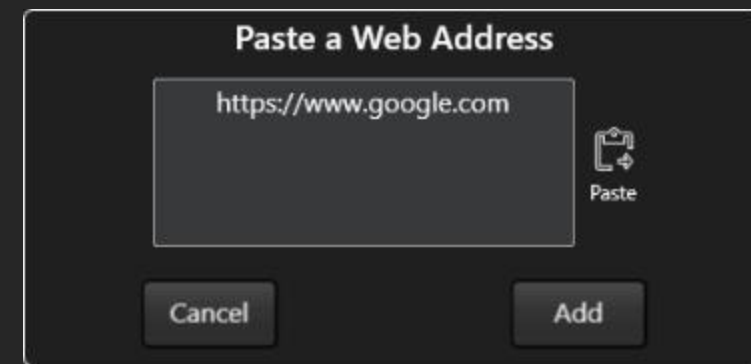
Action List of: Delay Example

	Delay	Repeat
Press and Release: F	2000	0
Press and Hold: F		
Wait: 2000		
Release: F		

Launch Website Action

Use the Launch Website Action to launch a website or Apps that rely on a URL.

First, copy the web address, then select Paste, then Add.



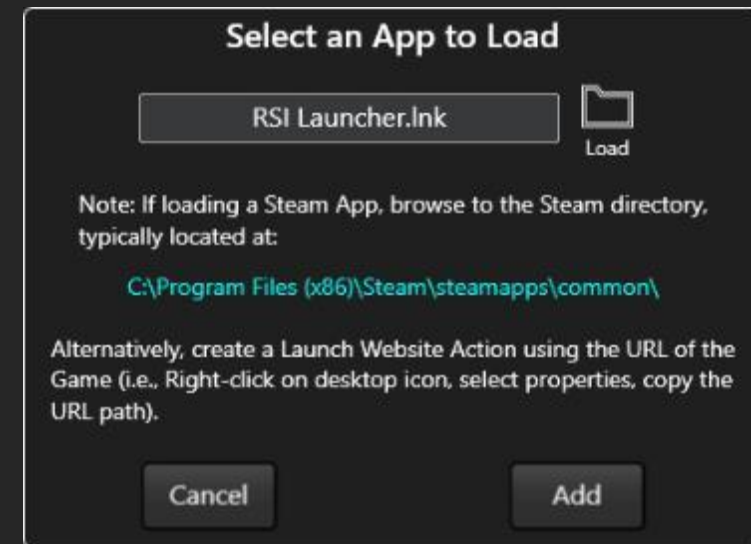
Load App Action

Use the Load App Action to load an App.

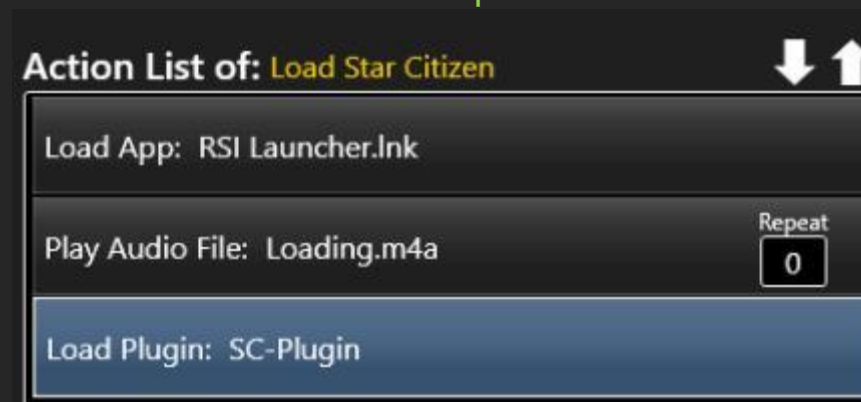
In the Launcher Plugin, use the Load App Action and Load Plugin Action together.

Select the Load button to find the App to load.

Note the use of Launch Website for some apps.



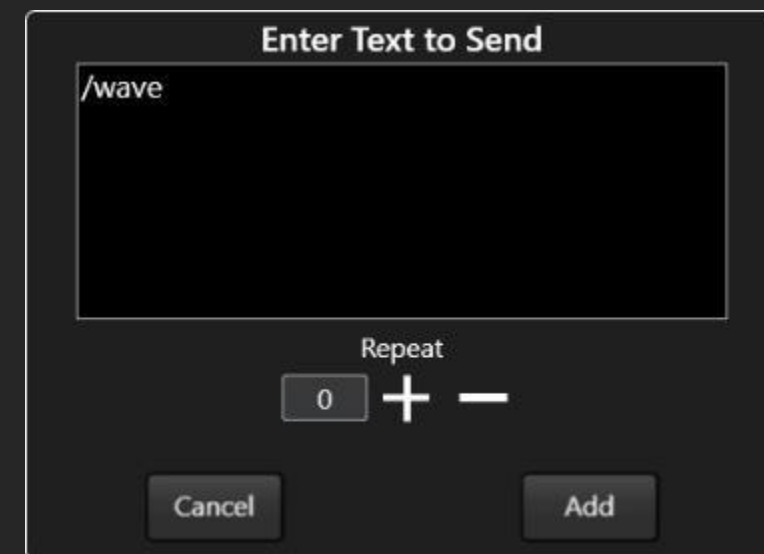
Example



Send Text Action

Use the Send Text Action to paste text anywhere that accepts typing, such as a chat window in a game.

Also, use for emotes that must be typed in a chat window.



Example: Emote

Example: Team Message

Action List of: Move to Bravo ↓ ↑

Press and Release: Enter	Delay: 5	Repeat: 0
Wait: 5		
Send Text: Move to Bravo!		Repeat: 0
Press and Release: Enter	Delay: 5	Repeat: 0

Action List of: Wave Emote ↓ ↑

Press and Release: Enter	Delay: 5	Repeat: 0
Wait: 5		
Send Text: /wave		Repeat: 0
Press and Release: Enter	Delay: 5	Repeat: 0
Change To: Last Active Command Set		

Change To Action

Use the Change To Action to easily switch between Command Sets with a Command.

Select the Command Set to be loaded. You can go back to the previous Command Set explicitly as well or use the Change To Last Command Set option.

Example use of Change To Last Command Set

There are two Command Sets, Pilot and G1000 Interface.

In the Pilot Command Set, a Command called: “Adjust Flight Reference” uses a Change To Action to open the G1000 Interface Command Set.

In the G1000 Interface Command Set, the command IDENT has a Change To Last Command Set Action at the end of the Action List.

Example

Action List of: Adjust Flight Reference

Action	Delay	Repeat
Execute Command: View 1	321	0
Wait: 800		
Press and Release: Right Mouse Button	25	0
Move to Absolute Position: 0,0		
Play Audio File: Ready.m4a		0
Change To: G1000 Interface		

Action List of: IDENT

Action	Delay	Repeat
Execute Command: Eight	144	0
Wait: 500		
Press and Release: F	25	0
Change To: Pilot		

Set **Core** as a Master Command Set.

Select a Command Set

- Cockpit Camera
- G1000 Interface

Change To Last Command Set

Change To Last Master Command Set

Clear All Key Monitors

Cancel Add

Change To Action – Command Set Layers & Master Command Sets

Select the **Set ___ as a Master Command Set** option if you are designing multi-layered Command Sets that enable interactive decision paths that may not always lead back to the desired Command Set. This will designate the current Command Set as a Master Command Set.

In the final Command Set or last Layer, select the **Change To Last Master Command Set** option to automatically return to the Master Command Set.

For example, a Display interface has multiple layers of information, each with similar bezel keys (1, 2, 3, etc.) that are assigned to Commands (Select 1, Select 2, Select 3, etc.).

On each Layer, the bezel keys represent different options as soft keys on the display. In Layer 1, the ENTER soft key is the #3 bezel key, where in Layer 2, the ENTER soft key is the #6 bezel key.

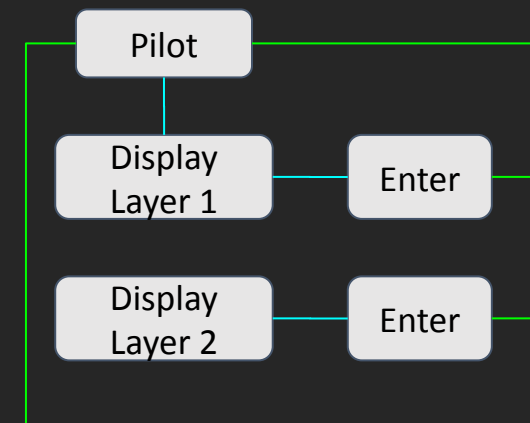
So, the Command Set for the first Display Layer has a Command called Enter, which moves the mouse to the #3 bezel key and clicks it, where in the Command Set for the second Display Layer has a Command called Enter, which moves the mouse to the #6 bezel key and clicks it.

Each Enter Command in either Command Set could have a Change To Action that loads the Last Master Command Set, in this case, Pilot.

All Change To Actions are eligible to be Master Command Sets, so the possibilities are interesting.



Example



Execute Command Action

Use the Execute Command Action to add a Command to the Action List.

This can be helpful by adding common functions to that need to be performed across different commands.

Select the Command to add to the Action List.

Example use of Execute Command

There is a need to arm five (5) missiles on the ship. Doing this requires a repetitive task.

Create a command called: Arm Missile. Add the actions required to Arm one (1) missile.

Create another Command called Arm 5 and add an Execute Command Action.

Repeat the Action five (5) times.

Important Notes:

The appropriate Delay time is automatically calculated. Do not decrease this value. Increase this value if other Commands require adjustment to Wait times. If the Base Command (i.e. Arm Missile) is modified, edit and save the associated Execute Command Action to recalculate the Delay.

Example: Arm Missile

Action List of: Arm Missile

	Delay	Repeat
Press and Release: D	5	0
Wait: 150		
Press and Release: Middle Mouse Button	5	0

Example: Arm 5

Action List of: Arm 5

	Delay	Repeat
Execute Command: Arm Missile	209	5

Load Plugin Action

Use the Load Plugin Action to load a Plugin as the last Action in the Action List.

This Action is typically used in the Launcher to load a Plugin and the respective Game or App with one Command.

However, the Load Plugin Action can be used to switch between Apps by creating the respective Commands in each App Plugin.

Example: Load a Game + Plugin

In the Launcher, create a command called: Load Star Citizen.

Add the "Loading" Audio File from the Emily Voice Pack.

Add a Load App Action that loads the Game, in this case Star Citizen®.

Add a Load Plugin Action to load the SC-Plugin.

Example: Load a App + Plugin from another App

In the Word Plugin, create a command called: Load PowerPoint.

Add the "Loading" Audio File from the Emily Voice Pack.

Add a Load App Action that loads the App, in this case Microsoft PowerPoint®.

Add a Load Plugin Action to load the PPT Plugin.

Important Notes:

Other actions can be added to the Action List after the Load App Action to perform tedious tasks such as clicking buttons or entering credentials.

Example: Load a Game + Plugin

Action List of: Load Star Citizen



Load App: RSI Launcher.Ink

Play Audio File: Loading.m4a

Repeat

0

Load Plugin: SC-Plugin

Example: Load a App + Plugin from another App

Action List of: Load Power Point



Play Audio File: Loading.m4a

Repeat

0

Load App: PowerPoint.Ink

Load Plugin: PPT Plugin