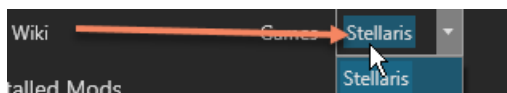


Table of contents

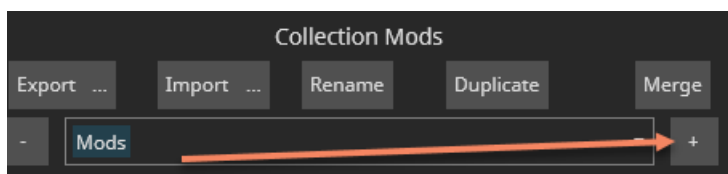
- **Home**
- [New User Checklist](#)
- Mod Management
 - [Options](#)
 - [Installed Mods](#)
 - [Collection Mods](#)
 - [Installing New Mods](#)
- Conflict Solver
 - [Conflicted Objects](#)
 - [Ignore Rules](#)
 - [Merge Viewer](#)
 - [Binary Merge Viewer](#)
 - [Mod Filter](#)
 - [Reset Conflicts](#)
 - [Database Search](#)
 - [Custom Patches](#)
- [Keyboard Shortcuts](#)
- [Add Irony To Steam](#)
- [FAQ](#)
- [Privacy Policy](#)

New User Checklist

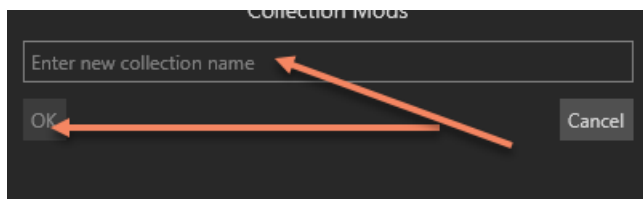
1. Go and check out the [home](#) page. It contains some useful info.
2. Go to the [releases](#) page and download the latest release
3. Select a game



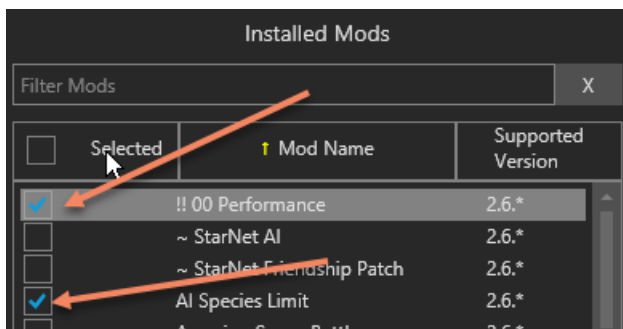
4. Create a new collection



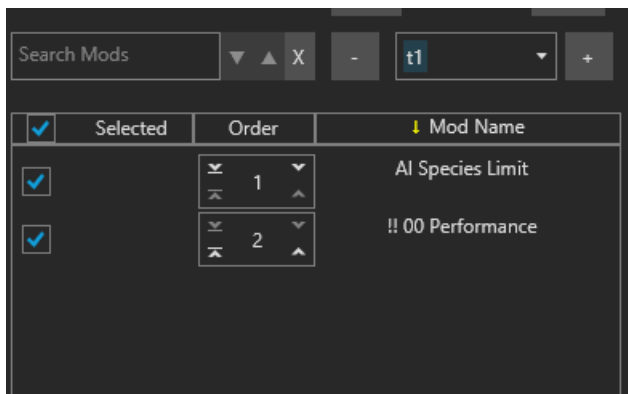
5. Enter a collection name



6. Select mods from the Installed Mods area



7. Note that they are visible on the right hand side now under Collection Mods



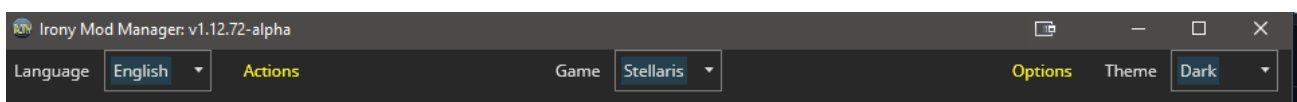
8. Perform any sorting you deem necessary

9. OPTIONAL: Run conflict solver (if available)

10. OPTIONAL: Merge any conflicts that you may find (if available). You can learn more under the Conflict Solver section on this wiki.

11. Press "Apply" or "Launch Game" button to apply the collection mod ordering to the game (and launch the game if "Launch Game" option was used)

Options



Language

Allows you to change the application language.

Actions

Allows you to perform the following:

- Open DLC Manager (see DLC Manager for more info)
- Open Irony wiki page in the browser
- Open Irony logs directory in file explorer (ex. windows explorer)
- Open Selected game error.log in default text editor

Game

Allows you to change to the game.

Options

Allows you to set game specific and global application options. See options screen for more info.

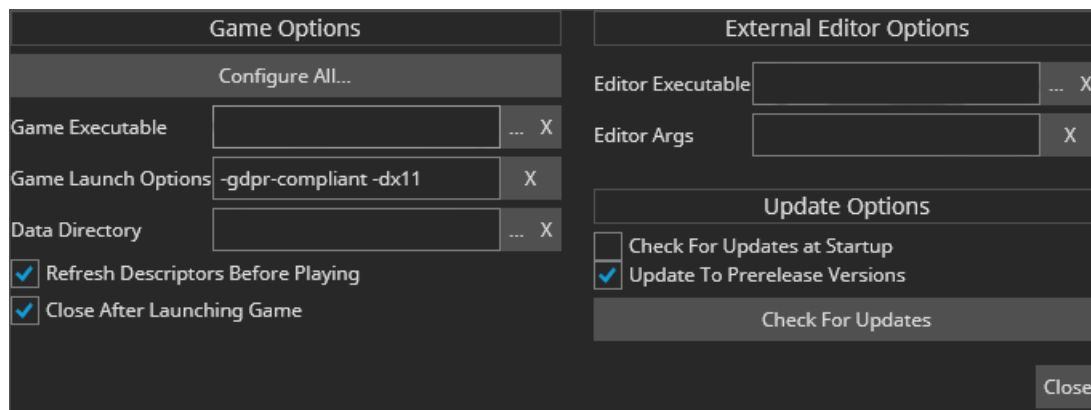
Themes

Allows you to select one of the available application themes.

DLC Manager

By toggling this actions option will open the DLC Manager screen where you can either check or uncheck a DLC. To apply settings press the close button.

Options screen



Game Options

- Configure all

This option raises the folder browse dialog and if you point it towards your game installation directory Irony should figure out automatically:

1. Executable path
2. Data directory path (ex. Documents\Paradox Interactive\Stellaris)
3. Executable launch arguments

NOTE: Ensure that you point towards root game install directory ex. ..\steamapps\common\Stellaris

- Game executable

Specify which executable Irony will run when launch game option is clicked

- Game args

Set any additional executable parameters you'd like Irony to send to the game executable

- Data directory

Root game data directory where the game holds its settings, mods etc. An example would be for Stellaris: Documents\Paradox Interactive\Stellaris

- Custom mod directory

If this option is set Irony will use this as an additional folder from where it will look for mods. If set it'll also use this folder to create merged mods and patch mods.

- Refresh descriptors before playing

Irony will delete and recreate all descriptors before launching the game

- Close After Launching Game

If disabled Irony will not shut itself down after you launch the game using Irony.

External Editor Options

This option is used in Conflict Solver which allows you to specify an external diff compare editor which you can launch using the

context menu on the Irony virtual definition.

- Editor Executable

The patch to the binary of the external diff editor (ex. winmerge.exe)

- Editor Args

Arguments that Irony will send to the external diff editor

NOTE: You can share your external editor args with others [here](#)

Update Options

- Check for updates at startup

Irony will at each app run check for available updates and notify you if there are any.

- Update to prerelease versions

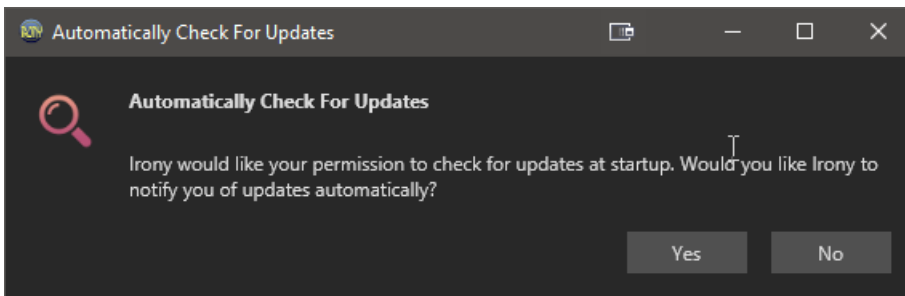
This will allow you to either just download stable releases or to update to latest alpha\beta\rc\stable releases.

- Download and install

If an update is available this option will initiate download and installation process.

How to update and how does it work

At first run you'll be greeted by this message (only once):

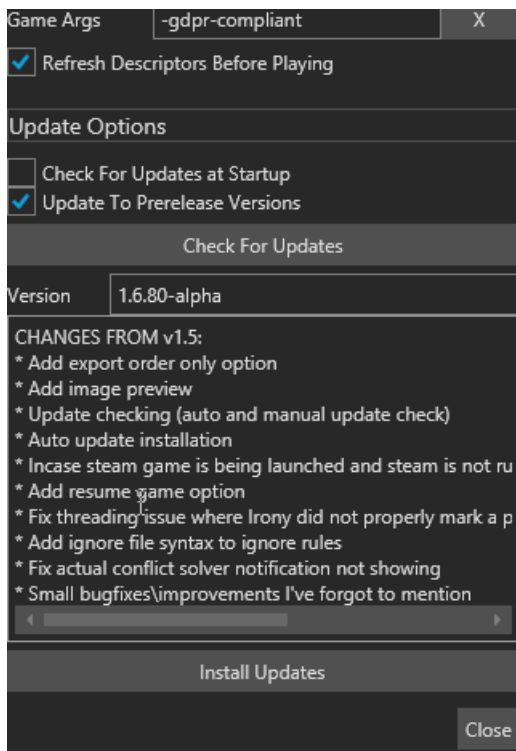


If you allow the prompt or if you do a manual update check it'll connect to this feed:

<https://bcssov.github.io/IronyModManager/appcast.xml>.

This is a simple xml holding all the download links and info about available updates. In short a simple GET request is performed at the mentioned address. If you selected yes at the initial prompt Irony will automatically at each app start check for updates. You can always turn it off in the options screen.

If there are updates available you should see something like this:

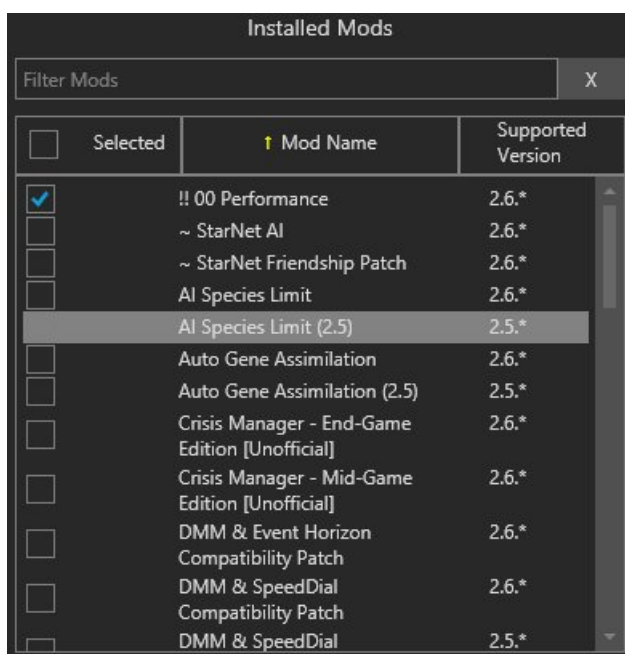


Toggling install updates Irony will connect to the Irony GitHub releases page and download the appropriate release package for your OS. And in case of Windows either a portable or installer version (depending on what you initially downloaded and installed). Irony will proceed and download the package to %appdata%\Mario\IronyModManager-Updates. It'll verify the downloaded archive signature to make sure it was downloaded correctly and then do one of the following:

1. Portable installation -> Irony will unpack the zip and run a console updater app. The console updater app will wait for 5 seconds before proceeding with the update (to allow Irony process to stop). It'll then copy all the downloaded files into the install folder and re-run Irony and console app will shut itself down. For OSX/Linux users it'll hopefully successfully set proper chmod +X permissions.
2. Installer installation -> Irony will unpack the zip and run the setup installer and that's done.

The downloaded files will remain there for about 3 days before Irony will delete them.

Installed Mods



Installed Mods section is where all of your installed mods are visible. This selection does not have any impact whether a mod will be enabled or disabled in the game.

Filtering Mods

By entering a query term in the Filter Mods text box you can filter mods by name or by their remote id. Querying will be performed by partial matching. This means that anything that matches the entered query term will be shown. Additional filters can be applied using the following syntax:

1. search term (supports OR operator syntax)
2. Use && to split various filters
3. Use || as an or operator
4. Use -- as a negation operator. All fields support negation operator.
5. achievements:[yes|true] to include [no|false] to exclude
6. selected:[yes|true] to include [no|false] to exclude
7. source:[steam|paradox|local] (supports OR operator syntax)
8. version:[version syntax] (supports OR operator syntax)

Example:

```
my mod && achievements:s:yes && source:steam && selected:no && version:2.0
```

Or operator example:

```
my mod || another mod && source:steam || paradox
```

Negate operator example:

```
--my mod && source:--local
```

In order to reset the textbox either hit the X button next to the textbox or simply delete all text.

Sorting Mods

Mods can be sorted by toggling the installed mods column headers. The grid supports sorting by all the column headers.

Moving the mod to the Collection Mods section

In order to move a mod to the Collection Mods section just toggle the selected checkbox. By performing this action the mod will be active in game. If you deselect the checkbox the mod will be removed from the Collection Mods section as well.

If you toggle the column header checkbox all visible mods will be selected. You can use this in combination with filters textbox to enable\disable certain mods from the Collection.

Saving Options

The application will remember all of your options as soon as you perform an operation. What is remembered is the following:

1. Sort column
2. Sort column sort order
3. Filter Mods text

Determining achievement compatibility

Irony will indicate next to the mod name an icon whether a mod is achievement compatible. Broken trophy icon indicates that a mod is not achievement compatible, while an intact trophy indicates that it is achievement compatible.

Context Actions

By right clicking on a Mod you can perform the following actions:

- Open Mod File/Directory

It opens the mod location in file explorer.

- Open Mod Url

It opens the mod homepage in the browser.

- Open in Steam

It opens the mod homepage in the steam app.

- Copy Mod Url

It will copy the mod homepage url to the clipboard.

- Check new mods

Irony will rescan the whole workshop directory and mods directory to see if there are any new mods added and generate a mod file for them.

- Delete And Reload

It will delete the hovered mod descriptor and regenerate the descriptor.

- Delete All And Reload

It will delete all visible mod descriptors and regenerate the descriptors.

- Lock

It will set the readonly descriptor file flag attribute for the hovered mod.

- Lock All

It will set the readonly descriptor file flag attribute for all visible mods.

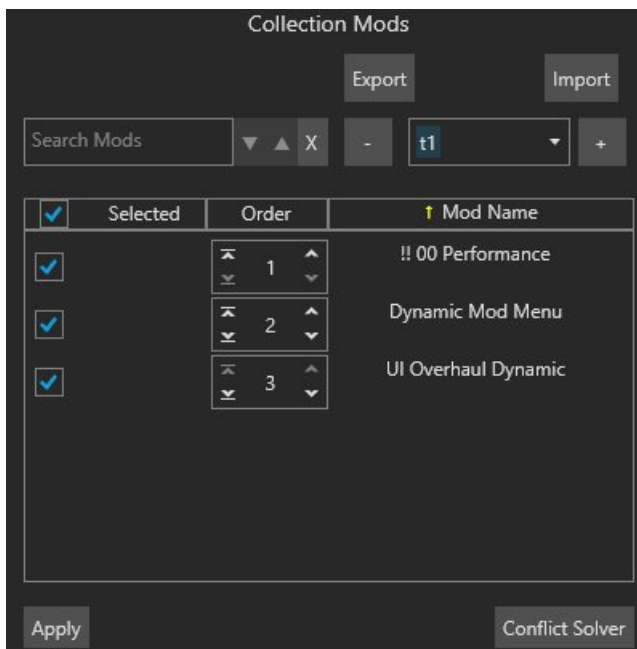
- Unlock

It will remove the readonly descriptor file flag attribute for the hovered mod.

- Unlock All

It will remove the readonly descriptor file flag attribute for all visible mods.

Collection Mods

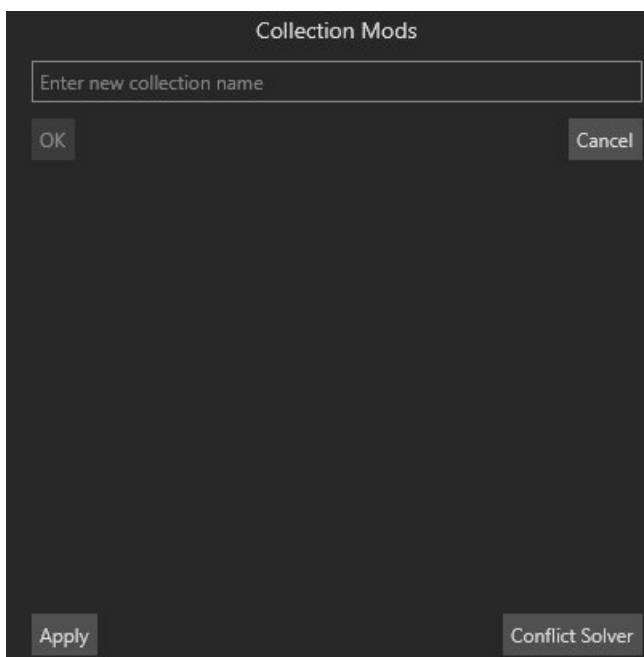


Collection Mods section is where you manage all mods that are going to be enabled in the game. In this section you can manage your load order, check for conflicts and of course apply the mod collection.

IMPORTANT: Before you do anything if the collections section dropdown is empty your first step is to create a new collection.

Creating a collection

First time you take a look at this section you will notice that the dropdown is empty. In order to be able to manage your mods press the "+" button next to the dropdown in order to create one. A new screen will be opened prompting you to enter a new collection name.



Deleting a collection

In order to delete a collection press the "-" button next to the dropdown. You will be prompted to confirm the action. This action is irreversible and will delete your collection patch mod as well.

Exporting Collection

Export button will prompt you on the location where to export your collection. Export function will:

1. Export the collection data (name, mod order)
2. Export the patch collection mod (if available)

Additional options can be accessed by toggling the  button next to the export button. The supported options are:

1. Export Order Only: This function will only export the collection name and mod order nothing else.
2. Export Whole Collection: This function will export your whole mod collection (Patch Mod, Load Order and all Mods)
3. Export to Paradox Launcher json (<2021.10): This function will export your mod order to a compliant format that Paradox Launcher can use (min version 2021.7.rc5 and max version 2021.9). Please note that Paradox Launcher only exports non local mods therefore Irony export will do the same.
4. Export to Paradox Launcher json (>=2021.10): This function will export your mod order to a compliant format that Paradox Launcher can use (min version 2021.10.rc). Please note that Paradox Launcher only exports non local mods therefore Irony export will do the same.

Notes:

- The collection which will be exported is the currently selected one in the dropdown
- Collection is exported as a zip file

Importing Collection

Import button will prompt you on the location of the exported collection. Once a collection is selected for import it will:

1. Import the collection data
2. Import the patch collection mod (if available)

Notes: If a collection with the same name exists that you're about to import you will be prompted to overwrite that collection. If you say yes to the prompt then these actions will overwrite any existing collection data by that name. What this means is that it will overwrite the existing collection mod order, and it will delete any existing patch collection mod and overwrite with the one from the package.

Importing other collections

Next to the import button you'll notice the  button. Clicking on it will allow present you with these options:

1. Import from Paradoxos. This will import a valid Paradoxos collection and its associated load order into Irony.
2. Import from the game. Irony will read the `dlc_load.json` and import its load order under a new collection named "Paradox".
3. Import from Paradox Launcher. Irony will import from the launcher SQLite database active playset; its name and load order.
4. Import from Paradox Launcher Beta. Irony will import from the launcher SQLite beta database active playset; its name and load order.
5. Import from Paradox Launcher Json. Irony will import from a exported Paradox Launcher json load order (minimum Paradox Launcher version 2021.7.rc5). Irony will accept any paradox launcher json format (<2021.10 and >=2021.10).

Duplicating a collection

Just click on the duplicate button and Irony will generate a new collection using the windows file explorer copy file naming scheme. Or in short a number will be appended to the copied collection.

Renaming a collection

Just click on the rename button and the same screen will be opened as when creating a new collection. Just enter a new name for your collection and hit the OK button. Your collection and patch collection will be modified accordingly.

Merging a collection

If you click on the merge button Irony will offer you a prompt whether you'd like to do a Basic or Compress merge.

NOTE: If a collection already exists with a desired name (Merged prefix) Irony will prompt you whether you'd like to overwrite an existing collection. Choosing "yes" means Irony will overwrite the collection. Choosing "no" Irony will append a number at the end to avoid overwriting existing collection.

Basic Merge

The following things will happen:

1. Irony will initiate file copy procedure following the mod load order and copy all the files into a huge mega mod.
2. Irony will create a new empty collection, the name will be the same as the name of the collection you're merging; with a Merged prefix being appended.
3. Irony will create a new mod under all which content will be dumped (the mod name will be the same as that of the merged collection)

Compress Merge

The following will happen:

Irony will take all mods from a collection and zip them in the mod folder individually (if you have 100 mods in a collection you'll now have 100 zipped local mods). Merged prefix will be appended to these individual mods along with the collection name. This behavior mimics <= 2.3 game behavior where all mods were zipped and is the cleanest "merge" type as the game will behave exactly as an "unmerged" collection as all mods are kept separately. The collection name will follow the basic merge naming convention.

Searching Mods

Unlike Filter Mods in Installed Mods section searching mods will highlight a match. Use the up\down buttons next to the textbox to navigate up\down. The search mods also performs a partial search by name or remote id. Additional filters can be applied using the following syntax:

1. search term (supports OR operator syntax)
2. Use && to split various filters
3. Use || as an or operator
4. Use -- as a negation operator. All fields support negation operator.
5. achievements:[yes|true] to include [no|false] to exclude
6. selected:[yes|true] to include [no|false] to exclude
7. source:[steam|paradox|local] (supports OR operator syntax)
8. version:[version syntax] (supports OR operator syntax)

Example:

```
my mod && achievements:s:yes && source:steam && selected:no && version:2.0
```

Or operator example:

```
my mod || another mod && source:steam || paradox
```

Negate operator example:

```
--my mod && source:--local
```

Sorting Mods

Mods can be sorted by toggling the collection mod name column header. The grid supports sorting by mod name column header only.

Reordering Mods

The reordering can be done by using the numeric up and down textbox buttons, by directly editing the position number in the textbox or by dragging and dropping the mod item.

Saving Options

The application will remember all of your options as soon as you perform an operation. What is remembered is the following:

1. Sort order
2. Search Mods text

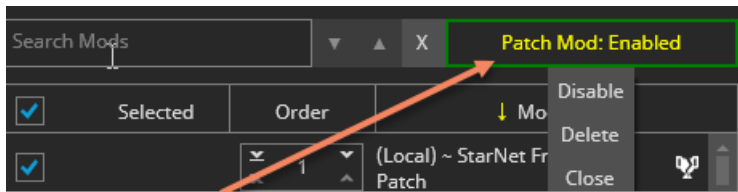
Verifying Hashes

Irony has the ability to verify game and mod file hashes which can be used for MP diagnostics in case you find yourself in a situation where the game hashes don't match. An example use case would be:

1. Via Irony you share the load order with your friends as well as the json hash report(s) (using Export Collection Hashes and or Export Game Hashes context menu option)
2. Your friends import the load order via Irony on their computers
3. Your friends use the "Verify Hashes" context menu option to validate their installs
4. If Irony report states that a certain file has any differences the player will sub/unsub from a mod and rerun the hash report
5. In more extreme cases the mod which is problematic will be manually shared between players

The following procedure can save bandwidth.

Patch Mod Controls



As of 1.12 version Irony will visually present you information that a Irony patch mod is created and allow you to either toggle on/off the patch mod or completely delete it.

Applying Collection

In order to reflect the collection on the game the collection must be applied this can be done by clicking the apply button. By doing so the changes should be reflected in the Paradox Launcher as well. In the newest Paradox Launcher versions (>= 2020.10) Irony creates a separate playset called "IronyModManager" and sets it as active in the Paradox Launcher.

NOTE: Making any changes via Paradox Launcher will not be reflected in the Irony Mod Manager. In order to make any changes to mod order you must use the Irony Mod Manager for that.

Launching game

Irony will apply the mod collection and then depending on what you set in the options section Irony will run the specified executable when you toggle the launch game command.

NOTE: If you have a steam game installation and steam is not running Irony will attempt to run steam before launching the game. This will allow steam to validate your DLC otherwise they will be unavailable to you.

Resume Game

Irony will perform the same procedures as when launching the game but it will append internally an additional game argument to signal to the game executable to load last saved game.

NOTE:

- If this option does not show, then there is no valid save game present in `continue_game.json`
- Irony will periodically verify if the resume game button needs to be shown or hidden

Conflict Solver

Clicking this button will initiate conflict solver. If this is your first time you will be prompted with a small dialog to select a mode

(Default, Advanced or Analyze Only):

Default

This is the mode that was used in 1.0 version and in this mode Irony will respect the descriptor.mod dependencies tag and ignore mods that superseded mods via this tag. Default without localization will behave the same it will only skip localization folders.

Advanced

In this mode Irony will not respect the dependencies tag and instead it will treat these as conflicts. Advanced without localization will behave the same it will only skip localization folders.

Analyze Only

This mode will just analyze conflicts and show them, it will not perform any fixes or patching. This mode in is read-only. Analyze without localization will behave the same it will only skip localization folders.

NOTE:

- Once a mod is set for the collection there is no supported way of switching to another version
- A prompt to run Analyze or selected mode will always be presented on concurrent runs
- This might take some time and might consume a lot of memory (depending on the number of mods being used). For more about conflict solver please see Conflict Solver section.
- If you ran the conflict solver once and you've modified the collection please rerun the conflict solver again to allow Irony to clean up the patch mod.

Context Actions

By right clicking on a Mod you can perform the following actions:

- Auto focus You can toggle whether the listbox will autoselect a moved object. This can result in the UI jumping to track the newly selected option.
- Copy collection to clipboard Irony will copy mod names into your clipboard so you can share with others for debugging purposes (if needed).
- Import mod order from clipboard Irony will check your clipboard for any valid mod names and if there are any present it'll import mod order from clipboard overwriting your existing mod selection and order. You can use this feature along with copy collection to clipboard; make your changes in some text editor copy that to clipboard and then toggle the import mod order option in Irony.
- Export Collection Hashes Irony will export all mods file hashes which can be used to verify that the mod files are the same on all computers (this should be used for MP syncing)
- Export Game Hashes Irony will export all game file hashes which can be used to verify that the mod files are the same on all computers (this should be used for MP syncing)
- Verify Hashes This option validates an existing hash `json` report exported using Export Game\Collection Hashes option. Once the validation finishes, and if there are any files that mismatch, it will show a complete report of mismatched mods and files.
- Open Mod File/Directory It opens the mod location in file explorer.
- Open Mod Url It opens the mod homepage in the browser.
- Open in Steam It opens the mod homepage in the steam app.
- Copy Mod Url It will copy the mod homepage url to the clipboard.

Installing New Mods

Irony Mod Manager can and will install new mods when they are detected.

Mod Installation Actions

The mods are installed on the following actions automatically:

1. At program startup
2. When a mod collection selection is changed
3. When descriptors are deleted

Manually they are performed when you toggle the "Check new mods" context menu option in the Installed Mods area.

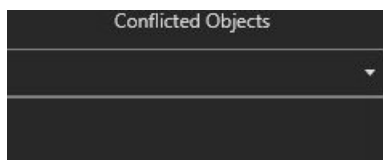
Mod Installation Locations

Mods are detected and installed from the following locations:

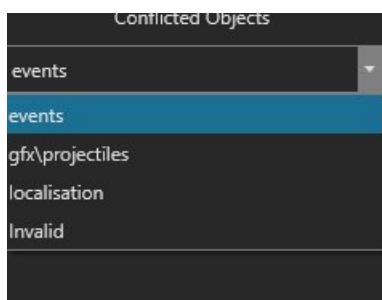
1. Steam workshop content directory
2. Documents\Paradox Interactive\Stellaris\mod directory

Irony Mod Manager will scan both of these directories for any directory and or zip file containing a file called "descriptor.mod".

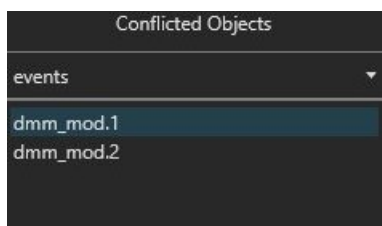
Conflicted Objects



Conflicted Objects view contains a game logic hierarchy of all detected conflicts. Selecting a conflict is easy as opening the dropdown and selecting a conflicted hierarchy folder.



After that use the listbox to select a definition.



Reset Indicators

Irony by default resets any conflicts if it detects they were modified (i.e a mod file was updated by mod author). The reset happens to bring the change to your attention. These reset conflicts are now indicated on the conflicted objects listbox.

- Red border around a conflicted object indicates that the reset conflict was previously resolved item
- Yellow border around a conflicted object indicates that the reset conflict was previously ignored item

To toggle the listbox to only show these items you can toggle this option in the Mod Filter.

Invalid Objects

From time to time Irony can detect what appears to be an invalid definition, this data is gathered by utilizing underneath excellent CWTools. Such errors indicate that a file potentially has an issue and may not perform in the game. These objects can be found in the dropdown under "Invalid" section. By selecting an invalid section you can see the details of the parsing error on the right hand side.

```
Error at line: 173 column: 3 file: gfx\particles\_disr_projectiles_particles.gfx. Please report this error to the author of the mod: Amazing Space Battles.
```

Error message says:

```
Error in gfx\particles\_disr_projectiles_particles.gfx: Ln: 173 Col: 10
(UTF16-Col: 3)
```

```
}
^
```

Note:

* The column count assumes a tab stop distance of 8 chars.

* The error occurred at the end of the input stream.

Expecting: closing brace, statement or whitespace

Other error messages:

The clause opened at ("gfx\particles_disr_projectiles_particles.gfx", Ln: 1, Col: 15) was not closed.

For mod developers

If you believe your mod was flagged incorrectly you can write a comment in your script `# Dear Irony please fallback to simple parser` and this will force Irony to not use CWTools parser but will instead switch to a less sensitive parser.

Sometimes you might also have a definition which you don't want picked up by Irony as this is a placeholder (contains only empty code). There are 2 ways to tell Irony to ignore some definitions:

1. Put anywhere in a file on a single line:

```
# Irony this is a placeholder file please ignore it
```

This will tell Irony to ignore any code in that file when it detects duplicates.

2. Selectively declare definitions by ids using csv. Put on a single line as well:

```
# Irony these are placeholder objects please ignore them: id1,id2
```

Please note that the following identifiers look for an exact match so be sure to copy and paste exactly. Finally note that sometimes Irony might ignore these comments in situations where it detects that your placeholder might win a conflict in a FIOS\LIOS situation.

Ignore Rules

Ignore rules can be accessed by pressing the Ignore Rules button in the Conflict Solver (it can be found next to the Ignore button).

Supported syntax is as follows:

```
# This is a comment it is indicated by placing the # in front of the text. This is not used by the ignore rules
. It can be useful to remind you why you placed an ignore rule there in the first place.
# Comments cannot be used inline with a rule, so a single line is either a comment or a rule
# Entries will be parsed by checking if a path starts with the following string below
events
# You can also specify more refined paths by slashes
common\defines
common/defines
# If you're a Linux user you might be used to this, which is also fine and is valid
!common\defines\subfolder
# And Irony will include this subfolder regardless of the main rule telling Irony to ignore anything
# Mods can also be excluded by name
modName:Name of the mod you'd like to exclude
# As of 1.25 you mod filter behavior changed, excluded mods are excluded always provided they match the count setting. IE. Irony if it finds a conflict which has 2 definitions from 2 different mods will not show a specified conflict if the count setting was set to 3. The new syntax:
modname:Name of the mod you'd like to exclude--count:3
# However there is also a more user friendly Mod Filter which you can use and just toggle on/off mods.
# With 1.6 Irony also supports wildcards as well, for example
localisation\*1_german.yml
# Will tell Irony to ignore all German Localisation files.
```

You can find more information on the Mod Filter/excluding mods by name [here](#)

Once you are done hit save and the conflict resolver will be refreshed and will ignore the matching rules.

You can share you ignore rule configurations [here](#)

Merge Viewer

Left Side	Right Side
Dynamic Mod Menu - main_gui_map_mode_ent UI Overhaul Dynamic - main_gui_map_mode_en IronyModManager_t1 - main_gui_map_mode_er	Dynamic Mod Menu - main_gui_map_mode_ent UI Overhaul Dynamic - main_gui_map_mode_en IronyModManager_t1 - main_gui_map_mode_er
<pre>1 guiTypes = { 2 containerWindowType = { 3 name = "main_gui_map_mode_ent 4 position = { x = 0 y = 0 } 5 size = { width = 32 height = 32 } 6 buttonType = { 7 name = "button" 8 quadTextureSprite = "GFX_button 9 position = { x = -12 y = -12 } 10 clicksound = "ui_click_map_mod</pre>	<pre>1 guiTypes = { 2 containerWindowType = { 3 name = "main_gui_map_mode_ent 4 position = { x = 0 y = 0 } 5 size = { width=32 height=32 } 6 buttonType = 7 { 8 name = "button" 9 quadTextureSprite = "GFX_button 10 position = { x = -12 y = -12 } 11 clicksound = "ui_click_map_mod</pre>
Resolve	Back

Definitions

Left Side	Right Side
AI Species Limit - dmm_mod.1 (FIOS)	AI Species Limit - dmm_mod.1 (FIOS)
Dynamic Mod Menu - dmm_mod.1	Dynamic Mod Menu - dmm_mod.1
IronyModManager_t1 - dmm_mod.1	IronyModManager_t1 - dmm_mod.1

This section will list all valid detected conflicts. It doesn't matter whether you pick a definition to be on the left or the right side.

On all text based conflicts you will notice a special definition usually called "IronyModManager_[name of the collection]" colored

green. This is a special virtual definition on which only merge actions **can be performed**. The virtual definition is created from the content of the definition colored orange in the list (unless conflict history is available). Once a conflict is resolved Irony Mod Manager will store the resolved code in a state.json file in the patch mod collection and if available this code will be used for virtual definition.

Definition color codes

- Green

This is a virtual definition on which you can perform editing actions

- Orange

This is the virtual definition which Irony has taken as the default one and copied its content (unless conflict history is available as previously stated). This definition was chosen by Irony as it believes it will be the one used by the game. On this definition you may also notice the following terms appended:

1. FIOS
2. LIOS
3. Load Order
4. Override (This is shown in advanced conflict solver mode and indicates that this definition wins due to dependencies tag)

These indicate the reason why the definition was chosen. For more information about this terminology please visit the [Stellaris Modding Wiki](#). If you are an experienced Stellaris modder and find that an item has been flagged as a false positive please report this on the [issue tracker](#). Irony only uses publicly available information from the modding wiki to make such assumptions and information can be erroneous at times.

Merge Viewer Actions (Non Virtual Definition)

The code is readonly and cannot be directly edited, this was done in order to make the tool friendly for inexperienced users. The merge process offers the following actions:

- Next\Prev conflict

Finds next\prev conflict.

- Copy text

Copies definition code to clipboard.

- Copy all

This action will copy all code blocks to the virtual definition. Consider this as a shortcut to CTRL-A and Copy this.

- Copy this

This action will copy the selected code block to the virtual definition.

- Copy this before line

This action will copy the selected code block to the virtual definition before the highlighted code block.

- Copy this after line

This action will copy the selected code block to the virtual definition after the highlighted code block.

Merge Viewer Actions (Virtual Definition)

The merge process offers the following actions:

- External Merge

If you configured an external editor on the options screen this context menu action will be visible. By clicking it Irony will launch the external diff viewer and wait for you to finish editing before taking back the changes made in the external editor. Be sure to save the changes via the external editor before toggling confirm button to signal Irony to take back the changes.

- Next\Prev conflict

Finds next\prev conflict.

- Copy text

Copies definition code to clipboard.

- Edit

Allows you to edit the virtual definition code directly in a textbox.

- Move up\down

Moves the selected code block up or down.

Resolve

Resolving a conflict will write the virtual definition code to the patch collection. The conflict will disappear from the conflicted objects tree view. Conflicted objects will only be shown again (when conflict solver is run again) under these conditions:

1. Mod collection has been changed (mod added\removed)
2. An existing definition was changed (due to a mod update)
3. Conflict was reset using "Reset Conflicts" feature

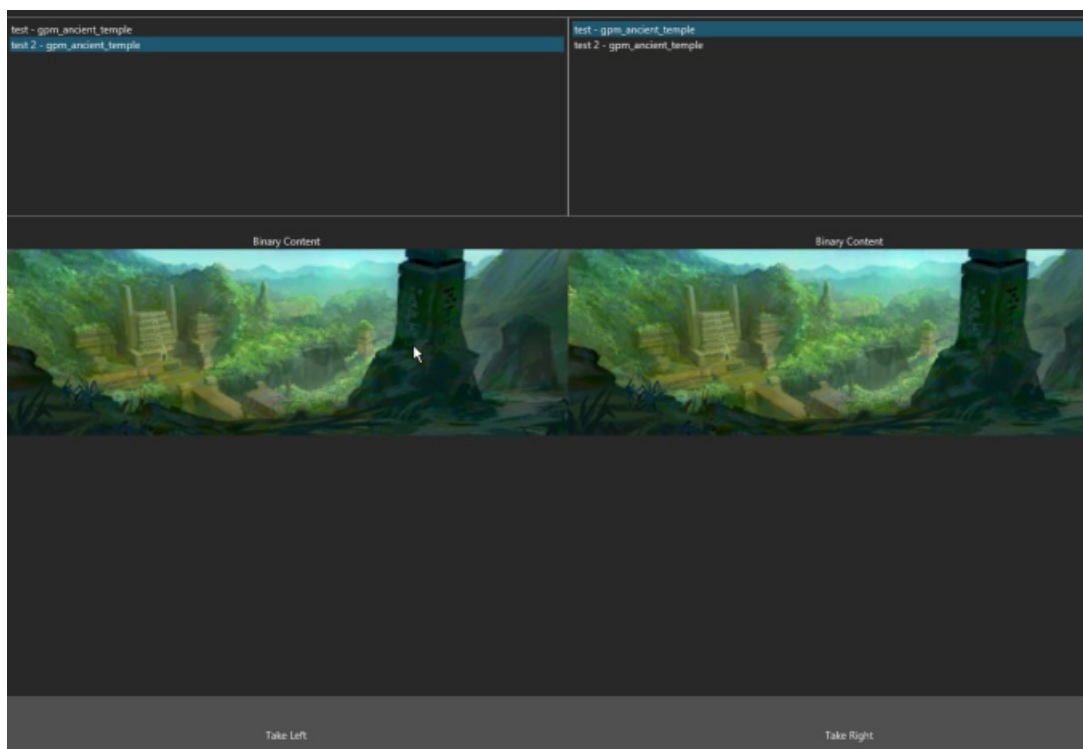
Ignore

Ignoring a conflict will tell Irony to ignore this conflict until the contents of the conflicted objects change again. After that point the conflict will be shown again. From that point on you can also ignore or resolve it again.

Back

Navigates back to the main screen.

Binary Merge Viewer



Definitions

See [Merge Viewer](#) for explanation on Definitions. The only exception is that no virtual definition is created for binary conflicts.

Resolve

In order to resolve a binary conflict first select "Take Left" or "Take Right" option. After that the Resolve button will be enabled. Clicking resolve will write the binary file to the patch collection mod.

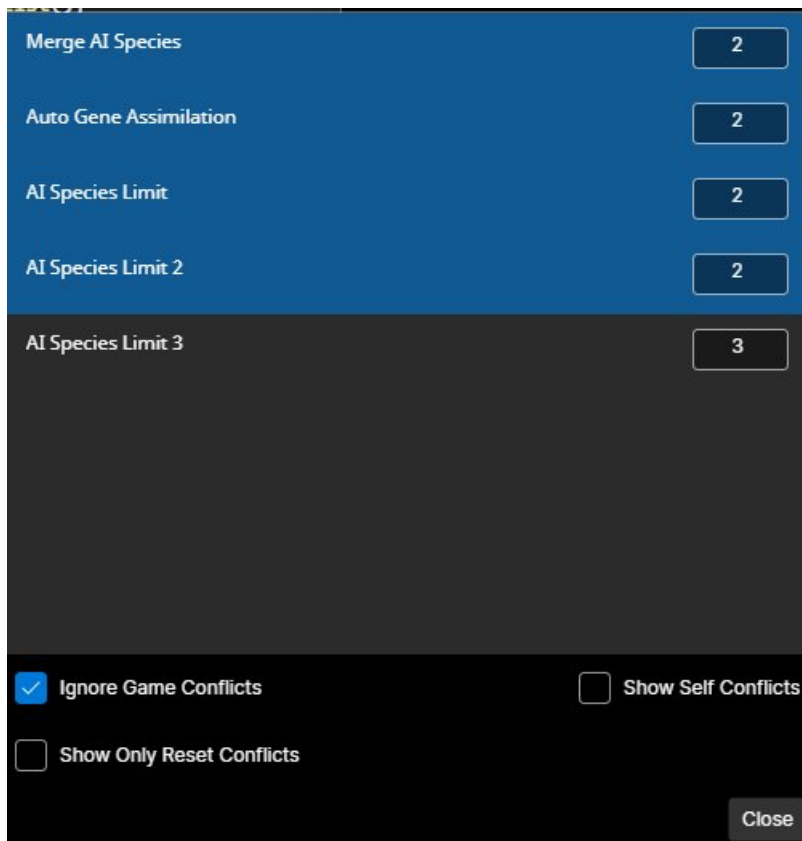
NOTE: As of version 1.6 Irony will show image previews in the binary merge viewer, in case some images are not shown please report this in the issue tracker with the mod name and definition name.

Mod Filter

This function can be accessed within the conflict solver just toggle the Mod Filter button. It allows you to turn off specific mods from showing in the conflicts. That is provided their count settings filter matches the criteria. IE. Irony if it finds a conflict which has 2 definitions from 2 different mods will not show a specified conflict if the count setting was set to 3.

Aside from this there are several options:

1. Ignore Game Conflicts (on by default). If you toggle this option off Irony will show conflicts with game objects which can increase conflict count by a significant margin. This is useful if you want to update a mod to a newer version of the game (ex. migrating mod from 3.2 version of Stellaris to 3.3 version).
2. Show Self Conflicts (off by default). If you toggle this option Irony will show conflicts in mods which conflict by itself. This means that certain mods declare the same object 2 or 3 times (be it intentional or accidental).
3. Show Only Reset Conflicts (off by default). If you toggle this option Irony will show conflicts only which were reset or will be reset (depending on the Conflict Solver mode). This option can be left on because it will only take effect if a conflict resolution was reset; otherwise it will be treated as being turned off internally. Please note that if this option is toggled on any other filters that were set in mod filters is ignored (in scenarios where the filter can take effect).



Merge AI Species	2
Auto Gene Assimilation	2
AI Species Limit	2
AI Species Limit 2	2
AI Species Limit 3	3

☒ Ignore Game Conflicts ☐ Show Self Conflicts ☐ Show Only Reset Conflicts

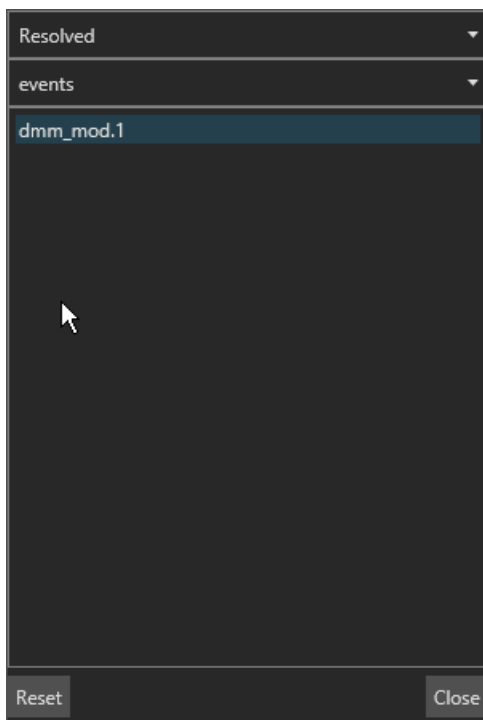
Close

Reset Conflicts

This function allows you to reset the states of previously ignored or resolved conflicts. To use:

1. Select the type of conflict you'd like to reset (Resolved or Ignored)
2. Select the folder where the conflict you're looking for is located

3. Select the definition you'd like to reset in the list box
4. Hit the reset button



Database Search

When Irony scans your mod collection structure it builds a database of all items. This database is exposed for you to search using the Database Search function. Irony indexes only the ids of the elements and not the entire code base. An example of an Id would be "anomaly.502" for example. Regardless of the number of items search should be pretty fast.



Custom Patches

This screen allows you to include your own patches in Irony Patch Mod. The recommended use case is to use this option for small scale patches which you do not believe are important enough for Irony to parse and include in its analysis. For anything else creating an external patch mod is recommended.

Custom Patches also supports patching Invalid definitions which can be found in the invalid section of the "mod folder" selector. By using this Irony will prefer to use the overridden definitions when performing an analysis of the mod collection for conflicts. Please note that once you create a custom patch for invalid objects you must re-run the conflict solver to allow the changes to take effect.

Keyboard Shortcuts

Main Screen

Collection Mods Area

Hotkeys react to hovered mod.

Hotkey	What the Hotkey Does
CTRL + UP	Move up mod
CTRL + DOWN	Move down mod
CTRL + SHIFT + Down Arrow	Move mod to top
CTRL + SHIFT + Bottom Arrow (⌵)	Move mod to bottom
CTRL + Z	Undo
CTRL + Y	Redo

Conflict Solver

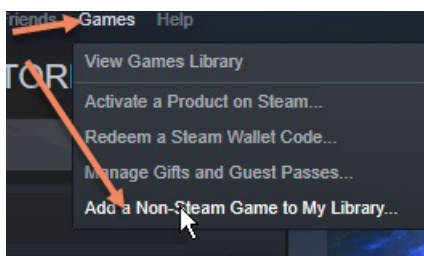
Hotkey	What the Hotkey Does
CTRL + UP	Find next conflict
CTRL + DOWN	Find prev conflict
CTRL + LEFT	Find next conflict (skips imaginary lines)
CTRL + Right	Find prev conflict (skips imaginary lines)
CTRL + SHIFT + UP	Scroll down the merge viewer
CTRL + SHIFT + DOWN	Scroll up the merge viewer
SHIFT + UP	Select next conflict
SHIFT + DOWN	Select prev conflict
CTRL + (1-0)	Select left side definition for comparison in merge viewer (1 being the first definition)
CTRL + SHIFT + (1-0)	Select right side definition for comparison in merge viewer (1 being the first definition)
CTRL + I	Ignore conflict
CTRL + R	Resolve conflict
CTRL + E	Enter edit mode (selects edit mode for the virtual definition)
CTRL + T	Copy text on the left side
CTRL + SHIFT + T	Copy text on the right side
CTRL + C	Perform copy this action
CTRL + V	Perform copy before line action
CTRL + B	Perform copy after line action
CTRL + Z	Undo

Hotkey	What the Hotkey Does
CTRL + Y	Redo
CTRL + X	Launch external editor

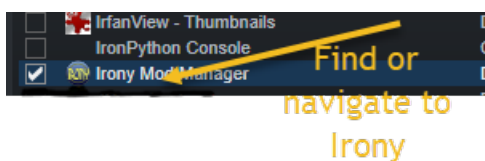
Add Irony To Steam

Let's say you want to add Irony to Steam and want Irony to startup with a specific game already selected. To achieve that follow these steps:

1. Fire up Steam
2. Toggle Games menu option
3. Click Add a Non-Steam Game to my Library



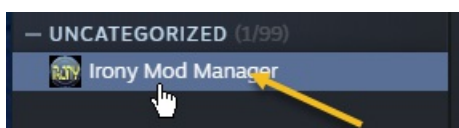
4. From the offered programs and games select Irony Mod Manager or alternatively use the browse option to navigate to Irony Mod Manager executable (IronyModManager.exe on Windows, IronyModManager on Linux\OSX)



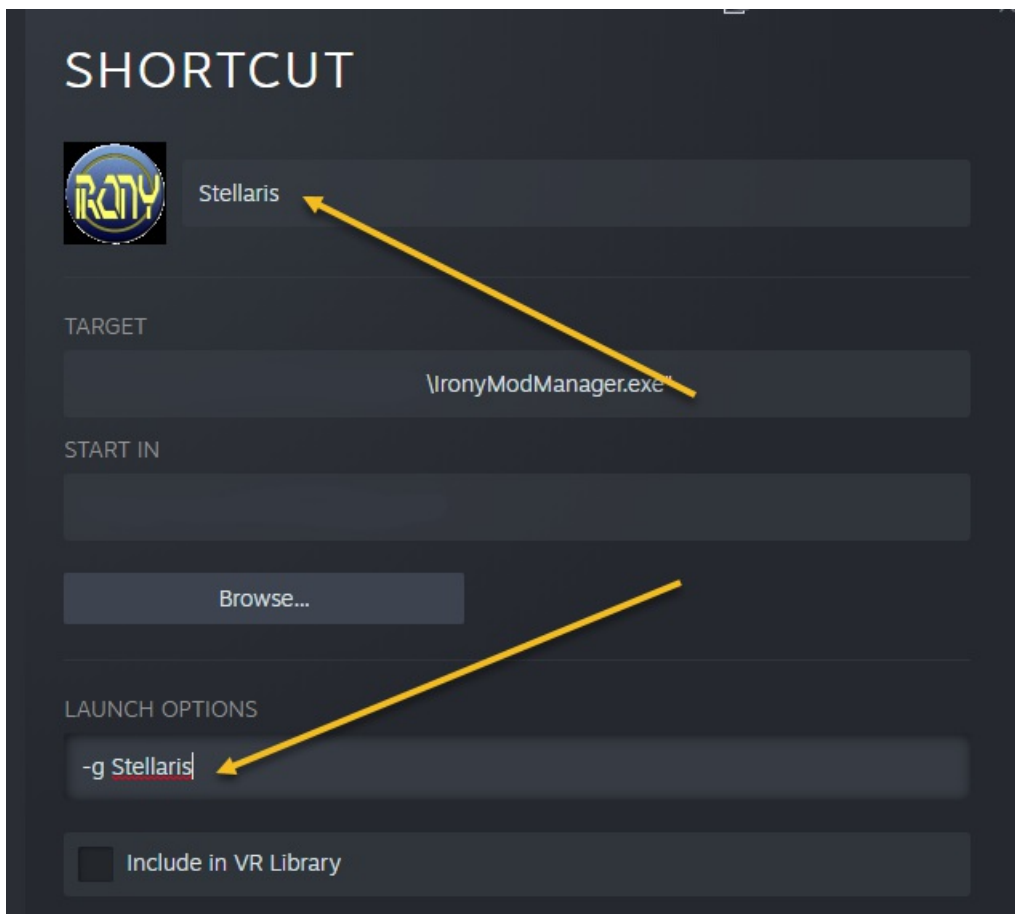
5. Once you're done click Add Selected Programs button



6. Irony Mod Manager should now be added to the list of games in your steam library. Right Click it.



7. Rename Irony Mod Manager to Stellaris and set the following launch options `-g Stellaris`



If you're doing this for another game just replace Stellaris launch option with one of the following:

- CK3
- EU4
- HOI4
- IR
- Stellaris

FAQ

Q: What mod load order does this mod manager use?

A: It uses the same load order as the official mod manager aka Paradox Launcher v2. For more information please refer the official guidelines.

##

Q: Does drag and drop support multiple items?

A: No, but it's scheduled for sometime in the future. The feature can be tracked [here](#).

##

Q: I modified the collection how do I purge the patch mod? Or how do I force Irony to clean up the patch mod? Or Irony patch mod seems to cause issues in my game!

A: If you ran the conflict solver once and you've modified the collection please rerun the conflict solver again to allow Irony to clean up the patch mod. Alternatively you can deactivate or delete the patch mod on the collections mods area.

##

Q: My Game or Stellaris is not detected on Linux. What's wrong?

A: Some Linux distros don't set an environment variable that is used to detect the LocalAppData directory. According to [this](#) `XDG_DATA_HOME` is not set.

##

Q: App Freezes on Linux. What's wrong?

A: This is caused by an Avalonia bug (nothing I can do at the moment) and on some distros the app hangs due to tooltips. In this case please edit the `appSettings.json` and set `Tooltips.Disable` flag to `true`. Alternatively you can try to edit one of the exposed `LinuxOptions` to see if different rendering methods fix the issue for you. Be sure to restart the app once you edit `appSettings.json`.

##

Q: Irony fails to launch of OSX Catalina. What seems to be wrong?

A: Please see [this ticket](#).

##

Q: Editing some conflicts is hard. Any tips?

A: Some conflicts are extremely complex and Irony might not be suited to fully handle this at times. You might want configure the External Editor in the options screen and use a third party tool to perform a diff merge (ex. KDiff, VS Code, WinMerge). Please see the Options and Merge Viewer sections for more info.

##

Q: Irony is crashing unexpectedly! Or Irony auto update doesn't work!

A: Please check whether your anti virus is blocking Irony. I could probably solve this by digitally signing the binaries but that costs quite a lot and I've already invested much of my spare time into this project as is.

##

Q: Irony is crashing on startup on Windows.

A: Try installing Microsoft Visual C++ 2017 Redistributable from [here](#). Or direct links [vc_redist.x86.exe](#) and [vc_redist.x64.exe](#).

##

Q: Can Irony upload to workshop?

A: No.

##

Q: How can I freeze my game state?

A: There are 3 options:

1. Merge -> Compress
2. Merge -> Basic
3. Duplicate collection -> Export -> Whole Collection -> Import collection (Now all your mods are local in the exact same state - zipped or unzipped just all are local)

##

Q: I'm on Mac and when I merge or export the whole collection I run out of RAM. What can I do?

A: OSX has a very low ulimit (256 by default) which means Irony cannot have many file handles open. It keeps files in memory instead while creating zips. There is now an appSettings option which you can use to toggle Irony to behave just as in Linux or Windows. There is a catch you must increase the ulimit before running Irony. To get started:

1. We need to turn this option on, go to where you have Irony and copy `appSettings.json` and rename it to `appSettings.override.json`

2. Open `appSettings.override.json` with a text editor and set `"UseFileStreams": false` to `"UseFileStreams": true` under `OSXOptions`
3. Open terminal and navigate to where you have Irony unpacked
4. Type `ulimit -n 200000`
5. Launch Irony by typing `./IronyModManager`

From this moment on to run Irony you'll need to repeat steps 4 and 5 before running Irony. While there are many ways to change ulimit this option is valid only for the current terminal session. The current terminal session means these changes are valid only while you have this specific terminal window open. Any other terminal window will not have these changes. To make changes permanent please use Google.

##

Q: Help! I'm on Wayland and Irony's window shows only a black screen on Linux.

A: There is either an issue with XWayland, an issue with your distro specifically, or you do not have XWayland installed. The two options provided below on running Irony with Wayland should fix this issue.

##

Q: I'm running a Linux distro under Wayland, how do I run Irony Mod Manager?

A: There are two options:

1. Ensure XWayland is installed via your distro's package manager (e.g. for Debian-based distros it is simply `xwayland`, while for Arch-based distros the corresponding package is `xorg-xwayland`)
2. In the `linux-x64` folder, open the file `appSettings.json` with an editor of your choice, and under the `"LinuxOptions"` block modify the setting `"DisplayServer"` from `"x11"` to either `"auto"` or `"wayland"` (`"auto"` should work but if you run into issues feel free to try `"wayland"` instead.)

##

Q: Is there a specific package of Irony Mod Manager in my Linux distro's package manager I can use that works with Wayland?

A: There may be community-built packages of Irony such as `irony-mod-manager-bin` in the Arch User Repository (AUR), but they are not guaranteed to work with Wayland (or in general), and any issues with these packages should be brought to the package maintainers themselves.

##

Privacy Policy

Software

Irony does not collect or transmit any personal information (or anything system related) without your consent. Even then no information you send is available to me as the developer of Irony. What information you transmit is your IP address which is available to GitHub.

- When you perform an update check a GET request is performed to this address:
<https://bcssov.github.io/IronyModManager/appcast.xml>.
- When you perform a version upgrade Irony downloads updates from the following address:
<https://github.com/bcssov/IronyModManager/releases/download/v{version}/{filename+extension}>

Technical details regarding updates are covered in the Options section and the only time Irony connects online is when it checks for updates (be it automatically or manually). Auto update checks can be turned off at any time in the Options screen also.

Site

I don't collect any information and any information that is gathered is available to GitHub and not me.

Removing Irony

1. If you wish to uninstall Irony first you need to determine which version of it you're using; either portable or installer based.
2. Portable: Delete the directory where you unpacked Irony.
3. Installer (Windows only): Go to add\remove programs and find Irony Mod Manager v{version-info} and select uninstall option.
4. Now you'll want to cleanup any leftovers (Irony settings and downloaded updates). These can be found in %appdata%\Mario directory. Simply remove this directory and you should be set.
5. (Optional) To remove patch mods go to your game mod directory and remove any folders and descriptor mods prefixed with IronyModManager.