Notable Files/Folders

If you were after build instructions, check out the compile Batocera page. If you were after a more general introduction to contributing code to Batocera, check out the coding rules page. Think of this more as a cheat sheet after you've got to grips with the basics.

Batocera is a very complex project, and it's easy to get lost in all the files and directories of both the source code and the live install. If you're a new developer, or just curious about which file does what, this page is a short list of some important files and their purposes.

Note: The final image location may differ between platforms, but they're usually in very similar spots. For instance, all files that pertain to config generators will always be in a `/usr/lib/python3.9/site-packages/configgen/generators/` sub-folder, but the opening path may be `/boot/` instead of `/` on some devices.

Emulator configuration

Emulator configuration files, including configuration generation. Batocera is a bit different from most static front-ends in that the configuration files for each emulator is generated upon the launch of the emulator, based on the settings the user has selected in EmulationStation (and by extension, `/userdata/system/batocera.conf`).

File purpose	Github source location	Final image location (x86_64)	Notes
Config generators (emulator defaults)	https://github.com/batocera-linux/batocera.linux/tree/master/package/batocera/core/batocera-configgen/configgen/configgen/generators	`/usr/lib/python3.9/site-packages/configgen/generators/`	The infamous Python configuration generators, bane of the end-user who's used to modifying INI and XML files directly. These are what create the necessary configuration files, controller profiles and any other necessary INI/CFG file an emulator uses upon launching an emulator. Used in conjunction with ES advanced per-system options to make these generated configs based on the options the user has set in ES. If creating a new one, don't forget to define it as well.
Emulator launch command array generator	https://github.com/batocera-linux/batocera.linux/blob/master/package/batocera/emulationstation/batocera-es-system/batocera-es-system.py	Generates `/usr/share/emulationstation/es_systems.cfg` and `es_features.cfg`	This one generates the run command (command array) that is written to `/usr/share/emulationstation/es_systems.cfg` in the final image.
Emulator config generator	https://github.com/batocera-linux/batocera.linux/tree/master/package/batocera/core/batocera-configgen/configgen/configgen/generators followed by /<system name>/<system name>Generator.py	`/usr/lib/python3.9/site-packages/configgen/generators/<system name>/<system name>Generator.py`	The "main" generator script for an emulator. This is the file that is specified by the emulator launcher specifier. This script can be separated into multiple scripts that call upon each other to separate and better organize the code.
Emulator patches	https://github.com/batocera-linux/batocera.linux/tree/master/package/batocera/emulators followed by /<emulator>	N/A, used during compiling the emulators at Fix Me!	Patches applied on top of the emulator binaries to make them work with Batocera.
EVMAPY keys files	https://github.com/batocera-linux/batocera.linux/tree/master/package/batocera/emulators followed by /<emulator>/<system>.<emulator>.keys	`/usr/share/evmapy/<system><emulator>.keys`	These are the pad2key maps that assign the Hotkey + button shortcuts to emulators that only support shortcuts via keyboard presses (and also others too because why not). Commonly referred to by the emulator config generators when they are generating the controller configuration.
Global parameters	https://github.com/batocera-linux/batocera.linux/blob/master/package/batocera/core/batocera-configgen/configs/configgen-defaults.yml	`/usr/share/batocera/configgen/configgen-defaults.yml`	The default global parameters that apply to all platforms. These configurations get applied first, followed by the architecture specific parameters. Only really used to select the default emulator for each system. Be careful, as if you select an emulator to be default that doesn't build on all platforms, compilation will fail. These have a lower priority than what the user has set in their `batocera.conf`.
Architecture parameters	https://github.com/batocera-linux/batocera.linux/blob/master/package/batocera/core/batocera-configgen/configs/ followed by configgen-defaults-[architecture_name].yml	`/usr/share/batocera/configgen/configgen-defaults-arch.yml`	The default platform parameters. These are used to specify certain platform-specific default settings to be applied globally or per system. These have a lower priority than what the user has set in their `batocera.conf` and should only be used if a particular platform should use something different from the global parameters.
Emulator makefiles	https://github.com/batocera-linux/batocera.linux/tree/master/package/batocera/emulators	N/A, only used in compiling Batocera	The makefiles that call upon the repos of the emulators included in Batocera and builds them into Fix Me!
Batocera Files (directories for emulators)	https://github.com/batocera-linux/batocera.linux/blob/master/package/batocera/core/batocera-configgen/configgen/configgen/batoceraFiles.py	Fix Me!	The paths used by emulators for their configuration files, saves, etc.
Shader sets	https://github.com/batocera-linux/batocera.linux/tree/master/package/batocera/emulators/retroarch/shaders/batocera-shaders/configs	`/usr/share/batocera/shaders/configs/`	Contains some of the preset shader sets included with Batocera, the ones available in ES's menu.
The Bezel Project systems	https://github.com/batocera-linux/batocera.linux/blob/master/scripts/linux/thebezelproject/tbp_systems.txt	Fix Me!	The list of all systems and their Bezel Project URL.

RetroArch

File purpose	Github source location	Final image location (x86_64)	Notes
RetroArch main generator	https://github.com/batocera-linux/batocera.linux/blob/master/package/batocera/core/batocera-configgen/configgen/configgen/generators/libretro/libretroGenerator.py	`/usr/lib/python3.9/site-packages/configgen/generators/libretro/libretroGenerator.py`	Calls all the other libretro-related generator files and produces the final command array.
RetroArch configuration file	https://github.com/batocera-linux/batocera.linux/blob/master/package/batocera/core/batocera-configgen/configgen/configgen/generators/libretro/libretroConfig.py	`/usr/lib/python3.9/site-packages/configgen/generators/libretro/libretroRetroarchCustom.py`, generates `/userdata/system/.configs/retroarch/retroarchcustom.cfg` upon drive initialization	RetroArch's main settings files. Includes input settings. The user can also manually edit in certain values to this file using RetroArch's "save configuration" or by hand, but most values get overwritten by this config generator.
RetroArch custom configuration generator	https://github.com/batocera-linux/batocera.linux/blob/master/package/batocera/core/batocera-configgen/configgen/configgen/generators/libretro/libretroRetroarchCustom.py	`/usr/lib/python3.9/site-packages/configgen/generators/libretro/libretroRetroarchCustom.py`, writes to `/userdata/system/.config/retroarch/retroarchcustom.cfg` as well	Appends all of Batocera's necessary settings to `retroarchcustom.cfg`. If a user has manually placed settings here that conflict with Batocera's required ones, they will be overwritten by this script.
RetroArch core options	https://github.com/batocera-linux/batocera.linux/blob/master/package/batocera/core/batocera-configgen/configgen/configgen/generators/libretro/libretroOptions.py	`/usr/lib/python3.9/site-packages/configgen/generators/libretro/libretroOptions.py`, generates and saves to `/userdata/system/configs/retroarch/cores/retroarch-core-options.cfg` upon emulator launch	The default core options that RetroArch uses based on the advanced per-system configuration set by the user in `batocera.conf` or ES. The user can also manually edit in certain values to this file using RetroArch's "save configuration" or by hand, but most values get overwritten by this config generator. If you're adding a Libretro core, you'll likely need to edit this file (unless the core has no core options).
RetroArch core option overrides	N/A, created by the user	`/userdata/system/.config/retroarch/config/`, created when the user saves core options in RetroArch	The core options saved by RetroArch. These have priority over `batocera.conf`.
RetroArch controller input generator settings	https://github.com/batocera-linux/batocera.linux/blob/master/package/batocera/core/batocera-configgen/configgen/configgen/generators/libretro/libretroControllers.py	`/usr/lib/python3.9/site-packages/configgen/generators/libretro/libretroControllers.py`	Fix Me!
RetroArch remap overrides	N/A, created by the user	`/userdata/system/.config/retroarch/config/remaps/`, created when the user saves a custom remap in RetroArch	The controller remaps saved by the user. These have the highest priority of any settings generated by Batocera and thus becomes immune to changes/updates made to the input generator (with respects to the Retropad at least).
Libretro cores	https://github.com/batocera-linux/batocera.linux/tree/master/package/batocera/emulators/retroarch/libretro	Compiled to `/usr/lib/libretro/`	The compiled cores used by RetroArch. Binaries should not be directly placed here, only compiled during the build process.
RetroArch shaders	https://github.com/batocera-linux/batocera.linux/tree/master/package/batocera/emulators/retroarch/shaders, which calls on https://github.com/libretro?q=shaders&type=&language=&sort=	`/usr/share/batocera/shaders/`	RetroArch's included shaders. Required to use shader sets properly.

EmulationStation options

The options displayed in the main menus to the user.

File purpose	Github source location	Final image location (x86_64)	Notes
ES settings configuration	https://github.com/batocera-linux/batocera.linux/blob/master/package/batocera/emulationstation/batocera-es-system/batocera-es-system.py	`/usr/share/emulationstation/es_systems.cfg`, can be appended to by the user with `/userdata/system/configs/emulationstation/es_settings.cfg`	Options the user has set from the main menu, related to only ES. Includes things like UI settings, custom game collections, etc. Used as a partial source to generate the basic `rom/<system>/` folder structure.
ES input profiles	https://github.com/batocera-linux/batocera.linux/blob/master/package/batocera/emulationstation/batocera-emulationstation/controllers/es_input.cfg	`/userdata/system/configs/emulationstation/es_input.cfg` (is added to by the user as they configure new controllers)	All the controller presets that ES automatically recognizes. The user can add more to this by configuring their controller in the **CONFIGURE A CONTROLLER** menu.
ES Main Menu labels	https://github.com/batocera-linux/batocera-emulationstation/tree/master/es-app/src/guis	Compiled into `/usr/bin/emulationstation`, not user-editable	The majority of the options shown in EmulationStation. Used in conjunction with Batocera scripts.
ES systems	https://github.com/batocera-linux/batocera.linux/blob/master/package/batocera/emulationstation/batocera-es-system/es_systems.yml	Recipe to `/usr/share/emulationstation/es_features.cfg`	The systems that ES recognizes and shows on the system list when the user has installed the appropriate ROMs. Contains some metadata about the system, such as full name and manufacture date. This is the file you'd want to edit if you want to put a comment in the `roms/<system>/_info.txt` file.
ES per-system games settings	https://github.com/batocera-linux/batocera.linux/blob/master/package/batocera/emulationstation/batocera-es-system/es_features.yml	Recipe to `/usr/share/emulationstation/es_features.cfg`	The configuration file EmulationStation uses to show which options are available for each system (in "features"). Also includes the advanced per-system settings (in "cfeatures" as their own unique entries). Used in conjunction with the config generators to define new options. The user may override this with a custom file.
Translations	https://github.com/batocera-linux/batocera-emulationstation/tree/master/locale	`/usr/share/locale/`	Files related to translation.

Batocera-specific data/meta information

Code related specifically to Batocera, not usually interchangeable with other distributions which use EmulationStation.

File purpose	Github source location	Final image location (x86_64)	Notes
Batocera user config template	https://github.com/batocera-linux/batocera.linux/blob/master/package/batocera/core/batocera-system/batocera.conf	`/userdata/system/batocera.conf`	Fix Me!
Batocera core directory	https://github.com/batocera-linux/batocera.linux/tree/master/package/batocera/core	Various, though most are in `/usr/bin/`	Where most of Batocera's core functionalities are!
Batocera scripts	https://github.com/batocera-linux/batocera.linux/tree/master/package/batocera/core/batocera-scripts/scripts	`/usr/bin/`	Where most bash scripts are! These are the files you're calling when you run commands via SSH or the terminal.
Emulator launcher specifier	https://github.com/batocera-linux/batocera.linux/blob/master/package/batocera/core/batocera-configgen/configgen/configgen/emulatorlauncher.py	`/usr/lib/python3.9/site-packages/configgen/emulatorlauncher.py`	Defines which emulators use which config generators upon launching. If you've added a new standalone system, you'll need to define it here. Also defines what conditions are sent to custom user scripts triggered by batocera-emulationstation.
Config gen definitions	https://github.com/batocera-linux/batocera.linux/blob/master/package/batocera/core/batocera-configgen/configgen/setup.py	Fix Me!	This is where all the available config generators are defined. If you've added a new generator, you'll need to define it here. This is also where you'll find the "defaults" for many emulators.
ES systems and their features	https://github.com/batocera-linux/batocera.linux/blob/master/package/batocera/emulationstation/batocera-es-system/es_systems.yml	Fix Me!, which generates `/usr/share/emulationstation/es_systems.cfg`	Source to the CFG file created for ES for its configured systems. This is the one you'll have to add a new entry to if you've added a new emulator/system to Batocera. Also contains the information used to generate the `_info.txt` files in the `/userdata/roms/` directories. You can specify which platforms this emulator is included in with the included emulators config file.
Supplementary emulator files	https://github.com/batocera-linux/batocera.linux/tree/master/package/batocera/emulationstation/batocera-es-system/roms	Fix Me!	Extra files/folders required for the system.
Batocera es_system generator	https://github.com/batocera-linux/batocera.linux/blob/master/package/batocera/emulationstation/batocera-es-system/batocera-es-system.py	Fix Me!	Generates the es_systems.cfg file. Generates roms folder and emulators folders. Generate the _info.txt file with the emulator information. Information from the emulators are being extracted from the file es_system.yml
Batocera decorations generator	https://github.com/batocera-linux/batocera.linux/blob/master/package/batocera/core/batocera-configgen/configgen/configgen/utils/bezels.py	Fix Me!	Generates the composite image used as the overlay (decoration). Recently, most of this was in libretro's configgen, but recent developments have made this independent of libretro.
Compiled component configuration	https://github.com/batocera-linux/batocera.linux/blob/master/package/batocera/core/batocera-system/Config.in	Fix Me!	This is where all the modules you want to compile with Batocera are specified. This can be used to exclude/include particular core utilities for certain platforms.
Batocera desktop shortcuts	https://github.com/batocera-linux/batocera.linux/tree/master/package/batocera/core/batocera-desktopapps	`/usr/share/applications`	The shortcuts shown in the "Applications" in the file manager's sidebar.

Batocera bundled content

All the "pre-installed" stuff that comes with a fresh Batocera install. Certain Batocera-specific media are stored as a copy at the Batocera-assets Github.

File purpose	Github source location	Final image location (x86_64)	Notes
Skeleton directory structure	Various scripts copy files to `datainit` during compilation	`/usr/share/batocera/datainit/`	The skeleton folder structure copied over to userdata partitions upon their first initialization (though some subfolders aren't). The script that handles this is at `/etc/init.d/S12populateshare`. Includes the generated (from `es_systems.yml` at build time) `info.txt` and static (always present) `readme.txt` files. When a drive is initialized, these files are copied with an underscore (_) added before their filename to the userdata partition.
ROM directory bundled files	https://github.com/batocera-linux/batocera.linux/tree/master/package/batocera/emulationstation/batocera-es-system/roms	`/usr/share/batocera/datainit/roms`, which is copied to `/userdata/roms/` on drive initialization	These are all the bundled ROMs with their metadata/images included with a freshly-flashed Batocera install. Also includes additional readme files that weren't generated by `es_systems.yml`. Emulators that fail to create their own directory can instead have it created here.
ES themes	Theme has its own repo, https://github.com/fabricecaruso/es-theme-carbon which is called from at https://github.com/batocera-linux/batocera.linux/blob/master/package/batocera/emulationstation/es-theme-carbon/es-theme-carbon.mk	`/usr/share/emulationstation/themes/`, user can append more themes by adding them to `/userdata/themes/`	Themes included with Batocera. Right now, only es-theme-carbon is included: https://github.com/fabricecaruso/es-theme-carbon
Batocera bezels (for RetroArch/MAME)	https://github.com/batocera-linux/batocera-bezel	`/usr/share/batocera/datainit/decorations/default/systems/`, where `default/` is symlinked to `default_unglazed/`	The "decorations" included and displayed in-game by default. Two files are required for a bezel, the PNG image and the respective INFO text file specifying its viewport dimensions.
Batocera controller overlays	https://github.com/batocera-linux/batocera-controller-overlays	`/usr/share/batocera/controller-overlays/`	The "system" decoration tattoos which show the current system's controls. Only one file is required for this (so far), a PNG with 225px width.
Batocera splash screen	https://github.com/batocera-linux/batocera.linux/tree/master/package/batocera/core/batocera-splash	`/usr/share/batocera/splash/`, custom splashes at `/userdata/splash/`	The default splash video/image used by Batocera. The user can choose to import their own splash video/image instead.
Background music	https://github.com/batocera-linux/batocera.linux/blob/master/package/batocera/emulationstation/es-background-musics/music	`/usr/share/batocera/music/`, custom music at `/userdata/music/`	Some bangin' tunes. The user can choose to import their own music instead.

Board patches

These are the board/platform specific patches.

File purpose	Github source location	Final image location (x86_64)	Notes
Architecture boot makefiles/binaries	https://github.com/batocera-linux/batocera.linux/tree/master/package/batocera/boot	Used in compiling Batocera; if binaries are included then `/boot/`	These are the makefiles for building the boot partition. Usually just a manner of copying files over from the below directory, but sometimes require a bit more complex instructions.
Boot partition patches	https://github.com/batocera-linux/batocera.linux/tree/master/board/batocera	Usually in boot, but patches can be applied to any location as needed	These are the patches made on a per-platform basis. Most of the time, they're just the simple files required to boot the system, but can also include special patches/configuration files needed to have particular components work on a particular platform. `fsoverlay` contains the files copied over to the boot partition. Try to keep this as small as possible, only the minimum required to boot the device, and handle the rest with packages.
Build flags	https://github.com/batocera-linux/batocera.linux/tree/master/configs	N/A, only used in compiling Batocera	Boot flags, which define what components will be built with your image depending on your chose architecture. If you're trying to port Batocera to a new architecture (device, platform, new bit mode, etc.) this is the file you'll need to edit. More information on the build configuration section on the buildroot compiling page.
Included emulators config file	https://github.com/batocera-linux/batocera.linux/blob/master/package/batocera/core/batocera-system/Config.in	N/A, used during compilation	This decides which emulators are included in which platform images. If your platform doesn't support a particular emulator or the system performs too poorly to be usable in any aspect, this is where you would go to disable it.
Boot scripts	Mixture of packages from https://github.com/batocera-linux/batocera.linux/tree/master/package/batocera/core and https://github.com/batocera-linux/batocera.linux/tree/master/package/batocera/core/batocera-scripts/scripts	`/etc/init.d/`	All the init.d scripts that run at boot time. This can be helpful in diagnosing what order Batocera is initializing modules/files in.