ABAT (A)tari Gamestation Pro (B)ox (A)rt (Tool)

ABAT is a simple, command-line, tool that lets you create ready-to-use box cover art images for the main menu of the Atari Gamestation Pro (AGP)[1] with a single command – and that only requires the name of the game you want the cover art for.

For example, the command:

```
abat.py cover “River Raid”
```

- Downloads a high-quality three-sides (back cover, spine, front cover) box art image for the Atari 2600 version of “River Raid”, from “The Cover Project”.
- Extracts the front cover from the downloaded file.
- Performs a high-quality resize to the proper size, aspect ratio and format for the AGP.
- Saves it to a file called “River Raid.png”, ready to use with the AGP.

Various options let you specify:

- What system to download box art for (Atari 2600, 5200 and 7800 … with 2600 being the default)
- What directory to place them in (default is the current directory).
- What you want the output file called. (default is the specified name as a .png file)
- Whether to keep the original high-resolution three-sides box art file or delete it.

Additional functions include:

- **Fetch** the primary, **and multiple alternate**, high-resolution, three-sides (back cover, spine, front cover) box art files.
- **Crop** the **front** or **back covers** from box art files automatically and save them to new, separate files.
- **Resize** an existing cover art image to the proper size, and aspect ratio, needed for the main menu on the AGP.

[1] Requires the custom Atari Gamestation Pro firmware, created by Atari Age user “Brad_from_the_80s”. See this thread for details.

Usage & Help

The main commands have help available.

For the tool, as a whole, the command:

```
abat.py --help
```

will display the main usage information:

```
Usage: abat.py [OPTIONS] COMMAND [ARGS]...

"Atari Gamestation Pro" (AGP) Box Art Tool (ABAT).

Creates box covers for the Atari Gamestation Pro “Atari Games" interface, with
a single command.

Separate utility functions to FETCH high-quality box art for a given Atari
system and game, CROP out front or back covers from a raw box art file, and
RESIZE covers to the native size for main AGSP interface.

Options:
  --help  Show this message and exit.

Commands:
  cover   Fetches, extracts, and sizes box cover files for the AGP.
  crop    Crops a front or back cover from the specified box art image.
  fetch   Fetches box art from "The Cover Project" for the specified game.
  resize  Resizes a cover image to the standard size for the AGP.
```

From there, you can get help for the individual commands (“`cover`”, “`crop`”, “`fetch`”, and “`resize`”, like this (replace **`cover`** with the appropriate command):

```
abat.py cover --help

Usage: abat.py cover [OPTIONS] GAME [DEST_PATH] [FILENAME]

  Fetches, extracts and sizes box cover files for the AGSP.

Options:
  -s, --system [2600|5200|7800]  Atari system  [default: 2600]
  -d, --delete                   Delete original box art file
  --help                         Show this message and exit.
```

Example Commands

Main Function – The “Cover” Command:

The main function of the tool is a simple way to have a single command get you to a “ready to use” cover file. That’s done with the “`cover`” command:

To fetch the primary cover for the Atari 5200 version of Pac-Man, have it cropped and resized ready for use as the box cover .PNG for the AGP’s main menu, have the file placed in the current directory, and to keep the high-resolution source image:

```
abat.py cover -s 5200 “pac man”
```

To do the same, but place the created file in specific folder, and give it a particular name:

```
abat.py cover -s 5200 “pac man” “/games/Atari 5200_c” pac-man.png
```

If you don’t want to keep the high-resolution art file that will be downloaded as part of creating the AGP’s file, add the “`-d`” switch:

```
abat.py cover -d -s 5200 “pac man” “/games/Atari 5200_c” pac-man.png
```

Other commands, which are intended for more manual application, and for creating scripts (e.g., in BASH, ZSH etc.) for which you want single, specific, operations include:

“Fetch”:

This will fetch box art for the specified system and game, *optionally* including “alternate” box art images (e.g., versions for different regions, eras and so on).

To get all of the box-art options for the 2600 version of River Raid, for example, include the “`-a`” option (which specifies to download alternate images as well as the main one):

```
abat.py fetch -a “River Raid” “/games/River Raid”
```

Note, this is the same as running the following, as the default system (“`-s`”) is the 2600:

```
abat.py fetch -a -s 2600 “River Raid” “/games/River Raid”
```

“Crop”:

This extracts, or crops, the front or back cover from a box-art file downloaded using the “fetch” option, or that was downloaded manually.

The full resolution of the original image is preserved during this operation.

You need to tell it the source file (the main box art file you want to extract the front or back cover from), and the name of the file in which to put the extracted/cropped cover.

For example, this command will crop out the front cover image from the file “tcp_2600_pacman_4.jpg” and place it in a new file “pac-man.png”.

```
abat.py crop “tcp_2600_pacman_4.jpg” “pac-man cover.png”
```

This is the same as the command below, since “`front`” is the default value for the “`-s`” (`side`) option:

```
abat.py crop -s front “tcp_2600_pacman_4.jpg” “pac-man cover.png”
```

“Resize”:

The “resize” command resizes the specified input image (the `source_file`) to the correct size for the AGP’s menu (236x332 pixels):

```
abat.py resize “pac-man cover.png” pac-man.png
```

NOTE: The **proper** aspect ratio for the box covers is 236x322 pixels, even though the images used by the AGP are 236x332. This can result in the cover being shifted or stretched with some source artwork (the shifting is more obvious than the stretching).

To address this, the “resize” command has a “pad” option, that will resize the original image to 236x322 and then add transparent padding at the top and bottom to bring the image up to the full 236x332 size while preserving the proper aspect ratio.

```
abat.py resize -p “pac-man cover.png” pac-man.png
```

*This padding mode is **always used** when using the “`cover`” (all-in-one) command, as it is the more “correct” way to handle the images.*

Details & Additional Considerations

The “GAME” Name/Argument

When specifying the name of a `GAME` (always the `GAME` argument for “`cover`” and “`fetch`” commands), it is important to understand there is **not** a 100% consistent, deterministic, way to determine how the files are named at “The Cover Project” automatically (most of them follow one of two patterns, which are automatically handled, but some are unique).

In general, if you type the full name of the game, without worrying about (i.e., you can either include or exclude them) punctuation, special characters or spaces, then if there IS box art for that game, it’ll be found.

So, “`ms pac-man`”, “`mspacman`”, “`Ms. Pac-Man`” are all equivalent here.

“Silent” Operation (and No Results)

Since this tool was built to operate primarily within other scripts, it currently displays no output – even if something goes wrong.

> A “verbose” error display option will be added to the next iteration, but for now it simply exits with either a return code of 0 (success) or 1 (something went wrong).

If you run a command, and get no file(s) as output, the most likely causes are:

- For “`fetch`” and “`cover`”, it was not possible to find a file at “The Cover Project” that could be matched with the supplied game name. If this happens, then *sometimes* running “`fetch`” with the “`-a`” option (fetch alternates) **will** work – this is a product of how files can be inconsistently named on the server.

  An example of this is with the 2600 version of Pac-Man. The alternate covers are named in the most common form, but the primary cover has a unique name. So just using the “cover” option will not work here.

- You specified an incorrect directory (must already exist) or filename.
- Something isn’t installed correctly.

Command Examples

All are shown in a form where the `abat.py` file takes advantage of the “shebang” entry in the script to save pre-pending every command with “`python `“ (e.g., “`python abat.py --help`” instead of just “`abat.py`”). If you have issues, put “`python `“ in front of every command.

Installation Etc.

Right now, it is assumed you know enough to install and run Python, install dependencies, and launch Python scripts/files from the command line.

If there is enough interest, I'll write up something for those not familiar with Python when I get a chance.

You'll require a current version of Python installed. Since I wrote this last night/this morning, I'm using the latest version (3.12.2). I've not tested with any other version.

I've only run this on macOS.

It *should* work fine under Linux or Windows, but I've not tried it.

It is **STRONGLY** recommended that you create a Python virtual environment for this. One of the dependencies, "Pillow", will **not** work side-by-side with the standard Python image library (PIL) – so if you install "Pillow" in a shared/global environment you may break something that relies on PIL.

Dependencies are just Click and Pillow.

You can install them with the command:

```
pip install -r requirements.txt
```