Nautilus Image Manipulator Documentation

Release 1.3

Emilien Klein

Sep 27, 2017

Contents

Version 1.3

Date Sep 27, 2017

Nautilus Image Manipulator is a Nautilus extension that lets you resize images and send them to friends and family, right from GNOME's default file manager. Just right-click on any photo or group of photos, and an option will appear that launches Nautilus Image Manipulator. A set of default settings make it easy to quickly resize pictures. You can also easily create your own profiles for later reuse.

This documentation contains information for end users, such as how to install the program and run it, and also information that would be useful for developers interested in expanding NIM's capabilities or packaging it for their preferred Linux distribution.

This documentation is a work on progress, so stay tuned!

Contents

Nautilus Image Manipulator's user documentation

Version 1.3

Date Sep 27, 2017

Nautilus Image Manipulator is a Nautilus extension that lets you resize images and send them to friends and family, right from GNOME's default file manager. Just right-click on any photo or group of photos, and an option will appear that launches Nautilus Image Manipulator.

The following pages are relevant for users of Nautilus Image Manipulator.

Installation

Nautilus Image Manipulator is written in Python and uses GTK+ version 3.

Prebuilt Packages

The preferred way to install Nautilus Image Manipulator is to get it from your distribution's software repository. Nautilus Image Manipulator is currently packaged in the following Linux distributions:

- Debian: Testing and Unstable
- Ubuntu: starting from 11.10 Oneiric Ocelot
- Arch

On Debian-based distributions, execute the following command to install Nautilus Image Manipulator:

```
sudo apt-get install nautilus-image-manipulator
```

Installing From Source

Dependencies

You will need the following software in order to run Nautilus Image Manipulator:

- Python 2 (2.6 or later)
- GTK+3
- Nautilus
- Nautilus-Python (most often packaged as python-nautilus)
- Python Imaging Library (python-imaging)
- Python binding to exiv2 (python-pyexiv2) (optional)

Note: you can use Nautilus Image Manipulator without Nautilus, but without the one feature that makes is interesting, i.e. the Nautilus extension...

Retrieving the source code

You can get the source in two ways:

- from a release tarball
- from the Bazaar source code repository:

```
bzr branch lp:nautilus-image-manipulator
```

Install the Nautilus extension

In order for Nautilus to display the option to use Nautilus Image Manipulator when right-clicking one or more images, you need to place the Nautilus extension file `nautilus_image_manipulator/nautilus-image-manipulator-extension.py` in one of these 2 directories:

- `/usr/share/nautilus-python/extensions` (for all users of the system)
- `~/.local/share/nautilus-python/extensions/` (only for your current user)

You can do this by creating a symbolic link like this:

```
ln -s nautilus_image_manipulator/nautilus-image-manipulator-extension.py ~/.local/
→share/nautilus-python/extensions
```

Don't forget to restart Nautilus for the new extension to be visible. You can do that by restarting your session, or by executing these commands:

```
killall nautilus; nautilus --no-desktop
```

Running it without the Nautilus extension

The script to run is `./bin/nautilus-image-manipulator`. You will have to pass one or more images files using the `-f` parameter. Example:

```
./bin/nautilus-image-manipulator -f ~/Images/733.jpg -f ~/Images/hyperion3_cassini_
→1024.jpg
```

Hint: add the -v parameter to display debug information. Can be useful when trying to determine what is going on.

Usage

Once Nautilus Image Manipulator is installed on your system, you'll most probably want to use it... Right?

Launching Nautilus Image Manipulator

Open Nautilus (the file manager in Gnome) and browse to a folder containing images. Select one or more, and right-click one. In the menu that appears, you will see a line that says "Resize images...":

When clicking on that menu option, Nautilus Image Manipulator will open and allow you to resize the selected images, and send them to friends and family if you so wish. This is how it looks like the first time that it's run:

Working with profiles

Resizing and sending images with Nautilus Image Manipulator is done using profiles. Those represent a set of parameters that will be applied to all the images being worked on.

Nautilus Image Manipulator comes with 4 default profiles to get you started:

- Send small images to 1fichier.com

    This profile will create images that are maximum 640 pixels wide and 640 pixels high, and the quality of the images is set to 90%. Those images are then zipped together in a file that will be sent to the file locker website http://www.1fichier.com.

- Create small images in the "resized" folder

    The images produced by this profile will be of the same size (640x640) and quality (90%) as the previous profile, but will be put inside the ./resized folder instead of being sent on the Internet.

- Send large images to 1fichier.com

    This profile will create images that are maximum 1280 pixels wide and 1280 pixels high, and the quality of the images is set to 95%. Those images are then zipped together in a file that will be sent to the file locker website http://www.1fichier.com.

- Create large images in the "resized" folder

    The images produced by this profile will be of the same size (1280x1280) and quality (95%) as the previous profile, but will be put inside the ./resized folder instead of being sent on the Internet.

Custom settings

TODO

Managing profiles
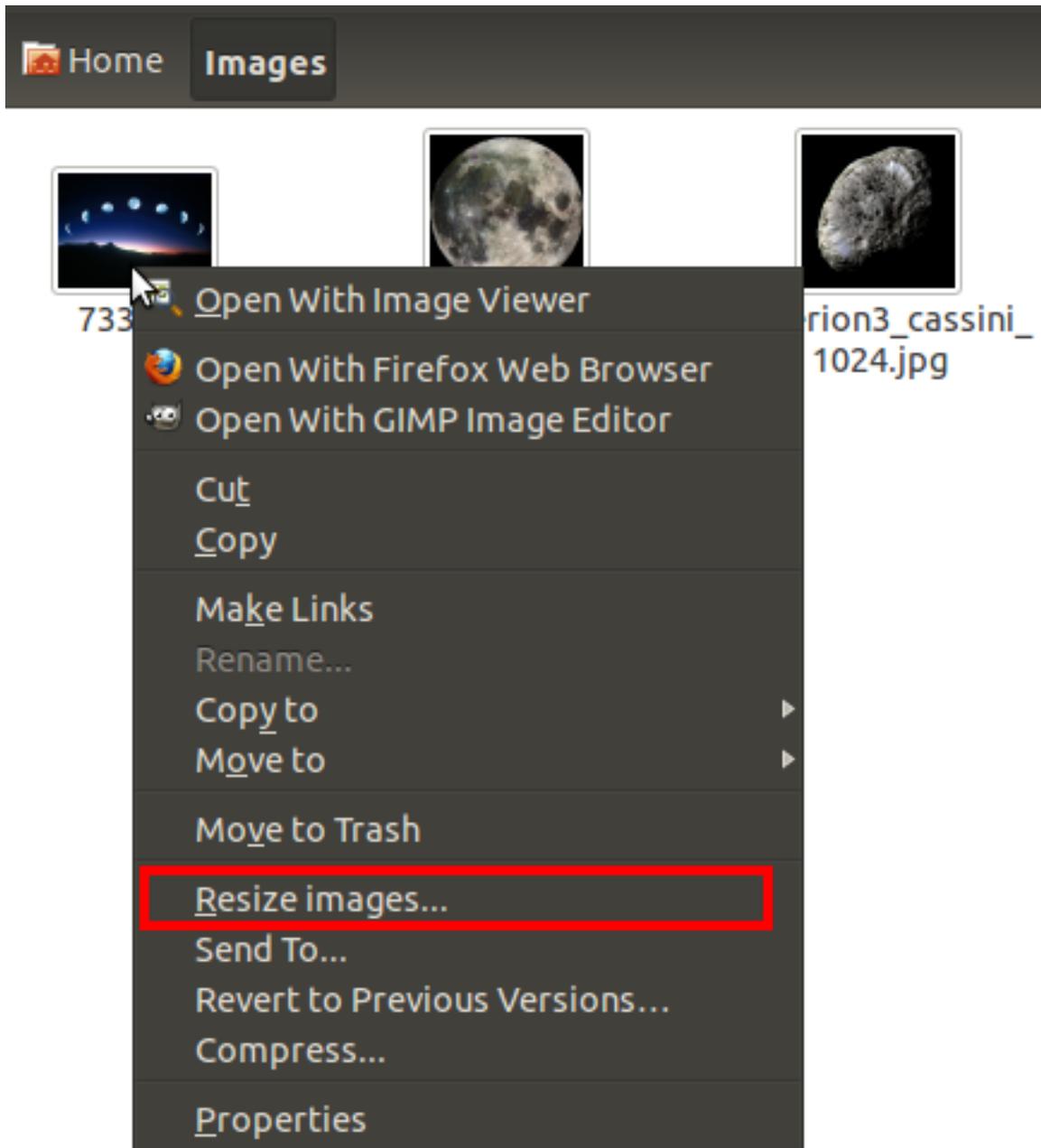
Adding a profile

TODO

Fig. 1.1: Right-clicking on images gives you the option to "Resize images..."
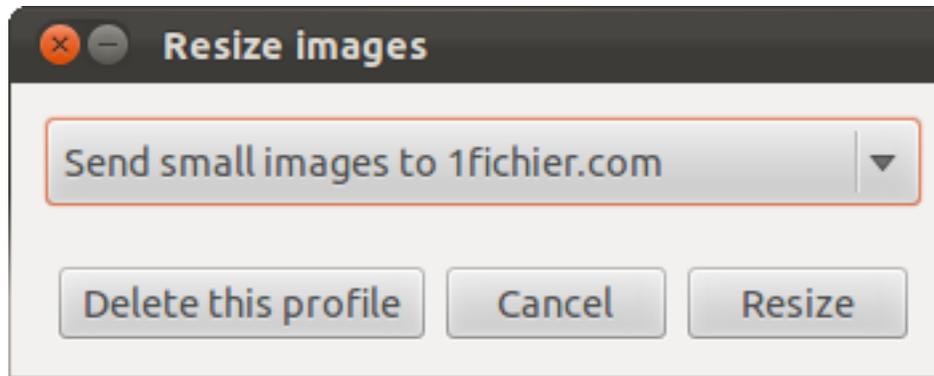
Fig. 1.2: How Nautilus Image Manipulator looks like the first time it's launched

Removing a profile

TODO

Technical documentation

Version 1.3

Date Sep 27, 2017

The following information is relevant for developers and people interested in packaging Nautilus Image Manipulator.

Working with the source code

Nautilus Image Manipulator is written in Python and uses GTK+ version 3.

Retrieving the source code

The code is hosted on Launchpad and is versioned using Bazaar. Execute the following command to get your copy of the main branch:

```
bzr branch lp:nautilus-image-manipulator
```

You can also browse the code if you'd just like to look at it online.

Running the software from source

The steps to install from source are explained here.

Modifying the GUI

The GUI is created using Glade. Open it up by running this command:

```
./extra/design.py
```

Updating the translation template

Whenever you have made changes in the GUI or the Python source code to user-facing strings, you should run the following command to update the translation template `po/nautilus-image-manipulator.pot`:

```
./extra/update_i18n.py
```

Updating the translations from Launchpad

The translation template `po/nautilus-image-manipulator.pot` is automatically read after each commit that is pushed to the main branch, and the strings are made available for translation on Launchpad. After strings have been translated, follow this process to update the translations in your branch:

- Request a download of all translated `.mo` files on Launchpad.

- You will receive an email from Launchpad containing a link to a tarball containing all the translated files, download it.

- Put the downloaded `launchpad-export.tar.gz` file in the `./po` folder

- Run the command `./extra/update_i18n.py` to unpack the files and compile them locally.

Note that aside from updating the `.mo` files that are already present in the branch, this process will also add new files for languages that are not yet versionned. These files are for languages where the number of translated strings are not yet sufficient to justify adding them to a new release. It's unlikely that a new language will be added before it has reached a translation rate of 80%-90%...

Releasing Nautilus Image Manipulator

These are the different steps to perform when releasing a new version of Nautilus Image Manipulator:

Update the version number

Change the version number in 3 places:

- `./setup.py` as "version" argument of "DistUtilsExtra.auto.setup"

- `./bin/nautilus-image-manipulator` as argument of "optparse.OptionParser"

- `./docs/conf.py` as the "version" configuration value

Update the CHANGELOG

Make sure the CHANGELOG file mentions the changes since the last release.

Create the release tarball

Place yourself in the root of the project and execute:

```
python setup.py sdist
```

This will create a tarball of the source code `./dist/nautilus-image-manipulator-<VERSION>.tar.gz`

Publish the new release on Launchpad

The previously created file should be uploaded to Launchpad as the source package of this new release:

- Go to https://launchpad.net/nautilus-image-manipulator/trunk/+addrelease
- Click on "Create milestone", enter the version number (for instance "1.3") as the name
- Enter today's date in "Date released"
- Add that release's text from the CHANGELOG in the "Changelog" section
- Click on "Create Release"
- Click on "Add download file"
- Enter "Source tarball" in "Description", select the tarball created in the previous step, and select "Code Release Tarball" as "File content type"
- Upload the new release!
- Go to https://launchpad.net/nautilus-image-manipulator/+announce and make an announcement!

Update the Debian package

Follow the Debian New Maintainers' Guide, starting with chapter 9. Updating the package.