

FITS Liberator Developers Guide

# Abstract

The Developers Guide is meant to help developers to quickly get up to speed on the FITS Liberator internals. It contains information about code design and project management.

# Glossary

AVM Astronomy Visualization Metadata

FITS Flexible Image Transport System

NSIS Nullsoft Scriptable Install System

# Contents

[1 Introduction 1](#_Toc196327467)

[2 Scientific Image Processing 2](#_Toc196327468)

[2.1 Image Sources 2](#_Toc196327469)

[3 Code Design 3](#_Toc196327470)

[3.1 Image Processing 3](#_Toc196327471)

[4 Release Management 5](#_Toc196327472)

[4.1 General steps 5](#_Toc196327473)

[4.2 Windows steps 5](#_Toc196327474)

1. Introduction

The ESO/ESA/NASA Photoshop FITS Liberator was originally a tool to import astronomical images into Photoshop. It has since evolved into a suite of tools designed to make it easier to create stunning astronomical imagery. Presently the suite contains three tools:

**FITS Liberator** is the main tool. It supports loading virtually all the FITS files in existence.

**Mosaicator** is a tool to stitch together separate image to form a larger mosaic. It uses coordinate information inside the images themselves to rotate and scale the images to fit together.

**Concatenator** is a Photoshop script that is used to merge several exposures together to form the final image while preserving and merging the metadata for each exposure.

More to come…

1. Scientific Image Processing

This chapter explains the theory behind the FITS Liberator image processing pipeline.

* 1. Image Sources

The source of the images used in astronomy

1. Code Design

In this chapter the overall code design and the design philosophy is described.

* 1. Image Processing

FITS Liberator is all about generating pretty images. To do that, the user needs some information about the original image. Statistics about the image is needed to make a good decision about which import settings to use. Figure 2.1 shows the dataflow of each pixel in the preview image. Ultimately the processing is controlled by the user interface and thereby the user. In Figure 2.1 the red boxes are blocks of data that is used to display the preview image, the histogram, and the statistics. The user can manipulate the blue boxes by means of user interface controls. If the user changes which image or plane to view, all the data in the red boxes needs to change, but if the user changes a view setting (like showing/hiding the mask for invalid pixels), only the mask bitmaps will need to change. Using the dependencies in Figure 1 one can make sure that a user action will only update the state needed rather than everything.



Figure 1: The pixel dataflow used in FITS Liberator.

* 1. Multithreading Support

FITS Liberator is designed to work with multiple processor cores at the same time. When the user initiates a long-running task a second background thread is spawned and set to work on the task. Initiating a long-running task will lock the user interface so that no further tasks can be initiated. This background thread will spawn additional threads for each processor core in the machine. Each will be given a piece of the task. Once the task is completed the background thread posts a message to the main thread that it has completed its work. This ensures that UI updates are only done on the main thread.

On Windows this is done using OpenMP while the Mac version uses Intel Threading Building Blocks. The former seems to get better performance, but does not work on Mac because the runtime implementation on Mac is utterly broken.

1. Release Management

This chapter documents all the steps developers need to go through to release a new version of FITS Liberator. FITS Liberator is released both as a manual installation where all the required files are delivered in a ZIP file and as a standalone automatic installer.

In the following all file references are relative to the root of the FITS Liberator source repository.

* 1. General steps

1. Edit the file /liberator/headers/version.h to reflect the new version number. Remember to modify both the string and numeric versions. Update the copyright information to be *2004-xxxx*, where *xxxx* is the release year.
2. Edit the *About* dialog image at /liberator/resources/About.psd to reflect the new version number and the copyright information as stated above.
3. Export a 24-bit PNG (with transparency) version of the about image overwriting the file /liberator/resources/About.png.
4. Update the release information in the files /liberator/resources/readme.txt and /liberator/resources/windows/readme.rtf. The contents of the two files should be the same with the latter file having better formatting.
   1. Windows steps
5. Build FITS Liberator using the *Setup* configuration. The setup program will end up in /liberator/binaries.
6. Create a *zip*-file with the files listed in Table 1 for the users electing to use the manual installation method.

Table .1: Manual installation files

|  |  |
| --- | --- |
| Source | Target |
| /liberator/binaries/FitsLiberator.8bi | $Photoshop/Plug-Ins/Format |
| /liberator/resources/photoshop/AVM-1-Creator.txt |  |
| /liberator/resources/photoshop/AVM-2-Content.txt |  |
| /liberator/resources/photoshop/AVM-3-Observation.txt |  |
| /liberator/resources/photoshop/AVM-4-Coordinates.txt |  |
| /liberator/resources/photoshop/AVM-5-Publisher.txt |  |
| /liberator/resources/photoshop/AVM-6-FitsLiberator.txt |  |