

# Developers

*Jeff Godwin*

The goal of this brief tutorial is to demonstrate the key concepts behind writing programs for Madagascar. Since the Madagascar API is well documented, we focus on a high level view of the development process and the core design functionality. For technical details of how to interface with the Madagascar API for a specific programming language, we refer the reader to the current online documentation mentioned at the end of this tutorial.

By the end of this tutorial, you should understand the basic design of Madagascar programs, and where to find more information about how to develop programs using your programming language of choice.

## CORE DESIGN

As previously mentioned, the main way for programs to communicate in Madagascar is through RSF files. Thus, Madagascar programs are expected to read RSF files, process them, and then write RSF files as outputs that can be then used in other RSF programs. At the highest level of abstraction, we can consider RSF programs as black boxes that simply input and output RSF files and do something to them in between. From a practical standpoint, your programs will first read RSF files from disk into memory (or progressively do so) as hypercubes. Once the file is read, the program processes the hypercubes using routines that you design or external libraries. After processing, you then write the hypercubes to RSF files on disk.

In pseudocode this process looks something like:

```
data = make_array()

input = rsf_input()

rsf_read(input,data)

. . . process data . . .

output = rsf_output()

rsf_write(output,data)
```

## CORE API

Madagascar provides core APIs for each language to ease the process of reading/writing files. Additionally, the core API provides functions to parse command line variables that can be used to control the execution of your programs.



Figure 1: A Madagascar program reads RSF files, processes them, and then outputs them at the most fundamental level.

In some languages, the API is extended to allow you to access commonly used functions from the RSF main library. For example, you can use the FFT library that is contained in Madagascar.

## PROGRAM DESIGN PHILOSOPHY

While Madagascar does not strictly have a design requirement for programs to enter the main distribution, there are some general guidelines to programs that we would like developers to follow. In particular, we would like developers to: design programs that have error handling and parameter checking, that accept command line arguments to control important parameters in the program, and write programs that are limited in scope. For example, a program that is limited in scope is a program that computes the Fourier transform of a real-valued signal and outputs a complex-valued RSF file. Conversely, a program that overreaches in its scope, would be a program that conducts a long series of processing completely in another language (say C or Fortran). You should avoid designing programs with too much scope, because you cannot fully leverage the advantages of SCons and Python, if everything is happening inside a C or Fortran program.

## FINAL THOUGHTS

With this background, and some additional information provided below, you should be able to start writing your own Madagascar programs to process data and implement algorithms that are not provided with Madagascar by default. We welcome you to the developer community, and would greatly appreciate it if you would consider releasing your programs to the community as a whole.

If you have further questions, please feel free to ask the RSF mailing lists.

## FURTHER INFORMATION

For more information about using the API for a particular language, please see: <http://ahay.org/wiki/Guide>

For more information about developing Madagascar programs in general see: <http://ahay.org/wiki/Addi>

For more information about contributing your programs see: [http://ahay.org/wiki/Contributing\\_new\\_](http://ahay.org/wiki/Contributing_new_)

For a full reference of the C API see: [http://www.ahay.org/RSF/book/rsf/manual/manual\\_html/](http://www.ahay.org/RSF/book/rsf/manual/manual_html/)