

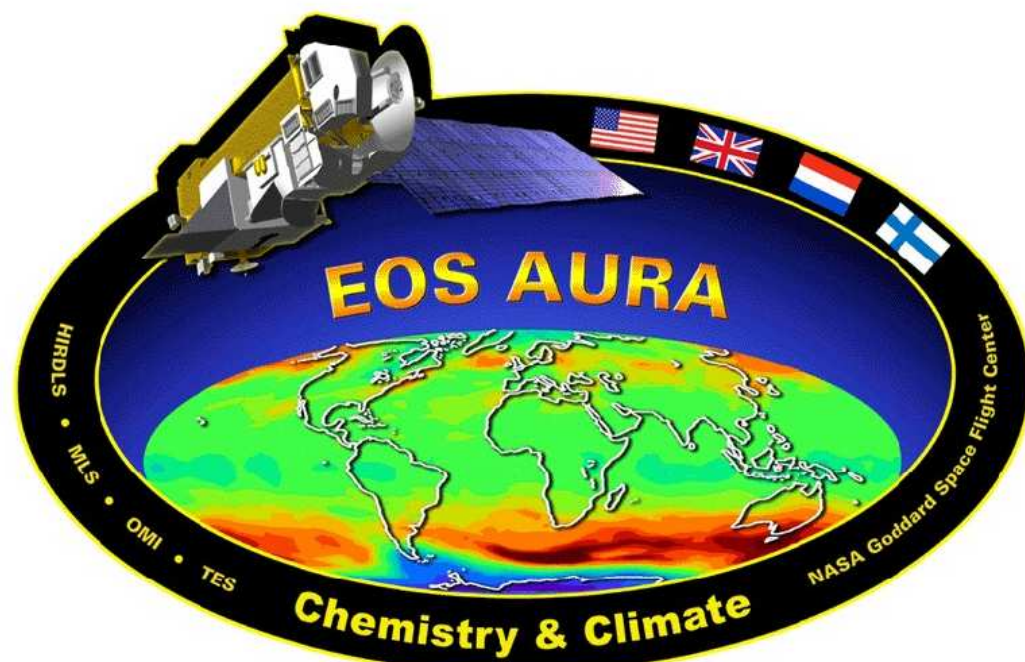
# SF23A-0043

## Overview of an Algorithm Plugin Package (APP)

Mike Linda  
SAIC  
7501 Forbes Blvd., Suite 103  
Seabrook, MD 20706  
Mike.Linda@gsfc.nasa.gov

Curt Tilmes  
NASA Goddard Space Flight Center  
Code 922  
Greenbelt, MD 20771  
Curt.Tilmes@nasa.gov

Albert J. Fleig  
PITA Analytic Sciences  
8705 Burning Tree Rd.  
Bethesda, MD 20817  
Albert.J.Fleig@gsfc.nasa.gov



### Summary

There are complexities in hosting science software for automated data production that are necessary and significant.

An Algorithm Plugin Package (APP) is science software that is combined with elements that permit the algorithm to interface with, and function within, a framework of a data processing system. The framework runs algorithms against large quantities of data.

This poster is an overview of how an APP interacts with MODAPS and OMIDAPS frameworks. Our exploration of APP structure may help builders of other production systems identify their common elements and reduce algorithm integration costs.

### APP Components

#### APP Science Source Code

Source code compiles into an APP Executable (Note 1).

#### Makefile

Makefile converts source code into an executable for the target SIPS (Note 5).

#### CI List

CI List documents version numbers of libraries and subcomponents of the APP.

#### Static APP Files

APP-unique lookup tables, coefficient files, static databases, and configuration files.

#### APP Script

APP-unique (yet mostly generic) script that ties the compiled executable into the framework and converts config files into a Runtime Control File (PCF in EOSDIS).

#### Production Rules

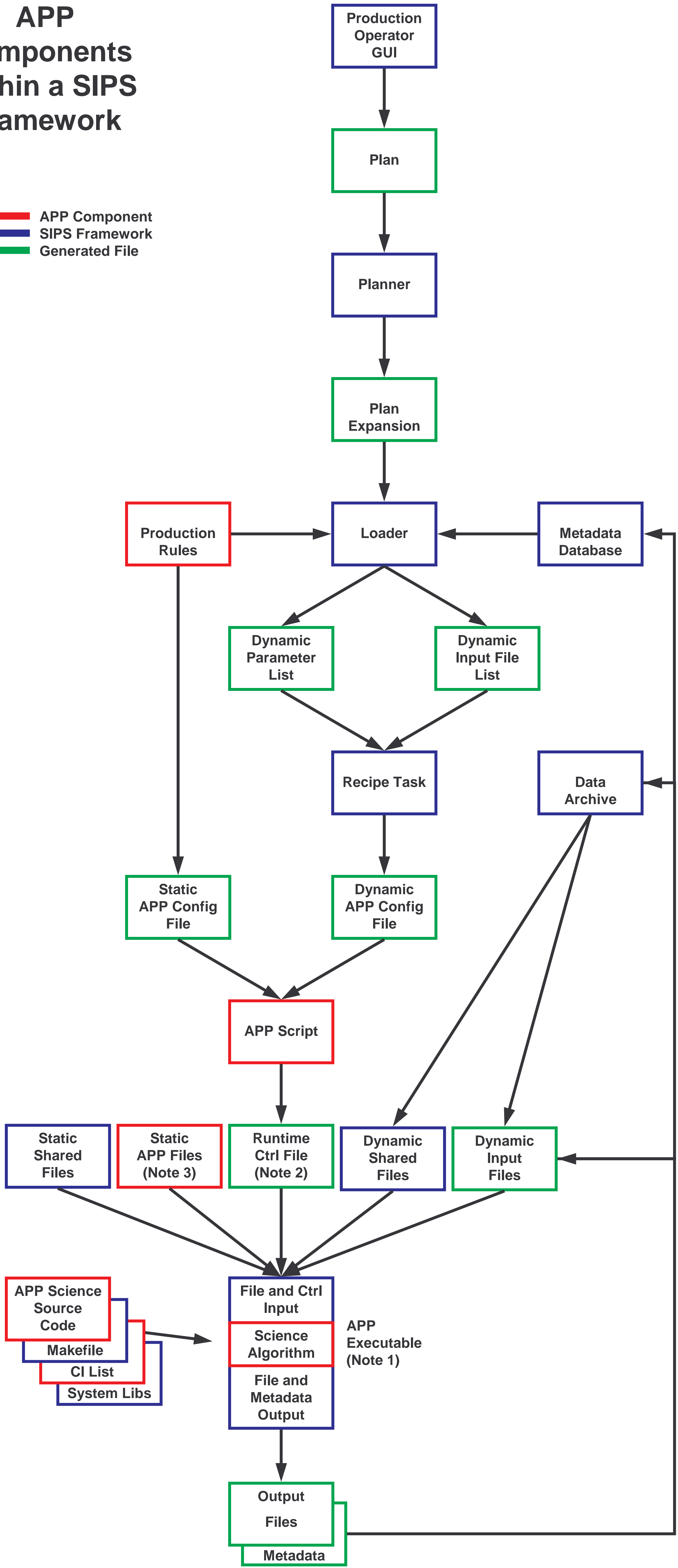
Instructions for generating lists of runtime file and parameter instances.

### Notes

- (1) APP Executable is the same as a Product Generation Executive (PGE) in EOSDIS. It is a combination of the science algorithm plus system components that facilitate I/O. For the SIPS, it is a black box.
- (2) Runtime Control File is the Process Control File (PCF) in EOSDIS.
- (3) In our system, Static APP Files are viewed as part of the APP source code. Static Files include the Metadata Configuration File (MCF) in EOSDIS.
- (4) SIPS stands for Science Investigator-led Processing System. OMIDAPS is OMI's SIPS and MODAPS is MODIS's SIPS
- (5) Although Makefiles may be APP-specific, we use a generic Makefile in OMIDAPS for most of the APPs.

### APP Components Within a SIPS Framework

APP Component  
SIPS Framework  
Generated File



### Context

The Ozone Monitoring Instrument (OMI) Data Processing System (OMIDAPS) is a framework for executing science data algorithms. The MODIS SIPS (MODAPS) is similar.

Science algorithm software that runs in our SIPS is made up of components that are integrated into an Algorithm Plugin Package (APP). The diagram on this poster illustrates how the components interact with the SIPS framework.

An APP Executable takes input files and a Runtime Control File; it produces output files and metadata. For each execution of the APP, a unique Runtime Control File must be generated. The Control File includes lists of the inputs and parameters that the APP needs for a single execution.

### APP Execution

Science data production is a series of APP executions against large numbers of input files.

1. Production begins when a SIPS operator submits a Plan (identifies a time span of desired collected data against which APPs should run).
2. Planner expands the Plan into a set of single APP execution definitions.
3. Loader interprets Production Rules and queries Metadata Database for available files.
4. Production Rules (structured abstract statements) tell the system the following:
  - Which files (relative to the output file time frame) are needed as inputs.
  - How to calculate dynamic parameters.
5. Loader calculates dynamic parameters and assembles a list of available input files.
6. Recipe Task controls APP execution. It runs the APP Script.
7. APP Script is mostly “boilerplate” that generates the Runtime Control File in a format that the APP Executable can read. The Control File includes lists of all that the APP Executable needs for a single execution.
8. APP Script retrieves needed input files, then it runs the APP Executable.
9. APP Executable uses dynamic and static files and parameters; it produces output files and metadata.
10. APP Script archives products and updates the Metadata Database.

### Conclusion

The science algorithm portion of data production is only a fraction of the overall software needed to process data.

OMIDAPS development is carried out under NASA contract number NAS5-00220.