



# INITIALIZATION TESTING

## YAX Team

*Evan Bolyen*

*Mike Deberg*

*Andrew Hodel*

*Hayden Westbrook*

## Contents

Introduction .....	1
Testing environment and purpose.....	2
Expected outcome and relevance.....	3
YAX Testing Pipeline.....	4





## Introduction

The purpose of the initialization step is to take an architectural configuration file provided to it by the user and build an exegraph that is representative of the described pipeline. It can use the exegraph to build a configuration file containing the requirements of the pipeline that can be presented to the user and populated.

The configuration file presented to the user should be a faithful representation of the pipeline described in the architectural configuration file with sections appearing in their order of operation with a breadth first pattern. When appropriate the configuration file will be populated with default or suggested parameters.

At this time the database schema is also constructed which will represent the currently configured pipeline.





## Testing environment and purpose

- YAX state system
  - This is to test just the initialization functionality of the state system and its capacity to bootstrap itself from a user provided architectural configuration file
- Testing pipeline
  - A “dirty DAG” has been created that will test multiple relationships that are expected in the final YAX pipeline.
  - These include but are not limited to:
    - Branching paths
    - Merging paths
    - Multiple independent modules
    - Multiple dependencies on one artifact
    - Multiple dependencies on one parameter
- Eight test modules
  - Modules have minimal functionality and function only to show evidence of the state systems successful use of them
  - Flow functionality that modules test includes but is not limited to:
    - Accepting parameters
    - Accepting artifacts
    - Accepting a combination of parameters and artifacts
    - Multiple modules dependent on the same artifact
    - Generate final output
    - Generate single and multiple artifacts
    - Generate a combination of final output and artifacts
- Thirteen test artifacts
  - Currently hold data for testing purposes only, can be expanded to allow to testing of later steps
- SQLite
  - Database is built to specifications of architectural configuration file.
  - No entries are created





## Expected outcome and relevance

The job of the state machine in the initialization step is to create a configuration file that the user will use to populate the required parameters of the individual modules.

This configuration file will contain:

- Parameter requests from the test modules
  - Listed in order of module use
    - Separate section for each module
  - Provided with default values where applicable

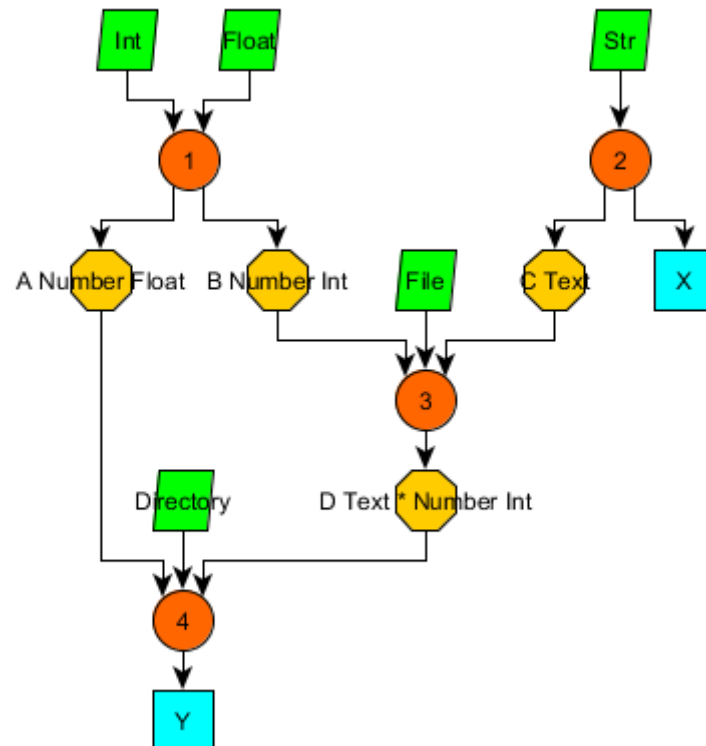
The test will also create the SQLite database schema to represent the indicated pipeline as well as tables, but no entries, in the database

Finally the test will generate the .yax directory used for internal operations of modules and components with for testing purposes will not be cleaned after operation.





## YAX Testing Pipeline



**FIGURE 1 YAX TEST PIPELINE DESCRIPTION**

In Figure 1 all five possible input parameters are utilized (represented in green). The four modules, 1, 2, 3, and 4, representing possible paths used in the YAX pipeline. There are two final output files, x and y, which are created at two different levels of the test pipeline (represented in blue). There are four internal artifacts, A, B, C, and D, which are produced and consumed by the various modules (represented in yellow).

