# Configuration File Documentation for Pipeline Manager

This document explains the proper structure and use of the configuration file for the Pipeline Manager. Each pipeline step is defined as part of a JSON array, and the configuration guides the execution of operations in sequence.

## Configuration Structure

The configuration file is a JSON object with a pipeline array containing a series of step configurations. Each step defines the operations to be performed, how input should be handled, and where output should be stored.

### Example Configuration File

{

"pipeline": [

{

"step\_name": "load\_pdf",

"module": "custom\_operations",

"function": "convert\_pdf\_to\_images",

"explicit\_input": "input\_dir/example.pdf",

"use\_intermediate\_file": false

},

{

"step\_name": "process\_images",

"module": "custom\_operations",

"function": "process\_image\_data",

"use\_intermediate\_file": true

},

{

"step\_name": "finalize\_results",

"module": "custom\_operations",

"function": "finalize\_output",

"skip\_step": false

}

]

}

## Step Configuration Fields

Each step in the pipeline is a JSON object with the following fields:

### 1. step\_name

* **Type**: String
* **Description**: A unique identifier for the step in the pipeline.
* **Example**: "step\_name": "process\_images"

### 2. module

* **Type**: String
* **Description**: The Python module containing the function to execute for this step.
* **Example**: "module": "custom\_operations"

### 3. function

* **Type**: String
* **Description**: The function within the specified module to execute.
* **Example**: "function": "process\_image\_data"

### 4. explicit\_input (Optional)

* **Type**: String (File path)
* **Description**: Specifies an external input file or path to use instead of the previous step's output.
* **Default**: null
* **Example**: "explicit\_input": "input\_dir/example.pdf"

### 5. use\_intermediate\_file (Optional)

* **Type**: Boolean
* **Description**: Indicates whether the output of this step should be written to an intermediate file.
* **Default**: false
* **Example**: "use\_intermediate\_file": true

### 6. skip\_step (Optional)

* **Type**: Boolean
* **Description**: Skips the execution of this step while passing the input data unchanged.
* **Default**: false
* **Example**: "skip\_step": true

### 7. skip\_sequencing (Optional)

* **Type**: Boolean
* **Description**: Skips the sequencing of nested data in the step.
* **Default**: false
* **Example**: "skip\_sequencing": true

## Input Handling

The pipeline supports various input types:

### 1. ****Explicit Input****

* Use the explicit\_input field to specify an input file or path for a particular step.

### 2. ****Intermediate Files****

* If use\_intermediate\_file is true, the step's output is saved as a JSON file in the intermediates folder. Subsequent steps can read from this file.
* Intermediate JSON files may include a filename field pointing to an external file (e.g., /tmp files).

## Example Use Cases

### Case 1: Convert a PDF to Images

{

"step\_name": "convert\_pdf",

"module": "custom\_operations",

"function": "convert\_pdf\_to\_images",

"explicit\_input": "input\_dir/example.pdf",

"use\_intermediate\_file": true

}

* **Purpose**: Converts a PDF file to a list of images.
* **Input**: input\_dir/example.pdf
* **Output**: Saved to an intermediate file.

### Case 2: Process Images Sequentially

{

"step\_name": "process\_images",

"module": "custom\_operations",

"function": "process\_image\_data",

"use\_intermediate\_file": true

}

* **Purpose**: Processes each image from the intermediate file generated in the previous step.

### Case 3: Finalize Results

{

"step\_name": "finalize\_results",

"module": "custom\_operations",

"function": "finalize\_output",

"skip\_step": false

}

* **Purpose**: Combines processed data into a final output.
* **Output**: Saved to the final output file.

## Notes

* Ensure all specified modules and functions are available and accessible.
* Validate file paths for explicit\_input and intermediate files.
* Use meaningful step\_name values for easier debugging and logging.

For further assistance, refer to the Pipeline Manager's logging output or contact the development team.