

Variables and Variable Managers

- [Overview](#)
- [Specifying Variables within Workflows](#)
- [Flow Chart Variables and the Flow Chart Variable Manager](#)
- [Action Variables and the Action Variable Manager](#)
- [Flow Context Variables and the Flow Context Variable Manager](#)
 - [Accessing Trigger and Action Results from the Flow Context Variable Manager - the "RESULT" Variable](#)
- [Runtime Variables](#)
 - [Transient Variables](#)
- [In Summary](#)

Overview

Flux variables and variable managers provide a powerful vehicle to pass data through a workflow to govern the processing and decision-making that takes place. Using variables can reduce the number of workflows that need to be created since an individual workflow can be made to "flex" depending upon the data it creates and encounters. Variables can be created within the context of a workflow, such as when a Flux File Trigger populates a list of file names for later processing, or variables can be supplied when the workflow first starts. Variables can be specified both within the workflow itself, or defined externally using Flux's runtime configuration.

Variable managers allow you to insert, store, and retrieve variables (i.e., data) within workflows. Variable managers provide the "pipe" to pass data between actions in a Flux workflow, such as file names, server names, or values required to run your batch and shell scripts. There are three variable managers available in Flux: flow chart, action, and flow context.

Variables in the variable managers can be inserted, stored, and retrieved using [Runtime Data Mapping](#), and simple data types such as character strings and number in the variable managers can also be easily accessed using [Variable Substitution](#). Flow chart variables can be also be set during the creation of the workflow, for example when provisioning a workflow with startup variables or running adhoc workflows.

In addition to the variable managers, Flux also provides variables to workflows at runtime from a Flux engine's runtime configuration. Runtime variables are a key feature of Flux. They allow your workflow designers to separate the specific values a workflow requires from the workflow itself. In this manner you can separate variables such as server names or file locations from the workflow itself. It is common for Flux users to have different runtime configuration files for different environments, and place these under version control. For instance, production, QA, and developer instances of Flux may all utilize the same workflows, but with different runtime configuration files.

Specifying Variables within Workflows

Variables are specified in workflows, either in configuration dialogs for Flux actions and triggers, prescripts or postscripts defined on actions and triggers, or in jobs specified in code, using the following convention: "\${Your_Variable_Name}". For runtime variables the convention is: "\${runtime Your_Variable_Name}".

Flow Chart Variables and the Flow Chart Variable Manager

Variables in the flow chart variable manager can be stored, retrieved and updated from anywhere within the workflow. The flow chart variable manager should be used when a variable is needed in several places throughout the lifetime of a workflow. Flow chart variable values are stored to the database at each [transaction break](#).

You can set flow chart variables when you create your workflow. Setting variables at the workflow creation time allows you to easily reuse values across your entire workflow while providing a single place to update the value when it needs changing. This is particularly useful when you are referencing the same values across several triggers or actions in a workflow (for example, you might create a flow chart variable to store the location of a mail server, then use [Variable Substitution](#) in your mail actions to reference that value).

Action Variables and the Action Variable Manager

Variables in an action's variable manager can only be accessed from the action itself. The action variable manager is typically only used internally within the action to store values that are vital to the action's execution. In most cases it is more appropriate to use the flow chart or flow context variable manager than the action's variable manager; generally speaking, you would only use this variable manager when writing a custom action (for more information on how the variable manager is used for custom actions, refer to the example code located in `/examples/software_developers/custom_action` under your Flux installation directory).

Flow Context Variables and the Flow Context Variable Manager

The flow context represents the flow of execution within the workflow. For example, when a workflow splits down multiple paths, Flux creates a separate flow context for each path. Variables stored or updated in flow context variables can be accessed by subsequent triggers or actions within the same flow context. Because the flow context represents the flow of execution at a particular instant in the workflow, it has a few advantages available over the flow chart variable manager: the special variable "RESULT" (which represents the result of the last action or trigger), and the ability to create transient variables (variables that are not persisted to the database).

Accessing Trigger and Action Results from the Flow Context Variable Manager - the "RESULT" Variable

Triggers and Actions that return results do so by storing the result value in the special flow context variable "RESULT". You can access this variable using normal techniques ([Variable Substitution](#), or, for more complex data types, [Runtime Data Mapping](#)).

Users who are comfortable with the Flux API can also access the result value in code.

Runtime Variables

Runtime variables and how they are configured has its own page of documentation. Click [here](#) to review.

Transient Variables

Transient variables are variables that are not stored in the database when a transaction break is encountered. Transient variables are useful when you have a significant amount of data (which can impact the performance of your workflow or Flux engine when being stored to the database), the variable contains sensitive data that you do not want in the database, or the variable does not follow the [Flux rules for persistence](#).

You can configure a trigger or action to return its result as a transient variable. You can also directly store transient variables in the flow context variable manager. Working with transient variables does require some comfort with using the Flux API; refer to the [flow context API documentation](#) for further instruction.

In Summary

Flux's variable managers and runtime configuration of variables provide a flexible and tailorable mechanism on which to develop powerful and reusable workflows. Leveraging variables requires a bit of forethought but the resultant flexibility pays benefits in reduced maintenance and increased reusability of your workflows.