

Runtime Data Mapping

What is Runtime Data Mapping?

Runtime data mapping is the process of transferring a piece of data from one place to another. runtime data mapping allows you to move data between action properties, your runtime configuration, and any of the Flux variable managers.

When Do I Need Runtime Data Mapping?

Generally, you'll only need runtime data mapping to pass data into action properties that cannot be easily set using [Variable Substitution](#). In practice, this usually means that you'll only use runtime data mapping for collections of data that are generated on-the-fly during workflow executions where the size is not known beforehand – this might be something like the collection of filenames returned from a file trigger, or the results of a database query returned from a database action.

Because variable substitution provides some visibility improvements over runtime data mapping, we recommend using variable substitution when possible; however, while variable substitution is suitable for gathering most of your dynamic data within an action, runtime data mapping provides the simplest solution for passing collections of data through your workflow.

What Is a Variable Manager?

Flux's variable managers are the storage depots for all of the data used in your workflows. Flux provides four such managers that you can use in your runtime data maps to send and receive data (for a more detailed discussion of the variable managers, see [Variables and Variable Managers](#)):

Action Property Variable Managers - Each trigger and action within your workflow has its own action property variable manager, which stores all of the property settings for the action. These are the same settings that you'll supply while building your workflow and configuring your triggers and actions – for example, the "Collection" of a For Each Collection Action, or the "Simple File Source" of a File Copy Action.

In a runtime data map, you'll use this variable manager to pass new data into your action properties – most often, to set one of the previously mentioned properties when you are working with collections of data and dynamically setting up file transfers.

Runtime Configuration - This variable manager comes from the [Runtime Configuration and Runtime Variables](#) for your engine.

Runtime variables within a runtime data map are specified by the the fully-qualified variable name (like */mynamespace/myvariable*). Within a runtime data map the namespace hierarchy IS NOT SEARCHED, which is different than variable substitution. For security reasons, the user running the workflow must have access to the variable manager for the namespace defined in */mynamespace/myvariable*.

Runtime variables are very rarely used in runtime data mapping as [Variable Substitution](#) can typically handle all runtime variable usage requirements. Variable substitution provides some visibility improvement over runtime data mapping, so we strongly suggest using variable substitution when possible.

Flow Chart Variables - Flow chart variables are variables that are created at [workflow design time](#). These are static variables that are defined to allow several actions in a workflow to share the same core set of data.

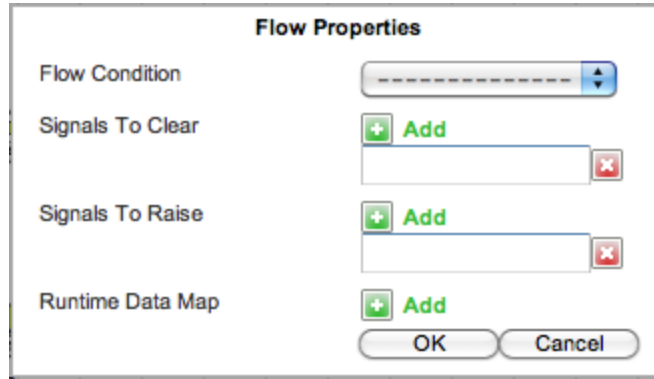
As with runtime configuration variables, we recommend using variable substitution whenever possible when working with flow chart variables.

Flow Context -The flow context is the variable manager that will most frequently hold dynamic data generated at runtime – in particular, the flow context stores the result of an action after it executes.

Most runtime data maps will transfer data from the flow context variable manager into an action property variable manager.

Usage

Using the designer, it is simple to specify runtime data mapping. Begin by either right clicking on the flow that you wish to map variables between and selecting "Edit runtime data map", or by clicking the button next to the "Runtime data map" field in a node's action properties. The window shown in the image below will appear:



Using the runtime data map editing tool is fairly straightforward. To specify a new entry, begin by clicking the "+" button. You will need to input a source variable manager and a destination variable manager to complete the map. The action variable manager is unavailable as a source variable manager, as the action will have already completed before this mapping could take place. Runtime configuration variables will be unavailable as a target in mapping, as they are specified before the engine is started and unable to be changed after. The "Type" specifies the variable manager from which to retrieve a variable, and the "name" is the name of the variable you want to retrieve.

For example, suppose you wanted to use map the results of a file exists trigger to the collection input of a for each collection element action. The type of the source variable manager would be set to "Flow Context" (where the results from nodes are put) and with the name input as RESULT.url_string_matches, RESULT.url_object_matches, RESULT.filename_matches, or RESULT.fileinfo_matches depending on which collection you want to use. The target type would be set to "Action" with "Collection" selected from the drop down box for the name. All possible entries for the type of an "Action" target will always be listed in a drop down box. Refer to the screenshot below for the proper configuration:

