

Operation Validity

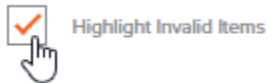
Introduction

Operations must be valid in order to be deployed. This page covers how to identify invalid operations and view the validation errors associated with them, as well as how to resolve validation errors by using a valid operation pattern. Examples of common operation arrangements are also provided.

Validation Errors

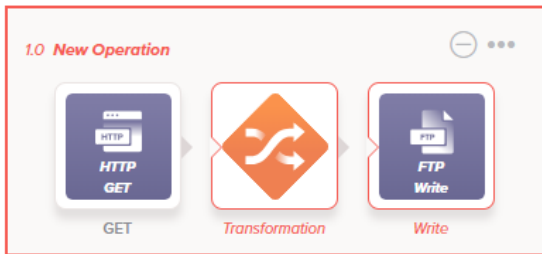
This section covers how to identify invalid operations and view the validation errors associated with invalid operations.

For new projects, invalid items are highlighted by default on the [design canvas](#), with the default selection of **Highlight Invalid Items**. To turn off this option, clear this selection:

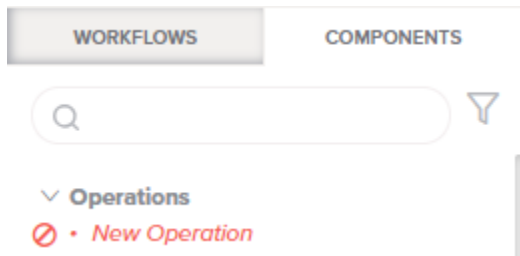


This view can also be toggled on using the **Invalid** filter in the **Components** tab of the [project pane](#) to filter on invalid items.

When this display option is selected, the names of invalid operations appear in italics and the color red on the design canvas and in the project pane. In addition, on the design canvas, blocks representing invalid operation steps are outlined with a red border and, when selected, the operation is outlined with a red border:



In the project pane, invalid operations that have an implicit error are shown with an error icon :



Click the error icon next to the operation name to display a message listing the validation errors for the operation. Operations with an implicit error are typically invalid because they do not meet established operation patterns that ensure the operation is supported and expected by the agent. These patterns are covered next under [Validation Patterns](#).

The error icon does not display if the reason the operation is invalid is because it contains other components with implicit errors. Project components used as any part of an operation must be valid for the operation to be valid. This includes components used as steps of an operation, as well as other components used in support of an operation. For example:

- A component used directly as a step in the operation, such as an activity, transformation, or script.
- An endpoint that an activity used in the operation is dependent upon.
- A component called by a script in the operation.

The validation rules depend on the type of component, and are covered collectively under [Component Validity](#).

On This Page

- [Introduction](#)
- [Validation Errors](#)
- [Validation Patterns](#)
 - [Frequently Asked Questions \(FAQ\)](#)
 - [Archive Pattern](#)
 - [Script Pattern](#)
 - [Transformation Pattern](#)
 - [Two-Target Archive Pattern](#)
 - [Two-Transformation Pattern](#)
 - [Salesforce Bulk Source Pattern](#)
 - [Salesforce Bulk Target Pattern](#)
- [Common Operation Pattern Examples](#)
 - [Script](#)
 - [Archive](#)
 - [Transformation](#)

Related Articles

- [Component Validity](#)
- [Design Canvas](#)
- [Operation Actions](#)
- [Operation Creation and Configuration](#)
- [Project Pane](#)
- [Workflow Validity](#)

Related Topics

- [Cloud Studio](#)
- [Connectors](#)
- [Operation Settings](#)
- [Operations](#)
- [Scripts](#)
- [Transformations](#)

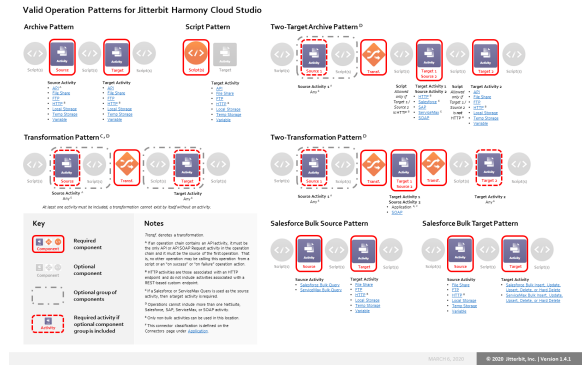
Last updated: Mar 06, 2020

Validation Patterns

Certain validation patterns must be followed for operations to be deployed to the Harmony cloud and executed on Harmony agents. These patterns ensure that all parts of a project are supported and expected by the agent:

- [Archive Pattern](#)
- [Script Pattern](#)
- [Transformation Pattern](#)
- [Two-Target Archive Pattern](#)
- [Two-Transformation Pattern](#)
- [Salesforce Bulk Source Pattern](#)
- [Salesforce Bulk Target Pattern](#)

The following diagram summarizes all valid operation patterns. Each pattern is also presented individually and described in text below the diagram, where **optional components appear in bold gray** and **required components appear in bold red**.



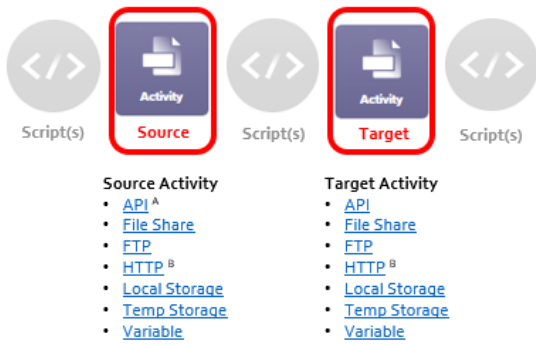
Frequently Asked Questions (FAQ)

As you design operations, it may be helpful to consider the answers to these frequently asked questions:

- **What's the difference between a source and a target?**
Activities are considered to be used as a *source* if they provide data within an operation. Activities are considered to be used as a *target* if they receive data within an operation. For additional explanation about sources vs. targets and the parts of the operation, see [Operation Creation and Configuration](#).
- **What patterns are valid with my endpoint?**
The patterns that can be used with each specific type of activity are documented within the individual activity pages under [Connectors](#). On each activity page, the specific patterns that can be used are included within the section "Next Steps," which is typically the last section on each activity page.
- **What if my use case doesn't fit a valid pattern?**
If a certain desired operation arrangement doesn't adhere to a valid pattern, you may be able to use a combination of operations that each follows a valid pattern. To do so, create each operation separately and then chain them together using [operation actions](#).
- **How can I remember these patterns?**
Operations that do not follow a valid pattern are flagged as invalid and are unable to be deployed. As you become familiar with the patterns, generalizations such as these may become apparent:
 - [File-based endpoints](#), such as FTP, HTTP, and Temporary Storage, are able to be used without a transformation.
 - [Server-based endpoints](#) for non-bulk activities, such as those for SOAP, Database, Epicor, ServiceNow, and Workday, require a transformation.
 - Scripts can be added almost anywhere.
- **Are there examples of how others have set up operations?**
For common examples using these patterns, see [Common Operation Pattern Examples](#) at the end of this page, or refer to each individual activity page under [Connectors](#).

Archive Pattern

Archive Pattern



Script(s) + **Source Activity** + Script(s) + **Target Activity** + Script(s)

In this pattern, the source and target activities can be associated with any of these endpoint types:

- [API](#)^A
- [File Share](#)
- [FTP](#)
- [HTTP](#)^B
- [Local Storage](#)
- [Temporary Storage](#)
- [Variable](#)

^A If an operation chain contains an API activity, it must be the only API or API SOAP Request activity in the operation chain and it must be the source of the first operation. That is, no other operation may be calling this operation from a script or an "on success" or "on failure" operation action.

^B HTTP activities are those associated with an HTTP endpoint and do not include activities associated with a REST-based custom endpoint.

Script Pattern

Script Pattern



- Target Activity**
- [API](#)
 - [File Share](#)
 - [FTP](#)
 - [HTTP](#)^B
 - [Local Storage](#)
 - [Temp Storage](#)
 - [Variable](#)

Script(s) + Target Activity

In this pattern, the target activity can be associated with any of these endpoint types:

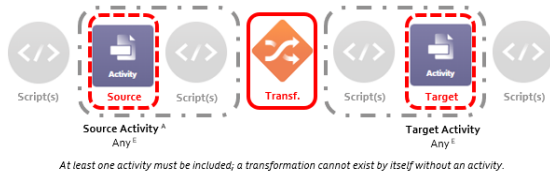
- [API](#)
- [File Share](#)
- [FTP](#)
- [HTTP](#)^B

- [Local Storage](#)
- [Temporary Storage](#)
- [Variable](#)

^B HTTP activities are those associated with an HTTP endpoint and do not include activities associated with a REST-based custom endpoint.

Transformation Pattern

Transformation Pattern ^{C, D}



Script(s) + (Group: Source Activity + Script[s]) + Transformation + (Group: Script(s) + Target Activity) + Script[s]

In this pattern, the source and/or target activities can be associated with any endpoint type, as long as at least one activity is included. A transformation cannot exist by itself in an operation without an activity.

^A If an operation chain contains an API activity, it must be the only API or API SOAP Request activity in the operation chain and it must be the source of the first operation. That is, no other operation may be calling this operation from a script or an "on success" or "on failure" operation action.

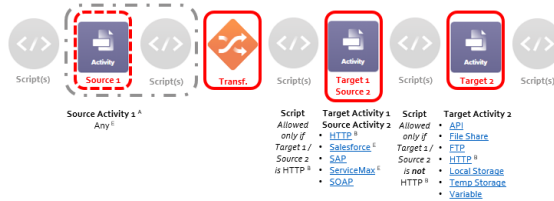
^C If a Salesforce or ServiceMax Query is used as the source activity, then a target activity is required.

^D Operations cannot include more than one NetSuite, Salesforce, SAP, ServiceMax, or SOAP activity.

^E Only non-bulk activities can be used in this location.

Two-Target Archive Pattern

Two-Target Archive Pattern ^D



Script(s) + (Group: Source Activity 1 + Script[s]) + Transformation + Script(s) + Target Activity 1 / Source Activity 2 + Script(s) + Target Activity 2 + Script(s)

In this pattern, the second target activity is used to archive a response from the middle activity, which functions both as the first target and as the second source.

The response from the middle activity passes through the raw response data to a second target without transforming it. This can be thought of as an archive or as a passing through of data (sometimes referred to as a passthrough).

The source and target activities must be associated with certain endpoint types depending on where they are used in the pattern:

- **Source Activity 1 ^A:** If used, the first source activity can be associated with any endpoint type.
- **Target Activity 1 / Source Activity 2:** The first target activity (also referred to as the second source activity) can be associated with any of these endpoint types:
 - [HTTP](#) ^B
 - [Salesforce](#) ^E
 - [SAP](#)
 - [ServiceMax](#) ^E
 - [SOAP](#)
- **Target Activity 2:** The second target activity can be associated with any of these endpoint types:
 - [API](#)
 - [File Share](#)
 - [FTP](#)

- [HTTP](#)^B
- [Local Storage](#)
- [Temporary Storage](#)
- [Variable](#)

^A If an operation chain contains an API activity, it must be the only API or API SOAP Request activity in the operation chain and it must be the source of the first operation. That is, no other operation may be calling this operation from a script or an “on success” or “on failure” operation action.

^B HTTP activities are those associated with an HTTP endpoint and do not include activities associated with a REST-based custom endpoint.

^D Operations cannot include more than one NetSuite, Salesforce, SAP, ServiceMax, or SOAP activity.

^E Only non-bulk activities can be used in this location.

Two-Transformation Pattern

Two-Transformation Pattern^D



Script(s) + (Group: **Source Activity 1** + Script(s)) + **Transformation 1** + **Target Activity 1 / Source Activity 2** + **Transformation 2** + Script(s) + **Target Activity 2** + Script(s)

In this pattern, the second transformation is used to take the response from the middle activity (which functions both as the first target and as the second source), transform it, and then optionally write it to a second target.

The source and target activities must be associated with certain endpoint types depending on where they are used in the pattern:

- **Source Activity 1**^A: If used, the first source activity can be associated with any endpoint type.
- **Target Activity 1 / Source Activity 2**: The first target activity (also referred to as the second source activity) can be associated with any of these endpoint types:
 - [Application](#)^{E, F}
 - [SOAP](#)
- **Target Activity 2**^E: If used, the second target activity can be associated with any endpoint type.

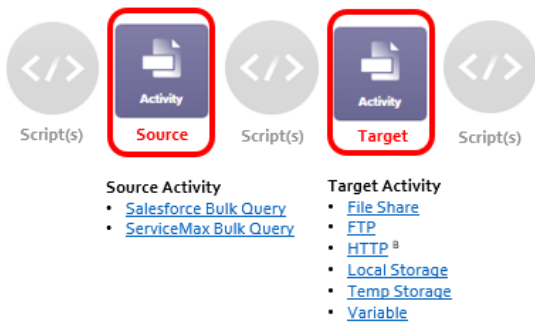
^A If an operation chain contains an API activity, it must be the only API or API SOAP Request activity in the operation chain and it must be the source of the first operation. That is, no other operation may be calling this operation from a script or an “on success” or “on failure” operation action.

^E Only non-bulk activities can be used in this location.

^F This connector classification is defined on the [Connectors](#) page under [Application](#).

Salesforce Bulk Source Pattern

Salesforce Bulk Source Pattern



Script(s) + **Source Activity** + Script(s) + **Target Activity** + Script(s)

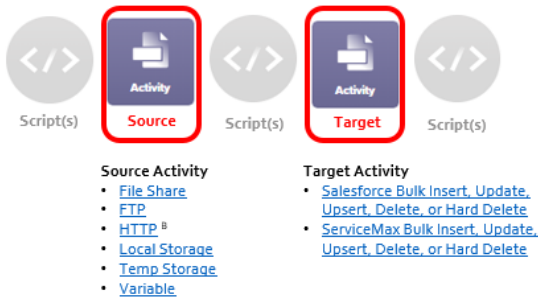
In this pattern, the source activity must be a [Salesforce Bulk Query activity](#) or a [ServiceMax Bulk Query activity](#). The target activity can be associated with any of these endpoint types:

- [File Share](#)
- [FTP](#)
- [HTTP](#)^B
- [Local Storage](#)
- [Temporary Storage](#)
- [Variable](#)

^B *HTTP activities are those associated with an HTTP endpoint and do not include activities associated with a REST-based custom endpoint.*

Salesforce Bulk Target Pattern

Salesforce Bulk Target Pattern



Script(s) + **Source Activity** + Script(s) + **Target Activity** + Script(s)

In this pattern, the source activity can be associated with any of these endpoint types:

- [File Share](#)
- [FTP](#)
- [HTTP](#)^B
- [Local Storage](#)
- [Temporary Storage](#)
- [Variable](#)

The target activity can be associated with any of these endpoint types:

- [Salesforce Bulk Insert](#)
- [Salesforce Bulk Update](#)
- [Salesforce Bulk Upsert](#)
- [Salesforce Bulk Delete](#)
- [Salesforce Bulk Hard Delete](#)
- [ServiceMax Bulk Insert](#)
- [ServiceMax Bulk Update](#)
- [ServiceMax Bulk Upsert](#)
- [ServiceMax Bulk Delete](#)
- [ServiceMax Bulk Hard Delete](#)

^B *HTTP activities are those associated with an HTTP endpoint and do not include activities associated with a REST-based custom endpoint.*

Common Operation Pattern Examples

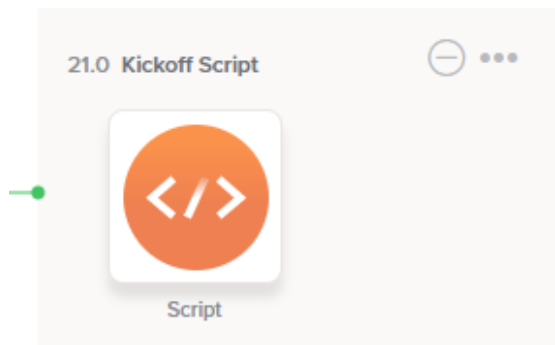
This section provides several common examples of operations configured using a valid operation pattern. These examples are simplified to demonstrate the basic building blocks of commonly built operations and are not intended to cover all possible combinations. Refer to [Validation Patterns](#) earlier on this page for all possible arrangements.

Script

An operation can simply contain a single [script](#). In this case, all actions in the operation are performed by the script.

This pattern is typically used at the beginning of a workflow to create a "launch script" that runs one or more of several branch operations based on a system variable or other data value.

Scripts may also be used anywhere in an operation to perform complex calculations that require custom logic.

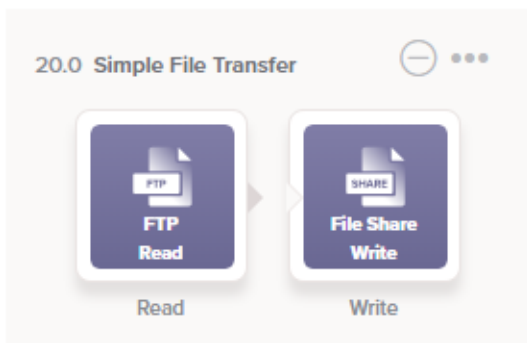


Archive

If all you need to do is to move files from a data source to a target, you can use an operation without a transformation.

The first activity is used as the source, providing data within the operation. The second activity is used as the target, receiving data within the operation.

For archiving data, the source and target activities are limited to those that interact with files. Instead of a source activity, you could also use a script.

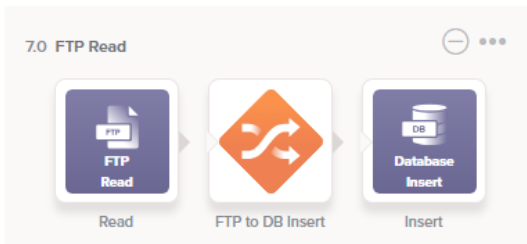


Transformation

Operations that make use of [transformations](#) serve a variety of use cases. For example, you can create operations that (1) transform data from a source to a target, (2) transform data, then write the response to another target, or (3) transform data from a web service call by converting the application call's response and writing to a target.

Operation Transforms Data from Source to Target

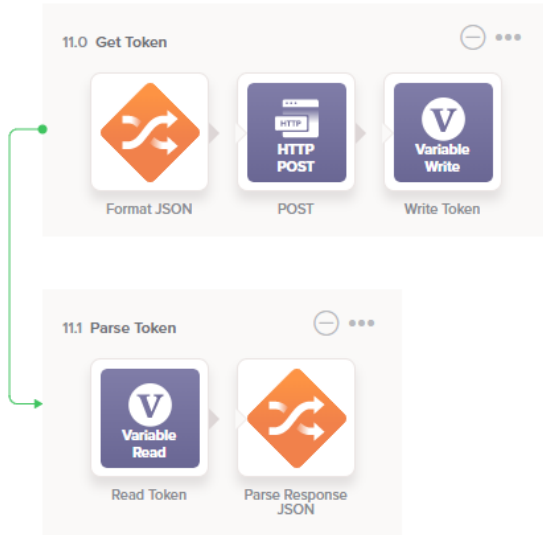
This example pulls data from a source activity that is then transformed and written to a target activity:



Operation Transforms Data, then Writes Response to Another Target

In this example, when an HTTP activity is used as the target of the transformation, such as HTTP PUT or HTTP POST, an additional target activity can be placed directly after it, or after another transformation, in order to write the HTTP response to another target.

This operation arrangement, used for getting a REST token, shows the HTTP response being written to a global variable for use in the next operation:



Operation Calls a Web Service

This example is for updating an application such as [Salesforce](#) through the application's API, or calling a [SOAP](#) web service method. The application call typically includes both a request and response data structure.

In this arrangement, the operation has three parts:

- The first source activity and transformation to construct the request data structure.
- The application call itself, which functions as the first target activity and second source activity.
- A second transformation and target to convert the application call's response and write the data or file to a target.

Application Call



SOAP Call

