

Component Dependencies, Deletion, and Removal

Introduction

Viewing the dependencies of a component lists the other project components that the component is dependent on. If other components are dependent on the component, those dependencies must be removed before a component can be deleted.

Viewing dependencies of and deletion of workflows and operations is covered on these pages:

- **Workflows:** [Workflow Dependencies and Deletion](#)
- **Operations:** [Operation Dependencies and Deletion](#)

Viewing dependencies for other project components, such as activities, transformations, scripts, schemas, project variables, global variables, email notifications, and schedules, is also available. Each of these components can also be deleted or removed from an operation using the same method, except for global variables, which are deleted by deleting the portion of a script where they are defined.

Definitions

It is important to understand the terminology when discussing dependencies. Two distinct phrases are used: "dependent on" and "dependency of." In these examples, Component A is said to be **dependent on** Component B. Component B is said to be **a dependency of** Component A:

- **Dependent on:** If a component is dependent on another component, it needs that component in order to function properly. A component that is dependent on another component cannot stand alone without that component. When Component A needs Component B in order to execute successfully, Component A is said to be dependent on Component B. Another way to say this is that Component A depends on Component B.
- **Dependency of:** If a component is a dependency of another component, it is needed by the first component in order for the first component to function properly. A component that is a dependency of another component is the component that is needed by another. When Component A needs Component B in order to execute successfully, Component B is said to be a dependency of Component A.

Viewing Dependencies, Deleting, or Removing

The options to view dependencies or delete most project components are accessible from the **Components** tab or **Workflows** tab of the [project pane](#):

- **Components:** In the **Components** tab of the project pane, hover over a component name and click the actions menu icon **☰** to open the actions menu. From the menu, select **View Dependencies** or **Delete**.
- **Workflows:** Components used as operation steps can have their dependencies viewed or can be deleted from **Workflows** tab of the [project pane](#). Hover over an operation name and click the actions menu icon **☰** to open the actions menu. From the menu, select **View Dependencies** or **Delete**.

In addition, unused project components, or components that aren't referenced by any other components in the current project, can be deleted in bulk. These components are displayed with a broken link icon  in the **Components** tab. To delete unused components in bulk, hover over the component category, then click the actions menu icon **☰** to open the actions menu. From the menu, select **Delete unused**:



Instead of deleting components used as operation steps from the project, you can remove them from being used in operation without deleting them. This can be done from the [design canvas](#). Within the operation, click on the component block to open the actions menu. From the menu, select **Remove**:

On This Page

- [Introduction](#)
- [Definitions](#)
- [Viewing Dependencies, Deleting, or Removing](#)
 - [View Dependencies](#)
 - [Delete](#)
 - [Remove](#)

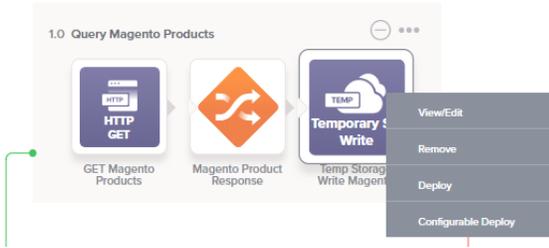
Related Articles

- [Component Reuse](#)
- [Design Canvas](#)
- [Operation Dependencies and Deletion](#)
- [Project Pane](#)
- [Workflow Dependencies and Deletion](#)

Related Topics

- [Cloud Studio](#)
- [Cloud Studio User Interface](#)
- [Connectors](#)
- [Notifications](#)
- [Operations](#)
- [Plugins](#)
- [Project Components](#)
- [Schemas](#)
- [Scripts](#)
- [Transformations](#)
- [Variables](#)
- [Workflows](#)

Last updated: Dec 11, 2019



View Dependencies

Upon selecting the **View Dependencies** option, the dependency view opens in the **Workflows** tab of the project pane:

Magento 2 Migration

Env: Production

WORKFLOWS | COMPONENTS

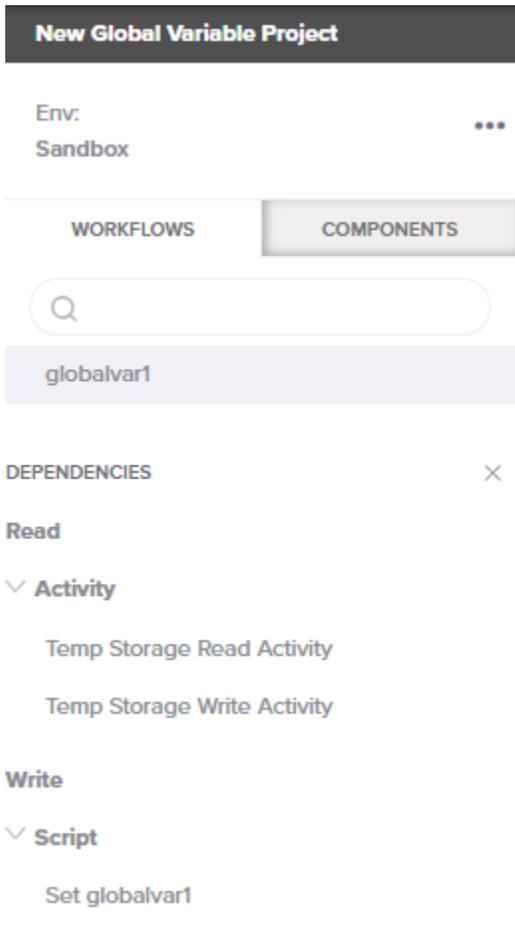
1.2 Magento Product Response

DEPENDENCIES

- Workflow
 - Magento Products to Postgres
- Operation
 - Query Magento Products
- Activity
 - GET Magento Products
 - Temp Storage Write Magento Product Resp...

In dependency view, the name of the selected component appears along the top, below the existing search and filter capabilities.

The component name is followed by a list of **Dependencies** that the component is dependent on. This list is organized by categories such as **Workflow**, **Operation**, and **Activity**. These categories will vary depending on the type of component being viewed. For global variables, the categories are further classified under **Read** or **Write** to indicate the access type of a particular variable reference:



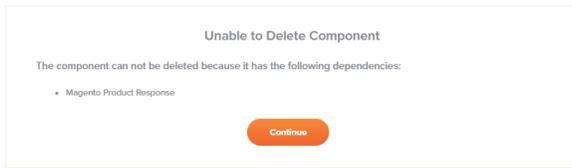
Each category can be expanded or collapsed using the disclosure triangles \vee \triangleright .

Only dependencies of the component that are a single level deep are displayed. For example, when viewing dependencies of the transformation "Magento Product Response," while the listed activity "GET Magento Products" is a dependency of the transformation, other components that are a dependency of the activity are not displayed. Neither are components that are dependent on the transformation displayed (that is, other components that the component itself is a dependency of are not listed).

To close out of dependency view, click the close icon \times .

Delete

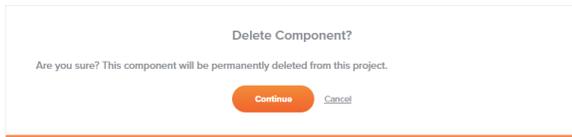
Upon selecting the **Delete** option, if there are other project components that are dependent on the component being deleted, those components will be listed:



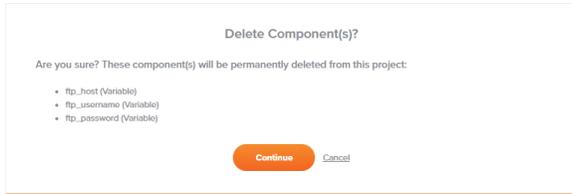
Note that the text in this message incorrectly states that dependencies of the component being deleted are listed. This is not true, and in fact any other components that are **dependent on** the component are actually what is actually listed. Click **Continue** to return to the project designer without any action taking place.

To make it so the component can be deleted, you must modify the project so that the component is not dependent on these other components. For example, if activity Component B needs transformation Component A to function properly, first remove transformation Component A from the operation and then try again to delete the activity Component B.

If there are no other components that have the component being deleted as a dependency of any component, a message will ask you to confirm you want to delete the component:



When deleting components in bulk, each component will be listed:



Click **Continue** to permanently delete the component(s).

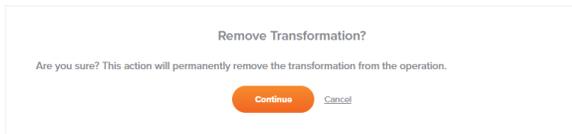
Remove

Upon selecting the **Remove** option for an activity, script, or transformation with both a source and a target schema [defined within the transformation](#), the component will automatically be removed from the operation. To reuse a component that you have removed from an operation, see [Component Reuse](#).



NOTE: When you remove a source or target activity from an operation, any associated mappings within the transformation, if present, will also be removed.

Upon selecting the **Remove** option for a transformation with one or both schemas [defined within an adjacent activity](#), a message will ask you to confirm you want to remove the transformation:



Click **Continue** to permanently remove the transformation from the operation. Transformations with one or both schemas [defined within an adjacent activity](#) are *not* reusable. In addition, when you remove a transformation of this type, any schemas provided by an adjacent activity will no longer be referenced by the transformation.

Removing components used as steps in an operation is allowed regardless of dependencies, as these components are not deleted from the project and are still available from the **Components** tab of the project pane.

However, depending on the dependencies that exist, upon removing the component from the operation, this may cause the operation or other components used as steps within the operation to become invalid. That is, removing components from an operation does not affect dependencies.