

Mapping Source Objects

Introduction

Source fields can be manually mapped to a target field in either [mapping mode](#) or [script mode](#). When you map a source field, the reference path of the source object is inserted into a script on the target field. Automatic mapping of a source object to a target field is covered in [Automapping](#).

Reference Paths

When a source object is mapped, its individual reference path is inserted into a script on the target field. As these paths are constructed automatically, you do not need to know how to construct them. This information is provided for reference only to aid in understanding how the path is constructed.

For reference, the syntax used for the path of the object includes these symbols:

| Symbol | Meaning |
|--------|--|
| [] | Square brackets are used to enclose nodes. Note that node paths cannot be mapped directly; only fields can be mapped. |
| . | A period following a node name indicates a loop node. |
| \$ | A dollar sign following a node name is used to indicate a <code>Once Only</code> or <code>[E]</code> element. This node cannot be a loop node. |
| # | A hash sign following a node name indicates that an array for the record value fields will be returned. This notation can be used only following a loop node before the period. |
| _ | An underscore preceding <code>Root</code> is used to reference the root in a complex CSV. This is not present for other data structures. An underscore may also simply be an underscore already present in a node or field name. |

As an example, the following are sample paths for a complex CSV data structure. The paths for other structures follow the same format except that `_Root` is not present for other data structures.

| Data Structure | Qualified Path | Meaning |
|-----------------------------|-----------------------------------|--|
| _Root (Once Only) | [_Root\$] | The <code>_Root</code> in a CSV data structure is always a node, and thus appears within brackets []. The ending dollar sign indicates a <code>Once Only</code> element. Node paths cannot be mapped. |
| Header (Once Only) | [_Root\$Header\$] | Sub-nodes use the reference of any preceding nodes, followed by the node name. The ending dollar sign indicates a <code>Once Only</code> or <code>[E]</code> element. Node paths cannot be mapped. |
| Value | [_Root\$Header\$Value] | A value field is not enclosed within square brackets, but still uses the reference of any preceding nodes, followed by the value name. This field path can be used for mapping. |
| Detail (One or More) | [_Root\$Header\$Detail.] | Sub-nodes use the reference of any preceding nodes, followed by the node name. The ending period indicates a loop node. Node paths cannot be mapped. |
| Attribute | [_Root\$Header\$Detail.Attribute] | An attribute field is not enclosed within square brackets, but still uses the reference of any preceding nodes, followed by the attribute name. This field path can be used for mapping. |

On This Page

- [Introduction](#)
- [Reference Paths](#)
- [Mapping to a Target Field](#)
- [Mapping to a Field with an Existing Mapping](#)
- [Removing a Mapping](#)
 - [Remove a Single Field Mapping](#)
 - [Remove Multiple Field Mappings](#)

Related Articles

- [Automapping](#)
- [Mapping Mode](#)
- [Nodes and Fields](#)
- [Script Editor](#)
- [Script Mode](#)
- [Transformation Mapping Validity](#)

Related Topics

- [Cloud Studio](#)
- [Scripts](#)
- [Transformation Mapping](#)
- [Transformations](#)

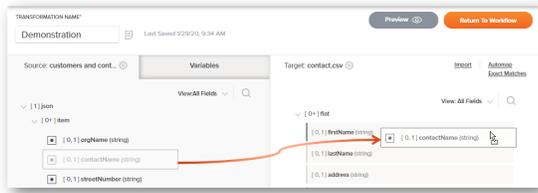
Last updated: Feb 12, 2020

| | | |
|--|--|---|
| <p>v a l u e [v]</p> | <p><code>_Root\$Header\$Detail#.Value</code></p> | <p>A value field is not enclosed within square brackets, but still uses the reference of any preceding nodes, followed by the value name. This specific example shows a hash sign preceding the loop node period, indicating that an array of the "Detail" record values will be returned. This field path can be used for mapping.</p> |
|--|--|---|

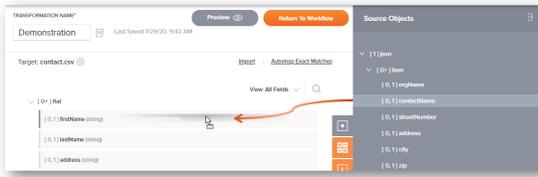
For additional information about data structures, mapped fields, loop nodes, and the display of nodes and fields, see [Nodes and Fields](#).

Mapping to a Target Field

To manually map a source field in [mapping mode](#), within the **Source** tab on the left, drag a source object on the left to an appropriate target field on the right:



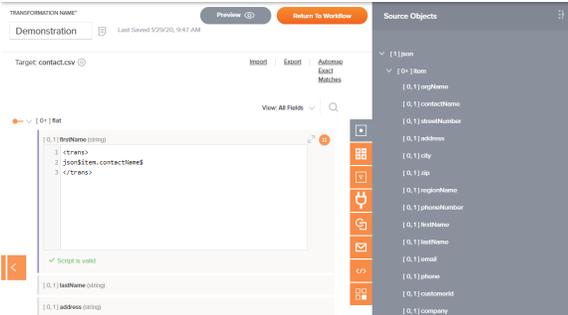
To manually map a source object in [script mode](#), within the **Source Objects** tab on the right, drag a source object on the right to an appropriate target field on the left:



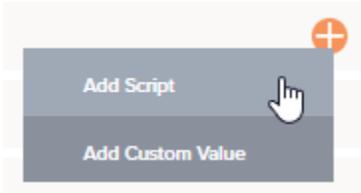
In both modes, the mapped source object will visually be displayed within the target field, with a script icon  added in the top right to indicate the field has a mapping:



To view or edit the mapping as a script, click the script icon  to open the editable script area for the target field in script mode:



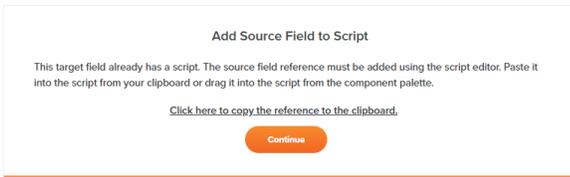
An alternative to drag-and-drop is to manually type or paste source object references into a target field script. To add a new script on a target field, in either mapping mode or script mode, hover over the target field and click the plus icon  in the upper right to open a menu of actions. From the menu select **Add Script**:



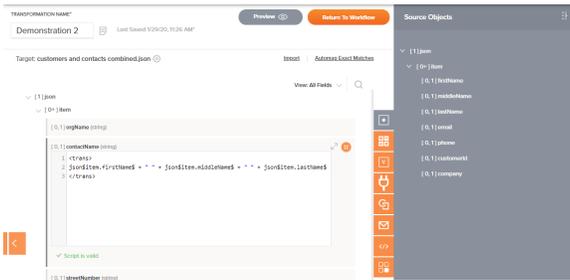
This opens the editable script area on the target field in script mode, where you can enter the source object reference(s) along with any other script logic. You can also drag the specific source object from the palette to the script. For a larger script area to work within, click the popout icon  in the upper right of the target field, which opens the full [script editor](#). Once within the full script editor, click the return icon  to come back to this screen.

Mapping to a Field with an Existing Mapping

When dragging and dropping a source object to a target field that already has a mapping, you will be presented with the option to copy the source object reference to your clipboard so that you can manually paste that reference into the script logic:



Upon clicking **Continue**, the target field will become expanded with an editable script area for you to paste in the source object reference in script mode. For mappings that contain multiple references, you must then add scripting logic to make the script valid. For example:



When the editable script area is open, you can drag and drop other source objects from the palette to the script to insert the reference path directly.

You could also manually type or paste source object references into a target field script if desired. For a larger script area to work within, click the popout icon  in the upper right of the target field, which opens the full [script editor](#). Once within the full script editor, click the return icon  to come back to this screen.

Removing a Mapping

You can remove target field mappings one at a time, or all at once for all field mappings contained within a node or all invalid field mappings contained within a node.

Remove a Single Field Mapping

To remove a single source object mapping from a target field, or *unmap* the field, hover over a source object within the target field and click the trash icon :



For mappings with multiple mapped objects or scripts, hover over the variable within the target field and click the edit icon , then edit the script to remove the variable mapping:



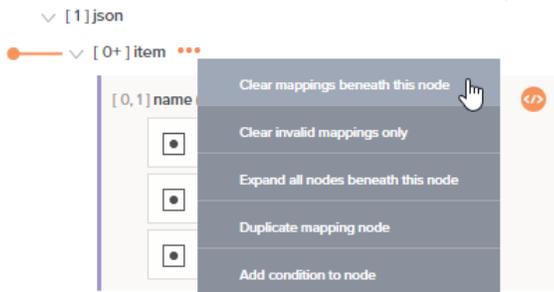
The editable script area is also accessible by clicking the script icon  on the target field.

Within the editable script area, you can also manually clear the contents of the target field script to remove the entire mapping.

Note that there is currently a known issue where manually clearing a mapping results in both the script and the mapping being reported as invalid with an error indicating that the script is unconfigured. If this happens, to restore the original unmapped status of the target field, use the actions menu on the node that contains the target field to select **Clear invalid mappings only**, as described in the next section [Remove Multiple Field Mappings](#).

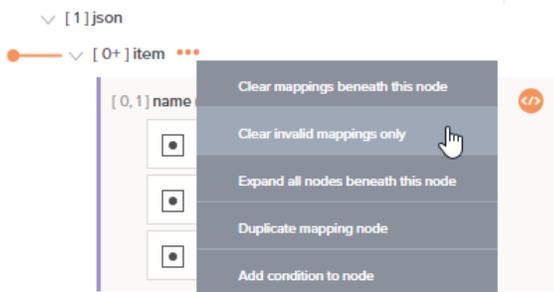
Remove Multiple Field Mappings

To remove all target field mappings contained within a node, from either [mapping mode](#) or [script mode](#), hover over a node name and click the actions menu icon  to open the actions menu. From the menu, select **Clear mappings beneath this node**:



This will delete all mappings on fields contained within the node, as well as all mappings on fields contained within sub-nodes of that node. Upon selecting this option, a message will ask you to confirm that you want to delete mappings. Note that using this option on a root node will remove all mappings from the transformation.

To remove only invalid target field mappings contained within a node, from either [mapping mode](#) or [script mode](#), hover over a node name and click the actions menu icon  to open the actions menu. From the menu, select **Clear invalid mappings only**:



This will delete all invalid mappings on fields contained within the node, as well as all invalid mappings on fields contained within sub-nodes of that node. For more information about invalid mappings, see [Transformation Mapping Validity](#).