

# Script Mode

## Introduction

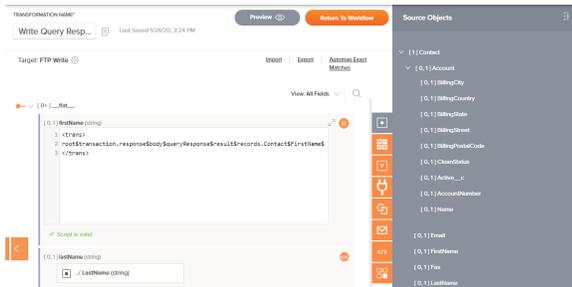
Script mode provides detailed views of fields and advanced tools for adding scripts to the mapping. Also see [Common Mode Elements](#) for elements of the user interface that are shared by multiple display modes.

## Accessing Script Mode

Script mode allows you to drill down to the script level of each target field and provides more control over the mapping of each field. You can enter script mode from any individual target field from within [mapping mode](#):

- For target fields that do not already have a mapping, within mapping mode hover over the target field, then click the plus icon  and select **Add Script** to enter script mode.
- For target fields that already have a mapping, within mapping mode click the script icon  to enter script mode.

In script mode, the target structure and mapping is displayed on the left side, and any components available to be used within the particular script within the mapping are shown in the component palette on the right side:

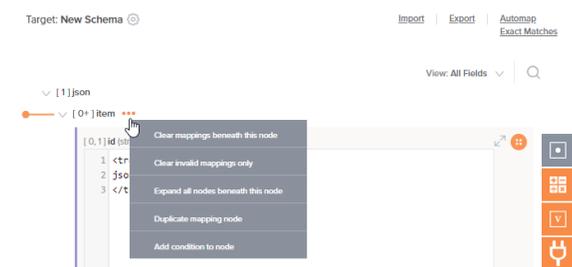


To return to [mapping mode](#) while in script mode, click the arrow on the left side of the screen:



## Target Nodes

A defined target schema is made up of nodes that can contain fields, as well as other, child nodes that can contain fields. Source data cannot be mapped to the target nodes themselves, but can be mapped only to the fields they contain. Once you have defined a target schema, to access options available on each node, hover over the target node, then click the actions menu icon  to show the actions menu. Each action is described below.



- **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 child 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.
- **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 child nodes of that node. For more information about invalid mappings, see [Transformation Mapping Validity](#).

**On This Page**

- [Introduction](#)
- [Accessing Script Mode](#)
- [Target Fields](#)

**Related Articles**

- [Automapping](#)
- [Common Mode Elements](#)
- [Conditional Mapping](#)
- [Mapping Custom Values](#)
- [Mapping Mode](#)
- [Mapping Source Objects](#)
- [Mapping Variables](#)
- [Mapping with Scripts](#)
- [Nodes and Fields](#)
- [Transformation Mapping Validity](#)

**Related Topics**

- [Cloud Studio](#)
- [Schemas](#)
- [Schemas Defined in a Transformation](#)
- [Schemas Defined in an Activity](#)
- [Scripts](#)
- [Transformation Display Modes](#)
- [Transformation Mapping](#)
- [Transformations](#)
- [Variables](#)

Last updated: Jan 28, 2020

- **Expand all nodes beneath this node:** This will expand all child nodes, making it easier to work with structures that have many nodes. By default, nodes are expanded up to 6 levels deep.
- **Duplicate mapping node:** Available only on nodes whose [cardinality](#) indicates they can occur more than once (with the exception of the root node), this will duplicate the node and its fields, as well as any child nodes and their fields. This option is typically used for the purposes of creating additional nodes to map a denormalized source to a hierarchical target.
- **Remove duplicated mapping node:** Available only on nodes that were created through the **Duplicate mapping node** functionality, this will remove the duplicated node and its fields, as well as any duplicated child nodes and their fields.
- **Add condition to node:** Available only on nodes that do not already have a condition applied, this is used to conditionally apply a mapping to the fields contained within a node. For details about creating the condition, see [Conditional Mapping](#).
- **Edit condition:** Available only on nodes that already have a condition applied, this opens script mode for editing the existing condition in script mode. For more details, see [Conditional Mapping](#).
- **Remove condition:** Available only on nodes that already have a condition applied, this removes the condition.

## Target Fields

For a transformation mapping to result in output data from the transformation, you must map to at least one field in the target schema. Once you have defined a target schema, you can map to the target fields in any of the ways covered below, including by mapping source objects, variables, scripts, or custom values to the target fields or adding other script logic:

- **Automapping**  
Automapping is used to automatically map source and target fields whose names are an exact match within identical data structures. You can perform automapping using the entire data structure or within selected source and target nodes.
- **Mapping Source Objects**  
In addition to automapping, source fields can be manually mapped to a target field. When you map a source field to a target field, the reference path of the source object is inserted into a script on the target field.
- **Mapping Variables**  
When you map a variable, the variable reference is inserted into a script on the target field. Variable references are constructed with a dollar sign (\$) preceding the variable name.
- **Mapping Custom Values**  
Custom values are simply hard-coded values that can be mapped to a target field.
- **Mapping with Scripts**  
All transformation mappings are made up of scripts on individual nodes and fields within a target schema. This page covers how to open transformation scripts and reference project components and functions within them, although any valid script logic can be used.