

# Data Transformation Objects

A Data Transformation function consists of several building blocks. It is ideal to get an understanding of these objects to understand how the API works. These include:

- [Step](#)
- [Flow](#)
- [Metadata fields](#)
- [Step options](#)
- [Wires](#)

## Step

A Step is a block of code which performs complex processing logic. It adheres to the Yellowfin Transformation Step API, as a result of which they can be connected to other unrelated blocks of code.

All steps implement the ETLStep Java interface.

An Inline Step (Inline Field Transformation Step) is a special type, which is applied to a single field and is not visible on the canvas. Inline Steps implement the ETLInlineStep interface.

[top](#)

---

## Flow

A Flow is the link which joins two steps. A step may have one or more Inputs and Outputs. Flows are identified using Input/Output FlowUUIDs.

[top](#)

---

## Metadata Fields

A metadata field represents a column of data which enters or leaves a step. The **ETLStepMetadataFieldBean** class represents a field.

The field object has a number of UUIDs. Here are the ones relevant to widget development:

Metadata field	Description
etlStepMetadataFieldUUID	This UUID is used to uniquely identify a metadata field.
groupUUID	Use this UUID to identify data fields from the same input step. It is usually used in multi-input steps.
linkFieldUUID	This UUID identifies the source field, or the field from which a field receives data.

Other relevant fields include:

Metadata field	Description
----------------	-------------

linkType	<p>This metadata field specifies if a data field meets the following conditions:</p> <ul style="list-style-type: none"><li>• Has a direct mapping to an input field.</li><li>• Was generated as a new field.</li><li>• Was generated to replace a field from the input.</li><li>• Was generated as a duplicate of another field.</li></ul> <p>For more information, see ETLLinkType in the API reference.</p>
stepIncludeField	<p>This should be set to false if the field is not to be produced from the step.</p>

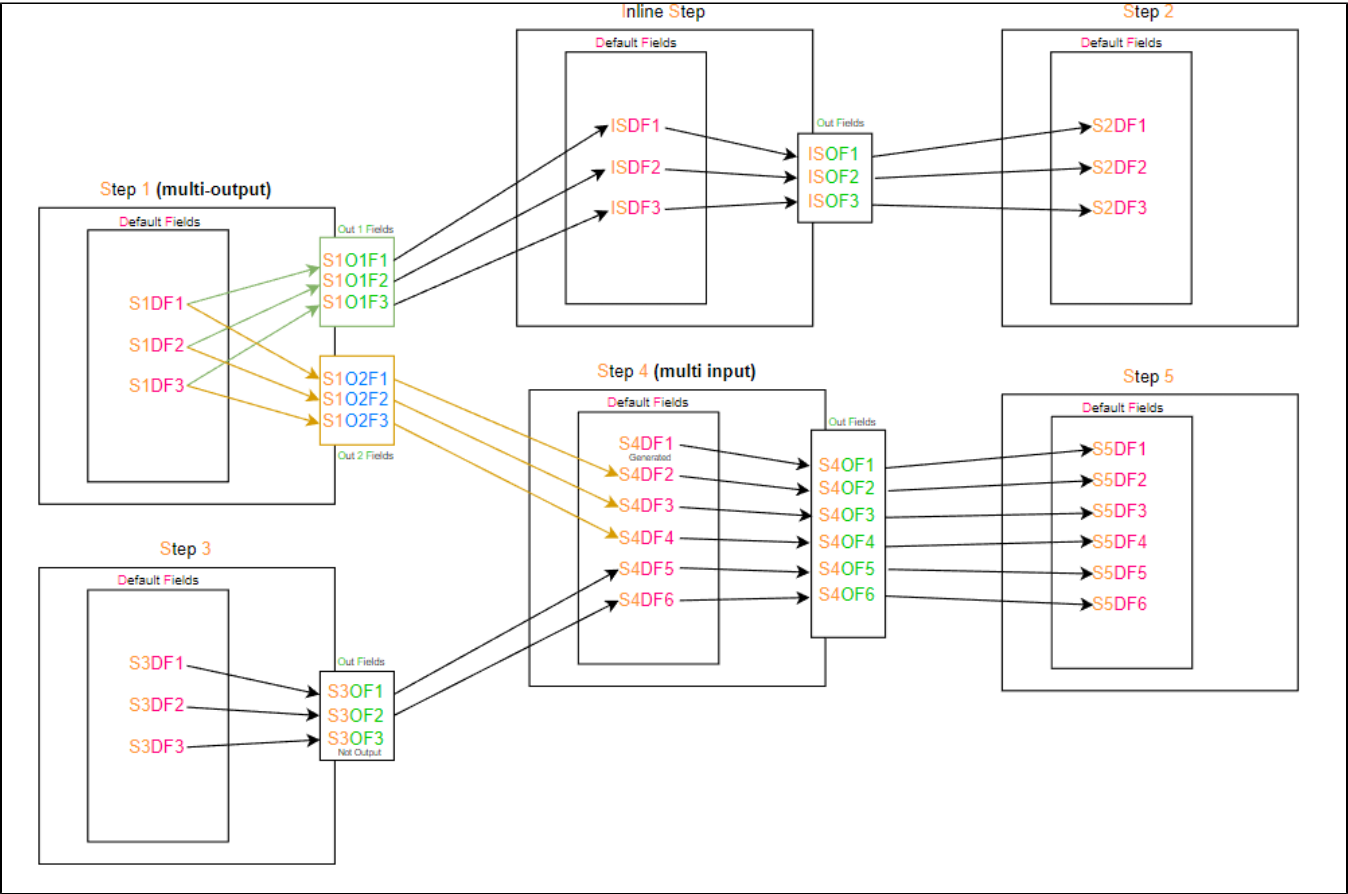
## Types of Metadata Fields

There are two types of metadata fields to be considered:

Type	Description
Default Metadata	<p>These are all input fields available to a step and any fields which the step generates.</p> <p>Default metadata fields are created when input step(s) are linked to the step, or when the step is configured.</p> <p>These fields remain with the step even when inputs and outputs are disconnected. Default metadata enable the step to be an independent pre-configured unit.</p>
Output Metadata	<p>Fields which are actually output from a step.</p> <p>Output metadata fields are created when the step is first linked to an output step.</p> <p>These are initially identical to the default metadata fields. The user can rename, reorder and exclude fields without affecting the default metadata or the output metadata of another output, in case of multiple output steps. Output metadata fields are deleted when the link to the output step is broken.</p>

## Field Linking

The following diagram explains how the fields are linked.



- Every step has Default and Output metadata fields. A target field is linked to a source field using linkFieldUUID. The source field's **fieldUUID** is the target field's **linkedFieldUUID**. For instance, S1DF1's UUID is the linkFieldUUID of S1O1F1.
- Step 1 is a multi-output step. There are two sets of output metadata fields which are linked to one default metadata.
- Step 1 is linked to an Inline Step. Inline steps are conceptually the same as any other transformation steps, but differs in the way they are applied in the UI.
- Step 3 has a field, S3OF3 which has no output; it is not linked to any field in Step 4.
- Step 4 is a multi-input step. Fields from Step 1 and Step 3 are linked. Default Field S4DF1 is generated by Step 4, so it's not linked to any field from the input steps. linkFieldUUID of generated fields is populated depending on how it is generated.
  - If the field is not related to a specific Default Metadata Field, linkFieldUUID will point to itself.
  - If the field is a duplicate of some default field or is replacing another, linkFieldUUID will be the FieldUUID of that default metadata field.

[top](#)

---

## Step Options

All configuration information of a Transformation Step is stored as Step Options. The API provides methods for saving and retrieving step options, which are stored as key-value pairs in a Map. User input may be directly saved as an option value, or inputs may be encoded as options. It is up to the step to set them up and consume them.

[top](#)

---

## Wires

One can resolve how an input field is linked to an output (via a default metadata field) using FieldUUID and linkFieldUUID. However, it is not always convenient to do this as it involves iterating over the fields. The Wire class provides an easy way to map these fields together. It may then be used to retrieve/send data. The API sets up the wires and a widget developer can use them to manipulate data without worrying about the order of fields in the default/output metadata collection.

Consider the following example:

[blocked URL](#)

The highlighted link in the above diagram is a wire in the Inline Step. The wire may be referenced from the Inline Step's implementation using any of its "joints": S1OF1, ISDF1 or ISOF1, which are the inputFieldUUID, fieldUUID (default metadata) and outputFieldUUID.

Generated fields will have only two joints: fieldUUID (default metadata) and outputFieldUUID.

[top](#)

