**Lookup**

The Lookup sage is a processing stage that is used to perform look up operations on a data set. This stage is one of three stages that join tables based on the values of key columns. Other two stages are Join and Merge stage. Generally speaking, Lookup is faster when the reference table is small enough to fit into RAM. If not, you need to use the Join stage.

The stage can be used for validation for a row. If no corresponding entry in a look up table on the key values, we can configure to output the row in a reject table.

It can also perform range lookup.

**Key Points**

The Lookup stage has a reference link, a single input link, a single output link and a single rejects link.

It does not required data on the input link or reference link to be sorted.

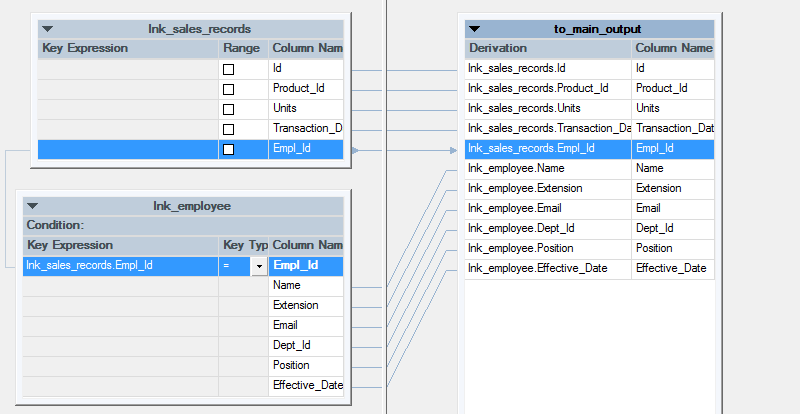
Lookup stage is a in-memory processing stage. Large look up table will result in the job failure if DataStage engine server runs out of memory.

The Key column names in main and lookup tables do not need to be the same as you map them in the stage.

Make sure to select the right Lookup Stage Conditions (see Example step 3).

Example

1. Map the key column and map the output.



1. Select Lookup Stage Conditions to specify the actions when Lookup condition is not met and Lookup fails.

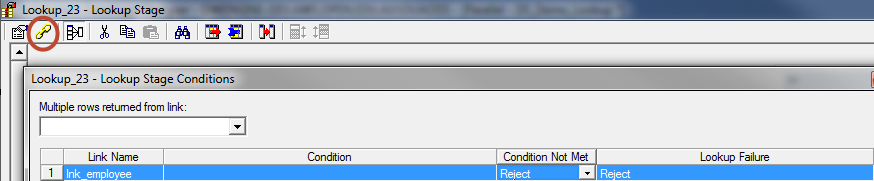
There are 4 options. Continue, Drop, Fail and Reject.

Continue: When the lookup table does not have the value appears in the main table, it will assign null values to the lookup table columns. In another word, this option works like Left Join.

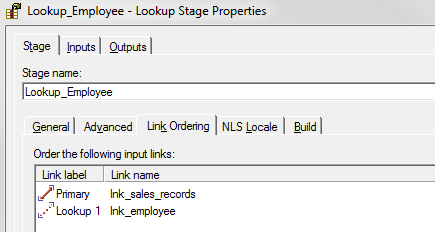
Drop: When the lookup table does not have the value appears in the main table, it will drop the row all together. In another word, this option works like Inner Join.

Fail: When the lookup table does not have the value appears in the main table, the job will fail. This is the default option for the Lookup stage.

Drop: When the lookup table does not have the value appears in the main table, it will output to the reject output (as in this example).



Make sure Link Ordering is correct



Input Partitioning usually works with auto.

