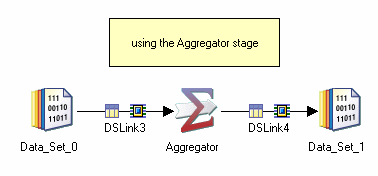
The Aggregator stage is a processing stage. It classifies data rows from a single input link into groups and computes totals or other aggregate functions for each group. The aggregator stage gives you access to grouping and summary operations.

The Aggregator stage is a processing stage. It classifies data rows from a single input link into groups and computes totals or other aggregate functions for each group. The summed totals for each group are output from the stage via an output link.



When you edit an Aggregator stage, the Aggregator stage editor appears. This is based on the generic stage editor described in [Stage editors](https://www.ibm.com/support/knowledgecenter/SSZJPZ_11.7.0/com.ibm.swg.im.iis.ds.parjob.dev.doc/topics/c_deeref_Stage_Editors.html?view=kc).

The stage editor has three pages:

* [**Stage Page**](https://www.ibm.com/support/knowledgecenter/SSZJPZ_11.7.0/com.ibm.swg.im.iis.ds.parjob.dev.doc/topics/c_deeref_Stage_Page_aggregator_stage.html?view=kc). This is always present and is used to specify general information about the stage.
* [**Input Page**](https://www.ibm.com/support/knowledgecenter/SSZJPZ_11.7.0/com.ibm.swg.im.iis.ds.parjob.dev.doc/topics/c_deeref_Inputs_Page_aggregator_stage.html?view=kc). This is where you specify details about the data being grouped or aggregated.
* [**Output Page**](https://www.ibm.com/support/knowledgecenter/SSZJPZ_11.7.0/com.ibm.swg.im.iis.ds.parjob.dev.doc/topics/c_deeref_Outputs_Page_aggregator_stage.html?view=kc). This is where you specify details about the groups being output from the stage.

The aggregator stage gives you access to grouping and summary operations. One of the easiest ways to expose patterns in a collection of records is to group records with similar characteristics, then compute statistics on all records in the group. You can then use these statistics to compare properties of the different groups. For example, records containing cash register transactions might be grouped by the day of the week to see which day had the largest number of transactions, the largest amount of revenue, and so on.

Records can be grouped by one or more characteristics, where record characteristics correspond to column values. In other words, a group is a set of records with the same value for one or more columns. For example, transaction records might be grouped by both day of the week and by month. These groupings might show that the busiest day of the week varies by season.

In addition to revealing patterns in your data, grouping can also reduce the volume of data by summarizing the records in each group, making it easier to manage. If you group a large volume of data on the basis of one or more characteristics of the data, the resulting data set is generally much smaller than the original and is therefore easier to analyze using standard workstation or PC-based tools.

At a practical level, you should be aware that, in a parallel environment, the way that you partition data before grouping and summarizing it can affect the results. For example, if you partitioned using the round robin method, records with identical values in the column you are grouping on would end up in different partitions. If you then performed a sum operation within these partitions you would not be operating on all the relevant columns. In such circumstances you might want to key partition the data on one or more of the grouping keys to ensure that your groups are entire.

It is important that you bear these facts in mind and take any steps you need to prepare your data set before presenting it to the Aggregator stage. In practice this could mean you use Sort stages or additional Aggregate stages in the job.

The example data is from a freight carrier who charges customers based on distance, equipment, packing, and license requirements.

They need a report of distance traveled and charges grouped by date and license type.

The following table shows a sample of the data:

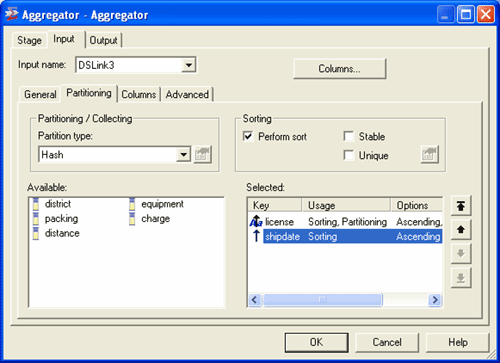
| **Ship Date** | **District** | **Distance** | **Equipment** | **Packing** | **License** | **Charge** |
| --- | --- | --- | --- | --- | --- | --- |
| ... |  |  |  |  |  |  |
| 2000-06-02 | 1 | 1540 | D | M | BUN | 1300 |
| 2000-07-12 | 1 | 1320 | D | C | SUM | 4800 |
| 2000-08-02 | 1 | 1760 | D | C | CUM | 1300 |
| 2000-06-22 | 2 | 1540 | D | C | CUN | 13500 |
| 2000-07-30 | 2 | 1320 | D | M | SUM | 6000 |
| ... |  |  |  |  |  |  |
| *Table 1. Sample of data* | | | | | | |

The stage will output the following columns:

| **Column name** | **SQL Type** |
| --- | --- |
| DistanceSum | Decimal |
| DistanceMean | Decimal |
| ChargeSum | Decimal |
| ChargeMean | Decimal |
| license | Char |
| shipdate | Date |
| *Table 2. Output column definitions* | |

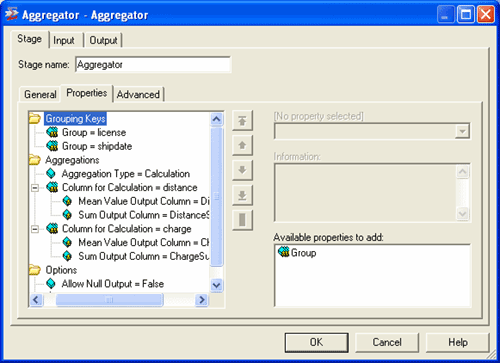
The stage first hash partitions the incoming data on the license column, then sorts it on license and date:

*Figure 1. Partitioning tab*



The properties are then used to specify the grouping and the aggregating of the data:

*Figure 2. Properties tab*

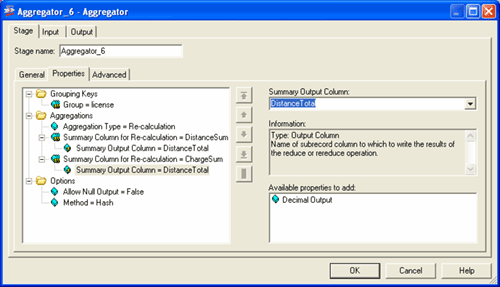


The following is a sample of the output data:

| **Ship Date** | **License** | **Distance Sum** | **Distance Mean** | **Charge Sum** | **Charge Mean** |
| --- | --- | --- | --- | --- | --- |
| ... |  |  |  |  |  |
| 2000-06-02 | BUN | 1126053.00 | 1563.93 | 20427400.00 | 28371.39 |
| 2000-06-12 | BUN | 2031526.00 | 2074.08 | 22426324.00 | 29843.55 |
| 2000-06-22 | BUN | 1997321.00 | 1958.45 | 19556450.00 | 19813.26 |
| 2000-06-30 | BUN | 1815733.00 | 1735.77 | 17023668.00 | 18453.02 |
| ... |  |  |  |  |  |
| *Table 3. Output data* | | | | | |

If you wanted to go on and work out the sum of the distance and charge sums by license, you could insert another Aggregator stage with the following properties:

*Figure 3. Second Aggregator stage*



This section specifies the minimum steps to take to get an Aggregator stage functioning.

**About this task**

InfoSphere® DataStage® has many defaults which means that it can be very easy to include Aggregator stages in a job. InfoSphere DataStage provides a versatile user interface, and there are many shortcuts to achieving a particular end, this section describes the basic method, you will learn where the shortcuts are when you get familiar with the product.

To use an aggregator stage:

* In the Stage page [**Properties Tab**](https://www.ibm.com/support/knowledgecenter/SSZJPZ_11.7.0/com.ibm.swg.im.iis.ds.parjob.dev.doc/topics/r_deeref_Properties_Tab_aggregator_stage.html?view=kc), under the Grouping Keys category:
  + Specify the key column that the data will be grouped on. You can repeat the key property to specify composite keys.

Under the Aggregations category:

* + Choose an aggregation type. Calculation is the default, and allows you to summarize a column or columns. Count rows allows you to count the number of rows within each group. Re-calculation allows you to apply aggregate functions to a column that has already been summarized.

Other properties depend on the aggregate type chosen:

* + If you have chosen the Calculation aggregation type, specify the column to be summarized in Column for Calculation. You can repeat this property to specify multiple columns. Choose one or more dependent properties to specify the type of aggregation to perform, and the name of the output column that will hold the result.
  + If you have chosen the Count Rows aggregation type, specify the output column that will hold the count.
  + If you have chosen the Re-calculation aggregation type, specify the column to be re-calculated. You can repeat this property to specify multiple columns. Choose one or more dependent properties to specify the type of aggregation to perform, and the name of the output column that will hold the result.
* In the Output page [**Mapping Tab**](https://www.ibm.com/support/knowledgecenter/SSZJPZ_11.7.0/com.ibm.swg.im.iis.ds.parjob.dev.doc/topics/r_deeref_Mapping_Tab_aggregator_stage.html?view=kc), check that the mapping is as you expect (InfoSphere DataStage maps data onto the output columns according to what you specify in the **Properties** tab).

The Aggregator stage: Stage page enables you to control aspects of the Aggregator stage.

The General tab allows you to specify an optional description of the stage. The Properties tab lets you specify what the stage does. The Advanced tab allows you to specify how the stage executes. The NLS Locale tab appears if your have NLS enabled on your system. It allows you to select a locale other than the project default to determine collating rules.

Use the Properties tab to specify how the Aggregator stage operates.

The Properties tab allows you to specify properties which determine what the stage actually does. Some of the properties are mandatory, although many have default settings. Properties without default settings appear in the warning color (red by default) and turn black when you supply a value for them.

The following table gives a quick reference list of the properties and their attributes. A more detailed description of each property follows.

| **Category/Property** | **Values** | **Default** | **Mandatory?** | **Repeats?** | **Dependent of** |
| --- | --- | --- | --- | --- | --- |
| Grouping Keys/[Group](https://www.ibm.com/support/knowledgecenter/SSZJPZ_11.7.0/com.ibm.swg.im.iis.ds.parjob.dev.doc/topics/r_deeref_Grouping_Keys_Category.html?view=kc) | Input column | N/A | Y | Y | N/A |
| Grouping Keys/[Case Sensitive](https://www.ibm.com/support/knowledgecenter/SSZJPZ_11.7.0/com.ibm.swg.im.iis.ds.parjob.dev.doc/topics/r_deeref_Grouping_Keys_Category.html?view=kc) | True/ False | True | N | N | Group |
| Aggregations/[Aggregation Type](https://www.ibm.com/support/knowledgecenter/SSZJPZ_11.7.0/com.ibm.swg.im.iis.ds.parjob.dev.doc/topics/r_deeref_Aggregations_Category.html?view=kc) | Calculation/ Recalculation/ Count rows | Calculation | Y | N | N/A |
| Aggregations/[Column for Calculation](https://www.ibm.com/support/knowledgecenter/SSZJPZ_11.7.0/com.ibm.swg.im.iis.ds.parjob.dev.doc/topics/r_deeref_Aggregations_Category.html?view=kc) | Input column | N/A | Y (if Aggregation Type = Calculation) | Y | N/A |
| Aggregations/[Count Output Column](https://www.ibm.com/support/knowledgecenter/SSZJPZ_11.7.0/com.ibm.swg.im.iis.ds.parjob.dev.doc/topics/r_deeref_Aggregations_Category.html?view=kc) | Output column | N/A | Y (if Aggregation Type = Count Rows) | Y | N/A |
| Aggregations/ [Summary Column for Recalculation](https://www.ibm.com/support/knowledgecenter/SSZJPZ_11.7.0/com.ibm.swg.im.iis.ds.parjob.dev.doc/topics/r_deeref_Aggregations_Category.html?view=kc) | Input column | N/A | Y (if Aggregation Type = Recalculation) | Y | N/A |
| Aggregations/ [Default To Decimal Output](https://www.ibm.com/support/knowledgecenter/SSZJPZ_11.7.0/com.ibm.swg.im.iis.ds.parjob.dev.doc/topics/r_deeref_Aggregations_Category.html?view=kc) | precision, scale | 8,2 | N | N | N/A |
| Aggregations/ [Corrected Sum of Squares](https://www.ibm.com/support/knowledgecenter/SSZJPZ_11.7.0/com.ibm.swg.im.iis.ds.parjob.dev.doc/topics/r_deeref_Calculation_and_Recalculation_Dependent_Properties.html?view=kc) | Output column | N/A | N | N | Column for Calculation & Summary Column for Recalculation |
| Aggregations/ [Maximum Value](https://www.ibm.com/support/knowledgecenter/SSZJPZ_11.7.0/com.ibm.swg.im.iis.ds.parjob.dev.doc/topics/r_deeref_Calculation_and_Recalculation_Dependent_Properties.html?view=kc) | Output column | N/A | N | N | Column for Calculation & Summary Column for Recalculation |
| Aggregations/ [Mean Value](https://www.ibm.com/support/knowledgecenter/SSZJPZ_11.7.0/com.ibm.swg.im.iis.ds.parjob.dev.doc/topics/r_deeref_Calculation_and_Recalculation_Dependent_Properties.html?view=kc) | Output column | N/A | N | N | Column for Calculation & Summary Column for Recalculation |
| Aggregations/ [Minimum Value](https://www.ibm.com/support/knowledgecenter/SSZJPZ_11.7.0/com.ibm.swg.im.iis.ds.parjob.dev.doc/topics/r_deeref_Calculation_and_Recalculation_Dependent_Properties.html?view=kc) | Output column | N/A | N | N | Column for Calculation & Summary Column for Recalculation |
| Aggregations/ [Missing Value](https://www.ibm.com/support/knowledgecenter/SSZJPZ_11.7.0/com.ibm.swg.im.iis.ds.parjob.dev.doc/topics/r_deeref_Calculation_and_Recalculation_Dependent_Properties.html?view=kc) | Output column | N/A | N | Y | Column for Calculation |
| Aggregations/ [Missing Values Count](https://www.ibm.com/support/knowledgecenter/SSZJPZ_11.7.0/com.ibm.swg.im.iis.ds.parjob.dev.doc/topics/r_deeref_Calculation_and_Recalculation_Dependent_Properties.html?view=kc) | Output column | N/A | N | N | Column for Calculation & Summary Column for Recalculation |
| Aggregations/ [Non-missing Values Count](https://www.ibm.com/support/knowledgecenter/SSZJPZ_11.7.0/com.ibm.swg.im.iis.ds.parjob.dev.doc/topics/r_deeref_Calculation_and_Recalculation_Dependent_Properties.html?view=kc) | Output column | N/A | N | N | Column for Calculation & Summary Column for Recalculation |
| Aggregations/ [Percent Coefficient of Variation](https://www.ibm.com/support/knowledgecenter/SSZJPZ_11.7.0/com.ibm.swg.im.iis.ds.parjob.dev.doc/topics/r_deeref_Calculation_and_Recalculation_Dependent_Properties.html?view=kc) | Output column | N/A | N | N | Column for Calculation & Summary Column for Recalculation |
| Aggregations/[Range](https://www.ibm.com/support/knowledgecenter/SSZJPZ_11.7.0/com.ibm.swg.im.iis.ds.parjob.dev.doc/topics/r_deeref_Calculation_and_Recalculation_Dependent_Properties.html?view=kc) | Output column | N/A | N | N | Column for Calculation & Summary Column for Recalculation |
| Aggregations/ [Standard Deviation](https://www.ibm.com/support/knowledgecenter/SSZJPZ_11.7.0/com.ibm.swg.im.iis.ds.parjob.dev.doc/topics/r_deeref_Calculation_and_Recalculation_Dependent_Properties.html?view=kc) | Output column | N/A | N | N | Column for Calculation & Summary Column for Recalculation |
| Aggregations/ [Standard Error](https://www.ibm.com/support/knowledgecenter/SSZJPZ_11.7.0/com.ibm.swg.im.iis.ds.parjob.dev.doc/topics/r_deeref_Calculation_and_Recalculation_Dependent_Properties.html?view=kc) | Output column | N/A | N | N | Column for Calculation & Summary Column for Recalculation |
| Aggregations/ [Sum of Weights](https://www.ibm.com/support/knowledgecenter/SSZJPZ_11.7.0/com.ibm.swg.im.iis.ds.parjob.dev.doc/topics/r_deeref_Calculation_and_Recalculation_Dependent_Properties.html?view=kc) | Output column | N/A | N | N | Column for Calculation & Summary Column for Recalculation |
| Aggregations/ [Sum](https://www.ibm.com/support/knowledgecenter/SSZJPZ_11.7.0/com.ibm.swg.im.iis.ds.parjob.dev.doc/topics/r_deeref_Calculation_and_Recalculation_Dependent_Properties.html?view=kc) | Output column | N/A | N | N | Column for Calculation & Summary Column for Recalculation |
| Aggregations/ [Summary](https://www.ibm.com/support/knowledgecenter/SSZJPZ_11.7.0/com.ibm.swg.im.iis.ds.parjob.dev.doc/topics/r_deeref_Calculation_and_Recalculation_Dependent_Properties.html?view=kc) | Output column | N/A | N | N | Column for Calculation & Summary Column for Recalculation |
| Aggregations/ [Uncorrected Sum of Squares](https://www.ibm.com/support/knowledgecenter/SSZJPZ_11.7.0/com.ibm.swg.im.iis.ds.parjob.dev.doc/topics/r_deeref_Calculation_and_Recalculation_Dependent_Properties.html?view=kc) | Output column | N/A | N | N | Column for Calculation & Summary Column for Recalculation |
| Aggregations/ [Variance](https://www.ibm.com/support/knowledgecenter/SSZJPZ_11.7.0/com.ibm.swg.im.iis.ds.parjob.dev.doc/topics/r_deeref_Calculation_and_Recalculation_Dependent_Properties.html?view=kc) | Output column | N/A | N | N | Column for Calculation & Summary Column for Recalculation |
| Aggregations/ [Variance divisor](https://www.ibm.com/support/knowledgecenter/SSZJPZ_11.7.0/com.ibm.swg.im.iis.ds.parjob.dev.doc/topics/r_deeref_Calculation_and_Recalculation_Dependent_Properties.html?view=kc) | Default/ Nrecs | Default | N | N | Variance |
| Aggregations/ [Calculation and Recalculation Dependent Properties](https://www.ibm.com/support/knowledgecenter/SSZJPZ_11.7.0/com.ibm.swg.im.iis.ds.parjob.dev.doc/topics/r_deeref_Calculation_and_Recalculation_Dependent_Properties.html?view=kc) | Input column | N/A | N | N | Column for Calculation or Count Output Column |
| Aggregations/Decimal Output | precision, scale | 8,2 | N | N | Calculation or Recalculation method |
| Options/[Group](https://www.ibm.com/support/knowledgecenter/SSZJPZ_11.7.0/com.ibm.swg.im.iis.ds.parjob.dev.doc/topics/r_deeref_Grouping_Keys_Category.html?view=kc) | hash/sort | hash | Y | Y | N/A |
| Options/[Allow Null Outputs](https://www.ibm.com/support/knowledgecenter/SSZJPZ_11.7.0/com.ibm.swg.im.iis.ds.parjob.dev.doc/topics/r_deeref_Options_Category_aggregator_stage.html?view=kc) | True/ False | False | Y | N | N/A |
| *Table 1. Properties* | | | | | |

Specifies the input columns you are using as group keys.

**Group**

Repeat the property to select multiple columns as group keys. You can use the Column Selection dialog box to select several group keys at once if required). This property has a dependent property:

* **Case Sensitive**

Use this to specify whether each group key is case sensitive or not, this is set to True by default, that is, the values "CASE" and "case" in would end up in different groups.

This property allows you to specify the type of aggregation operation your stage is performing.

**Aggregation type**

Choose from Calculate (the default), Recalculate, and Count Rows.

**Column for calculation**

The Calculate aggregate type allows you to summarize the contents of a particular column or columns in your input data set by applying one or more aggregate functions to it. Select the column to be aggregated, then select dependent properties to specify the operation to perform on it, and the output column to carry the result. You can use the Column Selection dialog box to select several columns for calculation at once if required).

**Count output column**

The Count Rows aggregate type performs a count of the number of records within each group. Specify the column on which the count is output.

**Summary column for recalculation**

This aggregate type allows you to apply aggregate functions to a column that has already been summarized. This is like calculate but performs the specified aggregate operation on a set of data that has already been summarized. In practice this means you should have performed a calculate (or recalculate) operation in a previous Aggregator stage with the Summary property set to produce a subrecord containing the summary data that is then included with the data set. Select the column to be aggregated, then select dependent properties to specify the operation to perform on it, and the output column to carry the result. You can use the Column Selection dialog box to select several columns for recalculation at once if required).

**Weighting column**

Configures the stage to increment the count for the group by the contents of the weight column for each record in the group, instead of by 1. Not available for Summary Column for Recalculation. Setting this option affects only the following options:

* Percent Coefficient of Variation
* Mean Value
* Sum
* Sum of Weights
* Uncorrected Sum of Squares

**Default to decimal output**

The output type of a calculation or recalculation column is double. Setting this property causes it to default to decimal. You can specify that individual columns have decimal output while others retain the default type of double. You can also set a default precision and scale.

The aggregate stage has two modes of operation: **hash** and **sort**.

## Method

Your choice of mode depends primarily on the number of groupings in the input data set, taking into account the amount of memory available. You typically use hash mode for a relatively small number of groups; generally, fewer than about 1000 groups per megabyte of memory to be used.

When using hash mode, you should hash partition the input data set by one or more of the grouping key columns so that all the records in the same group are in the same partition (this happens automatically if auto is set in the Partitioning tab). However, hash partitioning is not mandatory, you can use any partitioning method you choose if keeping groups together in a single partition is not important. For example, if you're summing records in each partition and later you'll add the sums across all partitions, you don't need all records in a group to be in the same partition to do this. Note, though, that there will be multiple output records for each group.

If the number of groups is large, which can happen if you specify many grouping keys, or if some grouping keys can take on many values, you would normally use sort mode. However, sort mode requires the input data set to have been partition sorted with all of the grouping keys specified as hashing and sorting keys (this happens automatically if auto is set in the Partitioningtab). Sorting requires a pregrouping operation: after sorting, all records in a given group in the same partition are consecutive.

The method property is set to **hash** by default.

You might want to try both modes with your particular data and application to determine which gives the better performance. You might find that when calculating statistics on large numbers of groups, sort mode performs better than hash mode, assuming the input data set can be efficiently sorted before it is passed to group.

## Allow null outputs

Set this to True to indicate that null is a valid output value when calculating minimum value, maximum value, mean value, standard deviation, standard error, sum, sum of weights, and variance. If False, the null value will have 0 substituted when all input values for the calculation column are null. It is False by default.

Some properties are dependents of both Column for Calculation and Summary Column for Recalculation.

These specify the various aggregate functions and the output columns to carry the results.

* **Corrected Sum of Squares**

Produces a corrected sum of squares for data in the aggregate column and outputs it to the specified output column.

* **Maximum Value**

Gives the maximum value in the aggregate column and outputs it to the specified output column.

* **Mean Value**

Gives the mean value in the aggregate column and outputs it to the specified output column.

* **Minimum Value**

Gives the minimum value in the aggregate column and outputs it to the specified output column.

* **Missing Value**

This specifies what constitutes a "missing" value, for example -1 or NULL. Enter the value as a floating point number. Not available for Summary Column to Recalculate.

* **Missing Values Count**

Counts the number of aggregate columns with missing values in them and outputs the count to the specified output column. Not available for recalculate.

* **Non-missing Values Count**

Counts the number of aggregate columns with values in them and outputs the count to the specified output column.

* **Percent Coefficient of Variation**

Calculates the percent coefficient of variation for the aggregate column and outputs it to the specified output column.

* **Range**

Calculates the range of values in the aggregate column and outputs it to the specified output column.

* **Standard Deviation**

Calculates the standard deviation of values in the aggregate column and outputs it to the specified output column.

* **Standard Error**

Calculates the standard error of values in the aggregate column and outputs it to the specified output column.

* **Sum of Weights**

Calculates the sum of values in the weight column specified by the Weight column property and outputs it to the specified output column.

* **Sum**

Sums the values in the aggregate column and outputs the sum to the specified output column.

* **Summary**

Specifies a subrecord to write the results of the calculate or recalculate operation to.

* **Uncorrected Sum of Squares**

Produces an uncorrected sum of squares for data in the aggregate column and outputs it to the specified output column.

* **Variance**

Calculates the variance for the aggregate column and outputs the sum to the specified output column. This has a dependent property:

* + **Variance divisor**

Specifies the variance divisor. By default, uses a value of the number of records in the group minus the number of records with missing values minus 1 to calculate the variance. This corresponds to a vardiv setting of Default. If you specify NRecs, InfoSphere® DataStage® uses the number of records in the group minus the number of records with missing values instead.

Each of these properties has a dependent property as follows:

* **Decimal Output**

By default all calculation or recalculation columns have an output type of double. This property allows you to specify that columns have an output type of decimal.

When you specify the decimal output, you can also specify precision and scale. **Precision** is the number of digits in a number. **Scale** is the number of digits to the right of the decimal point in a number. The default is 8,2.

In cases where the required output scale is low, set the precision and scale to p+4, s+4 to get accurate results. If a column has a precision and scale of 4,1, then in the decimal data type, set the precision and scale to 9,5.

For example, a column that has the values: " 004.0"," 010.0"," 004.0"," 006.0"," 010.0"," 008.0"," 009.0"," 007.0" " 010.0"," 007.0"," 010.0"," 007.0"," 010.0" . The precision value for the column is 4and the scale value is 1. The output is calculated as 7.8 if the precision and scale is set to 9,5. But if the precision and scale is set to 4,1, the output is 7.9. The more accurate calculation is 7.8.

This tab allows you to specify options.

This tab allows you to specify the following:

* **Execution Mode**. The stage can execute in parallel mode or sequential mode. In parallel mode the input data set is processed by the available nodes as specified in the Configuration file, and by any node constraints specified on the Advanced tab. In Sequential mode the entire data set is processed by the conductor node.
* **Combinability mode**. This is Auto by default, which allows InfoSphere® DataStage® to combine the operators that underlie parallel stages so that they run in the same process if it is sensible for this type of stage.
* **Preserve partitioning**. This is **Set** by default. You can select **Set** or **Clear**. If you select **Set** the stage will request that the next stage in the job attempt to maintain the partitioning.
* **Node pool and resource constraints**. Select this option to constrain parallel execution to the node pool or pools or resource pool or pools specified in the grid. The grid allows you to make choices from drop down lists populated from the Configuration file.
* **Node map constraint**. Select this option to constrain parallel execution to the nodes in a defined node map. You can define a node map by typing node numbers into the text box or by clicking the browse button to open the Available Nodes dialog box and selecting nodes from there. You are effectively defining a new node pool for this stage (in addition to any node pools defined in the Configuration file).

**Note**In the **Node map constraint** text box, you can enter jobs parameters as well as numbers. You can enter a single parameter, for example #testnode#, or you can enter a comma separated lists of parameters, for example #testnode#, #testnode2#. The browse button next to the text box will display a list of the node names from the last configuration file that was referenced by the job, but the browse button will not display the node names that were specified by the job parameters.

For the Aggregator stage, the NLS Locale tab appears if you have NLS enabled on your system. It lets you view the current default collate convention, and select a different one for this stage if required.

You can also use a job parameter to specify the locale, or browse for a file that defines custom collate rules. The collate convention defines the order in which characters are collated. The Aggregator stage uses this when it is grouping by key to determine the order of the key fields. Select a locale from the list, or click the arrow button next to the list to use a job parameter or browse for a collate file.

The Input page allows you to specify details about the incoming data set.

The Input page allows you to specify details about the incoming data set.

The General tab allows you to specify an optional description of the input link. The **Partitioning** tab allows you to specify how incoming data is partitioned before being grouped or summarized. The Columns tab specifies the column definitions of incoming data. The Advanced tab allows you to change the default buffering settings for the input link.

Details about Aggregator stage partitioning are given in the following section. See ["Stage Editors,"](https://www.ibm.com/support/knowledgecenter/SSZJPZ_11.7.0/com.ibm.swg.im.iis.ds.parjob.dev.doc/topics/c_deeref_Stage_Editors.html?view=kc) for a general description of the other tabs.

The Partitioning tab allows you to specify details about how the incoming data is partitioned or collected before it is grouped or summarized.

It also allows you to specify that the data should be sorted before being operated on.

By default the stage partitions in Auto mode. This attempts to work out the best partitioning method depending on execution modes of current and preceding stages and how many nodes are specified in the Configuration file.

If the Aggregator stage is operating in sequential mode, it will first collect the data before writing it to the file using the default Auto collection method.

The Partitioning tab allows you to override this default behavior. The exact operation of this tab depends on:

* Whether the Aggregator stage is set to execute in parallel or sequential mode.
* Whether the preceding stage in the job is set to execute in parallel or sequential mode.

If the Aggregator stage is set to execute in parallel, then you can set a partitioning method by selecting from the **Partition type**drop-down list. This will override any current partitioning.

If the Aggregator stage is set to execute in sequential mode, but the preceding stage is executing in parallel, then you can set a collection method from the **Collector type** drop-down list. This will override the default collection method.

The following partitioning methods are available:

* **(Auto)**. InfoSphere® DataStage® attempts to work out the best partitioning method depending on execution modes of current and preceding stages and how many nodes are specified in the Configuration file. This is the default partitioning method for the Aggregator stage.
* **Entire**. Each file written to receives the entire data set.
* **Hash**. The records are hashed into partitions based on the value of a key column or columns selected from the **Available**list.
* **Modulus**. The records are partitioned using a modulus function on the key column selected from the **Available** list. This is commonly used to partition on tag fields.
* **Random**. The records are partitioned randomly, based on the output of a random number generator.
* **Round Robin**. The records are partitioned on a round robin basis as they enter the stage.
* **Same**. Preserves the partitioning already in place.
* **Db2®**. Replicates the Db2 partitioning method of a specific Db2 table. Requires extra properties to be set. Access these properties by clicking the properties button.
* **Range**. Divides a data set into approximately equal size partitions based on one or more partitioning keys. Range partitioning is often a preprocessing step to performing a total sort on a data set. Requires extra properties to be set. Access these properties by clicking the properties button.

The following Collection methods are available:

* **(Auto)**. This is the default collection method for Aggregator stages. Normally, when you are using Auto mode, InfoSphere DataStage will eagerly read any row from any input partition as it becomes available.
* **Ordered**. Reads all records from the first partition, then all records from the second partition, and so on.
* **Round Robin**. Reads a record from the first input partition, then from the second partition, and so on. After reaching the last partition, the operator starts over.
* **Sort Merge**. Reads records in an order based on one or more columns of the record. This requires you to select a collecting key column from the **Available** list.

The Partitioning tab also allows you to specify that data arriving on the input link should be sorted before being written to the file or files. The sort is always carried out within data partitions. If the stage is partitioning incoming data the sort occurs after the partitioning. If the stage is collecting data, the sort occurs before the collection. The availability of sorting depends on the partitioning or collecting method chosen (it is not available for the default auto modes).

Select the check boxes as follows:

* **Perform Sort**. Select this to specify that data coming in on the link should be sorted. Select the column or columns to sort on from the **Available** list.
* **Stable**. Select this if you want to preserve previously sorted data sets. This is the default.
* **Unique**. Select this to specify that, if multiple records have identical sorting key values, only one record is retained. If stable sort is also set, the first record is retained.

If NLS is enabled an additional button opens a dialog box allowing you to select a locale specifying the collate convention for the sort.

You can also specify sort direction, case sensitivity, whether sorted as ASCII or EBCDIC, and whether null columns will appear first or last for each column. Where you are using a keyed partitioning method, you can also specify whether the column is used as a key for sorting, for partitioning, or for both. Select the column in the **Selected** list and right-click to invoke the shortcut menu.

In the Output page, you can specify details about data output from the Remove stage. The Aggregator stage does not transfer input data, instead the stage generates new columns.

The Output page allows you to specify details about data output from the Aggregator stage. The Aggregator stage can have only one output link.

The Aggregator stage does not transfer input data, instead the stage generates new columns. The Aggregator stage outputs nullable fields therefore columns that receive the output data must be nullable.

The General tab allows you to specify an optional description of the output link. The **Columns** tab specifies the column definitions of incoming data. The Mapping tab allows you to specify the relationship between the processed data being produced by the Aggregator stage and the Output columns. The Advanced tab allows you to change the default buffering settings for the output link.

Details about Aggregator stage mapping is given in the following section. See ["Stage Editors,"](https://www.ibm.com/support/knowledgecenter/SSZJPZ_11.7.0/com.ibm.swg.im.iis.ds.parjob.dev.doc/topics/c_deeref_Stage_Editors.html?view=kc) for a general description of the other tabs.

For the Aggregator stage, the Mapping tab allows you to specify how the output columns are derived, that is, what input columns map onto them or how they are generated.

The left pane shows the input columns or the generated columns. These are read only and cannot be modified on this tab.

The right pane shows the output columns for each link. This has a **Derivations** field where you can specify how the column is derived. You can fill it in by dragging columns over from the left pane, or by using the Auto-match facility.