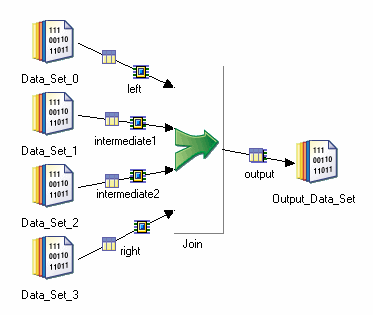
The Join stage is a processing stage that performs join operations on two or more data sets input to the stage and then outputs the resulting data set.

The Join stage is a processing stage. It performs join operations on two or more data sets input to the stage and then outputs the resulting data set. The Join stage is one of three stages that join tables based on the values of key columns. The other two are:

* [Lookup Stage](https://www.ibm.com/support/knowledgecenter/SSZJPZ_11.7.0/com.ibm.swg.im.iis.ds.parjob.dev.doc/topics/c_deeref_Lookup_Stage.html?view=kc)
* [Merge Stage](https://www.ibm.com/support/knowledgecenter/SSZJPZ_11.7.0/com.ibm.swg.im.iis.ds.parjob.dev.doc/topics/c_deeref_Merge_Stage.html?view=kc)

The three stages differ mainly in the memory they use, the treatment of rows with unmatched keys, and their requirements for data being input (for example, whether it is sorted). See ["Join Versus Lookup"](https://www.ibm.com/support/knowledgecenter/SSZJPZ_11.7.0/com.ibm.swg.im.iis.ds.parjob.dev.doc/topics/c_deeref_Join_Versus_Lookup.html?view=kc) for help in deciding which stage to use.

In the Join stage, the input data sets are notionally identified as the "right" set and the "left" set, and "intermediate" sets. You can specify which is which. It has any number of input links and a single output link.



The stage can perform one of four join operations:

* **Inner** transfers records from input data sets whose key columns contain equal values to the output data set. Records whose key columns do not contain equal values are dropped.
* **Left outer** transfers all values from the left data set but transfers values from the right data set and intermediate data sets only where key columns match. The stage drops the key column from the right and intermediate data sets.
* **Right outer** transfers all values from the right data set and transfers values from the left data set and intermediate data sets only where key columns match. The stage drops the key column from the left and intermediate data sets.
* **Full outer** transfers records in which the contents of the key columns are equal from the left and right input data sets to the output data set. It also transfers records whose key columns contain unequal values from both input data sets to the output data set. (Full outer joins do not support more than two input links.)

The data sets input to the Join stage must be key partitioned and sorted in ascending order. This ensures that rows with the same key column values are located in the same partition and will be processed by the same node. It also minimizes memory requirements because fewer rows need to be in memory at any one time. Choosing the auto partitioning method will ensure that partitioning and sorting is done. If sorting and partitioning are carried out on separate stages before the Join stage, InfoSphere® DataStage® in auto mode will detect this and not repartition (alternatively you could explicitly specify the Same partitioning method).

The Join 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_join_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_join_stage.html?view=kc). This is where you specify details about the data sets being joined.
* [**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_join_stage.html?view=kc). This is where you specify details about the joined data being output from the stage.

InfoSphere® DataStage® does not know how large your data is, so cannot make an informed choice whether to combine data using a join stage or a lookup stage. Here's how to decide which to use:

There are two data sets being combined. One is the primary or driving dataset, sometimes called the left of the join. The other data set(s) are the reference datasets, or the right of the join.

In all cases you are concerned with the size of the reference datasets. If these take up a large amount of memory relative to the physical RAM memory size of the computer you are running on, then a lookup stage might thrash because the reference datasets might not fit in RAM along with everything else that has to be in RAM. This results in very slow performance since each lookup operation can, and typically does, cause a page fault and an I/O operation.

So, if the reference datasets are big enough to cause trouble, use a join. A join does a high-speed sort on the driving and reference datasets. This can involve I/O if the data is big enough, but the I/O is all highly optimized and sequential. Once the sort is over the join processing is very fast and never involves paging or other I/O.

The following examples show what happens to two data sets when each type of join operation is applied to them.

Here are the two data sets:

| **Left Input Data Set** |  | **Right Input Data Set** |  |
| --- | --- | --- | --- |
| Status | Price | Price | ID |
| sold | 125 | 113 | NI6325 |
| sold | 213 | 125 | BR9658 |
| offered | 378 | 285 | CZ2538 |
| Pending | 575 | 628 | RU5713 |
| Pending | 649 | 668 | SA5680 |
| Offered | 777 | 777 | JA1081 |
| Offered | 908 | 908 | DE1911 |
| Pending | 908 | 908 | FR2081 |
| *Table 1. Data sets* | | | |

Price is the key column which is going to be joined on, and bold type indicates where the data sets share the same value for Price. The data sets are already sorted on that key.

This example shows the data set that is output if you perform an inner join on the Price key column

Here is the data set that is output if you perform an inner join on the Price key column:

| **Status** | **Price** | **ID** |
| --- | --- | --- |
| sold | 125 | NI6325 |
| Offered | 777 | JA1081 |
| Offered | 908 | DE1911 |
| Offered | 908 | FR2081 |
| Pending | 908 | DE1911 |
| Pending | 908 | FR2081 |
| *Table 1. Output data set* | | |

This example shows the data set that is output if you perform a left outer join on the Price key column.

Here is the data set that is output if you perform a left outer join on the Price key column:

| **Output Data Set** | | |
| --- | --- | --- |
| **Status** | **Price** | **ID** |
| sold | 125 | NI6325 |
| sold | 213 |  |
| offered | 378 |  |
| Pending | 575 |  |
| Pending | 649 |  |
| Offered | 777 | JA1081 |
| Offered | 908 | DE1911 |
| Offered | 908 | FR2081 |
| Pending | 908 | DE1911 |
| Pending | 908 | FR2081 |

This example shows the data set that is output if you perform a right outer join on the Price key column.

Here is the data set that is output if you perform a right outer join on the Price key column

| **Status** | **Price** | **ID** |
| --- | --- | --- |
|  | 113 | NI6325 |
| sold | 125 | BR9658 |
|  | 285 | CZ2538 |
|  | 628 | RU5713 |
|  | 668 | SA5680 |
| Offered | 777 | JA1081 |
| Offered | 908 | DE1911 |
| Offered | 908 | FR2081 |
| Pending | 908 | DE1911 |
| Pending | 908 | FR2081 |
| *Table 1. Output data set* | | |

Here is the data set that is output if you perform a full outer join.

Here is the data set that is output if you perform a full outer join on the Price key column:

| **Status** | **Price** | **Price** | **ID** |
| --- | --- | --- | --- |
|  |  | 113 | NI6325 |
| sold | 125 | 125 | BR9658 |
| sold | 213 |  |  |
|  |  | 285 | CZ2538 |
| offered | 378 |  |  |
| Pending | 575 |  |  |
|  |  | 628 | RU5713 |
| Pending | 649 |  |  |
|  |  | 668 | SA5680 |
| Status | Price | Price | ID |
| Offered | 777 | 777 | JA1081 |
| Offered | 908 | 908 | DE1911 |
| Offered | 908 | 908 | FR2081 |
| Pending | 908 | 908 | DE1911 |
| Pending | 908 | 908 | FR2081 |
| *Table 1. Output data set* | | | |

This section specifies the minimum steps to take to get a Join stage functioning.

**About this task**

InfoSphere® DataStage® has many defaults which means that Joins can be simple to set up. 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.

* 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_join_stage.html?view=kc) specify the key column or columns that the join will be performed on.
* 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_join_stage.html?view=kc) specify the join type or accept the default of Inner.
* In the Stage page [**Link Ordering Tab**](https://www.ibm.com/support/knowledgecenter/SSZJPZ_11.7.0/com.ibm.swg.im.iis.ds.parjob.dev.doc/topics/r_deeref_Link_Ordering_Tab_join_stage.html?view=kc), check that your links are correctly identified as "left", "right", and "intermediate" and reorder if required.
* Ensure required column meta data has been specified (this might be done in another stage).
* 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_join_stage.html?view=kc), specify how the columns from the input links map onto output columns.

This section specifies the minimum steps to take to get a Join stage functioning.

**About this task**

InfoSphere® DataStage® has many defaults which means that Joins can be simple to set up. 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.

* 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_join_stage.html?view=kc) specify the key column or columns that the join will be performed on.
* 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_join_stage.html?view=kc) specify the join type or accept the default of Inner.
* In the Stage page [**Link Ordering Tab**](https://www.ibm.com/support/knowledgecenter/SSZJPZ_11.7.0/com.ibm.swg.im.iis.ds.parjob.dev.doc/topics/r_deeref_Link_Ordering_Tab_join_stage.html?view=kc), check that your links are correctly identified as "left", "right", and "intermediate" and reorder if required.
* Ensure required column meta data has been specified (this might be done in another stage).
* 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_join_stage.html?view=kc), specify how the columns from the input links map onto output columns.

Use the Properties tab to specify how the Join 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** |
| --- | --- | --- | --- | --- | --- |
| Join Keys/[Key](https://www.ibm.com/support/knowledgecenter/SSZJPZ_11.7.0/com.ibm.swg.im.iis.ds.parjob.dev.doc/topics/r_deeref_Join_Keys_Category.html?view=kc#wp1237425__wp1238810) | Input Column | N/A | Y | Y | N/A |
| Join Keys/Case Sensitive | True/False | True | N | N | Key |
| Options/[Join stage: Options category](https://www.ibm.com/support/knowledgecenter/SSZJPZ_11.7.0/com.ibm.swg.im.iis.ds.parjob.dev.doc/topics/r_deeref_Options_Category_join_stage.html?view=kc#wp1238841) | Full Outer/ Inner/Left Outer/ Right Outer | Inner | Y | N | N/A |
| *Table 1. Properties* | | | | | |

Join keys are used to combine rows from multiple tables.

**Key**

Choose the input column you want to join on. You are offered a choice of input columns common to all links. For a join to work you must join on a column that appears in all input data sets, that is, have the same name and compatible data types. If, for example, you select a column called "name" from the left link, the stage will expect there to be an equivalent column called "name" on the right link.

You can join on multiple key columns. To do so, repeat the Key property. You can use the Column Selection dialog box to select several key columns at once if required).

Key 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 not be judged equivalent.

Use the Options category to specify how the Join stage operates.

**Join type**

Specify the type of join operation you want to perform. Choose one of:

* Full Outer
* Inner
* Left Outer
* Right Outer

The default is Inner.

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 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 **Propagate** by default. It adopts the setting which results from ORing the settings of the input stages, that is, if either of the input stages uses **Set** then this stage will use **Set**. You can explicitly select **Set** or **Clear**. Select **Set** to request that the next stage in the job attempts 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.

This tab allows you to specify which input link is regarded as the left link and which link is regarded as the right link, and which links are regarded as intermediate.

By default the first link you add is regarded as the left link, and the last one as the right link, with all other links labelled as Intermediate *N*. You can use this tab to override the default order.

In the example DSLink4 is the left link, click on it to select it then click on the down arrow to convert it into the right link.

For the Join 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 Join stage uses this when it is determining 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 sets. Choose an input link from the **Input name** drop down list to specify which link you want to work on.

The **Input page** allows you to specify details about the incoming data sets. Choose an input link from the **Input name** drop down list to specify which link you want to work on.

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 joined. 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 Join 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 data on each of the incoming links is partitioned or collected before it is joined.

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. Auto mode ensures that data being input to the Join stage is key partitioned and sorted.

If the Join 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 Join 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 Join 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 Join 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 collection method for the Join 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 the Join stage. Normally, when you are using Auto mode, InfoSphere DataStage will eagerly read any row from any input partition as it becomes available. In the case of a Join stage, Auto will also ensure that the collected data is sorted.
* **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 explicitly specify that data arriving on the input link should be sorted before being joined (you might use this if you have selected a partitioning method other than auto or same). 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 with the default auto methods).

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.

The **Output page** allows you to specify details about data output from the Join stage.

The Join stage can have only one output link.

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

Details about Join 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 Join stage, the Mapping tab allows you to specify how the output columns are derived, that is, what input columns map onto them.

The left pane shows the input columns from the links whose tables have been joined. These are read only and cannot be modified on this tab.

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