



tAzureSqlDWBulkExec

Function	<p>This component ingests data residing in Azure Blob Storage to LOAD data into Azure SQL Data Warehouse using PolyBase.</p> <p>For more information on the detailed steps behind this component, please refer to: Load data with PolyBase in SQL Data Warehouse.</p>
Purpose	<p>As a dedicated component, it allows gains in performance during Insert operations to Azure SQL Data Warehouse.</p>

tAzureSqlDWBulkExec properties

Basic settings	Property Type	Either Built-In or Repository . <ul style="list-style-type: none">• Built-In: No property data stored centrally.• Repository: Select the repository file where the properties are stored.
<i>Database settings</i>	<i>Host</i>	Type in the IP address or hostname of the database server.
	<i>Port</i>	Type in the listening port number of the database server.
	<i>Database</i>	Type in the name of the database.
	<i>Schema</i>	Type in the name of the schema.
	<i>Username and Password</i>	Type in the database user authentication data. To enter the password, click the [...] button next to the password field, and then in the pop-up dialog box enter the password between double quotes and click OK to save the settings.
	<i>Table</i>	Specify the name of the table to be written. Note that only one table can be written at a time.
	<i>Additional JDBC Parameters</i>	Specify additional JDBC properties for the connection you are creating. The properties are separated by semicolon; and each property is a key-value pair. For example, <i>encrypt=true; trustServerCertificate=false</i> .
	<i>Action on table</i>	On the table defined, you can perform one of the following operations: <ul style="list-style-type: none">• None: No operation is carried out.• Drop and create table: The table is removed and created again.• Create table: The table does not exist and gets created.• Create table if not exists: The table is created if it does not exist.

		<ul style="list-style-type: none"> • Drop table if exists and create: The table is removed if it already exists and created again. • Clear table: The table content is deleted. You have the possibility to rollback the operation. • Truncate table: The table content is truncated.
	<i>Schema and Edit schema</i>	A schema is a row description. It defines the number of fields (columns) to be processed and passed on to the next component. The schema is either Built-In or stored remotely in the Repository .
		Built-In: You create and store the schema locally for this component only. Related topic: see <i>Talend Studio User Guide</i> .
		Repository: You have already created the schema and stored it in the Repository. You can reuse it in various projects and Job designs. Related topic: see <i>Talend Studio User Guide</i> .
		<p>Click Edit schema to make changes to the schema. If the current schema is of the Repository type, three options are available:</p> <ul style="list-style-type: none"> • View schema: choose this option to view the schema only. • Change to built-in property: choose this option to change the schema to Built-in for local changes. • Update repository connection: choose this option to change the schema stored in the repository and decide whether to propagate the changes to all the Jobs upon completion. If you just want to propagate the changes to the current Job, you can select No upon completion and choose this schema metadata again in the [Repository Content] window.
<i>Azure Storage Connection</i>	<i>Account name</i>	Enter the name of the storage account you need to access. A storage account name can be found in the Manage Access Keys dashboard of the Microsoft Azure Storage system to be used.
	<i>Account Key</i>	Enter the key associated with the storage account you need to access. Two keys are available for each account and by default, either of them can be used for this access.
	<i>Protocol</i>	Select the protocol for this connection to be created.
	<i>Container</i>	<p>Enter the name of the container you need to write files in.</p> <p>This container must exist in the Azure Storage system you are using.</p>
	<i>Azure Storage Location</i>	Enter the path to the virtual blob folder or file in the remote Azure storage system.

Advanced settings		
<i>Load Parameters</i>	<i>File Format</i>	<p>Select the type of the file that contains the data to be loaded.</p> <p>PolyBase supports these file formats:</p> <ul style="list-style-type: none"> • Delimited text • Hive RCFile • Hive ORC • Parquet <p>For more information on supported file formats, please refer to Microsoft's documentation: https://msdn.microsoft.com/en-us/library/dn935026.aspx</p>
	<i>Field Separator</i>	Applies only to Delimited Text files. Specify one or more characters that mark the end of each field (column) in the text-delimited file.
	<i>Enclosed By</i>	Specify the string delimiter for data of type string in the Delimited Text file. The string delimiter is one or more characters in length. The default option is the Empty that equates to an empty string "".
	<i>Date Format</i>	Specify a custom format for all date and time data that might appear in a Delimited Text file. If the source file uses default datetime formats, this option is not necessary.
	<i>Use Default Type</i>	<p>Specify how to handle missing values in delimited text files when PolyBase retrieves data from the text file.</p> <ul style="list-style-type: none"> • False (default): Stores all missing values as NULL. Any NULL values that are stored by using the word NULL in the delimited text file will be imported as the string 'NULL'. • True: When retrieving data from the text file, stores each missing value by using the default value for the data type of the corresponding column in the external table definition. For example, replace a missing value with: <ul style="list-style-type: none"> ○ 0 if the column is defined as a numeric column. ○ Empty string "" if the column is a string column. ○ 1900-01-01 if the column is a date column.
	<i>Serde Method</i>	<p>Specify a Hive Serializer and Deserializer (SerDe) method. This option is only available for Hive RCFile files. PolyBase supports the following SerDe methods:</p> <ul style="list-style-type: none"> • Lazy Binary: org.apache.hadoop.hive.serde2.columnar.LazyBinaryColumnarSerDe • Columnar: org.apache.hadoop.hive.serde2.columnar.ColumnarSerDe

	<i>Compressed By</i>	Select this check box and from the list displayed select the compression type of the source file.
	<i>Data Import Reject Options</i>	<p>Select this check box to specify reject parameters that determine how PolyBase will handle dirty records it retrieves from the external data source. A data record is considered 'dirty' if its actual data types or the number of columns do not match the column definitions of the external table.</p> <p>Please refer to Microsoft's documentation for more information: https://msdn.microsoft.com/en-us/library/dn935021.aspx</p>
	<i>Reject Type</i>	Clarifies whether the Reject Value option is specified as a literal value or a percentage.
	<i>Reject Value</i>	<p>Specifies the value or the percentage of rows that can be rejected before the query fails.</p> <ul style="list-style-type: none"> • For Reject Type = value, must be an integer between 0 and 2,147,483,647. • For Reject Type = percentage, must be a float between 0 and 100.
	<i>Reject Sample Value</i>	This attribute is required when you specify Reject Type = percentage. It determines the number of rows to attempt to retrieve before the PolyBase recalculates the percentage of rejected rows.
<i>Data Warehouse Table Properties</i>	<i>Distribution Option</i>	<p>To understand how to choose the best distribution method and use distributed tables, see Microsoft's documentation: Distributing tables in Azure SQL Data Warehouse.</p> <p>DISTRIBUTION = HASH (distribution_column_name)</p> <p>Assigns each row to one distribution by hashing the value stored in distribution_column_name. The algorithm is deterministic which means it always hashes the same value to the same distribution. The distribution column should be defined as NOT NULL since all rows that have NULL will be assigned to the same distribution.</p> <p>DISTRIBUTION = ROUND ROBIN</p> <p>Distributes the rows evenly across all the distributions in a round-robin fashion. This is the default for SQL Data Warehouse.</p> <p>DISTRIBUTION = REPLICATE -- Applies only to Parallel Data Warehouse.</p>

		Stores one copy of the table in full on each Compute node. Within each Compute node, the table is stored in a SQL Server filegroup that spans the Compute node. This is the default for Parallel Data Warehouse.
	<i>Distribution Column Name</i>	This attribute is required when you specify Distribution Option = hash.
	<i>Table Option</i>	<p>For guidance on choosing the type of table, see Microsoft’s Documentation: Indexing tables in Azure SQL Data Warehouse.</p> <p>CLUSTERED COLUMNSTORE INDEX</p> <p>Stores the table as a clustered columnstore index. The clustered columnstore index applies to all of the table data. This is the default for SQL Data Warehouse.</p> <p>HEAP</p> <p>Stores the table as a heap. This is the default for Parallel Data Warehouse.</p> <p>CLUSTERED INDEX</p> <p>Stores the table as a clustered index with one or more key columns. This stores the data by row. Use Index Column(s) field to specify the name of one or more key columns in the index.</p>
	<i>Index Column(s)</i>	Specify the name of one or more key columns for the CLUSTERED INDEX Table Option . Default index order is ASC. Specific order can be specified in this field as: “indexcolumn1 ASC, indexcolumn2 DESC”.
	<i>Partition</i>	<p>Select this check box to partition the table.</p> <p>For guidance on using table partitions, see Microsoft’s documentation: Partitioning tables in SQL Data Warehouse.</p>
	<i>Partition Column Name</i>	Specify the column that SQL Data Warehouse will use to partition the rows. This column can be any data type. SQL Data Warehouse sorts the partition column values in ascending order. The low-to-high ordering goes from LEFT to RIGHT for the purpose of the Range specification.
	<i>Range</i>	Specify if the the boundary value belongs to the partition on the left (lower values) or on the right (higher values). The default is LEFT.
	<i>Partition for Values</i>	Specifies the boundary values for the partition.
	<i>tStatCatcher Statistics</i>	Select this check box to gather the Job processing metadata at the Job level as well as at each component level.

Global Variables	<p>ERROR_MESSAGE: the error message generated by the component when an error occurs. This is an After variable and it returns a string. This variable functions only if the Die on error check box is cleared, if the component has this check box.</p> <p>A Flow variable functions during the execution of a component while an After variable functions after the execution of the component.</p> <p>To fill up a field or expression with a variable, press Ctrl + Space to access the variable list and choose the variable to use from it.</p> <p>For further information about variables, see <i>Talend Studio User Guide</i>.</p>
Usage	<p>This component is mainly used to load data in bulk to the Azure SQL Data Warehouse.</p>
Log4j	<p>If you are using a subscription-based version of the Studio, the activity of this component can be logged using the <i>log4j</i> feature. For more information on this feature, see <i>Talend Studio User Guide</i>.</p> <p>For more information on the log4j logging levels, see the Apache documentation at http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html.</p>