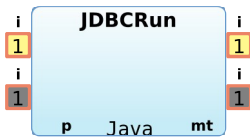


Operator JDBCRun

[IBMStreams.com.ibm.streamsx.jdbc Toolkit](#) > [com.ibm.streamsx.jdbc 1.4.0](#) > [com.ibm.streamsx.jdbc](#) > JDBCRun



The `JDBCRun` operator runs a user-defined SQL statement that is based on an input tuple. The statement is run once for each input tuple received. Result sets that are produced by the statement are emitted as output stream tuples. The `JDBCRun` operator is commonly used to update, merge, and delete database management system (DBMS) records. This operator is also used to retrieve records, create and drop tables, and to call stored procedures. # Behavior in a consistent region The `JDBCRun` operator can be used in a consistent region. It cannot be the start operator of a consistent region. In a consistent region, the configured value of the `transactionSize` is ignored. Instead, database commits are performed (when supported by the DBMS) on consistent region checkpoints, and database rollbacks are performed on consistent region resets. On drain: If there are any pending statements, they are run. If the statement generates a result set and the operator has an output port, tuples are generated from the results and submitted to the output port. If the operator has an error output port and the statement generates any errors, tuples are generated from the errors and submitted to the error output port. On checkpoint: A database commit is performed. On reset: Any pending statements are discarded. A rollback is performed. The new version of toolkit 1.3.x. supports also `optional type`. The SPL applications based on new JDBC toolkit and created with a new Streams that supports `optional type` are able to write/read 'null' to/from a nullable column in a table.

Summary

Ports

This operator has 2 input ports and 2 output ports.

Windowing

This operator does not accept any windowing configurations.

Parameters

This operator supports 26 parameters.

Required: [jdbcClassName](#), [jdbcDriverLib](#), [jdbcUrl](#)

Optional: [batchSize](#), [checkConnection](#), [commitInterval](#), [commitPolicy](#), [hasResultSetAttr](#), [isolationLevel](#), [jdbcPassword](#), [jdbcProperties](#), [jdbcUser](#), [keyStore](#), [keyStorePassword](#), [reconnectionBound](#), [reconnectionInterval](#), [reconnectionPolicy](#), [sqlFailureAction](#), [sqlStatusAttr](#), [sslConnection](#), [statement](#), [statementAttr](#), [statementParamAttrs](#), [transactionSize](#), [trustStore](#), [trustStorePassword](#)

Metrics

This operator does not report any metrics.

Properties

Implementation

Java

Input Ports

Ports (0)

The `JDBCRun` operator has one required input port. When a tuple is received on the required input port, the operator runs an SQL statement.

Properties

[Optional](#): false
[ControlPort](#): false
[WindowingMode](#): NonWindowed
[WindowPunctuationInputMode](#): Oblivious

Ports (1)

The `JDBCRun` operator has one optional input port. This port allows operator to change jdbc connection information at run time.

Properties

[Optional](#): true
[ControlPort](#): true
[WindowingMode](#): NonWindowed
[WindowPunctuationInputMode](#): Oblivious

Output Ports

Assignments

Java operators do not support output assignments.

Ports (0)

The `JDBCRun` operator has one required output port. The output port submits a tuple for each row in the result set of the SQL statement if the statement produces a result set. The output tuple values are assigned in the following order: 1. Columns that are returned in the result set that have same name from the output tuple 2. Auto-assigned attributes of the same name from the input tuple

Properties

[Optional](#): false

[WindowPunctuationOutputMode](#): Free

Ports (1)

The `JDBCRun` operator has one optional output port. This port submits tuples when an error occurs while the operator is running the SQL statement. The tuples deliver `sqlCode`, `sqlStatus` and `sqlMessage`.

Properties

[Optional](#): true

[WindowPunctuationOutputMode](#): Free

Parameters

This operator supports 26 parameters.

Required: [jdbcClassName](#), [jdbcDriverLib](#), [jdbcUrl](#)

Optional: [batchSize](#), [checkConnection](#), [commitInterval](#), [commitPolicy](#), [hasResultSetAttr](#), [isolationLevel](#), [jdbcPassword](#), [jdbcProperties](#), [jdbcUser](#), [keyStore](#), [keyStorePassword](#), [reconnectionBound](#), [reconnectionInterval](#), [reconnectionPolicy](#), [sqlFailureAction](#), [sqlStatusAttr](#), [sslConnection](#), [statement](#), [statementAttr](#), [statementParamAttrs](#), [transactionSize](#), [trustStore](#), [trustStorePassword](#)

batchSize

This optional parameter specifies the number of statement to execute as a batch. The default batch size is 1.

Properties

[Type](#): int32

[Cardinality](#): 1

[Optional](#): true

checkConnection

This optional parameter specifies whether a **checkConnection** thread should be start. The thread checks periodically the status of JDBC connection. The `JDBCRun` sends in case of any connection failure a `SqlCode` and a message to SPL application. The default value is `false`.

Properties

[Type](#): boolean

[Cardinality](#): 1

[Optional](#): true

commitInterval

This parameter sets a commit interval for the sql statements that are being processed and overrides the `batchSize` and `transactionSize` parameters.

Properties

[Type](#): int32

[Cardinality](#): 1

[Optional](#): true

commitPolicy

This parameter specifies the commit policy that should be used when the operator is in a consistent region. If set to *OnCheckpoint*, then commits will only occur during checkpointing. If set to *OnTransactionAndCheckpoint*, commits will occur during checkpointing as well as whenever the **transactionCount** or **commitInterval** are reached. The default value is *OnCheckpoint*. It is recommended that the *OnTransactionAndCheckpoint* value be set if the tables that the statements are being executed against can tolerate duplicate entries as these parameter value may cause the same statements to be executed if the operator is reset. It is also highly recommended that the **transactionCount** parameter not be set to a value greater than 1 when the policy is *OnTransactionAndCheckpoint*, as this can lead to some statements not being executed in the event of a reset. This parameter is ignored if the operator is not in a consistent region. The default value for this parameter is *OnCheckpoint*.

Properties

[Type](#): com.ibm.streamsx.jdbc.CommitPolicy (OnCheckpoint, OnTransactionAndCheckpoint)

[Cardinality](#): 1

[Optional](#): true

[ExpressionMode](#): CustomLiteral

hasResultSetAttr

This parameter points to an output attribute and returns true if the statement produces result sets, otherwise, returns false

Properties

[Type](#): rstring
[Cardinality](#): 1
[Optional](#): true

isolationLevel

This optional parameter specifies the transaction isolation level at which statement runs. If omitted, the statement runs at level READ_UNCOMMITTED.

Properties

[Type](#): rstring
[Cardinality](#): 1
[Optional](#): true

jdbcClassName

This required parameter specifies the class name for jdbc driver and it must have exactly one value of type rstring.

Properties

[Type](#): rstring
[Cardinality](#): 1
[Optional](#): false

jdbcDriverLib

This required parameter of type rstring specifies the path and the file name of jdbc driver libraris with comma separated in one string.

Properties

[Type](#): rstring
[Cardinality](#): 1
[Optional](#): false

jdbcPassword

This optional parameter specifies the user's password. If the jdbcPassword parameter is specified, it must have exactly one value of type rstring.

Properties

[Type](#): rstring
[Cardinality](#): 1
[Optional](#): true

jdbcProperties

This optional parameter specifies the path name of the file that contains the jdbc connection properties.

Properties

[Type](#): rstring
[Cardinality](#): 1
[Optional](#): true

jdbcUrl

This parameter specifies the database url that JDBC driver uses to connect to a database and it must have exactly one value of type rstring. The syntax of jdbc url is specified by database vendors. For example, jdbc:db2://<server>:<port>/<database>

. jdbc:db2 indicates that the connection is to a DB2 for z/OS, DB2 for Linux, UNIX, and Windows.

. server, the domain name or IP address of the data source.

. port, the TCP/IP server port number that is assigned to the data source.

. database, a name for the data source

Properties

[Type](#): rstring
[Cardinality](#): 1
[Optional](#): false

jdbcUser

This optional parameter specifies the database user on whose behalf the connection is being made. If the jdbcUser parameter is specified, it must have exactly one value of type rstring.

Properties

[Type](#): rstring
[Cardinality](#): 1
[Optional](#): true

keyStore

This optional parameter specifies the path to the keyStore. If a relative path is specified, the path is relative to the application directory. The **sslConnection** parameter must be set to `true` for this parameter to have any effect.

Properties

[Type](#): rstring
[Cardinality](#): 1
[Optional](#): true

keyStorePassword

This parameter specifies the password for the keyStore given by the **keyStore** parameter. The **sslConnection** parameter must be set to `true` for this parameter to have any effect.

Properties

[Type](#): rstring
[Cardinality](#): 1
[Optional](#): true

reconnectionBound

This optional parameter specifies the number of successive connection attempts that occur when a connection fails or a disconnect occurs. It is used only when the **reconnectionPolicy** parameter is set to `BoundedRetry`; otherwise, it is ignored. The default value is 5.

Properties

[Type](#): int32
[Cardinality](#): 1
[Optional](#): true

reconnectionInterval

This optional parameter specifies the amount of time (in seconds) that the operator waits between successive connection attempts. It is used only when the **reconnectionPolicy** parameter is set to `BoundedRetry` or `InfiniteRetry`; otherwise, it is ignored. The default value is 10.

Properties

[Type](#): float64
[Cardinality](#): 1
[Optional](#): true

reconnectionPolicy

This optional parameter specifies the policy that is used by the operator to handle database connection failures. The valid values are: `NoRetry`, `InfiniteRetry`, and `BoundedRetry`. The default value is `BoundedRetry`. If `NoRetry` is specified and a database connection failure occurs, the operator does not try to connect to the database again. The operator shuts down at startup time if the initial connection attempt fails. If `BoundedRetry` is specified and a database connection failure occurs, the operator tries to connect to the database again up to a maximum number of times. The maximum number of connection attempts is specified in the **reconnectionBound** parameter. The sequence of connection attempts occurs at startup time. If a connection does not exist, the sequence of connection attempts also occurs before each operator is run. If `InfiniteRetry` is specified, the operator continues to try and connect indefinitely until a connection is made. This behavior blocks all other operator operations while a connection is not successful. For example, if an incorrect connection password is specified in the connection configuration document, the operator remains in an infinite startup loop until a shutdown is requested.

Properties

[Type](#): rstring
[Cardinality](#): 1
[Optional](#): true

sqlFailureAction

This optional parameter has values of `log`, `rollback` and `terminate`. If not specified, `log` is assumed. If `sqlFailureAction` is `log`, the error is logged, and the error condition is cleared. If `sqlFailureAction` is `rollback`, the error is logged, the transaction rolls back. If `sqlFailureAction` is `terminate`, the error is logged, the transaction rolls back and the operator terminates.

Properties

[Type](#): rstring
[Cardinality](#): 1
[Optional](#): true

sqlStatusAttr

This parameter points to one or more output attributes and returns the SQL status information, including SQL code (the error number associated with the `SQLException`) and SQL state (the five-digit `XOPEN` `SQLState` code for a database error)

Properties

[Type](#): rstring
[Cardinality](#): 1
[Optional](#): true

sslConnection

This optional parameter specifies whether an SSL connection should be made to the database. When set to `true`, the **keyStore**, **keyStorePassword**, **trustStore** and **trustStorePassword** parameters can be used to specify the locations and passwords of the keyStore and trustStore. The default value is `false`.

Properties

[Type](#): boolean

[Cardinality](#): 1
[Optional](#): true

statement

This parameter specifies the value of any valid SQL or stored procedure statement. The statement can contain parameter markers

Properties

[Type](#): rstring
[Cardinality](#): 1
[Optional](#): true

statementAttr

This parameter specifies the value of complete SQL or stored procedure statement that is from stream attribute (no parameter markers).

Properties

[Type](#): rstring
[Cardinality](#): 1
[Optional](#): true
[ExpressionMode](#): Attribute

statementParamAttrs

This optional parameter specifies the value of statement parameters. The statementParameter value and SQL statement parameter markers are associated in lexicographic order. For example, the first parameter marker in the SQL statement is associated with the first statementParameter value.

Properties

[Type](#): rstring
[Cardinality](#): 1
[Optional](#): true

transactionSize

This optional parameter specifies the number of executions to commit per transaction. The default transaction size is 1 and transactions are automatically committed.

Properties

[Type](#): int32
[Cardinality](#): 1
[Optional](#): true

trustStore

This optional parameter specifies the path to the trustStore. If a relative path is specified, the path is relative to the application directory. The **sslConnection** parameter must be set to `true` for this parameter to have any effect.

Properties

[Type](#): rstring
[Cardinality](#): 1
[Optional](#): true

trustStorePassword

This parameter specifies the password for the trustStore given by the **trustStore** parameter. The **sslConnection** parameter must be set to `true` for this parameter to have any effect.

Properties

[Type](#): rstring
[Cardinality](#): 1
[Optional](#): true

[Libraries](#)

Operator class library

[Library Path](#): ../../impl/lib/com.ibm.streamsx.jdbc.jar