Cisco ParStream

Cisco ParStream DSA Link Guide

<October 25, 2017>

Document Information:

Title:	Cisco ParStream DSA Link Guide
Version:	3.2.0
Date Published:	<October 25, 2017>
Date Printed:	October 25, 2017

www.cisco.com

Table of Contents

1 Cisco ParStream DSA Link

This document describes the usage of the Cisco ParStream DSA Link.

1.1 Introduction

Distributed Services Architecture (DSA) from DGLogik, is an open source IoT platform that facilitates device inter-communication, logic, and applications at every layer of the Internet of Things infrastructure. For an introduction and documentation about DSA see `http://iot-dsa.org/`.

The Cisco ParStream DSA Link is able to insert data into single tables and query the database using SQL statements. Additionally, the link provides a historian interface to import time stamped single value data into a Cisco ParStream database.

1.2 Installation

The Cisco ParStream DSA Link can be easily installed by the DGLux5 web application via action *Install Link from ZIP* found at node *sys/links*.

From the technical side, the ZIP file will be unpacked into a sub-folder of `dglux-server/dslinks/` and started with the node description `dslink.json` as initial configuration by the DSA broker. The Cisco ParStream DSA Link is a Java application and the DSA broker will run the script `bin/parstream-dsa-link`. If this link terminates unexpectedly, the DSA broker will restart the Cisco ParStream DSA Link automatically.

Optionally, you can start the Cisco ParStream DSA Link as a stand-alone Java application with the standard DSA Link options:

`--name` Name of this Cisco ParStream DSA Link instance

`--broker` URL of the DSA broker (i.e. http://127.0.0.1:8080/conn or https://127.0.0.1:8443/conn)

`--nodes` Name of the node configuration file (default: nodes.json)

`--log` Logging level (default: info)

`--key` Name of the node key file (default: .key)

`--token` Token used by the DSA broker

In this case, you have to restart the link manually if the Java application terminates unexpectedly.

1.2.1 Minimum Requirements

- Cisco ParStream 4.4.0
- Java 8
- DSA Broker accepting sdk-dslink-java v0.15.0 connections
- Supported Operating System for Cisco ParStream Java Streaming Import

1.2.2 Networking

The Cisco ParStream DSA Link will connect to its DSA broker via tcp/http(s) communication. If the link was started from the DSA broker, the link will use a tcp connection to a local dynamic port of the DSA broker. If the link was started manually, the link has to use the public http or https DSA broker port.

The link will also connect to configured Cisco ParStream databases via JDBC and netcat ports.

1.2.3 Filesystem

The installed Cisco ParStream DSA Link consists of:

Artifact	Type	Brief Description
`bin`	folder	Contains the start scripts
`lib`	folder	Contains all depending libraries
`.key`	file	Contains the DSA broker token
`dslink-json`	file	Configuration for the initial link configuration and start
`nodes.json`	file	Configuration for the current link configuration and current metric information

Table 1.1: *Installation artifacts*

Upgrade Files

With a new version the installation zip file will contain a `"libnew"` folder, which will replace the `"lib"` folder while upgrading the link right before starting the link.

Logging Files

Normally the DSA broker will start the Cisco ParStream DSA Link, capture all output and log them to the file `"./dglux-server/logs/<link instance name>"`. If the link is started manually, the user has to take care to capture and log all output.

1.2.4 Configuration

The Cisco ParStream DSA Link node will be automatically configured with its first connection to its DSA broker. The node configuration is a list of connection configurations consisting of database table configuration and watch group configuration. The connection parameters are listed in Table 1.2.

The table configuration will be automatically configured while introspecting the database connection. The table configuration parameters listed in Table 1.3 are set initially to their default values. These parameters define the commit behavior for importing data into the Cisco ParStream database.

Note: Using the default configuration the Cisco ParStream DSA Link will commit every single row to the database. This will lead to performance problems, if the commit rate is too high. Therefore `$insertBufferMaxSize` or `$insertBufferMaxDelayMs` or both should be set to a proper value in order to perform bulk inserts into the database.

Watch group configurations are defined by the historian API provided by the DSA Java SDK. For partitioning new historian table with the standard historian API the default configuration listed in Table 1.4 will be used.

Warning: The historian API provides an additional message buffer. By default this buffer is unbound and will be flushed every 5 seconds by writing each message one by one into the database. This contradicts the table based buffer configuration. Setting the "Buffer Flush Time" of the watch group to 0 will disable the historian API buffer. However the table based buffer configuration of the Cisco ParStream DSA Link will be respected while importing historian data.

Connection Parameter	Brief Description
`$name`	Name of this connection
`$host`	Hostname for this connection
`$port`	Port for this connection
`$user`	Login name for the generic database account
`$pwd`	Passphrase for the generic database account
`$importPriority`	Import priority (normally set to "MEDIUM")

Table 1.2: *Brief Description of the Connection Configuration*

Table Import Parameter	Default Value	Brief Description
`$name`	N/A	Name of the table
`$insertBufferMaxSize`	`0`	Number of rows to be collected before auto-committing (disable buffering by row count if value is 0 or negative; any 32bit signed integer value is accepted)
`$insertBufferMaxDelayMs`	`0`	Milliseconds since last commit to wait before auto-committing pending rows (disable buffering by delay if value is 0 or negative; any 32bit signed integer value is accepted))
`$columnInfo`	N/A	Description of the table columns (will be set while introspecting this table)
`$outputChannel`	`PARSTREAM`	Type of output: PARSTREAM for streaming into a Cisco ParStream Server or CSVFILE for writing into a CSV File
`$outputCsvFolder`	N/A	The output folder to write CSV files, if outputChannel is CSVFILE
`$outputCsvPrefix`	table name	Prefix of the CSV file name

Table 1.3: *Brief Description of the Table Import Configuration*

Table Import Parameter	Default Value	Brief Description
`$defaultTimeSlicing`	`MONTH`	Segmentation granularity for new historian tables by time. (valid values are HOUR, DAY, WEEK or MONTH)
`$defaultWatchPathModulo`	`11`	If set, maximum number of partitions per time-slice, which are grouped by watched paths. (valid values are >0)

Table 1.4: *Brief Description of the Watch Groups Configuration*

1.3 Link Actions

The primary actions of this link are "inserting a single row" and "querying the database" using SQL. All available actions of the Cisco ParStream DSA Link are described in the following chapters.

1.3.1 Node Actions

Action Name	API Name	Brief Description
Add Database	addDb	Create new database connection

Add Database

Create a new database connection to a Cisco ParStream database node.

Parameter	Type	Default	Description
name	String		Name of the connection
host	String		IP address or host name of the Cisco ParStream database
port	String		Port or service name of the Cisco ParStream database
user	String	Null	Database account name for queries and imports
password	String	Null	Database account password
autoIntrospect	Boolean		Introspect tables after create this connection
importPriority	Enum		Import query priority (LOW, MEDIUM, HIGH)

1.3.2 Connection Actions

Action Name	API Name	Brief Description
Disconnect	disconnect	Close database connection
Introspect	introspect	Update table list
Query	query	Query database
Delete	delete	Delete this connection
Connect	connect	Connect to database

Disconnect

Disconnect will close all TCP/IP connections to the database. This includes netcat and JDBC connections.

Introspect

Introspect will override all table column information and might change the `insertRow` action parameters. In addition, all non existing tables are removed from the table list.

Query

Query the database by a single SQL statement.

Parameter	Type	Default	Description
sqlQuery	String		Single SQL query statement

This action will fail to process statements which do not return a proper result set, like DDL statements. However these SQL statements will be processed on the database.

Delete

Delete this database connection.

Connect

Connect to this configured database.

1.3.3 Table Actions

1.3.4 Node Actions

Action Name	API Name	Brief Description
Insert Row	insertRow	Insert a single row
Insert CSV	insertCSV	Insert CSV content into this table
Set insert buffer max size	setMaxBufferSize	Set row count for auto-commit
Set insert buffer max delay	setMaxBufferDelay	Set duration to wait for auto-commit

Insert Row

Insert values as a single row into this table. The parameter names are the column names of the target table.

Parameter	Type	Default	Description
<dynamic>	<dynamic>		Value of named column

The type of the field is deviated from the column type.

Result Name	Type	Description
Status	String	Result of insert invocation (FAILED, FAILED_FATAL, BUFFERED, INSERTED)

Insert CSV

Insert multiple rows from a CSV content into this table.

Parameter	Type	Default	Description
Csv	String		Standard comma separated format, as for RFC4180 but allowing empty lines.

The CSV columns must match the exact order defined by the table create statement. An optional CSV header containing the column names will be ignored. The type of the field is deviated from the column type.

Result Name	Type	Description
Status	String	Result of insert invocation (FAILED, FAILED_FATAL, BUFFERED, INSERTED)

Set insert buffer max size

Set row count for auto-commit.

Parameter	Type	Default	Description
Size	Number	Null	Maximum number of rows to wait before committing to database. Null or 0 will disable row buffering by row count.

See configuration `$insertBufferMaxSize` in Table 1.3 for parameter details.

Set insert buffer max delay

Set duration in milliseconds to wait before auto-committing pending rows.

Parameter	Type	Default	Description
Delay (ms)	Number	Null	Maximum number of milliseconds to wait before committing to database. Null or 0 will disable row buffering by delay time.

See configuration `$insertBufferMaxDelayMs` in Table 1.3 for parameter details.

1.3.5 Historian Actions

Action Name	API Name	Brief Description
Create Watch Group	createWatchGroup	Create a new watch group
Set Default Partitioning	setPartitioningDefaults	Set default partitioning configuration

Create Watch Group

Create a new watch group with additional table partitioning configuration and optional deletion job creation.

Parameter	Type	Default	Description
Name	String		Name of the new watch group
Buffer max Delay	Number	0	Maximum number of milliseconds to wait before committing to database. Null or 0 will disable row buffering by delay time.
Buffer max Size	Number	0	Maximum number of rows to wait before committing to database. Null or 0 will disable row buffering by row count.
Max Time Slices	Number	0	Maximum number of time-slices to retain in the database. Older time-slices will be purged periodically. If the number is zero no time-slices will be purged.
Time Slicing	Enum	<configuration>	Segmentation granularity for historian table by time (HOUR, DAY, WEEK, MONTH)
Buckets per Time Slice	Number	<configuration>	Maximum number of partitions grouped by watched paths. Null will disable partitioning by watched paths.

Each watch group will create a generic table as historian storage. The name of the watch group will be used as table name prefixed with `"WG_"`. The name must meet the requirements as a Cisco ParStream table name. A space is converted into an underscore character. If the parameter "max time slices" is a positive number, a deletion job will be created for this historian table.

Set Default Partitioning

Set default partitioning configuration for new historian tables.

Parameter	Type	Default	Description
Buckets per Time Slice	Number		Maximum number of partitions grouped by watched paths. Null will disable partitioning by watched paths.
Time Slicing	Number		Segmentation granularity for historian table by time (HOUR, DAY, WEEK, MONTH)

See Table 1.4 for parameter details.

1.3.6 Watch Group Actions

Action Name	API Name	Brief Description
Edit	edit	Edit watch group
Add Watch Path	addWatchPath	Add a new path to be monitored in this group
Delete	delete	Delete this watch group

Any of these action are predefined by the IOT-DSA Java Historian SDK.

1.3.7 Watch Path Metric Actions

Action Name	API Name	Brief Description
Get History	getHistory	Read historian data from the database
Unsubscribe	unsubscribe	Remove this path to be monitored in this group

Any of these action are predefined by the IOT-DSA Java Historian SDK.

Unsubscribe

Unsubscribe will not delete any data from this watch path in the historian table.

1.4 Known Issues for Version 3.2

- (PSDSLINK-138) updates of metric information will alter the link configuration
- (PSDSLINK-201) Get History cannot handle data rollup functionality correctly
  As workaround you can sort and rollup the table data in a dataflow.
- (PSDSLINK-202) Get History cannot handle named timeranges correctly

Americas Headquarters
Cisco Systems, Inc.
San Jose, CA

Asia Pacific Headquarters
Cisco Systems (USA) Pte. Ltd.
Singapore

Europe Headquarters
Cisco Systems International BV Amsterdam,
The Netherlands

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at **www.cisco.com/go/offices.**