

Using Digital Event Persistence Driver for JDBC

Version 10.3

October 2018

This document applies to webMethods Digital Event Persistence Version 10.3 and to all subsequent releases.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Copyright © 2018 Software AG, Darmstadt, Germany and/or Software AG USA Inc., Reston, VA, USA, and/or its subsidiaries and/or its affiliates and/or their licensors.

The name Software AG and all Software AG product names are either trademarks or registered trademarks of Software AG and/or Software AG USA Inc. and/or its subsidiaries and/or its affiliates and/or their licensors. Other company and product names mentioned herein may be trademarks of their respective owners.

Detailed information on trademarks and patents owned by Software AG and/or its subsidiaries is located at <http://softwareag.com/licenses>.

Use of this software is subject to adherence to Software AG's licensing conditions and terms. These terms are part of the product documentation, located at <http://softwareag.com/licenses> and/or in the root installation directory of the licensed product(s).

This software may include portions of third-party products. For third-party copyright notices, license terms, additional rights or restrictions, please refer to "License Texts, Copyright Notices and Disclaimers of Third Party Products". For certain specific third-party license restrictions, please refer to section E of the Legal Notices available under "License Terms and Conditions for Use of Software AG Products / Copyright and Trademark Notices of Software AG Products". These documents are part of the product documentation, located at <http://softwareag.com/licenses> and/or in the root installation directory of the licensed product(s).

Use, reproduction, transfer, publication or disclosure is prohibited except as specifically provided for in your License Agreement with Software AG.

Table of Contents

About this Guide.....	5
Document Conventions.....	5
Online Information and Support.....	6
Data Protection.....	7
About the Digital Event Persistence Driver for JDBC.....	9
Implementation Classes for the JDBC Driver.....	10
The JDBC Connection URL for Elasticsearch.....	10
Establishing a Connection to an Elasticsearch Cluster and Index.....	11
Before You Begin.....	12
Establishing a Connection Using the JDBC Driver Manager.....	12
Establishing a Connection Using the DataSource Class.....	12
Object Mapping Between Elasticsearch and JDBC Metadata.....	15
SQL Syntax for the JDBC Driver.....	17

About this Guide

This guide provides you with information about how to use the webMethods Digital Event Persistence driver for Java Database Connectivity (JDBC). Developers can use the driver with Software AG products, such as Apama, Integration Server, and MashZone NextGen, to view events persisted with Digital Event Persistence to Elasticsearch 2.3.2 and 5.6.4.

Document Conventions

Convention	Description
Bold	Identifies elements on a screen.
Narrowfont	Identifies service names and locations in the format <i>folder.subfolder.service</i> , APIs, Java classes, methods, properties.
<i>Italic</i>	Identifies: Variables for which you must supply values specific to your own situation or environment. New terms the first time they occur in the text. References to other documentation sources.
Monospace font	Identifies: Text you must type in. Messages displayed by the system. Program code.
{ }	Indicates a set of choices from which you must choose one. Type only the information inside the curly braces. Do not type the { } symbols.
	Separates two mutually exclusive choices in a syntax line. Type one of these choices. Do not type the symbol.
[]	Indicates one or more options. Type only the information inside the square brackets. Do not type the [] symbols.

Convention	Description
...	Indicates that you can type multiple options of the same type. Type only the information. Do not type the ellipsis (...).

Online Information and Support

Software AG Documentation Website

You can find documentation on the Software AG Documentation website at "<http://documentation.softwareag.com>". The site requires credentials for Software AG's Product Support site Empower. If you do not have Empower credentials, you must use the TECHcommunity website.

Software AG Empower Product Support Website

If you do not yet have an account for Empower, send an email to "empower@softwareag.com" with your name, company, and company email address and request an account.

Once you have an account, you can open Support Incidents online via the eService section of Empower at "<https://empower.softwareag.com/>".

You can find product information on the Software AG Empower Product Support website at "<https://empower.softwareag.com/>".

To submit feature/enhancement requests, get information about product availability, and download products, go to "[Products](#)".

To get information about fixes and to read early warnings, technical papers, and knowledge base articles, go to the "[Knowledge Center](#)".

If you have any questions, you can find a local or toll-free number for your country in our Global Support Contact Directory at "https://empower.softwareag.com/public_directory.asp" and give us a call.

Software AG TECHcommunity

You can find documentation and other technical information on the Software AG TECHcommunity website at "<http://techcommunity.softwareag.com>". You can:

- Access product documentation, if you have TECHcommunity credentials. If you do not, you will need to register and specify "Documentation" as an area of interest.
- Access articles, code samples, demos, and tutorials.
- Use the online discussion forums, moderated by Software AG professionals, to ask questions, discuss best practices, and learn how other customers are using Software AG technology.
- Link to external websites that discuss open standards and web technology.

Data Protection

Software AG products provide functionality with respect to processing of personal data according to the EU General Data Protection Regulation (GDPR). Where applicable, appropriate steps are documented in the respective administration documentation.

1 About the Digital Event Persistence Driver for JDBC

■ Implementation Classes for the JDBC Driver	10
■ The JDBC Connection URL for Elasticsearch	10

The Digital Event Persistence (DEP) driver for Java Database Connectivity (JDBC) enables you to view persisted events, using common tools that support the JDBC API. The JDBC driver provides access to digital events that have been persisted to Elasticsearch, using DEP. In addition, you can use the driver with Software AG MashZone NextGen to create dashboards against persisted historical events.

Implementation Classes for the JDBC Driver

The JDBC driver provides the following implementation classes:

- `com.softwareag.evp.jdbc.EvpDriver` - an implementation class for the driver.
- `com.softwareag.evp.jdbc.EvpDataSource` - a datasource implementation class for the driver.

The JDBC Connection URL for Elasticsearch

Use the following JDBC URL format to connect to an Elasticsearch cluster:

```
jdbc:evp:elasticsearch:5.6.4://host:port/cluster_name/index_name
```

where

- *host* is the hostname of the Elasticsearch server.
- *port* is the port number of the Elasticsearch server.
- *cluster_name* is the name of the Elasticsearch cluster.
- *index_name* is the name of the Elasticsearch index.

For example, you can have the following URL:

```
jdbc:evp:elasticsearch:5.6.4://localhost:9300/my_cluster/myindex
```

2 Establishing a Connection to an Elasticsearch Cluster and Index

■ Before You Begin	12
■ Establishing a Connection Using the JDBC Driver Manager	12
■ Establishing a Connection Using the DataSource Class	12

Before You Begin

Before you can establish a connection to an Elasticsearch cluster and index, you must place the `evp-jdbc-driver.jar` file in the classpath of your project. The `evp-jdbc-driver.jar` is located in the `Software AG_directory\common\EventPersistence\JDBC` directory.

If you use the driver in MashZone NextGen, you can specify the driver properties using a query string in the database URL, for example:

```
jdbc:evp:elasticsearch:5.6.4://localhost:9300/my_cluster/
myindex?installDir=path_to_JDBC_driver_directory
```

Establishing a Connection Using the JDBC Driver Manager

The examples in this section show you how to establish a connection to an Elasticsearch index and cluster, using the JDBC Driver Manager.

First, the following code shows how you pass the required property `"installDir"` and use the JDBC Driver Manager to create a connection to a locally installed Elasticsearch and the index `orders`.

```
String url = "jdbc:evp:elasticsearch:5.6.4://localhost:9300/cluster_name/orders";
Properties info = new Properties();
info.setProperty("installDir", path_to_JDBC_driver_installation_directory);
```

If the Elasticsearch instance you are connecting to is enabled for Secure Sockets Layer (SSL) with the Search Guard plugin, you must pass these additional properties:

```
info.setProperty("useSSL", true);
info.setProperty("keystore", "path_to_the_JKS_keystore_file");
info.setProperty("keystorePassword", "unencrypted_JKS_keystore_password");
info.setProperty("truststore", "path_to_the_JKS_truststore_file");
info.setProperty("truststorePassword", "unencrypted_JKS_truststore_password")
```

Then, use the following code to establish the connection:

```
try (Connection connection = DriverManager.getConnection(url, info)) {
    assertTrue(connection != null);
}
```

Establishing a Connection Using the DataSource Class

The examples in this section show you how to establish a connection to an Elasticsearch index and cluster, using the JDBC DataSource Implementation class.

First, the following code shows how you pass the required property `"installDir"` and use the JDBC DataSource Implementation class to create a connection to a locally installed Elasticsearch and the index `orders`.

```
EvpDataSource dataSource = new EvpDataSource();
dataSource.setDatabaseURL("jdbc:evp:elasticsearch:5.6.4://localhost:9300/
cluster_name/orders");
dataSource.setInstallDir(path_to_JDBC_driver_installation_directory);
```

If the Elasticsearch instance you are connecting to is enabled for Secure Sockets Layer (SSL) with the Search Guard plugin, you must make these additional method calls:

```
dataSource.setUseSSL(true);  
    dataSource.setKeystore("path_to_the_JKS_keystore_file");  
    dataSource.setKeystorePassword("unencrypted_JKS_keystore_password");  
    dataSource.setTruststore("path_to_the_JKS_truststore_file");  
    dataSource.setTruststorePassword("unencrypted_JKS_truststore_password");
```

Then, use the following code to establish the connection:

```
try(Connection connection = dataSource.getConnection()) {  
    assertTrue(connection != null);  
}
```


3 Object Mapping Between Elasticsearch and JDBC Metadata

The Java Database Connectivity (JDBC) specification defines a set of entities that correspond to specific Elasticsearch objects. These common entities are catalogs, schemas, tables, indices, and columns. The JDBC API provides interfaces to access metadata using standard methods.

The following table lists the common JDBC entities and their corresponding Elasticsearch objects.

JDBC Object	Elasticsearch Object
catalog	index
schema	"default"
table	type
column	field

Note: For Elasticsearch, the metadata API calls return no information on indices because all fields of an Elasticsearch document are indexed.

4 SQL Syntax for the JDBC Driver

Supported Syntax

The Digital Event Persistence (DEP) driver for Java Database Connectivity (JDBC) supports the following syntax:

- Column lists to specify the columns that the query returns, or * for all columns.
- Aggregate functions such as `count`, `sum`, `min`, `max`, and `avg`.
- Column aliases for renaming columns or assigning a name to an aggregate result in the results.
- Grouping of results for aggregate functions.
- Ordering of results for instance and aggregate queries.
- LIMIT clause for returning the first *n* results of a query.
- Standard comparison operators for filtering: `=`, `!=`, `<>`, `>`, `>=`, `<`, and `<=`.
- BETWEEN operator for timestamp, string, and numeric range comparisons.
- Logical operations such as `and` | `or`.
- Parentheses () to specify precedence.
- Date-based queries for `TIMESTAMP` column types.

Syntax Grammar

The following snippet provides an abbreviated grammar for the supported syntax:

```
SELECT * | column_list, function_list
FROM TABLE table
[WHERE search_condition]
[GROUP BY column_list]
[ORDER BY column_list]
[LIMIT n]
;
```

Unsupported Syntax and Features

This JDBC driver does not support the following features:

- Joins
- Nested SELECT statements
- IN and LIKE operators
- INSERT, UPDATE, and DELETE

In addition, the driver does not support the following digital event field types. These field types do not appear in any of the JDBC metadata APIs and they are not available via query:

- Any
- Array
- Map