

# DatabaseTable Connector

## Status

<b>Functionality</b>	Stable
<b>Support status</b>	Supported
<b>Support provided by</b>	Evolveum
<b>Origin</b>	Identity Connector Framework (Sun), many improvements by Evolveum
<b>Target systems</b>	Generic database table (JDBC)

## Description

Generic connector for simple database tables.

<b>Protocol</b>	Database-specific. Connector is using JDBC.
<b>Framework</b>	ConnId 1.4.x
<b>Bundle name</b>	com.evolveum.polygon.connector-databasetable
<b>Connector name</b>	org.identityconnectors.databasetable.DatabaseTableConnector
<b>Additional requirements</b>	Requires correct JDBC driver to be present on classpath

## Capabilities and Features

<b>Schema</b>	YES	
<b>Provisioning</b>	YES	
<b>Live Synchronization</b>	YES	Using last modification timestamps
<b>Password</b>	YES	
<b>Activation</b>	SIMULATED	Configured capability
<b>Scripting</b>	NO	

## History

This is almost entirely the original version from Sun ICF.

## Versions

Version	Origin	Binary	Sources	Build Date	Framework version	Description
1.1.0. e6296	OpenICF trunk version, built and tested by Evolveum	<a href="#">download jar</a>	<a href="#">OpenICF: trunk</a> , revision 6296	Jan 2013		Well tested (now obsolete)
1.4.0.49	Evolveum	<a href="#">download jar</a>	<a href="#">Evolveum OpenICF git repository (tag connector-databasetable-v1.4.0.49)</a>	16 Jan 2015		Bundled with midPoint.
1.4.2.0	Evolveum	<a href="#">download jar</a>	<a href="#">Evolveum OpenICF git repository (tag connector-databasetable-v1.4.2.0)</a>	June 2016	1.4.2.0	Current version. Bundled with midPoint 3.4.

1.4.3.0	Evolveum	<a href="#">download jar</a>	<a href="#">Evolveum OpenICF git repository (tag connector-databasetable-v1.4.3.0)</a>	October 2018	1.4.2.0	Sync Order Column (contributed by Martin Lizner) Current version. Bundled with midPoint 3.9.
1.4.4.0	Evolveum	<a href="#">download jar</a>	<a href="#">Evolveum OpenICF git repository (tag connector-databasetable-v1.4.4.0)</a>	July 2020	1.4.2.0	Exception handling for creating new connection

## Interoperability

This connector is supposed to work with all relational databases that follow the SQL standard and that provide a JDBC driver. The driver is **not** part of the connector distribution (except for PostgreSQL driver, which is distributed with midPoint). The deployer is responsible for using correct JDBC driver.

## Support

Only standard SQL behavior is supported. This means standard SQL operations and standard database schemas. Non-standard database operations and vendor-specific behavior is not supported.

Problems originating in JDBC driver, problem that are caused by non-compliance to SQL standard or vendor-specific behavior are **not** covered by the support.

The [bundled support](#) does not include support for any specific database server. Support for specific database servers needs to be explicitly negotiated in the support contract.

## Documentation

Unfortunately, there is no comprehensive documentation for this connector.

## Resource Examples

- [DatabaseTable](#)

## Notes

DatabaseTable connector now assumes that UID is the same as NAME. See <https://jira.evolveum.com/browse/MID-1127>.

The following two configuration properties were reported to resolve problems with Oracle DATE attribute

```
<icfsdbt:nativeTimestamps>true</icfsdbt:nativeTimestamps>
<icfsdbt:allNative>true</icfsdbt:allNative>
```

For PostgreSQL TIMESTAMP or TIMESTAMPTZ attribute only allNative configuration property is needed

```
<icfsdbt:allNative>true</icfsdbt:allNative>
```

In addition, the DATE, TIMESTAMP or TIMESTAMPTZ column must be provisioned as string with correct formatting, e.g.: `return new Date().format("yyyy-MM-dd HH:mm:ss")`, otherwise "ORA-01861: literal does not match format string" occurs (or similar exception in PostgreSQL).

TIMESTAMPTZ is stored including the timezone. DatabaseTable connector doesn't work with timezone directly, but the value is automatically converted to midPoint local timezone. Therefore, you always work with local timezone in mappings and it will be automatically stored correctly in the database.

PostgreSQL server might be configured to close the idle connection after several minutes. That will raise following exception in the DatabaseTable connector "DB error: org.postgresql.util.PSQLException(An I/O error occurred while sending to the backend.)->java.net.SocketException(Connection reset)". Solution for this problem is configure jdbc driver to send keepalive packets by adding "keepalives=1" and "keepalives\_idle=XXX" (XXX is number of seconds after which keepalive packet will be sent) to jdbcUrlTemplate. For example:

```
<icfsdbt:jdbcUrlTemplate>jdbc:postgresql://localhost:5432/databaseName?keepalives=1&keepalives_idle=120<
/icfsdbt:jdbcUrlTemplate>
```

