



Simba JDBC Bridge



Installation Guide

Version 1.0

April 2023

Copyright

This document was released in April 2023.

Copyright ©2014-2023 Magnitude Software, Inc., an insightsoftware company. All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without prior written permission from Magnitude, Inc.

The information in this document is subject to change without notice. Magnitude, Inc. strives to keep this information accurate but does not warrant that this document is error-free.

Any Magnitude product described herein is licensed exclusively subject to the conditions set forth in your Magnitude license agreement.

Simba, the Simba logo, SimbaEngine, and Simba Technologies are registered trademarks of Simba Technologies Inc. in Canada, the United States and/or other countries. All other trademarks and/or servicemarks are the property of their respective owners.

All other company and product names mentioned herein are used for identification purposes only and may be trademarks or registered trademarks of their respective owners.

Information about the third-party products is contained in a third-party-licenses.txt file that is packaged with the software.

Contact Us

Magnitude Software, Inc.

www.magnitude.com

About This Guide

Purpose

The *Simba JDBC Bridge Installation Guide* explains how to use Simba JDBC Bridge Installer and corresponding command lines to setup server as service and support JDBC access to a local ODBC driver. Simba JDBC Bridge is configured to use localhost as the listening address. It is essential that the client use localhost and the configured port to connect to the bridge.

Audience

The guide is intended for end users of the Simba JDBC Bridge.

Knowledge Prerequisites

To use the Simba JDBC Bridge, the following knowledge is helpful:

- Familiarity with the platform on which you are using the Simba JDBC Bridge
- Ability to use the data store to which the Simba JDBC Bridge is connecting
- An understanding of the role of JDBC technologies in connecting to a data store
- Experience creating and configuring JDBC connections
- Exposure to SQL

Document Conventions

Italics are used when referring to book and document titles.

Bold is used in procedures for graphical user interface elements that a user clicks and text that a user types.

Monospace font indicates commands, source code or contents of text files.

Note:

A text box with a pencil icon indicates a short note appended to a paragraph.

Important:

A text box with an exclamation mark indicates an important comment related to the preceding paragraph.

Contents

Platform Support	5
Installer Setup	6
Bridge as a Service Setup	7
Service Setup Parameters	9
LicenseFile	9
ListenPort	9
LogLevel	10
LogPath	10
ServerDSN	11
Other Service Setup Parameters	12
BinPath	12
DisplayName	13
Configuring SimbaClient for JDBC	14
Connection URL	14
Linking to the connector class	14
SimbaClient for JDBC Configuration Properties	14
General configuration properties	15
Third-Party Trademarks	19

Platform Support

Each machine where you use the Simba JDBC Bridge must have Java Runtime Environment (JRE) 11.0 installed.

The Simba JDBC Bridge supports Windows 64-bit and ODBC 64-bit drivers.

The connector complies with the JDBC 4.3 data standard.

Installer Setup

Start by downloading and extracting the package onto your computer. Then, run the installer for Simba JDBC Bridge.

To install the product using the setup wizard on Windows:

1. To start the setup wizard, double-click the `SimbaJDBCBridge.msi` file.
2. In the setup wizard, click **Next**.
3. Select the check box to accept the terms of the license agreement, and then click **Next**.
4. Choose the location on your computer where you want Simba JDBC Bridge to be installed:
 - To accept the default installation directory, click **Next**. By default, the installation location is `C:\Program Files\insightSoftware`
 - Or, to install the application to a different location, click **Change**, then type or browse to the location you want, and then click **OK**. To accept this installation directory, click **Next**.
5. Click **Install**. If you are prompted to allow the program to make changes to your computer, click **Yes**.

When the installation is completed, click **Finish**.

You can now use Simba JDBC Bridge to create a service.

Bridge as a Service Setup

To create a bridge as a service in Windows, use the `sc` command.

The `sc` command is a command-line utility that allows users to manage Windows services, including creating, modifying, and deleting them.

For example, this is a basic syntax:

```
sc create [service name] [binPath= ] [DisplayName= ] [start= ]  
[depend= ] [obj= ] [password= ]
```

Parameters within square brackets are arguments for `sc create`. You can also configure Simba JDBC Bridge parameters, for more information, see [Service Setup Parameters](#) on page 9.

For example, a command using Simba JDBC Bridge parameters:

```
sc create SimbaBridge binpath=  
"C:\server\lib\SimbaBridge64.exe -LogPath C:\server\lib\log -  
LicenseFile C:\Users\Administrator\bridge.lic -ServerDSN  
SQLServer"
```

This command creates the `SimbaBridge` service with designated 64-bit binary path, Server DSN, log path, and license file path. If you would like the service to automatically start when Windows starts, use `start=auto`. For explanations of the settings of each parameter, see [Service Setup Parameters](#) on page 9.

To start the service use the following command:

```
sc start SimbaBridge
```

Now the service is running and the JDBC Client can be used to connect the configured ODBC driver.

To stop the service use the following command:

```
sc stop SimbaBridge
```

Note:

The service should be stopped before upgrading Simba JDBC Bridge.

To delete the service use the following command:

```
sc delete SimbaBridge
```

i Note:

The service should be deleted before uninstalling.

Service Setup Parameters

The `sc create` command allows user to attach additional parameters to setup the service and Simba JDBC Bridge has its own specific parameters to configure the service.

Note:

Property values are case-sensitive.

LicenseFile

Default Value	Required	Allowed Value
Unspecified	No	Valid file path or unspecified.

Description

Specifies the directory where the license file is located.

Example:

```
-LicenseFile "C:\Users\Administrator\bridge.lic"
```

Defaults to checking `bridge.lic` in the current working directory of server. If configuring the license path fails or an expired license is used, the server starts but connections are rejected with a license error.

ListenPort

Default Value	Required	Allowed Value
1543	No	1 to 65535 (TCP/IP port number range).

Description

The local port for SimbaServer to bind to.

Example:

```
-ListenPort 1583
```

ServerDSN is required for a successful connection. Simba JDBC Bridge does not start without proper DSN setup.

LogLevel

Default Value	Required	Allowed Value
LOG_OFF	No	See below.

Description

Use this parameter to control the granularity of the messages and events that are logged.

Example:

```
-LogLevel LOG_ERROR
```

Set to one of the following numbers:

- 0 or LOG_OFF: Disable all logging.
- 1 or LOG_FATAL: Enable logging on the FATAL level, which logs very severe error events that will lead the connector to abort.
- 2 or LOG_ERROR: Enable logging on the ERROR level, which logs error events that might still allow the connector to continue running.
- 3 or LOG_WARNING: Enable logging on the WARNING level, which logs events that might result in an error if action is not taken.
- 4 or LOG_INFO: Enable logging on method entry and exit points, and parameter values for debugging.
- 5 or LOG_DEBUG: Enable logging on the DEBUG level, which logs detailed information that is useful for debugging the connector.
- 6 or LOG_TRACE: Enable logging on all method entry points.

LogPath

Default Value	Required	Allowed Value
Unspecified	No	Valid directory path or unspecified.

Description

The full path to the folder where the connector saves log files when logging is enabled.

Example:

```
-LogPath "C:\Simba Technologies\Temp"
```

If this value is not set, the log files are written to the current working directory. When it's running as a service, the current working directory of the SimbaBridge process is:

```
C:\Windows\system32.
```

ServerDSN

Default Value	Required	Allowed Value
None	Yes	Valid DSN string.

Description

The ODBC DSN used to connect to the bridged ODBC driver.

Example:

```
-ServerDSN SQLServer
```

ServerDSN is required for a successful connection. Simba JDBC Bridge does not start without proper DSN setup.

Other Service Setup Parameters

The `sc create` command has its own arguments that are not Simba JDBC Bridge specific to configure the service. There are many arguments and some common ones are listed here.

Note:

Property values are case-sensitive.

BinPath

Default Value	Required	Allowed Value
None	Yes	Valid path to SimbaBridge binary and command line options.

Description

The path to the service binary file.

Example:

```
binpath= "C:\server\lib\SimbaBridge64.exe -ServerDSN  
SQLServer"
```

There is no default value for this argument and this string must be supplied. The bin path should be pointing to where Simba JDBC Bridge is. The double quotes are needed if the parameter contains whitespace and the parameters for the bridge go in here.

Note:

Multiple service instances can be created by adding unique `-ListenPort [portnum]` values to the `binPath` and unique Service Name values.

DisplayName

Default Value	Required	Allowed Value
Unspecified	No	Valid informative string.

Description

The descriptive user-friendly name for the service that is displayed in the Services control panel.

Example:

```
displayname= "JDBCBridge"
```

Multiple data sources can be configured by setting up more than one service with different names, listen ports, and `ServerDSNs`.

Configuring SimbaClient for JDBC

SimbaClient for JDBC configuration properties control logging, server discovery, and security. Configuration properties are either added to the connection URL or implemented programmatically in the JDBC application.

Connection URL

The following is the connection URL format for the JDBC client:

```
jdbc:simba://localhost:[port];[property]=[value];[property]=[value]
```

For example:

```
jdbc:simba://localhost:1543;UID=BartonL;PWD=sneaky;LogLevel=0;LogPath=C:\Simba Technologies\Temp
```

Linking to the connector class

You must link the JDBC application to the correct connector class:

For JDBC 4.3 connectors:

```
com.simba.client.core.jdbc43.SCJDBC43DataSource
```

SimbaClient for JDBC Configuration Properties

All configuration properties are configured either programmatically or on the connection string. The following table summarizes the configuration properties.

Note:

For properties that list a maximum value of `UINT_MAX`, this equals a value of $2^{32}-1$.

Property	Description
Timeout Properties:	
<code>LoginTimeout</code>	The time to wait for a response during login.
<code>ConnectionTimeout</code>	The time to wait for a response from the server.

Property	Description
Logging configuration properties:	
LogLevel	Controls the granularity of the messages and events that are logged.
LogPath	Specifies the directory where the log files are created.
Other:	
D2OJDBCFunctionCatalogMode	Controls the behavior of DatabaseMetadata getProcedures or getProcedureColumns and getFunctions or getFunctionColumns for JDBC clients.

General configuration properties

Along with the specific properties that apply to the client itself, any other properties are passed along to the ODBC driver, in the connection string used to connect to it.

The following are general configuration properties.

Note:

Property values are case-sensitive.

LoginTimeout

Default Value	Required	Allowed Value
60	No	0 to UINT_MAX seconds.

Description

The time, in seconds, to wait for a response from the server after a login request is made by the client.

Example:

```
LoginTimeout=10
```

A value of 0 means no timeout. The value of this property is used to set the value of `SQL_ATTR_LOGIN_TIMEOUT`. A log in timeout may occur earlier than the time specified by this value. For example, if a DNS lookup failure occurs, the log in attempt times out immediately.

ConnectionTimeout

Default Value	Required	Allowed Value
0	No	0 to <code>UINT_MAX</code> seconds.

Description

The time, in seconds, to wait for a response from the server after a login request is made by the client.

Example:

```
ConnectionTimeout=10
```

A value of 0 means no timeout.

LogLevel

Default Value	Required	Allowed Value
OFF	No	See below.

Description

Use this parameter to control the granularity of the messages and events that are logged.

Example:

```
LogLevel=ERROR
```

Note:

Log files are not created if this value is set to `OFF`.

Set to one of the following values:

- **OFF**: Disable all logging.
- **FATAL**: Enable logging on the FATAL level, which logs very severe error events that will lead the connector to abort.
- **ERROR**: Enable logging on the ERROR level, which logs error events that might still allow the connector to continue running.
- **WARNING**: Enable logging on the WARNING level, which logs events that might result in an error if action is not taken.
- **INFO**: Enable logging on the INFO level, which logs general information that describes the progress of the connector.
- **DEBUG**: Enable logging on method entry and exit points, and parameter values for debugging.
- **TRACE**: Enable logging on all method entry points.

LogPath

Default Value	Required	Allowed Value
Unspecified	No	Valid directory path or unspecified.

Description

The full path to the folder where the connector saves log files when logging is enabled.

Example:

```
LogPath=C:\Simba Technologies\Temp
```

If this value is not set, the log files are written to the current directory of the JVM using the client.

D2OJDBCFunctionCatalogMode

Default Value	Required	Allowed Value
NotSupported	No	See below.

Description

Controls the behaviour of `DatabaseMetadata.getProcedures` or `getProcedureColumns` and `getFunctions` or `getFunctionColumns` for JDBC clients.

Example:

`D2OJDBCFunctionCatalogMode=EmulateFunctionCatalogs`

Set to one of the following numbers:

- `NotSupported`: The default value, if not specified. Methods `getFunctions` or `getFunctionColumns` returns an error.
- `NoFunctions`: `getFunctions` or `getFunctionColumns` returns an empty result and no error.
- `EmulateFunctionCatalogs`: `SimbaBridge` emulates `getFunctions` or `getFunctionColumns` by splitting the results of the underlying ODBC driver's `SQLProcedures` or `SQLProcedureColumns` between `getProcedures` or `getProcedureColumns` and `getFunctions` or `getFunctionColumns`. Information for procedures for which `SQLProcedures` returns `SQL_PT_FUNCTION` in the `PROCEDURE_TYPE` column is returned by `getFunctions` or `getFunctionColumns`, and information for all other procedures is returned by `getProcedures` or `getProcedureColumns`.

Third-Party Trademarks

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Microsoft and Windows Server are trademarks or registered trademarks of Microsoft Corporation or its subsidiaries in Canada, United States and/or other countries.

All other trademarks are trademarks of their respective owners.