



Simba OEM ODBC Drivers

Installation Guide

June 2026

Copyright

This document was released in June 2026.

Copyright ©2014-2026 insightsoftware. 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 insightsoftware.

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

Any insightsoftware product described herein is licensed exclusively subject to the conditions set forth in your insightsoftware 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

www.insightsoftware.com

About This Guide

Purpose

The *Simba OEM ODBC Drivers Installation Guide* explains how to install OEM editions of the Simba ODBC drivers using ZIP and tarball packages.

Audience

The guide is intended for developers integrating the OEM ODBC drivers into their products.

Knowledge Prerequisites

To install the drivers, the following knowledge is helpful:


- Familiarity with the platform on which you are using the driver
- An understanding of the role of ODBC technologies and driver managers in connecting to a data source
- Experience creating and configuring ODBC connections


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.

This guide uses the following placeholders in file and folder names:

- *[Bitness]* is the number indicating the architecture that the driver supports (32-bit or 64-bit). For example, 32.
- *[DataSource]* is the data source that the driver connects to, in title case. For example, Hive.
- *[datasource]* is the data source that the driver connects to, in lower case. For example, hive.

- *[DATASOURCE]* is the data source that the driver connects to, in upper case. For example, HIVE.
- *[Vendor]* is the name of the company that is distributing the driver, in title case. For example, Simba.
- *[vendor]* is the name of the company that is distributing the driver, in lower case. For example, simba.
- *[VENDOR]* is the name of the company that is distributing the driver, in upper case. For example, SIMBA.
- *[Version]* is the version number of the driver. For example, 2.1.3.1001.

For example, if each placeholder is defined as above, then the actual name of a file referred to as *[Vendor] [DataSource] ODBC_ [Version] _Windows .zip* would be SimbaHiveODBC_2.1.3.1001_Windows.zip. Similarly, a file referred to as *simba . [datasource] odbc .ini* would be simba.hiveodbc.ini.

Contents

Copyright	2
About This Guide	3
Purpose	3
Audience	3
Knowledge Prerequisites	3
Document Conventions	3
Introduction	6
Installing an OEM Driver	7
Installing from a Windows ZIP Package	7
Installing from a macOS Tarball Package	9
Installing from a Linux Tarball Package	10
Windows Package	13
Registry Entries	14
Example of a .reg File	15
macOS Package	17
Linux Package	19
Configuration Files for macOS and Linux	21
Example of an odbc.ini File	22
Example of an odbcinst.ini File	24
Example of a Driver-Wide Configuration File	25
Third-Party Trademarks	26

Introduction

Simba offers ODBC drivers with various packaging and branding options at different price points.

The packaging options include the following:

- ZIP packages
- Tarball packages (`.tar.gz` files)
- Installers (`.msi`, `.rpm`, and `.dmg` files)

Each packaging option can be Simba-branded, or custom-branded with the name of another vendor.

This *Simba OEM ODBC Drivers Installation Guide* explains how to install ZIP and tarball packages. For detailed information about system requirements, how to use the installers, or how to configure and use the driver after either installation process, see the *Installation and Configuration Guide* for the driver.

Installing an OEM Driver

Each Simba OEM ODBC driver is delivered in a ZIP or tarball file. Depending on the platform that the driver is intended for, the name of the file is as follows:

- **Windows:**
 - For most drivers: `[Vendor][DataSource]ODBC_[Version]_Windows.zip`
 - For some drivers that connect to Google data sources: `[Vendor]ODBCDriverfor[DataSource]_[Version]_Windows.zip`
- **macOS:**
 - For most drivers: `[Vendor][DataSource]ODBC-[Version]-OSX.tar.gz`
 - For some drivers that connect to Google data sources: `[Vendor]ODBCDriverfor[DataSource]-[Version]-OSX.tar.gz`
- **Linux:**
 - For most drivers: `[Vendor][DataSource]ODBC-[Version]-Linux.tar.gz`
 - For some drivers that connect to Google data sources: `[Vendor]ODBCDriverfor[DataSource]-[Version]-Linux.tar.gz`

Each file contains setup files, documentation, and additional ZIP or tarball files that contain the driver files. The Windows and Linux packages have two inner ZIP/tarball files, each containing either the 32-bit or 64-bit driver files. The macOS package has one inner tarball file only, because the macOS driver supports both 32- and 64-bit applications. For more information about the contents of each package, see the following sections:

- [Windows Package](#)
- [macOS Package](#)
- [Linux Package](#)

For installation instructions, see the following sections:

- [Installing from a Windows ZIP Package](#)
- [Installing from a macOS Tarball Package](#)
- [Installing from a Linux Tarball Package](#)

Installing from a Windows ZIP Package

To install the driver, first create the installation directory for the driver by extracting the driver files from the ZIP archive and copying them to the appropriate location. Then, configure the Windows Registry to recognize the driver and point to the driver files by updating and running the appropriate `.reg` file from the ZIP archive.

To install an OEM driver from a Windows ZIP package:

1. If it does not already exist, create the directory where you want to install the driver.



Important:

If you plan to install both the 32-bit and 64-bit versions of the driver on the same machine, make sure to use a different installation directory for each installation to prevent file name conflicts.

2. Extract the main ZIP file to a convenient temporary location.
3. Extract the inner ZIP file corresponding to the version of the driver that you want to install (32-bit or 64-bit) into a convenient temporary location.
4. Navigate to the subfolder that you extracted from the inner ZIP file in the previous step, and then copy all the files and folders to the installation directory.
5. Navigate to the temporary location where you extracted the main ZIP file in step 2, and then copy the `[DataSource]ODBC.did` file to the `\lib` subfolder in the installation directory.
6. Navigate to the `\setup` subfolder that was extracted from the main ZIP file in step 2, and identify the appropriate `.reg` file for your installation, given the bitness of the driver being installed and the bitness of your Windows machine.

For example, if you are installing a 32-bit driver on a 64-bit machine, then `Setup-32bitDriverOn64Windows.reg` is the file that you need to modify in the next step.

7. Open the `.reg` file in a text editor, and verify the name of the `.dll` file that the driver uses. Look for a line looks similar to the following, where `[FileName]` is the actual name of the `.dll` file:

```
"Driver"=<INSTALLDIR>\lib\[FileName].dll
```

If the name of the file is `RDFODBC_sb[Bitness].dll`, then you must rename the `[DataSource]ODBC.did` from step 5 to `RDFODBC.did`. Otherwise, do not change the name of the `[DataSource]ODBC.did` file.

Additionally, for drivers that support the use of `.ini` files in Windows, you may need to configure driver-wide settings in a different `.ini` file depending on the name of the `.dll` file that the driver uses. If your driver uses the `RDFODBC_sb[Bitness].dll` file, then driver-wide settings are read from `[vendor].rdfodbc.ini` instead of `[datasource]odbc.ini`.

8. In your text editor, make the following changes to the `.reg` file:
 - Update the branding information. The vendor name and data source name mentioned in the `.reg` file must match the names shown in the ZIP file name.

For example, if you are installing a driver from a `MagnitudeHiveODBC_2.1.3.1001_Windows.zip` package, then make sure that "Magnitude" is used as the vendor name instead of "Simba", and that "Hive" is the only data source mentioned in the `.reg` file.

- Replace each instance of `<INSTALLDIR>` with the actual file path to the installation directory.



Important:

- You must specify the path to the installation directory regardless of whether you installed the driver to the default Simba location or a custom location.
- Be aware that any backslashes (\) in the directory path must be escaped with a second backslash. For example, the directory `C:\Program Files\Magnitude` would be specified as `C:\\Program Files\\Magnitude` in the `.reg` file.
- In the `HKEY_LOCAL_MACHINE\SOFTWARE\[Vendor]\[DriverName]\Driver` section, set the `SwapFilePath` property to a writable directory where temporary files can be stored.

For an example of an updated `.reg` file, see [Example of a .reg File](#).

9. To run the `.reg` file and create the necessary registry keys, double-click the file.
10. If you are installing a driver that includes the Schema Editor application, and you modified the name of the driver `.dll` file, make sure that the new `.dll` file name includes the original `.dll` name.

For example, if the original `.dll` is named `MongoDBODBC_sb64.dll`, then you can rename the file to `YourCompanyMongoDBODBC_sb64.dll` but not `YourCompanyMongoDBODBC64.dll`.

You can now configure a connection and use the driver to access your data. For more information, see the *Installation and Configuration Guide* for your driver.

Installing from a macOS Tarball Package

To install the driver, first create the installation directory for the driver by extracting the driver files from the tarball file and copying them to the appropriate location. Then, update the configuration file for driver settings so that your driver manager can recognize and work with the driver.

To install an OEM driver from a macOS tarball package:

1. If it does not already exist, create the directory where you want to install the driver.
2. Extract the main tarball file to a convenient temporary location.
3. Extract the inner tarball file into a convenient temporary location.
4. Navigate to the subfolder that you extracted from the inner tarball file in the previous step, and then copy all the files and folders to the installation directory.
5. Navigate to the temporary location where you extracted the main tarball file in step 2, and then copy the `[DataSource]ODBC.did` file to the `/lib` subfolder in the installation directory.
6. To verify the name of the `.dylib` file that the driver uses, which determines the `.did` file name that the driver requires and the `.ini` file that the driver uses for driver-wide configurations, do the following:
 - a. Navigate to the `/setup` subfolder that was extracted from the main tarball file in step 2, and then open the `odbc.ini` file in a text editor.
 - b. In the `odbc.ini` file, look for a line that looks similar to the following, where `[FileName]` is the actual name of the `.dylib` file:

```
Driver=[INSTALLDIR]/lib/[FileName].dylib
```

If the name of the file is `librdfsodbc_sbu.dylib`, then you must rename the `[DataSource]ODBC.did` from step 5 to `RDFSODBC.did`. Otherwise, do not change the name of the `[DataSource]ODBC.did` file.

Additionally, if the name of the file is `librdfsodbc_sbu.dylib`, this indicates that your driver uses the `[vendor].rdfsodbc.ini` file in the `/lib` subfolder for driver-wide configurations. In this case, skip step 7 below. Otherwise, your driver uses the `[vendor].[datasource]odbc.ini` file for driver-wide configurations.

- c. After verifying the name of the `.dylib` file, close the `odbc.ini` file.
7. If your driver uses the `[vendor].[datasource]odbc.ini` file, do the following:
 - a. In the `/setup` subfolder, find this driver-wide configuration file. If the `[vendor]` part of the name is "simba", but you are installing a driver from a custom-branded package, then rename the file to use your vendor name.
 - b. Copy the `[vendor].[datasource]odbc.ini` file to the `/lib` subfolder in the installation directory.
8. Open the driver-wide configuration file in a text editor, and then do the following:
 - Update the branding information. The vendor name and data source name mentioned in this `.ini` file must match the names shown in the tarball file name.

For example, if you are installing a driver from a `MagnitudeHiveODBC-2.1.3.1001-OSX.tar.gz` package, then make sure that "Magnitude" is used as the vendor name instead of "Simba", and that "Hive" is the only data source mentioned in the `.ini` file.

- Replace each instance of `<INSTALLDIR>` with the actual file path to the installation directory.

**Important:**

You must specify the path to the installation directory regardless of whether you installed the driver to the default Simba location or a custom location.

- Set the `SwapFilePath` property to a writable directory where temporary files can be stored.

For an example of an updated driver-wide configuration file, see [Example of a Driver-Wide Configuration File](#).

You can now configure a connection and use the driver to access your data.

In most cases, you can refer to your driver's *Installation and Configuration Guide* for detailed information about using these files to configure the driver and set up connections. However, some drivers are not yet generally available for macOS and Linux, so the guides for those drivers may not provide complete information about using the driver on those platforms. If necessary, see [Configuration Files for macOS and Linux](#) for more information.

Installing from a Linux Tarball Package

To install the driver, first create the installation directory for the driver by extracting the driver files from the tarball file and copying them to the appropriate location. Then, update the configuration file for driver settings so that your driver manager can recognize and work with the driver.

To install an OEM driver from a Linux tarball package:

1. If it does not already exist, create the directory where you want to install the driver.



Important:

If you plan to install both the 32-bit and 64-bit versions of the driver on the same machine, make sure to use a different installation directory for each installation to prevent file name conflicts.

2. Extract the main tarball file to a convenient temporary location.
3. Extract the inner tarball file corresponding to the version of the driver that you want to install (32-bit or 64-bit) into a convenient temporary location.
4. Navigate to the subfolder that you extracted from the inner tarball file in the previous step, and then copy all the files and folders to the installation directory.
5. Navigate to the temporary location where you extracted the main tarball file in step 2, and then copy the `[DataSource]ODBC.did` file to the `/lib` subfolder in the installation directory.
6. To verify the name of the `.so` file that the driver uses, which determines the `.did` file name that the driver requires and the `.ini` file that the driver uses for driver-wide configurations, do the following:
 - a. Navigate to the `/setup` subfolder that was extracted from the main tarball file in step 2, and then open the `odbc.ini` file in a text editor.
 - b. In the `odbc.ini` file, look for a line that looks similar to the following, where `[FileName]` is the actual name of the `.so` file:

```
Driver=[INSTALLDIR]/lib/[FileName].so
```

If the name of the file is `librdfodbc_sb[Bitness].so`, then you must rename the `[DataSource]ODBC.did` from step 5 to `RDFODBC.did`. Otherwise, do not change the name of the `[DataSource]ODBC.did` file.

Additionally, if the name of the file is `librdfodbc_sb[Bitness].so`, this indicates that your driver uses the `[vendor].rdfodbc.ini` file in the `/lib` subfolder for driver-wide configurations. In this case, skip step 7 below. Otherwise, your driver uses the `[vendor].[datasource]odbc.ini` file for driver-wide configurations.

- c. After verifying the name of the `.so` file, close the `odbc.ini` file.
7. If your driver uses the `[vendor].[datasource]odbc.ini` file, do the following:
 - a. In the `/setup` subfolder, find this driver-wide configuration file. If the `[vendor]` part of the name is "simba", but you are installing a driver from a custom-branded package, then rename the file to use your vendor name.
 - b. Copy the `[vendor].[datasource]odbc.ini` file to the `/lib` subfolder in the installation directory.
8. Open the driver-wide configuration file in a text editor, and then do the following:
 - Update the branding information. The vendor name and data source name mentioned in this `.ini` file must match the names shown in the tarball file name.

For example, if you are installing a driver from a `MagnitudeHiveODBC-2.1.3.1001-OSX.tar.gz` package, then make sure that "Magnitude" is used as the vendor name instead of "Simba", and that "Hive" is the only data source mentioned in the `.ini` file.

- Replace each instance of <INSTALLDIR> with the actual file path to the installation directory.

**Important:**

You must specify the path to the installation directory regardless of whether you installed the driver to the default Simba location or a custom location.

- Set the `SwapFilePath` property to a writable directory where temporary files can be stored.

For an example of an updated driver-wide configuration file, see [Example of a Driver-Wide Configuration File](#).

You can now configure a connection and use the driver to access your data.

In most cases, you can refer to your driver's *Installation and Configuration Guide* for detailed information about using these files to configure the driver and set up connections. However, some drivers are not yet generally available for macOS and Linux, so the guides for those drivers may not provide complete information about using the driver on those platforms. If necessary, see [Configuration Files for macOS and Linux](#) for more information.

Windows Package

Each OEM ODBC driver for Windows is distributed in a ZIP file with one of the following names:

- For most drivers: `[Vendor][DataSource]ODBC_[Version]_Windows.zip`
- For some drivers that connect to Google data sources: `[Vendor]ODBCDriverfor[DataSource]_[Version]_Windows.zip`

The ZIP file is structured as follows:

- `[Vendor][DataSource]ODBC_[Version]_Windows` or `[Vendor]ODBCDriverfor[DataSource]_[Version]_Windows`
 - `\docs`
 - `\setup`
 - `[DataSource]ODBC.did`
 - `Simba[DataSource]ODBC32_[Version].zip` or `SimbaODBCDriverfor[DataSource]32_[Version].zip`
 - `\Simba[DataSource]ODBC32_[Version]` or `\SimbaODBCDriverfor[DataSource]32_[Version]`
 - `\ErrorMessages`
 - `\lib`
 - `third-party-licenses.txt`
 - `\[DriverSpecificFolders]`
 - `Simba[DataSource]ODBC64_[Version].zip` or `SimbaODBCDriverfor[DataSource]64_[Version].zip`
 - `\Simba[DataSource]ODBC64_[Version]` or `\SimbaODBCDriverfor[DataSource]64_[Version]`
 - `\ErrorMessages`
 - `\lib`
 - `third-party-licenses.txt`
 - `\[DriverSpecificFolders]`

The major components of the Windows package are described below:

File or Folder Name	Description
<code>\docs</code>	This folder contains Simba-branded driver documentation. Refer to this documentation for information such as the release history of the driver, instructions for configuring a connection, and details about each supported connection property.

File or Folder Name	Description
\setup	This folder contains Simba-branded Windows registry files. These files contain registry entries that are required for the driver. You must update the branding and installation directory paths in the appropriate file, and then run it to create the necessary registry keys. For more information, see Registry Entries and Example of a .reg File .
[DataSource]ODBC.did	This file is a vendor-branded .did file that determines which vendor name and driver name the driver uses. This file must be copied into the \lib folder.
Simba[DataSource]ODBC32_ [Version].zip or SimbaODBCDriverfor [DataSource]32_[Version].zip and Simba[DataSource]ODBC64_ [Version].zip or SimbaODBCDriverfor [DataSource]64_[Version].zip	These inner ZIP files contain driver-specific components, including the driver binaries, error messages, and third-party software licenses. You must extract the ZIP file corresponding to the bitness of the driver that you want to install (32-bit or 64-bit).
\ErrorMessages	This folder contains all the error messages that the driver may produce. You must copy this folder to the installation directory.
\lib	This folder contains all the necessary driver binary files. You must copy this folder to the installation directory.
third-party-licenses.txt	This file contains the licenses for all the third-party software that is used by the driver. You must copy this file to the installation directory.
\[DriverSpecificFolder]	This placeholder in the ZIP file structure shown above refers to additional components that may be required by certain drivers. If your ZIP package includes these components, you must copy them to the installation directory.

Registry Entries

The \setup folder in the OEM package for Windows contains three Windows registry files, one for each possible combination of bitnesses between the driver and the machine. These files contain all the necessary registry entries for registering the driver on a Windows machine:

- Setup-32bitDriverOn32Windows.reg
- Setup-32bitDriverOn64Windows.reg
- Setup-64bitDriverOn64Windows.reg

The Setup-32bitDriverOn32Windows.reg and Setup-64bitDriverOn64Windows.reg files are identical except for the driver .dll file name mentioned within, which varies depending on the bitness of the driver.

The Setup-32bitDriverOn64Windows.reg file is different from the other two files because the registry entries must be added under the HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node registry key instead of the standard HKEY_LOCAL_MACHINE\SOFTWARE key.

When you run one of the .reg files, entries are created in the following registry keys within HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node or HKEY_LOCAL_MACHINE\SOFTWARE:

Registry Key Name	Entries Created
ODBC\ODBC.INI	A sample DSN is installed under this registry key. These entries are not required if your application connects to data sources using DSN-less connection strings. However, having the sample DSN helps end users to quickly configure, test, and troubleshoot the driver.
ODBC\ODBCINST.INI	The ODBC driver is installed under this registry key. These entries are required to register the driver with the Windows ODBC Driver Manager.
[Vendor]\[Vendor] [DataSource] ODBC Driver	Configuration information needed by the driver is installed under this registry key. These entries are required for the proper operation of the driver.

Example of a .reg File

When installing an OEM driver on a Windows machine, you must update the branding information and install paths in the appropriate .reg file, and then run it to create registry keys.

The following is an example of an updated .reg file for installing the 32-bit Impala ODBC driver on a 32-bit Windows machine, given the company name "Magnitude" and the installation directory C:\Program Files\Magnitude\Magnitude Impala ODBC Driver:

```
REGEDIT4
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBC.INI]
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBC.INI\ODBC Data Sources]
```

```
"Magnitude Impala"="Magnitude Impala ODBC Driver"
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBC.INI\Magnitude Impala]
```

```

"Driver"="Magnitude Impala ODBC Driver"
"Description"="Magnitude Impala ODBC Driver DSN"
"Host"="your_impala_server"
"Port"="21050"
"RowsFetchedPerBlock"="10000"
"UseNativeQuery"="0"
[HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBCINST.INI]
[HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBCINST.INI\ODBC Drivers]
"Magnitude Impala ODBC Driver"="Installed"
[HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBCINST.INI\Magnitude Impala ODBC Driver]
"Description"="Magnitude Impala ODBC Driver 1.2"
"Driver"="C:\\Program Files\\Magnitude\\Magnitude Impala ODBC Driver\\lib\\ImpalaODBC_sb32.dll"
"Setup"="C:\\Program Files\\Magnitude\\Magnitude Impala ODBC Driver\\lib\\ImpalaODBC_sb32.dll"
[HKEY_LOCAL_MACHINE\SOFTWARE\Magnitude]
[HKEY_LOCAL_MACHINE\SOFTWARE\Magnitude\Magnitude Impala ODBC Driver]
[HKEY_LOCAL_MACHINE\SOFTWARE\Magnitude\Magnitude Impala ODBC Driver\Driver]
"DriverManagerEncoding"="UTF-16"
"ErrorMessagesPath"="C:\\Program Files\\Magnitude\\Magnitude Impala
ODBC Driver\\ErrorMessages"
"LogLevel"="0"
"LogNamespace"=""
"LogPath"=""
"SwapFilePath"="C:\\Temp"

```


macOS Package

Each ODBC driver for macOS is distributed in a tarball file with one of the following names:

- For most drivers: `[Vendor] [DataSource]ODBC-[Version]-OSX.tar.gz`
- For some drivers that connect to Google data sources: `[Vendor]ODBCDriverfor [DataSource]-[Version]-OSX.tar.gz`

The tarball file is structured as follows:

- `[Vendor] [DataSource]ODBC-[Version]-OSX` or `[Vendor]ODBCDriverfor [DataSource]-[Version]-OSX`
 - `/docs`
 - `/setup`
 - `[DataSource]ODBC.did`
 - `Simba [DataSource]ODBC-[Version].tar.gz` or `SimbaODBCDriverfor [DataSource]-[Version].tar.gz`
 - `/Simba [DataSource]ODBC-[Version]` or `/SimbaODBCDriverfor [DataSource]-[Version]`
 - `/ErrorMessage`s
 - `/lib`
 - `third-party-licenses.txt`
 - `/[DriverSpecificFolders]`

The major components of the macOS package are described below:

File or Folder Name	Description
<code>/docs</code>	This folder contains Simba-branded driver documentation. Refer to this documentation for information such as the release history of the driver, instructions for configuring a connection, and details about each supported connection property.
<code>/setup</code>	This folder contains sample configuration files. For more information, see Configuration Files for macOS and Linux .
<code>[DataSource]ODBC.did</code>	This file is a vendor-branded <code>.did</code> file that determines which vendor name and driver name the driver uses. This file must be copied into the <code>/lib</code> folder.

File or Folder Name	Description
<code>Simba[DataSource]ODBC-</code> <code>[Version].tar.gz</code> or <code>SimbaODBCDriverfor</code> <code>[DataSource]-</code> <code>[Version].tar.gz</code>	<p>This inner tarball file contains driver-specific components, including the driver binaries, error messages, and third-party software licenses. You must extract this tarball file.</p>
<code>/ErrorMessage</code> s	<p>This folder contains all the error messages that the driver may produce. You must copy this folder to the installation directory.</p>
<code>/lib</code>	<p>This folder contains all the necessary driver binary files.</p> <p>You must copy this folder to the installation directory.</p>
<code>third-party-licenses.txt</code>	<p>This file contains the licenses for all the third-party software that is used by the driver. You must copy this file to the installation directory.</p>
<code>/[DriverSpecificFolder]</code>	<p>This placeholder in the tarball file structure shown above refers to additional components that may be required by certain drivers. If your tarball package includes these components, you must copy them to the installation directory.</p>

Linux Package

Each ODBC driver for Linux is distributed in a tarball file with one of the following names:

- For most drivers: `[Vendor][DataSource]ODBC-[Version]-Linux.tar.gz`
- For some drivers that connect to Google data sources: `[Vendor]ODBCDriverfor[DataSource]-[Version]-Linux.tar.gz`

The tarball file is structured as follows:

- `[Vendor][DataSource]ODBC-[Version]-Linux` or `[Vendor]ODBCDriverfor[DataSource]-[Version]-Linux`
 - `/docs`
 - `/setup`
 - `[DataSource]ODBC.did`
 - `Simba[DataSource]ODBC32-[Version].tar.gz` or `SimbaODBCDriverfor[DataSource]32-[Version].tar.gz`
 - `/Simba[DataSource]ODBC32-[Version]` or `/SimbaODBCDriverfor[DataSource]32-[Version]`
 - `/ErrorMessages`
 - `/lib`
 - `/ThirdParty`
 - `third-party-licenses.txt`
 - `/[DriverSpecificFolders]`
 - `Simba[DataSource]ODBC64-[Version].tar.gz` or `SimbaODBCDriverfor[DataSource]64-[Version].tar.gz`
 - `/Simba[DataSource]ODBC64-[Version]` or `/SimbaODBCDriverfor[DataSource]64-[Version]`
 - `/ErrorMessages`
 - `/lib`
 - `/ThirdParty`
 - `third-party-licenses.txt`
 - `/[DriverSpecificFolders]`

The major components of the OEM package are described below:

File or Folder Name	Description
<code>/docs</code>	This folder contains Simba-branded driver documentation. Refer to this documentation for information such as the release history of the driver, instructions for configuring a connection, and details about each supported connection property.

File or Folder Name	Description
/setup	This folder contains sample configuration files. For more information, see Configuration Files for macOS and Linux .
[DataSource]ODBC.did	This file is a vendor-branded .did file that determines which vendor name and driver name the driver uses. This file must be copied into the /lib folder.
Simba [DataSource]ODBC32- [Version].tar.gz or SimbaODBCDriverfor [DataSource]32- [Version].tar.gz and Simba [DataSource]ODBC64- [Version].tar.gz or /SimbaODBCDriverfor [DataSource]32- [Version]	These inner tarball files contain driver-specific components, including the driver binaries, error messages, and third-party software licenses. You must extract the tarball file corresponding to the bitness of the driver that you want to install (32-bit or 64-bit).
/ErrorMessages	This folder contains all the error messages that the driver may produce. You must copy this folder to the installation directory.
/lib	This folder contains all the necessary driver binary files. You must copy this folder to the installation directory. The /ThirdParty subfolder inside this folder contains third-party libraries that are dynamically linked to the driver library.
third-party-licenses.txt	This file contains the licenses for all the third-party software that is used by the driver. You must copy this file to the installation directory.
/ [DriverSpecificFolder]	This placeholder in the tarball file structure shown above refers to additional components that may be required by certain drivers. If your tarball package includes these components, you must copy them to the installation directory.

Configuration Files for macOS and Linux

The `/setup` folders in the macOS and Linux packages contain the following sample configuration files:

- An `odbc.ini` file, which is used to create DSNs and specify connection properties.
- An `odbcinst.ini` file, which is used to define the driver in the ODBC driver manager.
- A configuration file that is used to specify driver-wide configuration settings, which apply to every connection that uses the driver:
 - If your driver uses the `librdfodbc_sbu.dylib` or `librdfodbc_sb[Bitness].so` library files (for macOS or Linux, respectively), then the driver-wide configuration file is named `[vendor].rdfodbc.ini`.
 - Otherwise, the driver-wide configuration file is named `[vendor].[datasource]odbc.ini`.

**Important:**

If necessary, rename the `[vendor].[datasource]odbc.ini` file to use the appropriate branding information. For example, the file for the Simba Hive ODBC Driver with SQL Connector should be `simba.hiveodbc.ini`.

For information about how to verify which library file and driver-wide configuration file your driver uses, see step 6 in [Installing from a macOS Tarball Package](#) and [Installing from a Linux Tarball Package](#).

You can refer to these files for examples of the necessary configuration settings, or copy them to the appropriate locations and then use them to configure the driver. ODBC driver managers check specific default locations for these files, so if you choose to store these files in a different location, you must set environment variables to point to the full path and name of each file. However, be aware that the `[vendor].rdfodbc.ini` file in particular must be stored in its default location.

File Name	Default Location	Environment Variable for Specifying Custom Locations
<code>odbc.ini</code>	The home directory.	ODBCINI
<code>odbcinst.ini</code>	The home directory.	ODBCINSTINI if you are using iODBC. ODBCSYSINI if you are using unixODBC.

File Name	Default Location	Environment Variable for Specifying Custom Locations
<code>[vendor].[datasource].odbc.ini</code>	The <code>/lib</code> subfolder in the driver installation directory.	<p>An environment variable using one of the following naming formats:</p> <ul style="list-style-type: none"> ▪ <code>[VENDOR]_[DATASOURCE]_ODBC_INI</code> ▪ <code>[VENDOR][DATASOURCE]ODBCINI</code> ▪ <code>[VENDOR][DATASOURCE]INI</code> <p>To confirm the naming format that your driver uses for this environment variable, see the <i>Installation and Configuration Guide</i> for the driver.</p> <p>For example, the MongoDB driver uses an environment variable with the first naming format, so if this driver is branded for a vendor named "Magnitude", the environment variable would be <code>MAGNITUDE_MONGODB_ODBC_INI</code>.</p>
<code>[vendor].rdfodbc.ini</code>	The <code>/lib</code> subfolder in the driver installation directory.	<p>N/A</p> <p>This file must remain in its default location.</p>

In most cases, you can refer to your driver's *Installation and Configuration Guide* for detailed information about using these files to configure the driver and set up connections. However, some drivers are not yet generally available for macOS and Linux, so the *Installation and Configuration Guide* for those drivers may not provide complete information about using the driver on those platforms. For examples that demonstrate how to use these files to configure the driver, see the following:

- [Example of an odbc.ini File](#)
- [Example of an odbcinst.ini File](#)
- [Example of a Driver-Wide Configuration File](#)

Example of an odbc.ini File

Use the `odbc.ini` file to define DSNs. A DSN is a data structure that stores connection information so that it can be used by the driver to connect to a data source.

The contents of this configuration file are divided into the following sections:

- A single [ODBC Data Sources] section, which lists the defined DSNs. Each line in this section consists of a DSN name, an equal (=) sign, and the name of the driver that uses the DSN.
- For each DSN, a section containing the connection information that is used when you connect with the DSN. The name of the section is the name of the DSN enclosed in brackets ([]). Each line in the section consists of a key name for a connection property, an equal (=) sign, and the value that you want to specify for this property.

**Note:**

For detailed information about all the connection properties that the driver supports, see "Driver Configuration Options" in the *Installation and Configuration Guide* for the driver.

The following is an example of an updated `odbc.ini` file that contains two DSNs on a macOS machine. One DSN connects to a Hive database and authenticates the connection using a user name and password, while the other DSN connects to a MongoDB database without authenticating the connection.

[ODBC Data Sources]

Hive Auth DSN=Magnitude Hive ODBC Driver

MongoDB No Auth DSN=Magnitude MongoDB ODBC Driver

[Hive Auth DSN]

Driver=/Library/magnitude/hive/lib/libhiveodbc_sbu.dylib

Host=192.168.222.160

Port=10000

UID=yourusername

PWD=yourpassword

[MongoDB No Auth DSN]

Driver=/Library/magnitude/mongodb/lib/libmongododbc_sbu.dylib

Server=192.168.222.160

Port=28018

Database=TestData

An equivalent `odbc.ini` file for a Linux machine would be nearly identical to the example shown above. To modify this example to be suitable for Linux, change the `Driver` property in each DSN section to specify the full path and name of the driver library file for Linux instead of the one for macOS.

Example of an odbcinst.ini File

You only need to configure the `odbcinst.ini` file if you plan to connect to your data source using a DSN-less connection string instead of a DSN. Use this file to define the driver name so that your ODBC driver manager can detect the correct driver when you specify the driver name in your connection string.



Note:

For detailed information about the accepted syntax for connection strings, see "Using a Connection String" in the *Installation and Configuration Guide* for the driver.

The contents of this configuration file are divided into the following sections:

- A single [ODBC Drivers] section, which lists the ODBC drivers that are installed on your machine. Each line in this section consists of a driver name, an equal (=) sign, and the value "Installed".
- For each driver, a section that defines the driver name to ODBC driver managers. The name of the section is the name of the driver enclosed in brackets ([]). Each line in the section consists of a key name (either `Description` or `Driver`), an equal (=) sign, and the value that you want to specify for this key. The `Description` setting is optional and can be set to any value, but the `Driver` setting is mandatory and must indicate the full path and name of the driver library file.

The following is an example of an updated `odbcinst.ini` file that defines the Magnitude Hive ODBC Driver and the Magnitude MongoDB ODBC Driver on a macOS machine:

[ODBC Drivers]

Magnitude Hive ODBC Driver=Installed

Magnitude MongoDB ODBC Driver=Installed

[Magnitude Hive ODBC Driver]

Description=The Magnitude-branded ODBC driver for connecting to Hive databases

Driver=/Library/magnitude/hive/lib/libhiveodbc_sbu.dylib

[Magnitude MongoDB ODBC Driver]

Description=The Magnitude-branded ODBC driver for connecting to MongoDB databases

Driver=/Library/magnitude/mongodb/lib/libmongododbc_sbu.dylib

When you write DSN-less connection strings, you can set the `Driver` connection property inside the string to `Magnitude Hive ODBC Driver` or `Magnitude MongoDB ODBC Driver`.

An equivalent `odbcinst.ini` file for a Linux machine would be nearly identical to the example shown above. To modify this example to be suitable for Linux, change the `Driver` property in each driver name section to specify the full path and name of the driver library file for Linux instead of the one for macOS.

Example of a Driver-Wide Configuration File

The driver-wide configuration file contains settings that apply to every connection that uses the driver. Depending on the specific driver that you are configuring, this file might be named `[vendor].[datasource]odbc.ini` or `[vendor].rdfodbc.ini`. For more information, see [Configuration Files for macOS and Linux](#).

When installing an OEM driver on a macOS or Linux machine, you must update the branding information, installation directory path, and `SwapFilePath` setting in the driver-wide configuration file.

The following is an example of an updated file for installing the Impala ODBC driver, given the company name "Magnitude" and the installation directory `opt/magnitude/impala`:

```
[Driver]
ErrorMessagesPath=opt/magnitude/impala/ErrorMessage
LogLevel=0
LogPath=
SwapFilePath=/tmp
```

You can also use the file to configure certain driver-wide features, such as the error logging feature. The following is an example of an updated file that also enables the error logging feature:

```
[Driver]
ErrorMessagesPath=opt/magnitude/impala/ErrorMessage
LogLevel=1
LogPath=/var/impalaodbc/logfiles
SwapFilePath=/tmp
```

Third-Party Trademarks

Linux is the registered trademark of Linus Torvalds in Canada, United States and/or other countries.

Mac, macOS, Mac OS, and OS X are trademarks or registered trademarks of Apple, Inc. or its subsidiaries in Canada, United States and/or other countries.

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

MongoDB and Mongo are trademarks or registered trademarks of MongoDB, Inc. or its subsidiaries in Canada, the United States and/or other countries.

Apache Hive, Apache Impala, Apache, Hive, and Impala are trademarks or registered trademarks of The Apache Software Foundation or its subsidiaries in Canada, United States and/or other countries.

All other trademarks are trademarks of their respective owners.