Progress

# Progress DataDirect Autonomous REST Connector for ODBC

## Quick Start for Windows

*Release 8.0.0*

Progress DataDirect

# Quick Start: Progress DataDirect Autonomous REST Connector for ODBC on Windows

The Progress DataDirect Autonomous REST Connector for ODBC is a driver that supports SQL read-only access to REST API data sources. This quick start guide provides the basic information necessary to install, configure, and connect to REST API data sources on Windows platforms. To take full advantage of the features and functionality available for your driver, refer to the Progress DataDirect Autonomous REST Connector for ODBC User's Guide and Reference.

**Note:** OEM CUSTOMERS: Refer to the *Progress DataDirect for ODBC Drivers Distribution Guide* for information on installing, branding, unlocking, and distributing your branded drivers.

This quick start covers the following topics:

## Before You Start

Before you get started, you need the following:

- Appropriate user permissions to modify your environment and to read, write, and execute various files in the DataDirect for ODBC installation directory.
- Connection information:
  - **Authentication Method**: The method your endpoints use to authenticate users. The Autonomous REST Connector supports basic, token based (HTTP Header and URL based), and no authentication.
  - **User Name** and **Password**: If required by your authentication method, the user ID and password used to access your data source.

  - **URL Parameter**: For URL-based authentication, the name of the URL parameter used to pass the security token.
  - **HTTP Header Name**: For basic and HTTP header authentication methods, the name of your HTTP header used for authentication, if it differs from the default value of `Authorization`.
  - **HTTP Endpoints**: A list of the HTTP endpoints that you want to map to the relational view. Note that additional endpoints can be added during later sessions.
  - **(Optional) Location to store internal driver files**: The path you want the driver to use to generate and store internal configuration files. This location is not used when the driver uses memory to store mapping information for the session (the default). When configured to generate internal files by setting the `CreateMap` option to either **1 - ForceNew** or **2 - NotExist**, the driver will attempt to use the following directory by default:

    *application_data_folder*\Local\Progress\DataDirect\AutoREST_Schema\

- For licensed installations, you will also need the following information that was provided by Progress DataDirect:
  - **Control Number** (IPE Key)
  - **Serial Number**

# Requirements and Support

**Driver/Client Software Requirements:**

- Java Virtual Machine (JVM) that is Java SE 8 or higher, including Oracle JDK, OpenJDK, and IBM SDK (Java) distributions.

**Supported Data Sources:**

The driver supports a variety of REST API deployments that employ:

- Payloads expressed in the JSON format.
- Basic, token (URL or header-based), or no authentication.

For the latest versions supported by the driver, refer to the supported configurations page on the Progress DataDirect Web site: https://www.progress.com/supported-configurations/datadirect.

# Downloading the Driver

**To download the ODBC Autonomous REST Connector:**

1. Visit the Progress DataDirect download site at:

   http://forms.progress.com/forms/driverdownload

2. Complete the questionnaire. To download the driver, provide the following values:
   - When prompted for your data source, select **Autonomous REST API**.

- When prompted for an interface, select **ODBC**.
- When prompted for your OS and architecture, select the platform and bitness used by your application.

3. Fill-in the registration form with your contact information.
4. Review the *End User License Agreement*. If you agree, check the corresponding box; then, click **Download**.

The installer program has been downloaded. See "Installing the ODBC Driver on Windows" for instructions on installing the driver.

### See also

Installing the ODBC driver on Windows on page 5

# Installing the ODBC driver on Windows

**Note:** OEM CUSTOMERS: Refer to the *Progress DataDirect for ODBC Drivers Distribution Guide* for information on installing, branding, unlocking, and distributing your branded drivers.

This section provides instructions for installing the driver using a GUI. For complete installation instructions, refer to the Progress DataDirect for ODBC Drivers Installation Guide.

**To install the drivers from downloaded files:**

1. Unzip the files, maintaining the directory structure in the zip file, to a temporary directory. For example: `c:\TEMP`.
2. From Windows Explorer, navigate to the temporary directory that contains the unzipped installer files; then, double-click the installer program:

   For 32-bit drivers, the installer program is named:

   `PROGRESS_DATADIRECT_ODBC_nn_WIN_32_INSTALL.exe`

   For 64-bit drivers, the installer program is named:

   `PROGRESS_DATADIRECT_ODBC_nn_WIN_64_INSTALL.exe`

   where:

   *nn*

   is the product version number. For example, this value is 8.0 for the 8.0.*x* version of the product.

3. The Introduction window for the product installation appears. Click **Next** to continue.
4. The License Agreement window appears. Make sure that you read and understand the license agreement. To continue with the installation, select the **I accept the terms of the License Agreement** option; then, click **Next**.

   **Note:** You can exit the installer program at any time by clicking **Cancel** or return to the previous window by clicking **Previous**.

5. The Install Directory window appears. In the **Where Would You Like to Install?** field, type the path, including the drive letter, of the product installation directory or click the **Choose...** button to browse to and select an installation directory.

   - If you are running the 32-bit installer on 64-bit Windows, the default value for the installation directory is `C:\Program Files (x86)\Progress\DataDirect\ODBC`.
   - For all other installations, the default value for the installation directory for is `C:\Program Files\Progress\DataDirect\ODBC`

   Verify that you have entered (or selected) the correct installation directory. Then, click **Next** to continue.

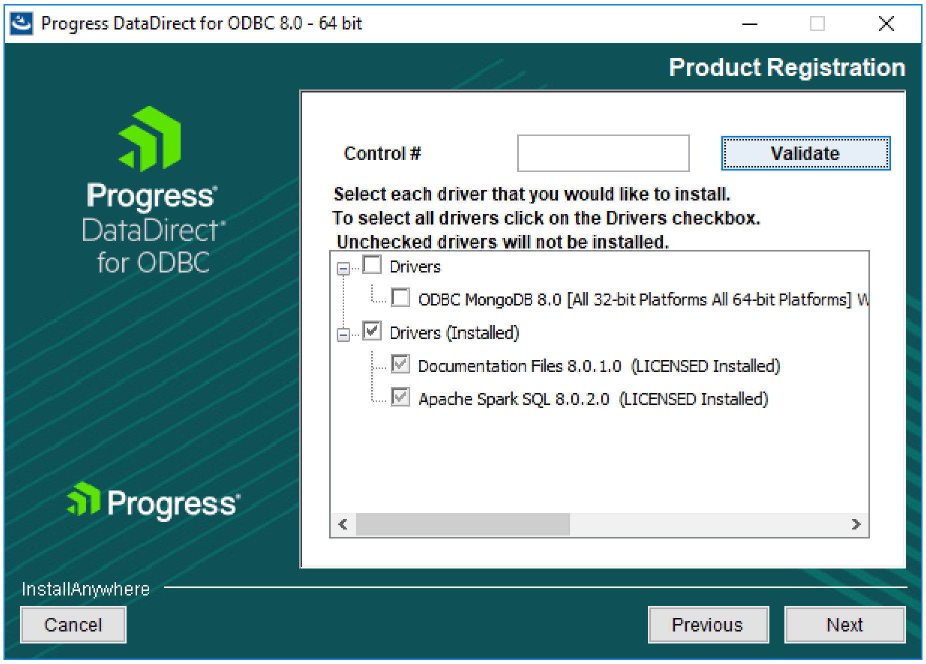
   To restore the installation directory to its default setting, click **Restore Default**.

6. Choose the type of installation to perform. Select one of the following options:

   - **Evaluation Installation (will expire in 15 days)**. Select this option to install an evaluation version of the driver. Click **Next** to continue with the installation. Skip to Step 10 on page 7.
   - **OEM/Licensed Installation**. Select this option if you purchased a licensed version of one or multiple drivers. Click **Next**. Proceed to the next step.

   **Note:** OEM CUSTOMERS: Refer to the *Progress DataDirect for ODBC Drivers Distribution Guide* for information on installing, branding, unlocking, and distributing your branded drivers.

7. The Product Registration window appears.



Progress DataDirect for ODBC 8.0 - 64 bit

Product Registration

Control #

Select each driver that you would like to install.
To select all drivers click on the Drivers checkbox.
Unchecked drivers will not be installed.

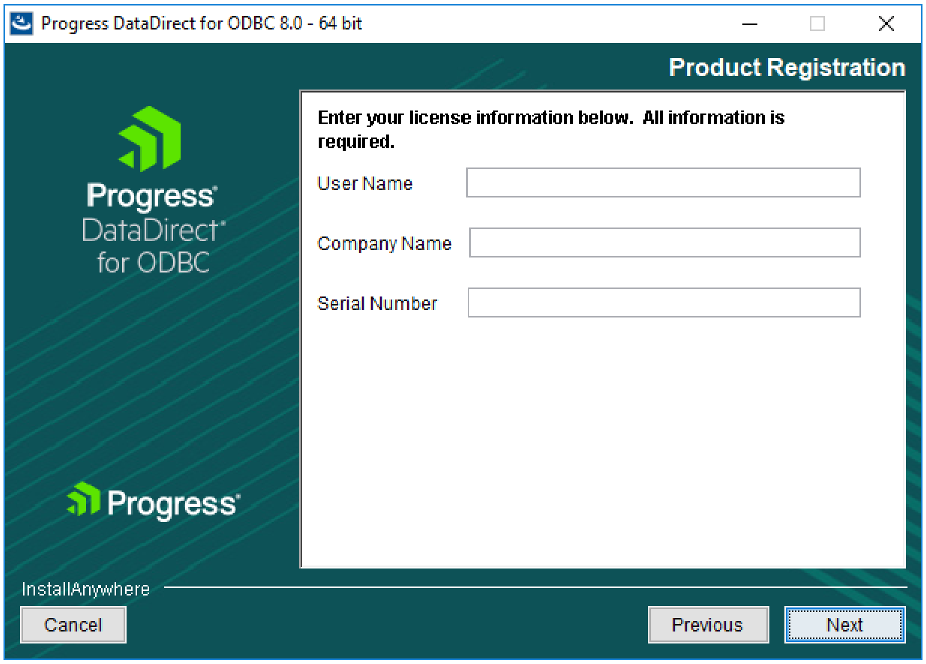
- □ Drivers
  - □ ODBC MongoDB 8.0 [All 32-bit Platforms All 64-bit Platforms] W
- ☑ Drivers (Installed)
  - ☑ Documentation Files 8.0.1.0 (LICENSED Installed)
  - ☑ Apache Spark SQL 8.0.2.0 (LICENSED Installed)

   Type the control number that was provided by Progress DataDirect in the Control # field, and click the **Validate** button. A tree menu of drivers with valid licenses appears in the selection box. You can add one or multiple control numbers, one at a time.

8. From the tree menu, select the drivers that you want to install. Drivers that are already installed are listed in the Drivers (Installed) tree and cannot be deselected. To remove installed drivers, you must uninstall the product. Click **Next** to continue.

**Note:** If you are installing a different version of a currently installed driver, the installer will overwrite the existing driver files with the version that you are installing.

9. The Product Registration window appears.



Enter your license information below. All information is required.

User Name

Company Name

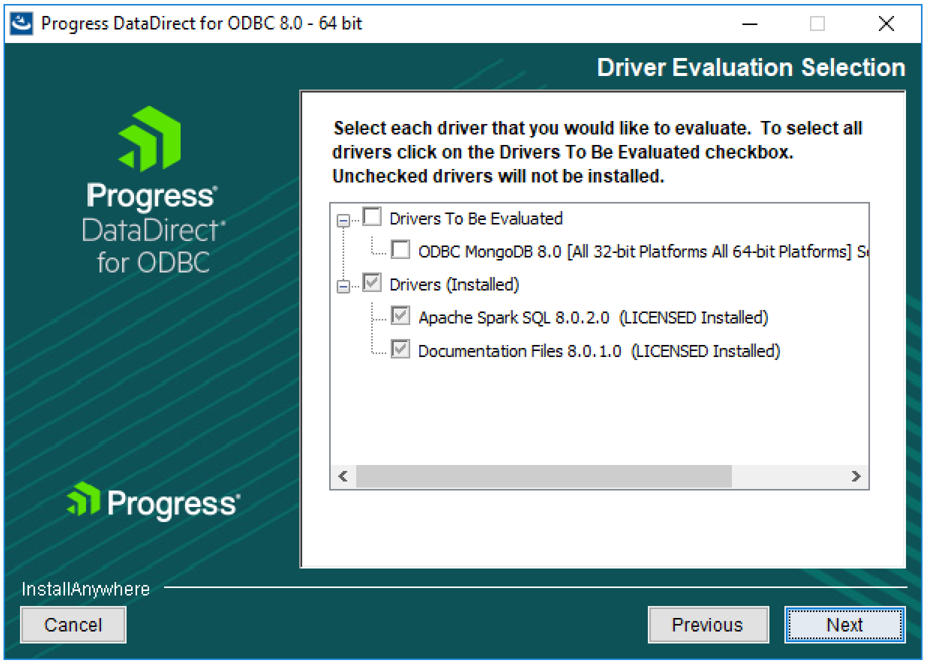
Serial Number

Provide the following information:

a) Type your name and company name in the corresponding fields.

b) Type the serial number that was provided by Progress DataDirect. Click **Next** to continue. Skip to Step 11 on page 8.

10. The Driver Evaluation Selection window appears.



Progress DataDirect for ODBC 8.0 - 64 bit
Driver Evaluation Selection
Progress DataDirect for ODBC
Progress
InstallAnywhere
Cancel
Previous
Next

Select each driver that you would like to evaluate. To select all drivers click on the Drivers To Be Evaluated checkbox. Unchecked drivers will not be installed.

- □ Drivers To Be Evaluated
  - □ ODBC MongoDB 8.0 [All 32-bit Platforms All 64-bit Platforms] S
- ☑ Drivers (Installed)
  - ☑ Apache Spark SQL 8.0.2.0 (LICENSED Installed)
  - ☑ Documentation Files 8.0.1.0 (LICENSED Installed)

From the **Drivers To Be Evaluated** tree, select the drivers that you want to install. Drivers that are already installed are listed in the Drivers (Installed) tree and cannot be deselected.

11. The Create Default Data Source window appears. Select the Create Default Data Source box to define a user data source in `HKEY_CURRENT_USER\SOFTWARE\ODBC\ODBC.INI` for each DataDirect driver that you install.

⚠ **Warning:** If you select Create Default Data Sources, existing data sources in your registry with the same DataDirect default name will be overwritten. To maintain your current DataDirect default data sources, rename them before you continue.

Then, click **Next** to continue.

12. The Pre-Installation Summary window provides the opportunity to review the information you have entered. When you are satisfied with your installation or branding option selections, click **Install** to begin the installation.
13. When the installation finishes, the Install Complete window appears. You may be prompted to restart your system, but it is not required to use the product. Select one of the following options:
    - **Yes, restart my system**. Select this option if you are ready to restart your system. The installer will restart your system when you exit the program.
    - **No, I will restart my system myself**. Select this option if you prefer to use the product without restarting the system, or if you are not ready to restart the system.
14. Click **Done** to exit the installer program.

This completes the installation.

A Progress DataDirect program group is created as part of the installation. This program group provides the following shortcuts:

- ODBC Administrator
- ODBC Driver Help

- ODBC Performance Tuning Wizard
- ODBC Readme
- Processor Information Utility
- Uninstall Progress DataDirect *for* ODBC *x.x*- 32 bit [32-bit only]
- Uninstall Progress DataDirect *for* ODBC *x.x*- 64 bit [64-bit only]
- XML Persistence Demo [32-bit only]

Before you begin using your driver, see "Setting the Library Path Environment Variable."

### See also

Setting the Library Path Environment Variable on page 9

# Setting the Library Path Environment Variable

Before you can use your driver, you must set the PATH environment variable to include the path of the `jvm.dll` file of your Java™ Virtual Machine (JVM).

# Creating an Input REST File

The driver supports an optional input REST file that allows you to specify multiple endpoints, define POST requests, and configure paging. You will need to create an input REST file if your session does any of the following:

- Accesses multiple endpoints
- Issues POST requests
- Accesses endpoints that require paging
- Accesses endpoints that use custom HTTP-headers

If your session does not require a REST file, you can specify your endpoint, for GET requests, using the REST Sample Path (`Sample`) option instead. For sessions that use a REST file, you will need to create a REST file and specify its location using the REST Config File (`Config`) option.

The input REST file is a simple text file that uses the *`file_name`*`.rest` naming convention. To configure the file, you will need to populate the contents with a list of comma-separated endpoints and requests using the formats described in this section. A sample REST file, named `example.rest`, is installed in the *`install_dir`*`\restfiles` directory. For additional examples, see "Sample Input REST File."

The following example demonstrates the basic format used in the REST file when mapping a table to the schema:

```
{
 "<table_name1>":"<endpoint1>",
 "<table_name2>":"<endpoint2>",
 "<table_name3>":"<endpoint3>"
}
```

## Specifying Endpoints for GET Request with Unparameterized Paths

To specify endpoints for unparameterized GET requests, use the following format:

```
"<table_name>":"<host_name>/<endpoint_path>",
```

*table_name*

is the name of the relational table to which the driver maps the endpoint. For example, `country`.

*host_name*

(optional) is the protocol and host name components of the URL endpoint. For example, `http://example.com`. You can omit this value by specifying the host name using the Host Name (`HostName`) option.

*endpoint_path*

is the path component of the URL endpoint. For example, `countries`.

For example, the following demonstrates a GET request that will map to the `countries` table.

```
"countries":"http://example.com/countries/",
```

## Specifying Endpoints for GET Request with Parameterized Paths

To specify parameterized GET requests, use the following format:

```
"<table_name>":"<host_name>/<endpoint_path1>/{<param_name>:<param_value>}[/<endpoint_path2>]",
```

*table_name*

is the name of the relational table to which the driver maps the endpoint. For example, `states`.

*host_name*

(optional) is the protocol and host name components of the URL endpoint. For example, `http://example.com`. You can omit this value by specifying the host name using the Host Name (`HostName`) option.

*endpoint_path*

is the path component of the URL endpoint. For example, `states`.

*param_name*

is the parameter identifier used for filtering the request. For example, `countryCode`.

*param_value*

is the parameter value used for filtering the request. For example, `USA`.

**Note:** When generating the relational map, the driver maps parameter identifiers specified in GET requests as columns, while the corresponding parameter values become the default values for each row within these columns. This allows data to be filtered by these parameters using the Where clause.

For example, the following demonstrates a GET request that will map to the `states` table.

```
"states":"http://example.com/states/get/{countryCode:USA}/all",
```

## Specifying Endpoints for GET Requests with Query Parameters

Use the following format to specify endpoints for GET requests with argument parameters. Multiple argument parameters within the same endpoint are separated by an ampersand (&).

```
"<table_name>":"<host_name>/<endpoint_path>?<parameter>=<value>[&...]",
```

*table_name*

is the name of the relational table to which the driver maps the endpoint. For example, `timeseries`.

*host_name*

(optional) is the protocol and host name components of the URL endpoint. For example, `http://example.com`. You can omit this value by specifying the host name using the User Name (`UserName`) option.

*endpoint_path*

is the path component of the URL endpoint. For example, `times`.

*parameter*

is the argument parameter component of the *parameter=value* pair used for filtering the request. For example, `interval`.

*value*

is the value argument parameter used for filtering the request. For example, `5min`.

For example, the following demonstrates a GET request that will map to the `timeseries` table.

```
"timeseries":"https://www.example.com/times/query?interval=5min&symbol=USA&function=TIME_SERIES_WEEKLY",
```

## Specifying Endpoints for GET Requests with Custom HTTP Headers

Some endpoints employ custom HTTP headers to filter data returned by a GET request. This type of filtering is typically used to create multiple unique reports/tables from the same endpoint. To use custom headers, you must define the request in the input REST file. The REST file entry is comprised of a path and header object. The path object contains the URL endpoint used in requests, while the header object defines the headers and provides value arguments used to filter the request.

In addition to filtering requests, the header object can be used to specify a value for the Accept header if the default, `application/json`, is not accepted by the endpoint. This scenario sometimes occurs when accessing a vendor endpoint that uses a proprietary Accept header.

An entry for a GET request using custom HTTP headers takes the following form:

```
"table_name":{
     "#path": "<host_name>/<endpoint_path>",
     "#headers":{
          "<header1>":"<value1>",
          "<header2>":"<value2>",
          "<header3>":"<value3>"
     }
}
```

*table_name*

is the name of the relational table to which the driver maps the endpoint. For example, `people`.

*host_name*

(optional) is the protocol and host name components of the URL endpoint. For example, `http://example.com`. You can omit this value by specifying the host name using the Host Name option.

*endpoint_path*

is the path component of the URL endpoint. For example, `times`.

*header*

is the HTTP header component of the *header=value* pair used for filtering the request. For example, `X-Subway-Payment`.

When overriding the Accept header, this value is `Accept`.

*value*

is the value argument for the HTTP header used for filtering the request or, if overriding the default Accept header, the value of the Accept header for the endpoint. For example, `token`.

For example, the following demonstrates an entry for a GET request that defines custom HTTP headers.

```
"people":{
     "#path": "http://example.com/people",
     "#headers":{
          "Accept":"application/calendar+json"
          "X-Subway-Payment":"token",
          "X-Laundry-Service":"dryclean",
          "X-Favorite-Food":"pizza"
     }
},
```

## Defining a POST Request

To use POST requests, you must define the request in the REST file in the JSON format. The definition entry is comprised of a path and body. The path contains the URL endpoint and the body used in requests, while the body defines documents and provides sample values. The driver uses these sample values to define which data type to be used when executing a POST request. An entry for a POST request takes the following form:

```
"table_name": {
          "#path": "<host_name>/<endpoint_path>",
          "#post": {
               "<field1>":"<value1>",
               "<field2>":"<value2>",
          }
     },
```

*table_name*

is the name of the relational table to which the driver maps the endpoint. For example, `countries2`.

*host_name*

(optional) is the protocol and host name components of the URL endpoint. For example, `http://example.com`. You can omit this value by specifying the host name using the Host Name (`HostName`) option.

*endpoint_path*

is the path component of the URL endpoint. For example, `country`.

*document*

is the document name of the *document=value* pair. For example, `START_DATE`.

*value*

is the sample value the driver uses to determine the data type to use when executing a POST to that document. For example, `2018-08-31`.

For example, the following demonstrates an entry for a POST request that will map to the `countries2` table.

```
"countries2": {
          "#path": "http://example.com/country/get/all",
          "#post": {
               "start_date":"2018-08-31",
               "end_date":"2018-09-01",
               "departments":"[engineering,marketing,sales]",
               "tags":"[blue,green,red]"
          }
     },
```

## Configuring Paging

The driver supports two types of paging: offset and page numbering paging. To configure paging, specify values for the properties in the following tables that correspond to the type of paging you want to employ. Paging properties can be set for individual GET or POST requests by specifying these options in the body object. If paging properties are not specified, the driver will attempt to retrieve the first page for data sources that require paging.

The following demonstrates configuring row offset paging for an unparametrized GET request:

```
 "table_name": {
           "#path": "<host_name>/<endpoint_path>",
           "#maximumPageSize":1000,
           "#firstRowNumber":1,
           "#pageSizeParameter":"maxResults",
           "#rowOffsetParameter":"startAt"
  },
```

*table_name*

is the name of the relational table to which the driver maps the endpoint. For example, `countries2`.

*host_name*

(optional) is the protocol and host name components of the URL endpoint. For example, `http://example.com`. You can omit this value by specifying the host name using the Host Name (HostName) option.

*endpoint_path*

is the path component of the URL endpoint. For example, `country`.

**Table 1: Row Offset Paging Properties**

| Property | Description |
|---|---|
| `#maximumPageSize` | The maximum page size in rows. |
| `#firstRowNumber` | The number of the first row. The default is `0`; however, some systems begin numbering rows at 1. |
| `#pageSizeParameter` | The name of the URI parameter that contains the page size. |
| `#rowOffsetParameter` | The name of the URI parameter that contains the starting row number for this set of rows. |

**Table 2: Page Number Paging Properties**

| Property | Description |
|---|---|
| `#maximumPageSize` | The maximum page size in rows. |
| `#firstPageNumber` | The number of the first page. The default is `0`; however, some systems begin numbering pages at 1. |
| `#pageSizeParameter` | The name of the URI parameter that contains the page size. |
| `#pageNumberParameter` | When requesting a page of rows, this is the name of the URI parameter to contain the page number. |

## Sample Input REST File

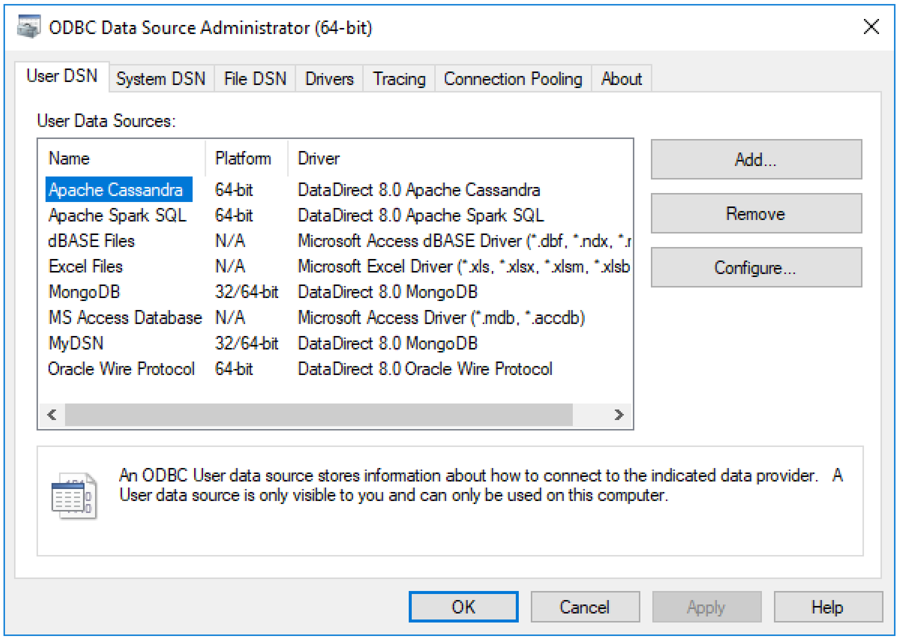
The following is a sample input REST file that demonstrates GET requests, POST requests, and a request configured for paging:

```
{
    // a simple GET request without parameters to sample:
    "countries":"http://example.com/country",

    // A GET request with a parameter in the path:
    "states":"http://example.com/states/get/{countryCode:USA}/all",

    // A GET request with parameters as arguments

"timeseries":"https://www.example.com/times/query?interval=5min&symbol=USA&function=TIME_WEEKLY",

// A GET request with parameters as arguments
"timeseries":"https://www.example.com/times/query?interval=5min&symbol=USA&function=TIME_WEEKLY",


  // A GET request with custom HTTP headers
    "people":{
         "#path": "http://example.com/people",
         "#headers":{
              "Accept":"application/calendar+json"
              "X-Subway-Payment":"token",
              "X-Laundry-Service":"dryclean",
              "X-Favorite-Food":"pizza"
         }
    },

    // A POST with parameters in the body
    "countries2": {
          "#path": "http://example.com/country",
          "#post": {
                "start_date":"2018-08-31",
                "end_date":"2018-09-01",
                "departments":"[engineering,marketing,sales]",
                "tags":"[blue,green,red]"
         }
    },

    // A GET with paging configured
    "products": {
          "#path": "http://example.com/products",
          "#maximumPageSize":1000,
          "#firstRowNumber":1,
          "#pageSizeParameter":"maxResults",
          "#rowOffsetParameter":"startAt"
    },
}
```

# Configuring a Data Source

To configure a basic data source to test your connection:

1. From the Progress DataDirect program group, start the ODBC Administrator.



User Data Sources:

| Name | Platform | Driver |
|---|---|---|
| Apache Cassandra | 64-bit | DataDirect 8.0 Apache Cassandra |
| Apache Spark SQL | 64-bit | DataDirect 8.0 Apache Spark SQL |
| dBASE Files | N/A | Microsoft Access dBASE Driver (*.dbf, *.ndx, *.r |
| Excel Files | N/A | Microsoft Excel Driver (*.xls, *.xlsx, *.xlsm, *.xlsb |
| MongoDB | 32/64-bit | DataDirect 8.0 MongoDB |
| MS Access Database | N/A | Microsoft Access Driver (*.mdb, *.accdb) |
| MyDSN | 32/64-bit | DataDirect 8.0 MongoDB |
| Oracle Wire Protocol | 64-bit | DataDirect 8.0 Oracle Wire Protocol |

An ODBC User data source stores information about how to connect to the indicated data provider. A User data source is only visible to you and can only be used on this computer.

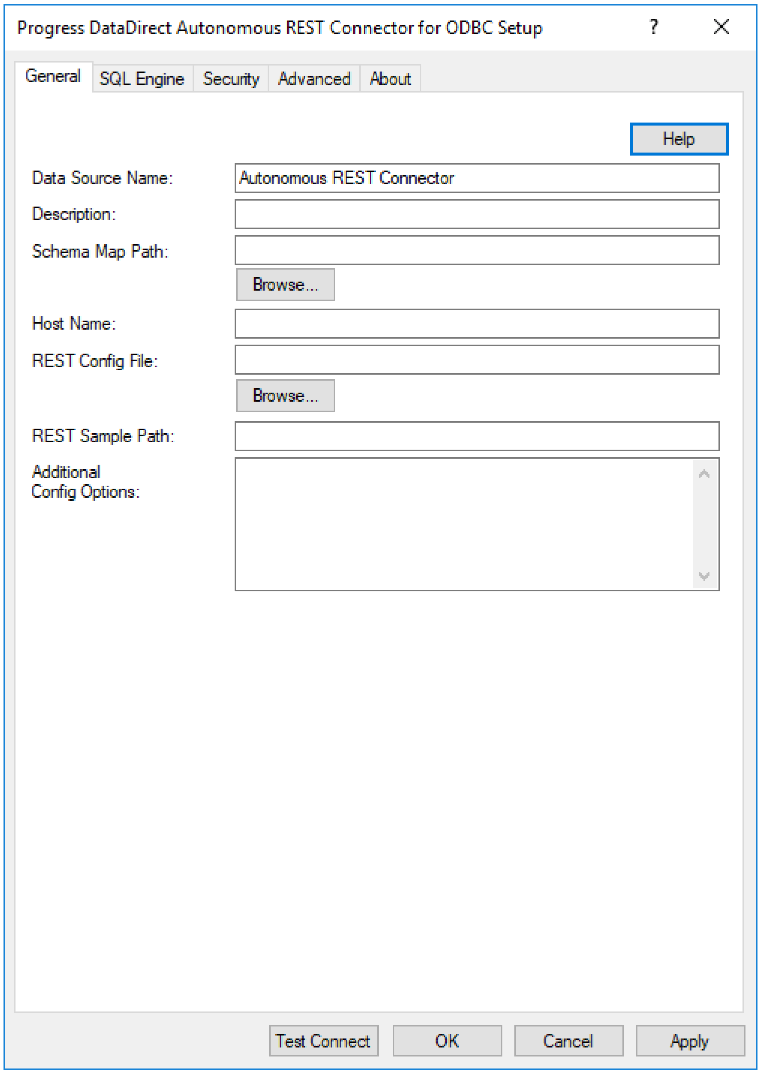
2. Click either the **User DSN**, **System DSN**, or **File DSN** tab to display a list of data sources.

   - **User DSN**: If you installed a default DataDirect ODBC user data source as part of the installation, select the appropriate data source name and click **Configure** to display the driver Setup dialog box.

   If you are configuring a new user data source, click **Add** to display a list of installed drivers. Select the Salesforce driver and click **Finish** to display the driver Setup dialog box.

   - **System DSN**: To configure a new system data source, click **Add** to display a list of installed drivers. Select the Salesforce driver and click **Finish** to display the driver Setup dialog box.
   - **File DSN**: To configure a new file data source, click **Add** to display a list of installed drivers. Select the Salesforce driver and click **Advanced** to specify attributes; otherwise, click **Next** to proceed. Specify a name for the data source and click **Next**. Verify the data source information; then, click **Finish** to display the driver Setup dialog box.

3. The General tab of the Setup dialog box appears by default.



Data Source Name: Autonomous REST Connector

Description:

Schema Map Path:

Host Name:

REST Config File:

REST Sample Path:

Additional Config Options:

On the General tab, provide the following information:

- **Data Source Name**: Type a string that identifies this data source configuration, such as Accounting.
- **Description**: Type an optional long description of a data source name, such as My Accounting Database.

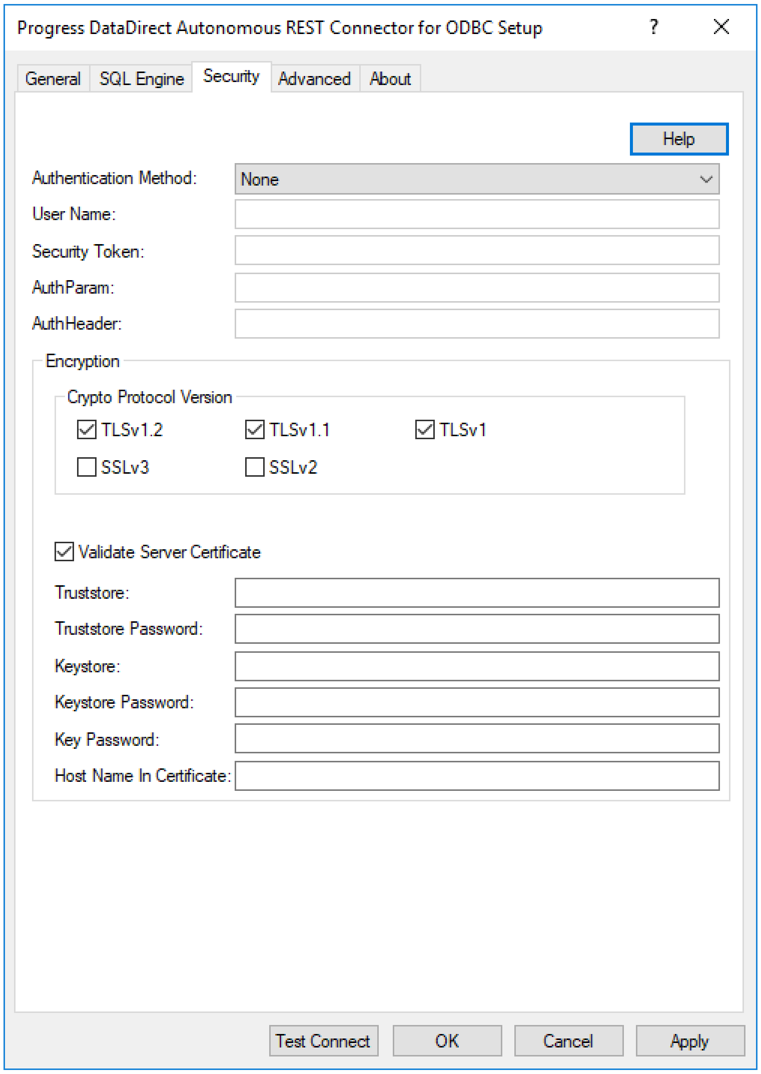
4. Provide a value for one of the following options:

   - For sessions using an input REST file (sessions with multiple endpoints, POST requests, paging, or custom headers):

     **REST Config File**: Type the name and location of the input REST file that contains the list of endpoints to sample and paging configuration information. For example, `C:/path/to/myrest.rest`.
   - For sessions that issue GET requests to a single endpoint:

     **REST Sample Path**: Type the endpoint that the driver connects to and samples. For example, `http://example.com/countries/`.

5. Click the Security tab.



Authentication Method: None

User Name:

Security Token:

AuthParam:

AuthHeader:

Encryption

Crypto Protocol Version

☑ TLSv1.2 ☑ TLSv1.1 ☑ TLSv1

□ SSLv3 □ SSLv2

☑ Validate Server Certificate

Truststore:

Truststore Password:

Keystore:

Keystore Password:

Key Password:

Host Name In Certificate:

From the Authentication Method drop-down, select the method used by your REST service:

- **None**: Select this method if your service does not require authentication; then, skip to Step 9 on page 19.
- **Basic**: Select this method if your service authenticates using user IDs, passwords, and HTTP headers; then, proceed to the next step.
- **HttpHeader**: Select this method if your service authenticates using security tokens passed via HTTP headers; then, proceed to Step 7 on page 18.
- **UrlParameter**: Select this method if your service authenticates using security tokens passed via URLs; then, proceed to Step 8 on page 19.

6. Provide values for the following options to configure basic authentication:

   - **User Name**: Enter your logon ID for the REST service.
   - **AuthHeader**: Enter the name of the HTTP header used for authentication. The default is `Authorization`.

7. Provide values for the following options to configure HTTP Header based authentication:

   - **Security Token**: Enter the security token required to make a connection to your endpoint. For example, `XaBARTsLZReM`.

- **AuthHeader**: Enter the name of the HTTP header used for authentication. The default is `Authorization`.

8. Provide values for the following options to configure URL parameter based authentication:
    - **AuthParam**: Enter the name of the URL parameter used to pass the security token. For example, `apikey`.
    - **SecurityToken**: Enter the security token required to make a connection to your endpoint. For example, `XaBARTsLZReM`.
9. To save your settings, click **Apply**; then, **OK**.

You are now ready to test the connection.

**Note:** Testing your connection with the **Test Connect** button only partially confirms that the driver is properly configured. Since REST services lack traditional database structures, you will need to retrieve data from your endpoint to confirm that you have successfully connected to your data. The Example application described in "Testing the Connection" allows you to quickly fetch data from your endpoint, thereby testing your connection.

### See also

Testing the Connection on page 19

# Testing the Connection

The driver installation includes an ODBC application called Example that can be used to connect to a data source and execute SQL.

**To run the Example program:**

1. After you have configured the data source, navigate to the *`installation_directory`*`\samples\example` directory.
2. Double-click the `Example.exe` file. A command prompt opens.
3. Follow the prompts to enter your data source name, user name, and password. If successful, a `SQL>` prompt appears.
4. At the prompt, enter SQL statements to test your connection. For example:

```
SELECT * FROM INFORMATON_SCHEMA.SYSTEM_TABLES
```

The results of your query are displayed. If example is unable to connect, the appropriate error message is returned.

# Accessing Data with Third-party Applications

The following sections guide you through using the driver to access your data with some common third-party applications:

- Accessing Data in Tableau on page 20
- Accessing Data in Microsoft Excel on page 23
- Accessing Data in Microsoft Excel from the Query Wizard on page 25

## Accessing Data in Tableau

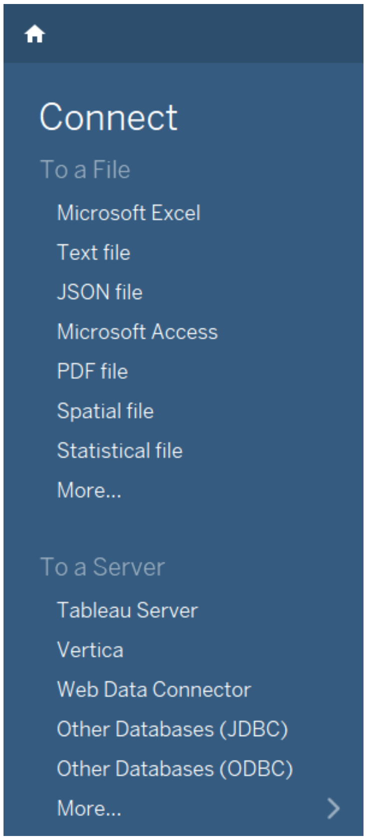
After you have configured your data source, you can use the driver to access your REST service data with Tableau. Tableau is a business intelligence software program that allows you to easily create reports and visualized representations of your data. By using the driver with Tableau, you can improve performance when retrieving data while leveraging the driver's relational mapping tools.

To use the driver to access data with Tableau:

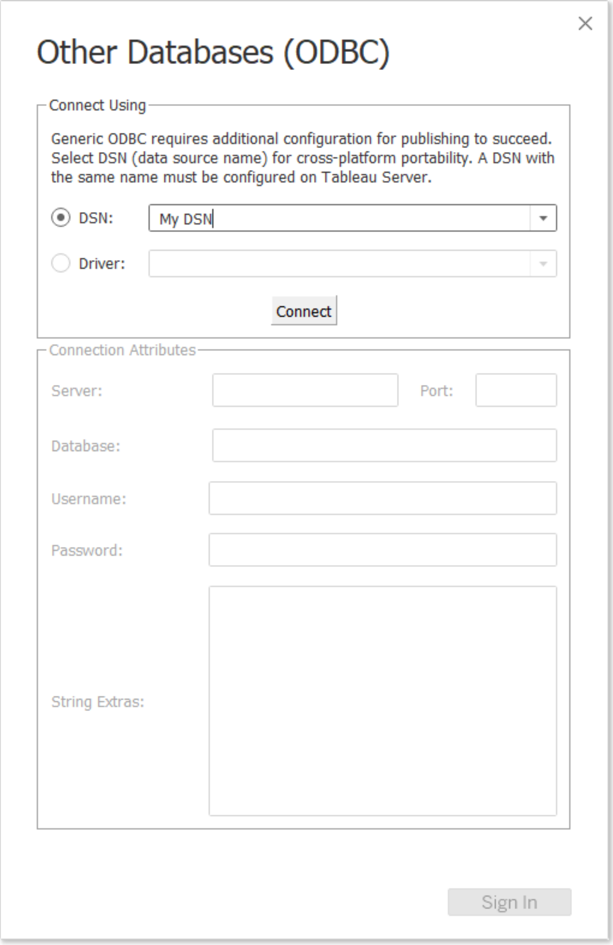
1. Navigate to the `\tools\Tableau` subdirectory of the Progress DataDirect installation directory; then, locate the Tableau data source file, `DataDirect AutoREST.tdc`.
2. Copy the `DataDirect AutoREST.tdc` into the following directory:

   `C:\Users\`*`user_name`*`\Documents\My Tableau Repository\Datasources`

3. Open Tableau. If the **Connect** menu does not open by default, select **Data > New Data Source** or the Add New Data Source button to open the menu.



4. From the **Connect** menu, select **Other Databases (ODBC)**.
5. The **Server Connection** dialog appears.



Other Databases (ODBC)

Connect Using

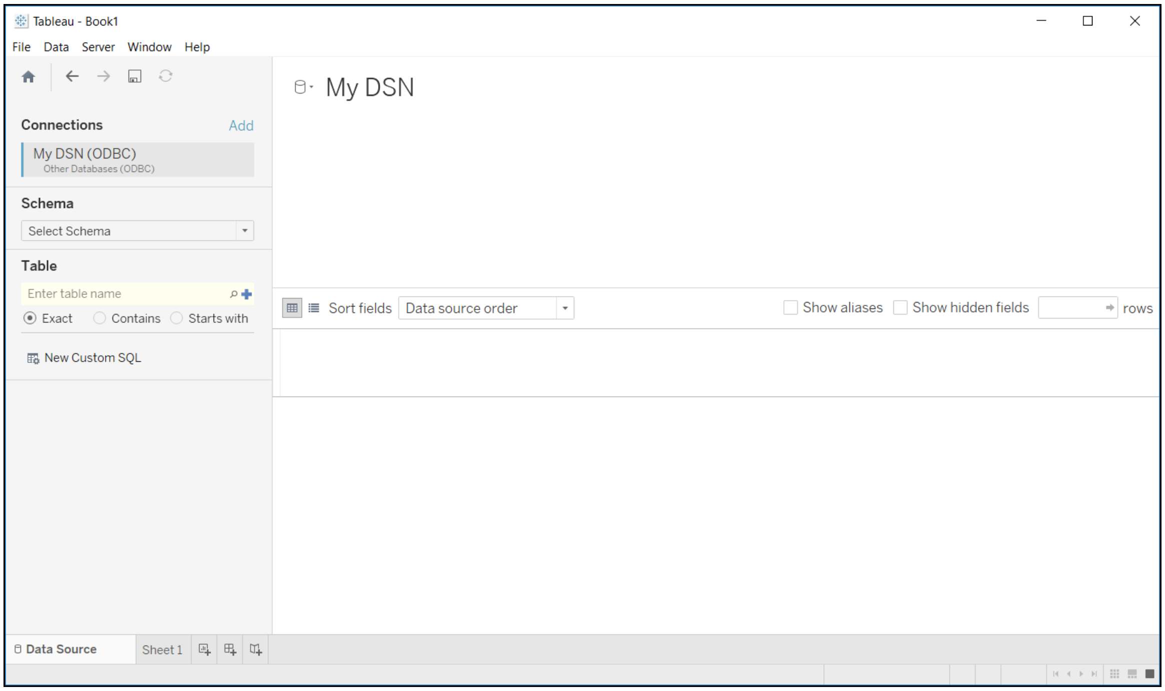
Generic ODBC requires additional configuration for publishing to succeed. Select DSN (data source name) for cross-platform portability. A DSN with the same name must be configured on Tableau Server.

● DSN: My DSN

○ Driver:

In the DSN field, select the data source you want to use from the drop-down menu. For example, **My DSN**. Then, click **Connect**. The Logon to Autonomous REST Connector dialog appears pre-populated with the connection information you provided in your data source.

6. If required, type your user name and password; then, click **OK**. The Logon dialog closes. Then, click **OK** on the Server Connection dialog.
7. The **Data Source** window appears.



By default, Tableau connects live, or directly, to your data. We recommend that you use the default settings to avoid extracting all of your data. However, if you prefer, you can import your data by selecting the **Extract** option at the top of the dialog.

8. In the Schema field, select the database you want to use. The tables stored in this database are now available for selection in the Table field.

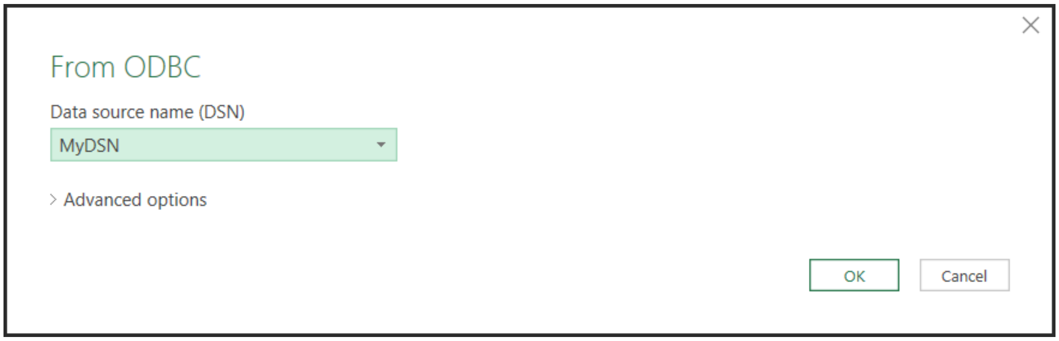
You have successfully accessed your data and are now ready to create reports with Tableau. For more information, refer to the Tableau product documentation at: http://www.tableau.com/support/help.

# Accessing Data in Microsoft Excel

After you have configured your data source, you can use the driver to access your data with Microsoft Excel from the Data Connection Wizard. Using the driver with Excel provides improved performance when retrieving data, while leveraging the driver's relational-mapping tools.

To use the driver to access data with Excel from the Data Connection Wizard:

1. Open your workbook in Excel.
2. From the **Data** menu, select **Get Data**>**From Other Sources**>**From ODBC**.
3. The **From ODBC** dialog appears.



From ODBC

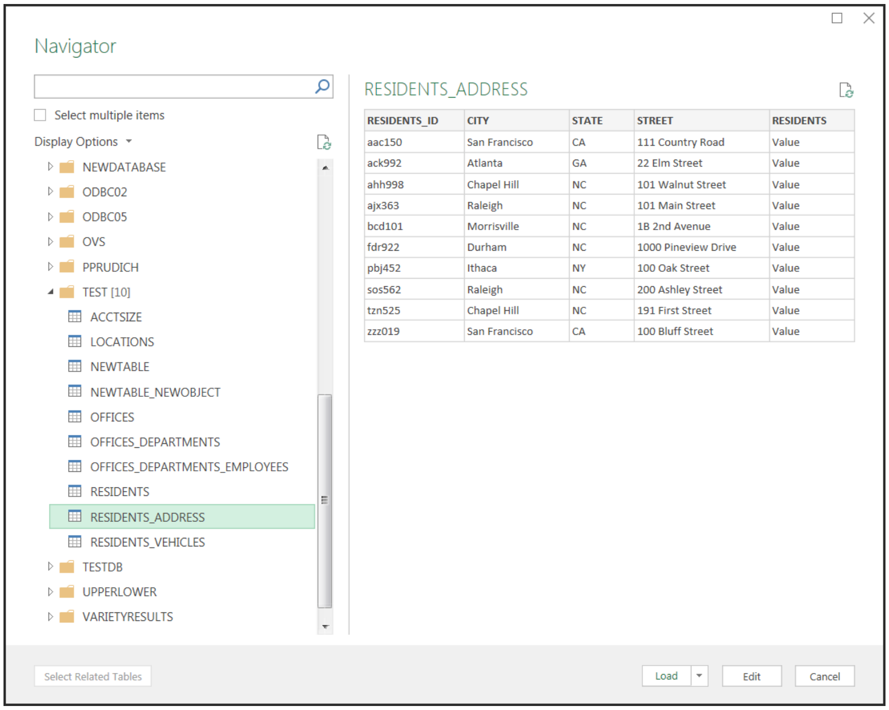
Data source name (DSN): MyDSN

Select your data source from the Data Source Name (DSN) drop down; then, click **OK**.

4. You are prompted for logon credentials for your data source:

    - If your data source does not require logon credentials or if you prefer to specify your credentials using a connection string, select **Default or Custom** from the menu on the left. Optionally, specify your credential-related properties using a connection string in the provided field. Click **Connect** to proceed.
    - If your data source uses Windows credentials, select **Windows** from the menu; then, provide your credentials. Optionally, specify a connection string with credential-related properties in the provided field. Click **Connect** to proceed.
    - If your data source uses credentials stored on the database, select **Database**; then, provide your user name and password. Optionally, specify a connection string in the provided field. Click **Connect** to proceed.

5. The **Navigator** window appears.



Navigator

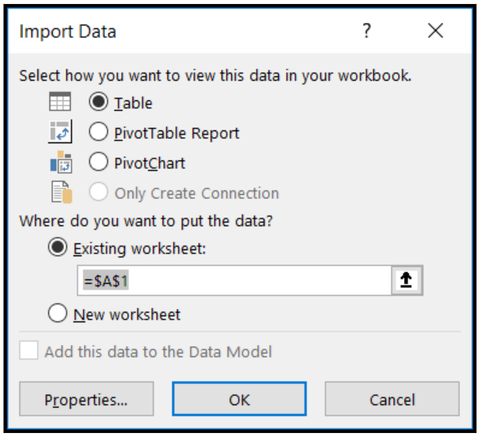
RESIDENTS_ADDRESS

| RESIDENTS_ID | CITY | STATE | STREET | RESIDENTS |
|---|---|---|---|---|
| aac150 | San Francisco | CA | 111 Country Road | Value |
| ack992 | Atlanta | GA | 22 Elm Street | Value |
| ahh998 | Chapel Hill | NC | 101 Walnut Street | Value |
| ajx363 | Raleigh | NC | 101 Main Street | Value |
| bcd101 | Morrisville | NC | 1B 2nd Avenue | Value |
| fdr922 | Durham | NC | 1000 Pineview Drive | Value |
| pbj452 | Ithaca | NY | 100 Oak Street | Value |
| sos562 | Raleigh | NC | 200 Ashley Street | Value |
| tzn525 | Chapel Hill | NC | 191 First Street | Value |
| zzz019 | San Francisco | CA | 100 Bluff Street | Value |

From the list, select the tables you want to access. A preview of your data will appear in the pane on the right. Optionally, click **Edit** to modify the results using the Query Editor. Refer to the Microsoft Excel product documentation for detailed information on using the Query Editor.

6. Load your data:
   - Click **Load** to import your data into your work sheet. Skip to the end.
   - Click **Load**>**Load To** to specify a location to import your data. Proceed to the next step.

7. The **Import Data** window appears.



Import Data

Select how you want to view this data in your workbook.

● Table
○ PivotTable Report
○ PivotChart
○ Only Create Connection

Where do you want to put the data?

● Existing worksheet:
=$A$1
○ New worksheet

□ Add this data to the Data Model

Select the desired view and insertion point for the data. Click **OK**.

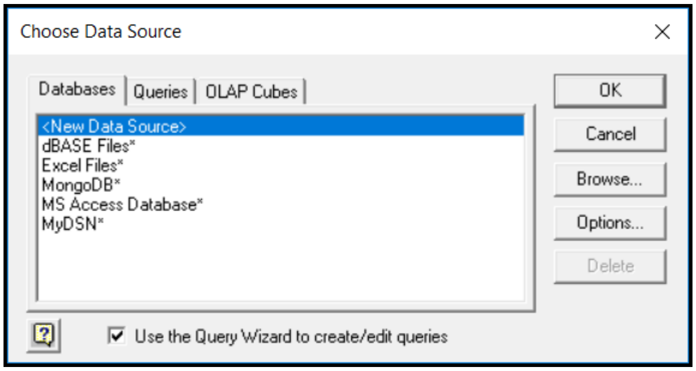
You have successfully accessed your data in Excel. For more information, refer to the Microsoft Excel product documentation at: https://support.office.com/.

## Accessing Data in Microsoft Excel from the Query Wizard

After you have configured your data source, you can use the driver to access your data with Microsoft Excel from the Query Wizard. Using the driver with Excel provides improved performance when retrieving data, while leveraging the driver's relational-mapping tools.

To use the driver to access data with Excel from the Query Wizard:

1. Open your workbook in Excel.
2. From the **Data** menu, select **Get Data**>**From Other Sources**>**From Microsoft Query**.
3. The **Choose Data Source** dialog appears.



Choose Data Source

<New Data Source>
dBASE Files*
Excel Files*
MongoDB*
MS Access Database*
MyDSN*

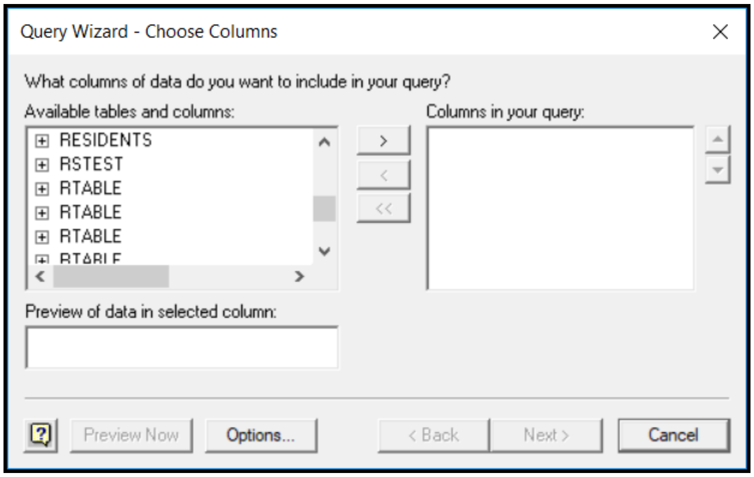
☑ Use the Query Wizard to create/edit queries

From the Databases list, select your data source. For example, **MyDSN**. Click **OK**.

4. The logon dialog appears pre-populated with the connection information you provided in your data source. If required, type your password. Click **OK** to proceed.

   **Note:** The logon dialog may reappear if Excel needs to access additional information from the data source. If this occurs, re-enter your password; then, click **OK** to proceed to the next step.

5. The **Query Wizard - Choose Columns** window appears.

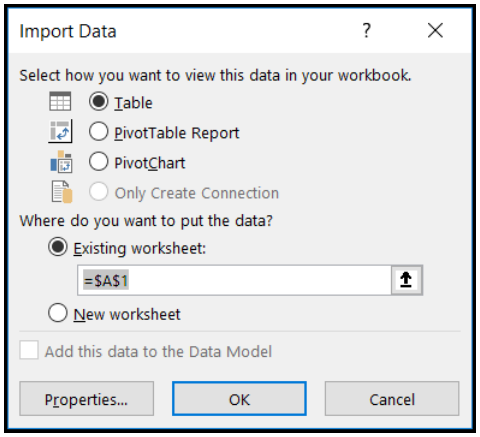


Query Wizard - Choose Columns

What columns of data do you want to include in your query?

   Choose the columns you want to import into your workbook. To add a column, select the column name in Available tables and columns pane; then, click the **>** button. After you add the columns you want to include, click **Next** to continue.

6. Optionally, filter your data using the drop-down menus; then, click **Next**.
7. Optionally, sort your data using the drop-down menus; then, click **Next**.
8. Select "Return Data to Microsoft Excel"; then, click **Finish**.
9. The **Import Data** window appears.



Import Data

Select how you want to view this data in your workbook.

● Table
○ PivotTable Report
○ PivotChart
○ Only Create Connection

Where do you want to put the data?

● Existing worksheet:
=$A$1
○ New worksheet

□ Add this data to the Data Model

   Select the desired view and insertion point for your data. Click **OK**.

You have successfully accessed your data in Excel using the Query Wizard. For more information, refer to the Microsoft Excel product documentation at: https://support.office.com/.

# Troubleshooting Setup/Connection Issues

This section describes common setup/connection issues you may encounter while trying to establish a database connection with the driver as well as some potential reasons for these issues. If you are experiencing a problem not described in this section, comprehensive troubleshooting resources are available in the "Troubleshooting" section of the User's Guide and Reference for your driver.

In addition, Progress DataDirect offers a variety of options to meet your technical support needs. See Contacting Technical Support on page 28 for details.

## Common Setup/Connection Issues

You are experiencing a setup/connection issue if you are encountering an error or hang while you are trying to make a database connection with the ODBC driver or are trying to configure the ODBC driver. Some common errors that are returned by the ODBC driver if you are experiencing a setup/connection issue include:

- `Specified driver could not be loaded.`
- `Data source name not found and no default driver specified.`
- `Unable to connect to destination.`
- `Invalid username/password; logon denied.`

## Troubleshooting the Issue

Some common reasons that setup/connection issues occur are:

- The library path environment variable is not set correctly. Before you can use the driver, you must set the PATH environment variable to the path of the `jvm.dll` file of your Java Virtual Machine (JVM). For Windows platforms, the library path environment variable is `PATH`.
- The database and/or listener are not started.
- The ODBC driver's connection options are not set correctly when configuring the driver. For example, the host name or port number are not correctly configured. See "Configuring a Data Source" for a list of connection options that are required for the driver to connect properly to the underlying database.

### See also

Configuring a Data Source on page 15

# Additional Resources

In addition to this quick start, the following resources enable you to take full advantage of the features and support offered for your driver.

- Product Documentation Library contains a comprehensive set of product documentation, including the following guides:
  - Progress DataDirect Connect Series for ODBC Installation Guide details requirements and procedures for installing the product.
  - Progress DataDirect Autonomous REST Connector for ODBC User's Guide and Reference guides you through using and configuring the driver, provides detailed reference information, and explains the tools used to troubleshoot common problems.
- Progress Support Knowledgebase provides answers to questions, access to technical documentation, release notes, product alerts and other support information.
- Progress Community allows you to contribute, share, and network with other Progress users and employees.
- Technical Support provides technical support services, including maintenance services and opening a support case.

## Contacting Technical Support

Progress DataDirect offers a variety of options to meet your support needs. Please visit our Web site for more details and for contact information:

https://www.progress.com/support

The Progress DataDirect Web site provides the latest support information through our global service network. The SupportLink program provides access to support contact details, tools, patches, and valuable information, including a list of FAQs for each product. In addition, you can search our Knowledgebase for technical bulletins and other information.

When you contact us for assistance, please provide the following information:

- Your number or the serial number that corresponds to the product for which you are seeking support, or a case number if you have been provided one for your issue. If you do not have a SupportLink contract, the SupportLink representative assisting you will connect you with our Sales team.
- Your name, phone number, email address, and organization. For a first-time call, you may be asked for full information, including location.
- The Progress DataDirect product and the version that you are using.
- The type and version of the operating system where you have installed your product.
- Any database, database version, third-party software, or other environment information required to understand the problem.
- A brief description of the problem, including, but not limited to, any error messages you have received, what steps you followed prior to the initial occurrence of the problem, any trace logs capturing the issue, and so on. Depending on the complexity of the problem, you may be asked to submit an example or reproducible application so that the issue can be re-created.

- A description of what you have attempted to resolve the issue. If you have researched your issue on Web search engines, our Knowledgebase, or have tested additional configurations, applications, or other vendor products, you will want to carefully note everything you have already attempted.
- A simple assessment of how the severity of the issue is impacting your organization.