isCOBOL Evolve: UDBC

Universal Database Connector

veryant

Table of Contents

Key Topics

- Getting Started
- UDBC SQL Server Engine
- ODBC Driver for UDBC SQL Server
- JDBC Driver for UDBC SQL Server
- ODBC Driver for isCOBOL File Server
- Troubleshooting

Overview

isCOBOL UDBC (Universal DataBase Connectivity) allows you to integrate COBOL files into a relational database-like environment.

It enables you to apply SQL to your COBOL files resulting in data that is accessed and managed as in RDBMS systems.

Windows-based applications like Microsoft Office and Crystal Reports, applications developed in ODBC supported environments such as Visual Basic, and Java applications that use JDBC standards will be able to access your COBOL files.

The isCOBOL UDBC solution provides two products:

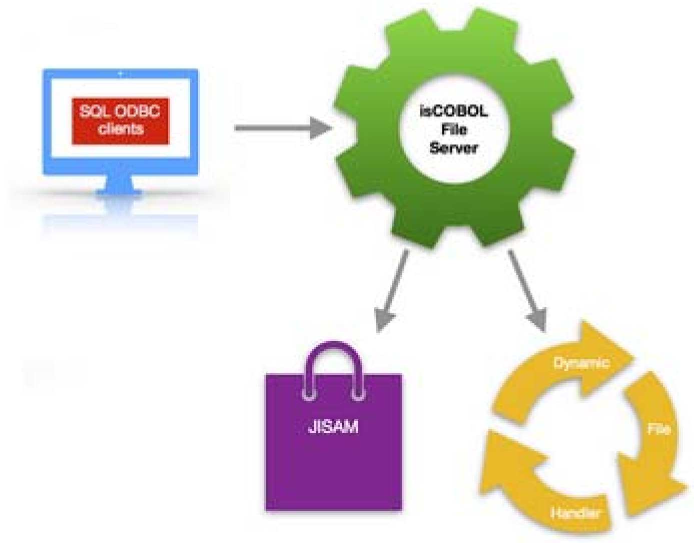
- The ODBC Driver for isCOBOL File Server, a ODBC driver that can be installed on every Windows machine and allows applications like Microsoft Office and Crystal Reports to access your COBOL files. This solution is preferable in most of the cases as it's the simplest to set up.
- The isCOBOL UDBC software kit, that is composed of a SQL Server engine, a command line SQL client, a ODBC driver and a JDBC driver. This solution is usually suggested when the clients are not Windows or, more generally speaking, when a JDBC access is required.

ODBC Driver for isCOBOL File Server

Overview

The isCOBOL Driver for File Server is a communications interface designed to give you easy access to your COBOL data files from popular Microsoft Windows applications. The driver interface enables you to benefit from modern PC productivity tools without having to restructure your existing data files.

isCOBOL Driver for File Server connects directly to the isCOBOL File Server without communicating with the UDBC SQL Engine. This characteristic makes it lighter solution than the ODBC Driver for UDBC SQL Server to interface File Server data from ODBC.



Getting Started

The setup of an isCOBOL UDBC environment requires the following steps:

1. Download and install the Java Runtime Environment (JRE)
2. Download and install isCOBOL UDBC
3. Activate the License

In order to activate your isCOBOL Evolve products, you will need the e-mail you received from Veryant containing your license key. Contact your Veryant representative for details.

Download and install the Java Runtime Environment (JRE)

JRE version 8, 11 or 17 must be installed on your machine in order to use isCOBOL UDBC. For best results and performance, install the latest update build available for your platform.

Self-extracting setups are provided for the Windows platform

On Unix/Linux platforms Java may be already installed. If it's not the case, you can install it using the appropriate system commands (e.g. yum, or apt-get).

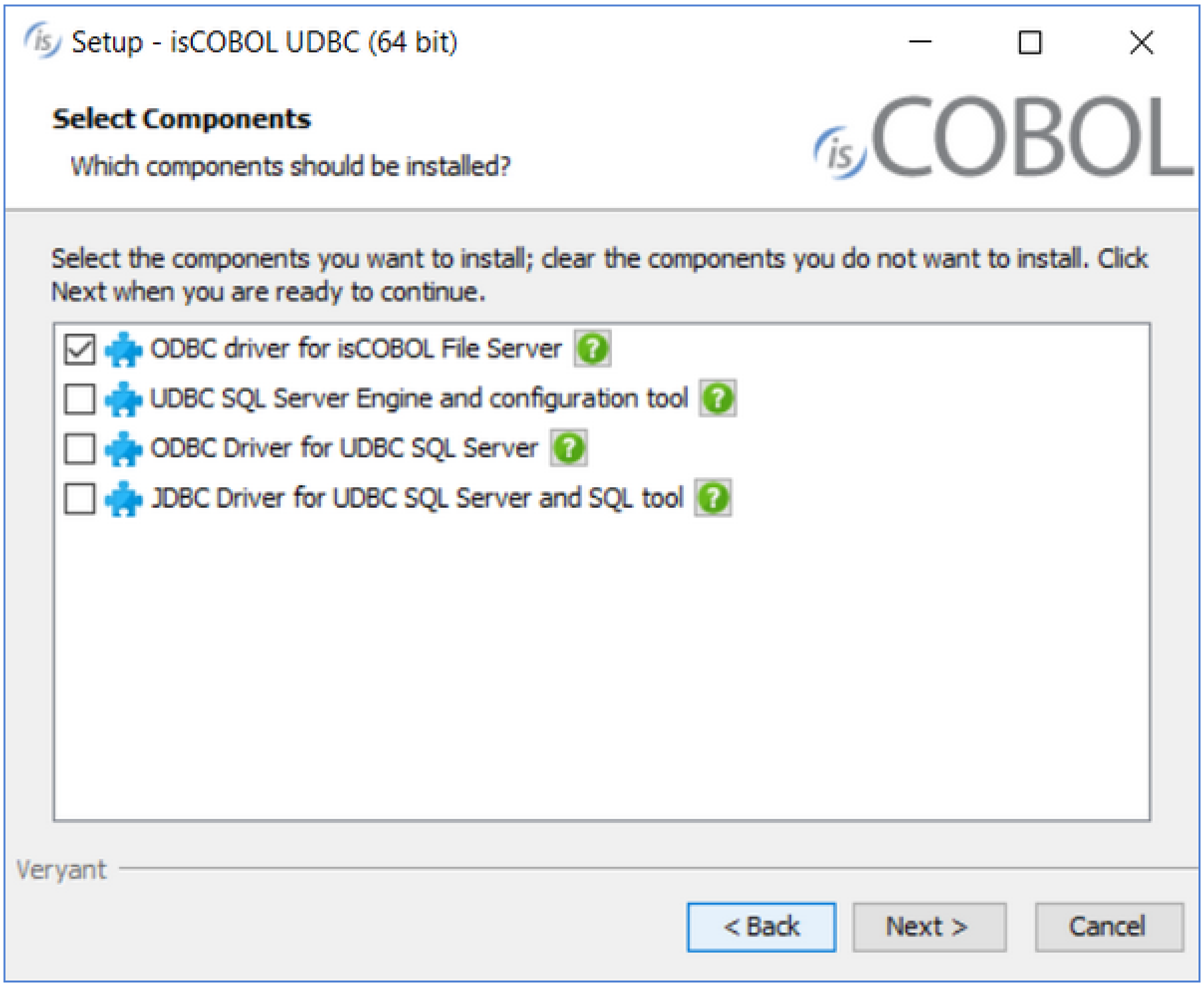
Download and install isCOBOL UDBC

Windows

1. If you haven't already done so, Download and install the Java Runtime Environment (JRE).
2. Go to "https://support.veryant.com".
3. Sign in with your User ID and Password.
4. Click on the "Download Software" link.
5. Scroll down to the list of files for Windows x64 64-bit or Windows x86 32-bit. Select isCOBOL_UDBC*yyyyR*_*n*_Windows*arc*.msi, where *yyyy* is the year, *r* is the release number, *n* is the build number and *arc* is the system architecture.
6. Run the downloaded installer to install the files.

Note - If your Windows has the option "Run as Administrator", you should run the setup with that option, otherwise the setting of environment variables might silently fail.

7. Select the desired items from the list of products when prompted.


Setup - isCOBOL UDBC (64 bit)

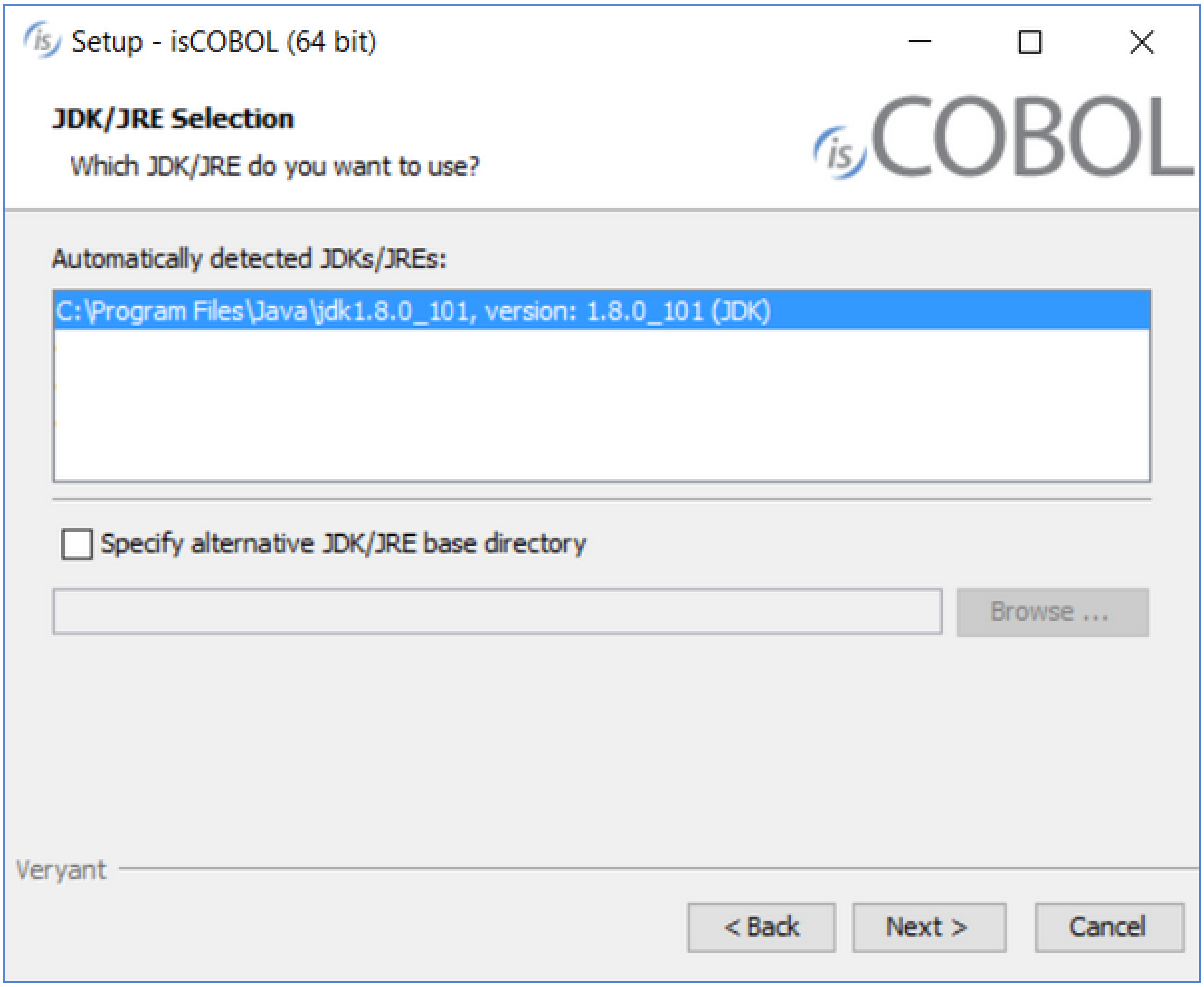
Select Components

Which components should be installed?

Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.

- [✓] ODBC driver for isCOBOL File Server
- [ ] UDBC SQL Server Engine and configuration tool
- [ ] ODBC Driver for UDBC SQL Server
- [ ] JDBC Driver for UDBC SQL Server and SQL tool

8. Select your JDK/JRE when prompted


Setup - isCOBOL (64 bit)

JDK/JRE Selection

Which JDK/JRE do you want to use?

Automatically detected JDKs/JREs:

C:\Program Files\Java\jdk1.8.0_101, version: 1.8.0_101 (JDK)

☐ Specify alternative JDK/JRE base directory

9. Follow the wizard procedure to the end. In the process you will be asked to provide the installation path ("C:\Veryant" by default) and license keys. You can skip license activation and perform it later, as explained in Activate the License.

Quiet mode

The isCOBOL UDBC setup supports the msi quiet mode. Settings can be driven with a response file.

A response file is a text file with name-value pairs that represent installer variables.

A response file is generated automatically after an installation is finished. The generated response file is found in the *.install4j* directory of the isCOBOL UDBC and is named *response.varfile*.

When an installer is executed, it checks whether a file with the same name and the *.varfile* extension can be found in the same directory and loads that file as the response file. For example, if an installer is named *foo_setup.msi* on Windows, the response file next to it has to be named *foo_setup.varfile*.

For more information about msi setups and their command line options, see Microsoft Standard Installer Command-Line Options.

Distribution Files

For information on a specific distribution file, please see the README file installed with the product.

Activate the License

If you provided license keys during the installation, on Windows, you should skip reading this chapter.

The ODBC Driver for isCOBOL File Server looks for the following configuration property for the license key:

```
iscobol.udbc.license.2023=<license_key>
```

The key should be stored in one of the following files (if they exist):

1. \etc\iscobol.properties in the drive where the working directory is
2. C:\Users\<username>\iscobol.properties (the setup wizard saves licenses here, if you don't skip activation)
3. iscobol.properties found in the Java Classpath
4. a custom configuration file passed on the command line
5. %ISCOBOL%\iscobol.properties

NOTE - Files are listed in the order they're processed. If the license key appears in more than one of the above files, then the last occurrence is considered.

File Server startup

The ODBC Driver for isCOBOL File Server requires isCOBOL File Server to be up and running.

The isCOBOL File Server is provided along with isCOBOL.

Assuming that you have correctly installed isCOBOL and its thin client, you can start the File Server with the following command:

```
iscserver -as -fs
```

For more information about the iscserver command, see isCOBOL File Server usage.

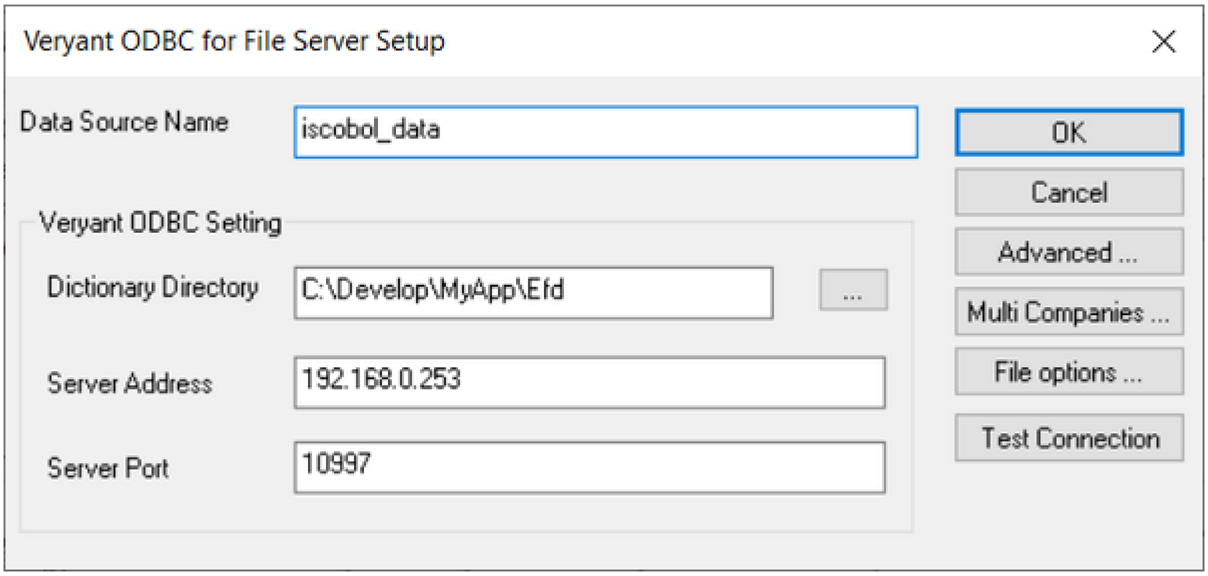
How to create a DSN

1. Open Control Panel, Administrative Tools, ODBC Data Sources
2. Select either the User DSN or System DSN page.
3. Click on the "Add" button.
4. Select "Veryant ODBC Driver for File Server" between the available drivers.

Note - if you're running the 32 bit ODBC on a 64 bit Windows, you have to open the 32 bit ODBC manager by running *C:\Windows\SysWOW64\odbcad32.exe.*

The following sections explain the DSN configuration panels of the isCOBOL Driver for File Server.

How to configure the DSN (Main Setup)



Data Source Name: iscobol_data

Dictionary Directory: C:\Develop\MyApp\Efd

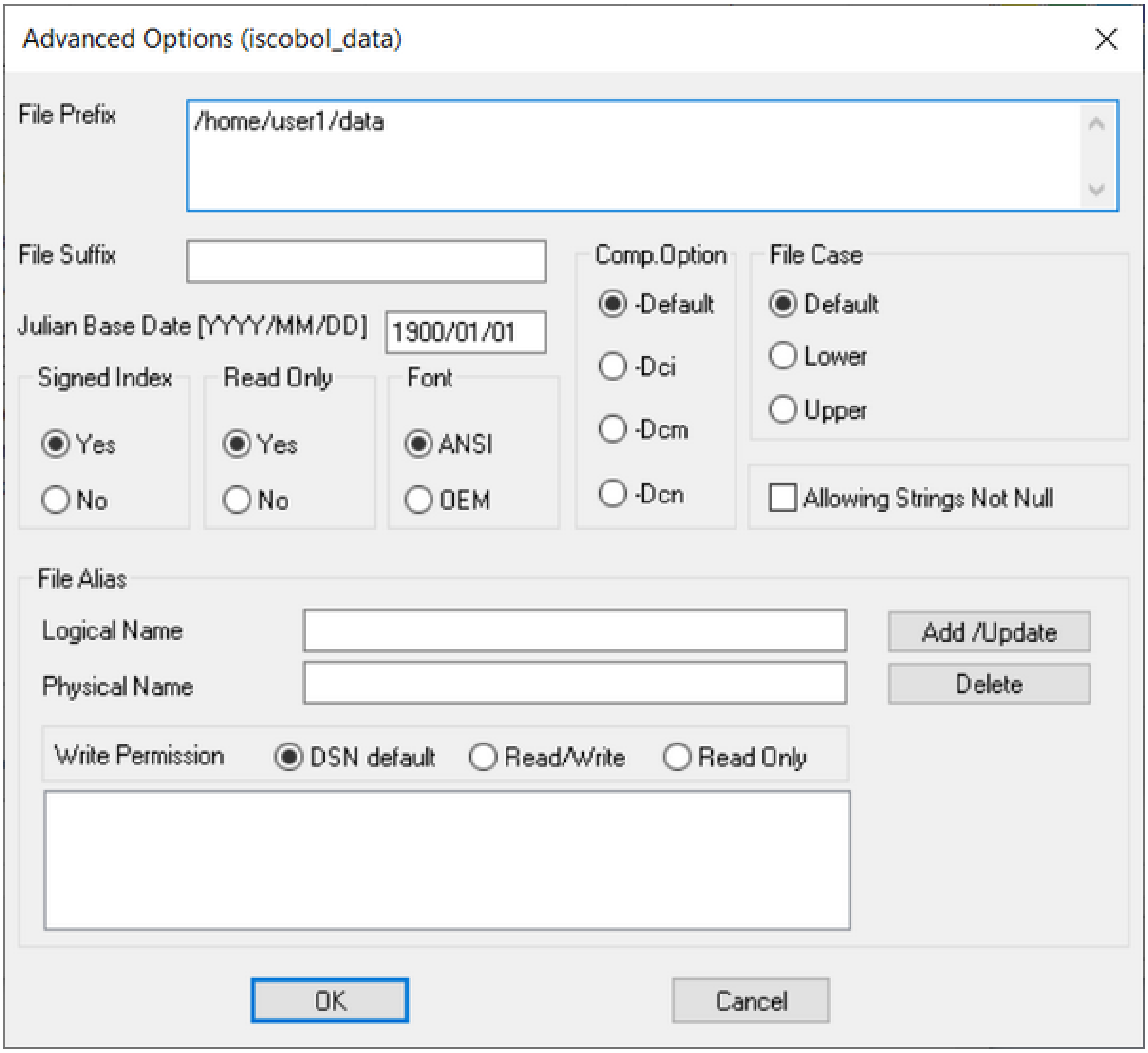
Server Address: 192.168.0.253

Server Port: 10997

- **Data Source Name** is the name of the Data Source. ODBC programs will connect to this Data Source to work with the files.
- **Dictionary Directory** is the directory containing the EFD (extend file descriptors) of the files of our DSN. EFD files are xml files generated by the isCOBOL compiler if you compile with the -efd option.

  If the value of this field starts with a colon (:) it means that the directory is not a local folder, but it’s a remote folder instead. Such folder is searched on the server specified by *Server Address*. The driver expects to find a file named "efdrmt.dir" (lowercase) in that remote folder. This file must list the names of the xml files that have to be accessed by the DSN. The xml files included in the "efdrmt.dir" file must reside in the same directory as the "efdrmt.dir" file itself.
- **Server Address** is the address of the machine where the isCOBOL File Server is active and listening. If omitted, then localhost is assumed.
- **Server Port** is the port of the machine where the isCOBOL File Server is active and listening. If omitted, then 10997 is assumed.

How to configure the DSN (Advanced Setup)


Advanced Options (iscobol_data)

File Prefix: /home/user1/data

File Suffix:

Julian Base Date [YYYY/MM/DD]: 1900/01/01

Signed Index: [✓ Yes] No

Read Only: [✓ Yes] No

Font: [✓ ANSI] OEM

Comp.Option: [✓ -Default] -Dci -Dcm -Dcn

File Case: [✓ Default] Lower Upper

☐ Allowing Strings Not Null

File Alias

Logical Name:

Physical Name:

Write Permission: [✓ DSN default] Read/Write Read Only

- **File Prefix** is the list of the directories where physical files are stored on the server.
- **File Suffix** is the suffix of the files. The extension appended by the file handler (e.g. ".dat" appended by c-tree and JIsam) must not be specified here. Only the custom extension appended by the COBOL programs that created the files must be specified here.
- **Julian Base Date** allows the julian base date to be used in defining julian date formats to be specified. The information in this field must have the following format:

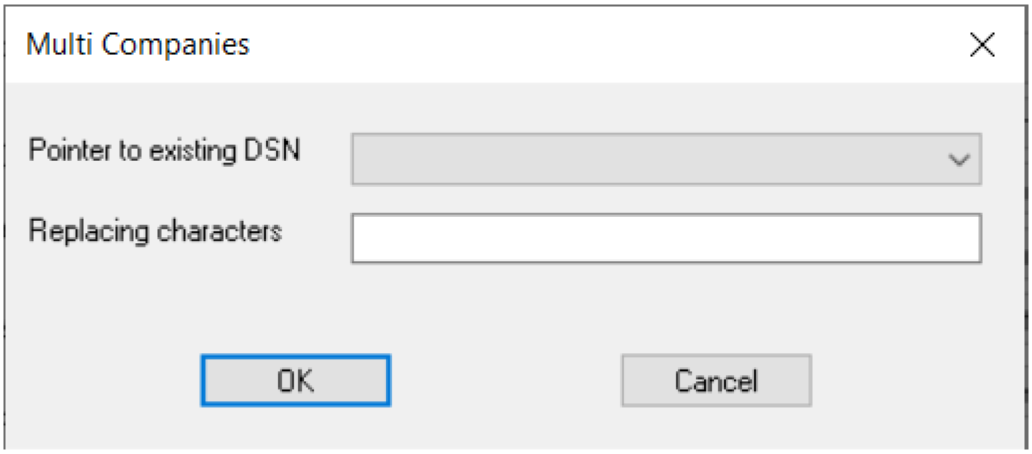
```
yyyy/mm/dd
```

where yyyy is the year, mm is the month, and dd is the day.

By default, 1900/01/01 is the base date (January 1, 1900). 0001/01/01 is the minimum base date allowed.

- **Signed Index.** If your application has any numeric fields that belong to a key or index, you may have a problem with record ordering when a negative value occurs in the field. (A negative value results in an index that is not ordered.) Set this option to "No" to disqualify numeric signed fields from being indexes. There is some performance degradation when this option is set to "No." Since it is the case that most signed fields included in a key do not contain negative numbers, you might want to set this option to "Yes" to avoid the performance penalty.
- **Read Only** establishes the default read/write permission for all the files belonging to this DSN. Select "Yes" if the files are read-only. Select "No" if they are read and write approved.
- **Font** specifies the character set to be used for accepting and displaying data on the screen.
- **Comp. Option.** If you specified the "-Dci" or "-Dcm" compiler option when compiling your program, click the radio button for that option here. This tells the driver which data storage convention was used to create your data files.
- **File Case** specifies the case of the file names in your destination directory (applies to UNIX file names only). By default, the file names are assumed to be in upper case. If your filenames are lower case, you must click the "Lower" radio button here, or the driver will not be able to find your files.
- **Allowing Strings Not Null** has been provided to avoid problems with old versions of Ms Access.
- **File Alias** allows you to create aliases between physical names (the name of the disk file) and logical names (the name used by the ODBC client to reach the file).

How to configure the DSN (Multi Companies)



Using multi-company support option, you can manage customer data files from several companies with a single DSN. The data files from each customer (company) are structured the same and even named the same, but stored in a different directory.

The name of the company directories can assume several formats. Usually the name is composed of two different parts: a fixed prefix (or suffix), like acct for example, and a variable that is used to identify the company itself. This variable can be a progressive digit or a part of the name of the company. The directory name can also reflect the name of the company itself, barring operating system naming restrictions.

- **Pointer to existing DSN** specifies the data source name you assigned the master DSN.
- **Replacing characters** is the string of characters that needs to be replaced, followed by an equals sign and the string of characters to be used for this company. For example, when creating the Acme Glass DSN, type:

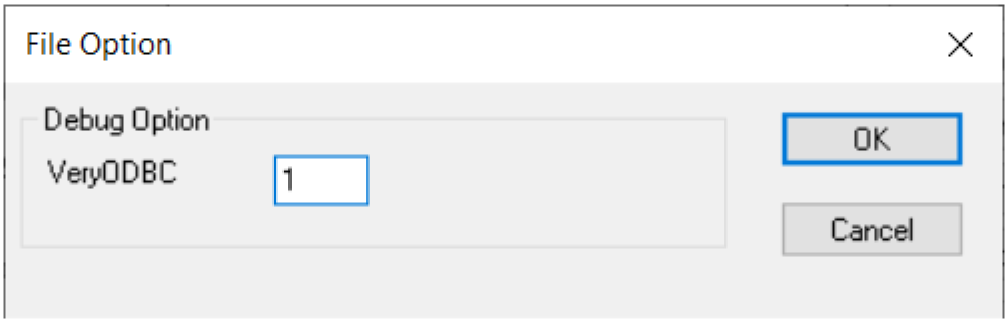
```
xxx=acme
```

If more than one variable was used, separate the variables by a space. For example:

```
xxx=acme yyy=data
```

When the driver opens the data source, it replaces the general string found in File Prefix with the specific string in order to obtain the correct pathname.

How to configure the DSN (File Options)


File Option

Debug Option

VeryODBC 1

Use the File Options dialog to turn on driver activity logging. "1" in the entry field turns logging on, "0" disables it. The log file *VeryantODBC.log* is created in the temp directory. You can retrieve the temp directory by callng the GetTempPath function, which looks for environment variables in the following order and uses the first path found:

- The path specified by the TMP environment variable.
- The path specified by the TEMP environment variable.
- The path specified by the USERPROFILE environment variable.
- The Windows directory.

Troubleshooting

These are the most common errors you can run into using the isCOBOL Driver for File Server.

100	A duplicate key was detected where duplicates are not allowed.

102	This indicates that a routine was called with an illegal parameter. For example, specifying a key number that is larger than the number of keys in the file would result in this error.
104	An attempt was made to open more files than the system allows open at once.
105	The file content can’t be read. This error is returned when one of these three conditions is true: • the index file is corrupt, • the index file is encrypted, • the physical disk file does not match the type of file being opened. If the index file is corrupt, it should be reconstructed using the appropriate host system utility. The exact nature of this corruption is identified by "f_int_errno."
107	The requested record is locked by another process or (for some systems) by another file handle used by this process.
111	The requested record was not found.
112	The current key of reference was in the "undefined" state when the "i_next" or "i_previous" routines were called. There is no current record from which to read forwards or backwards.
113	The file is locked against the current open mode.
116	The system ran out of dynamic memory.
117	An operation was requested that the current open mode doesn't allow. For example, attempting to add a record to a file that is open for input-only would result in this error.
126	The requested operation is not supported by this host system.
127	The disk became full while adding the new record.
128	The size of the new record doesn't match the type of file being opened.
129	The system ran out of lock-table entries.
130	The file is missing.
131	Invalid permission for the operation.
147	Mismatch between physical file logical-info and the info included in the EFD dictionary

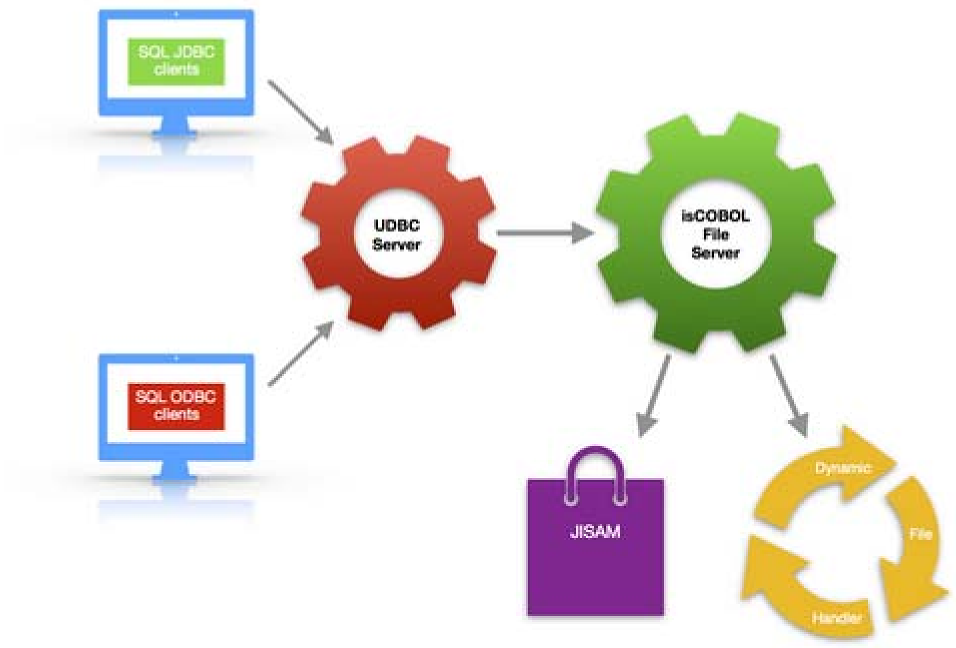
isCOBOL UDBC

Overview

The UDBC Server receives connections from SQL clients and provides them with access to COBOL files through the isCOBOL File Server. See isCOBOL File Server for more information on how to set up the isCOBOL File Server in order to allow remote access to files.

EFD dictionaries for each file are required. Compile COBOL programs with the -efd option to create an EFD dictionary for each file described in the FILE SECTION.

The below picture summarizes how SQL clients access COBOL files.



Getting Started

The setup of an isCOBOL UDBC environment requires the following steps:

1. Download and install the Java Runtime Environment (JRE)
2. Download and install isCOBOL UDBC
3. Activate the License

In order to activate your isCOBOL Evolve products, you will need the e-mail you received from Veryant containing your license key. Contact your Veryant representative for details.

Download and install the Java Runtime Environment (JRE)

JRE version 8 or 11 must be installed on your machine in order to use isCOBOL UDBC. For best results and performance, install the latest update build available for your platform.

Self-extracting setups are provided for the Windows platform

On Unix/Linux platforms Java may be already installed. If it's not the case, you can install it using the appropriate system commands (e.g. yum, or apt-get).

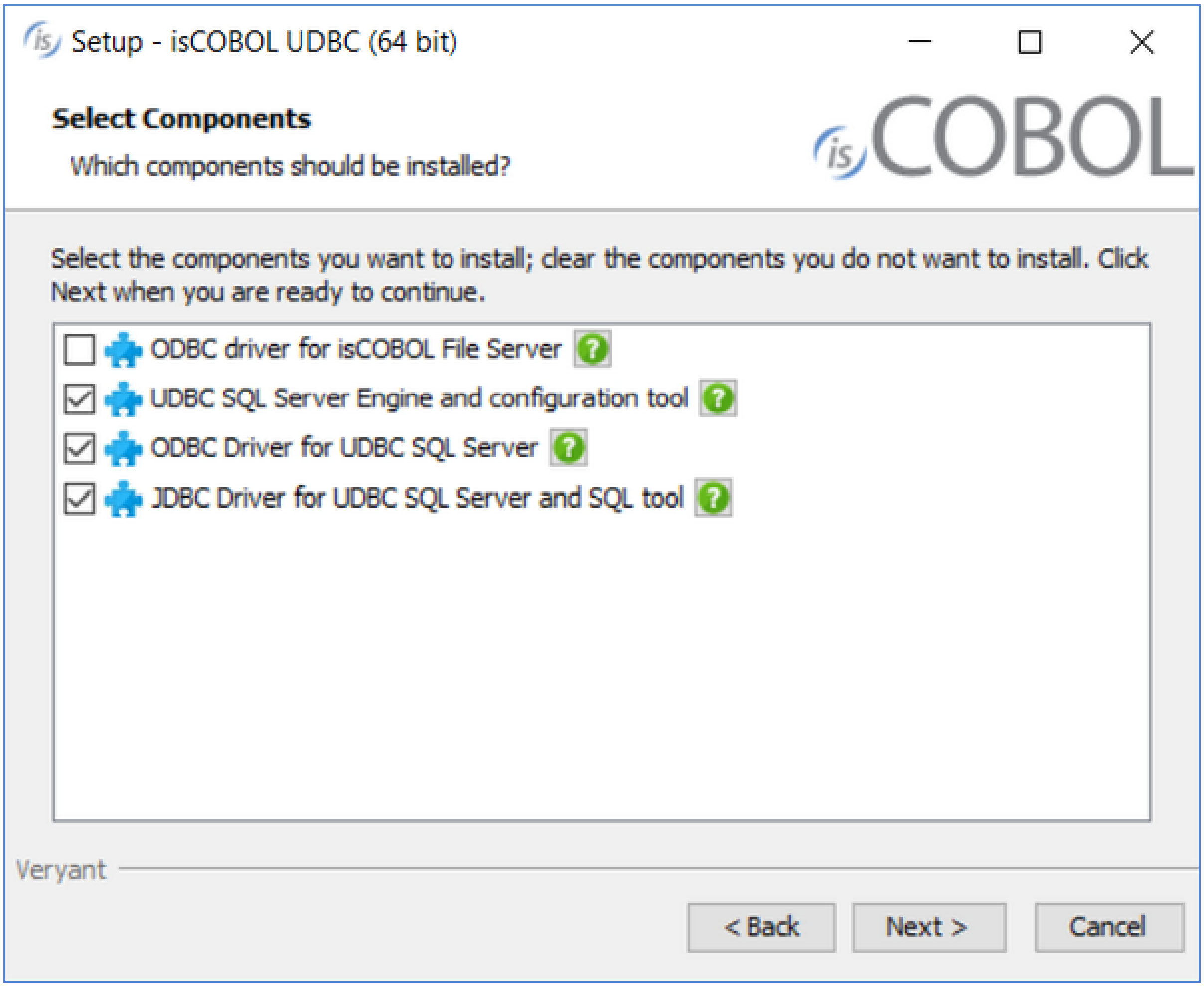
Download and install isCOBOL UDBC

Windows

1. If you haven't already done so, Download and install the Java Runtime Environment (JRE).
2. Go to "https://support.veryant.com".
3. Sign in with your User ID and Password.
4. Click on the "Download Software" link.
5. Scroll down to the list of files for Windows x64 64-bit or Windows x86 32-bit. Select isCOBOL_UDBC*yyyyR*_*n*_Windows*arc*.exe, where *yyyy* is the year, *r* is the release number, *n* is the build number and *arc* is the system architecture.
6. Run the downloaded installer to install the files.

Note - If your Windows has the option "Run as Administrator", you should run the setup with that option, otherwise the setting of environment variables might silently fail.

7. Select the desired items from the list of products when prompted.


Setup - isCOBOL UDBC (64 bit)

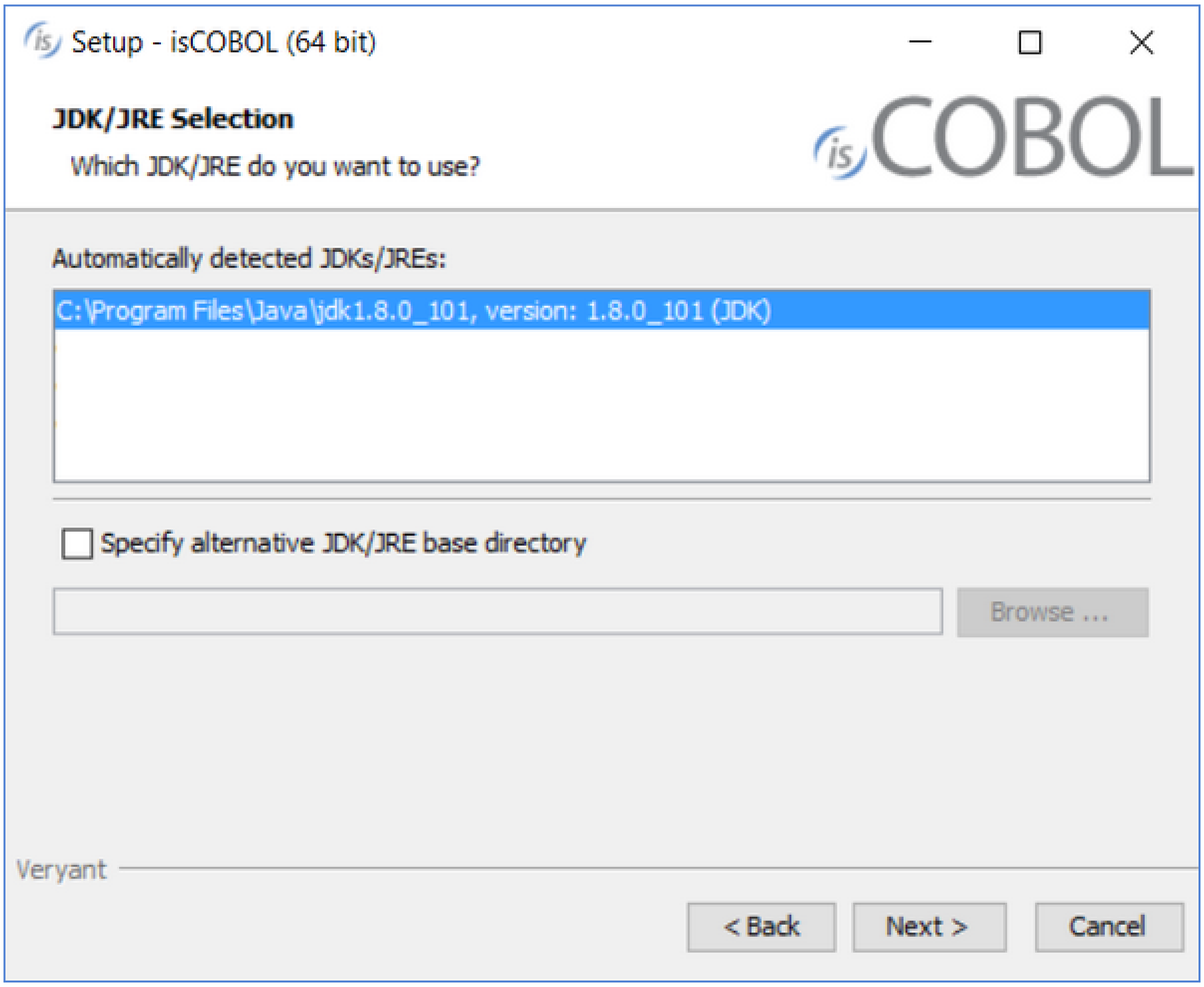
Select Components

Which components should be installed?

Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.

- [ ] ODBC driver for isCOBOL File Server
- [✓] UDBC SQL Server Engine and configuration tool
- [✓] ODBC Driver for UDBC SQL Server
- [✓] JDBC Driver for UDBC SQL Server and SQL tool

8. Select your JDK/JRE when prompted



Setup - isCOBOL (64 bit)

JDK/JRE Selection

Which JDK/JRE do you want to use?

Automatically detected JDKs/JREs:

C:\Program Files\Java\jdk1.8.0_101, version: 1.8.0_101 (JDK)

☐ Specify alternative JDK/JRE base directory

9. Follow the wizard procedure to the end. In the process you will be asked to provide the installation path ("C:\Veryant" by default) and license keys. You can skip license activation and perform it later, as explained in Activate the License.

Quiet mode

The isCOBOL UDBC setup supports the msi quiet mode. Settings can be driven with a response file.

A response file is a text file with name-value pairs that represent installer variables.

A response file is generated automatically after an installation is finished. The generated response file is found in the *.install4j* directory of the isCOBOL UDBC and is named *response.varfile*.

When an installer is executed, it checks whether a file with the same name and the *.varfile* extension can be found in the same directory and loads that file as the response file. For example, if an installer is named *foo_setup.msi* on Windows, the response file next to it has to be named *foo_setup.varfile*.

For more information about msi setups and their command line options, see Microsoft Standard Installer Command-Line Options.

Linux, FreeBSD, Mac OSX and SunOS

1. If you haven't already done so, Download and install the Java Runtime Environment (JRE).
2. Go to "https://support.veryant.com".
3. Sign in with your User ID and Password.
4. Click on the "Download Software" link.
5. Scroll down, and select the appropriate .tar.gz file for the product and platform you require.
6. Extract all contents of the archive. For example,
   on Linux 32 bit:

```
gunzip isCOBOL_UDBC_2023_R1_*_Linux.32.i586.tar.gz
tar -xvf isCOBOL_UDBC_2023_R1_*_Linux.32.i586.tar
```

on Linux 64 bit:

```
gunzip isCOBOL_UDBC_2023_R1_*_Linux.64.x86_64.tar.gz
tar -xvf isCOBOL_UDBC_2023_R1_*_Linux.64.x86_64.tar
```

on FreeBSD:

```
gunzip isCOBOL_UDBC_2023_R1_*_FreeBSD.64.tar.gz
tar -xvf isCOBOL_UDBC_2023_R1_*_FreeBSD.64.tar
```

on Mac OSX:

```
gunzip isCOBOL_UDBC_2023_R1_*_MacOSX.64.x86_64.tar.gz
tar -xvf isCOBOL_UDBC_2023_R1_*_MacOSX.64.x86_64.tar
```

on SunOS:

```
gunzip isCOBOL_UDBC_2023_R1_*_SunOS.64.tar.gz
tar -xvf isCOBOL_UDBC_2023_R1_*_SunOS.64.tar
```

Other Unix

isCOBOL UDBC is available only for the following UNIX platforms:

- Linux 32 bit
- Linux 64 bit
- FreeBSD
- Mac OSX 64 bit
- SunOS

No other UNIX porting is available.

Distribution Files

For information on a specific distribution file, please see the README file installed with the product.

Activate the License

If you provided license keys during the installation, on Windows, you should skip reading this chapter.

isCOBOL UDBC looks for the following configuration property for the license key:

```
iscobol.udbc.license.2023=<license_key>
```

The key should be stored in one of the following files (if they exist):

Windows

1. \etc\iscobol.properties in the drive where the working directory is
2. C:\Users\<username>\iscobol.properties (the setup wizard saves licenses here, if you don't skip activation)
3. iscobol.properties found in the Java Classpath
4. a custom configuration file passed on the command line
5. %ISCOBOL%\iscobol.properties

Unix/Linux

1. /etc/iscobol.properties
2. $HOME/iscobol.properties
3. iscobol.properties found in the Java Classpath
4. a custom configuration file passed on the command line
5. $ISCOBOL/iscobol.properties

NOTE - Files are listed in the order they're processed. If the license key appears in more than one of the above files, then the last occurrence is used.

Testing the product using sample data

File Server startup

isCOBOL UDBC requires isCOBOL File Server to be up and running.

The isCOBOL File Server is provided along with isCOBOL.

Assuming that you have correctly installed isCOBOL and its thin client, you can start the File Server with the following command:

```
iscserver -fs
```

This guide assumes that you're running isCOBOL UDBC on the same machine as the isCOBOL File Server.

Sample data files and EFD dictionaries are installed with isCOBOL under *sample\as\fs\odbc*. Data files have to be created using the batch file *create.bat* available in that folder.

Database configuration

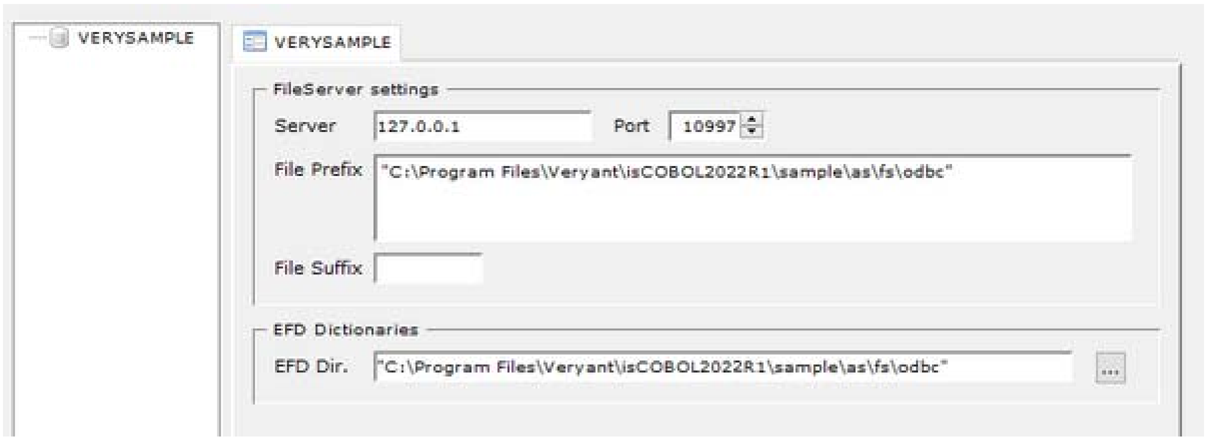
Windows

Run the UDBC Configuration Tool using the link available in Start -> Programs -> isCOBOL UDBC 2023R1

Create a new database in the following way:

1. Click on *File / New Database*
2. Type the desired name (e.g. "VERYSAMPLE")
3. Press Enter to confirm

Complete the mandatory fields as follows:



Server 127.0.0.1 Port 10997
File Prefix "C:\Program Files\Veryant\isCOBOL2022R1\sample\as\fs\odbc"
File Suffix
EFD Dir. "C:\Program Files\Veryant\isCOBOL2022R1\sample\as\fs\odbc"

Click on the *Save* button.

Linux/Unix

A sample configuration file is provided with the product. You will find a file named *udbc.ini* in the folder vUDBC2023R1/etc.

Copy that file either under /etc or in the user home directory.

Edit it and ensure that the entries EfdDirectory and FilePrefix point to the proper directory. The directory is *$ISCOBOL/sample/as/fs/odbc* and contains both data files and EFD dictionaries. Data files have to be created using the script *create.sh* available in that folder.

If a graphical desktop is available, you can manage the ini file with the following graphical utility:

```
/opt/vUDBC2023R1/bin/vudbccfg
```

Thin Client and Code Prefix:

VUDBCCFG can be used in thin client environment as well. Use this command to start it:

```
iscclient -hostname <server-ip> -port <server-port> -utility COBVUDBCCFG
```

The utility will configure databases on the server machine.

First query on isCOBOL UDBC

Windows

1. Run the UDBC Server using the link available in Start -> Programs -> isCOBOL UDBC 2023R1

A correct startup will produce a similar output:

```
Veryant VDBC 2023R1  Server console
Starting service:
 C:\Veryant\isCOBOL_UDBC2023R1\bin\vsqld.exe  6789
 on port 6789 ... Succesful, Mainpid is: 0
```

2. Run the Utility Shell using the link available in Start -> Programs -> isCOBOL UDBC 2023R1 -> Utility

In the Utility Shell, start the command-line SQL tool by issuing the command:

```
visql -cstring jdbc:veryant:127.0.0.1:6789:VERYSAMPLE -u admin
```

Where VERYSAMPLE is the name of the database.

You will be prompted for a password, type ‘admin’.

Type a query to select data from one of the archives available in the database:

```
select * from file1
GO
```

Check the result.

Linux/Unix

1. Run the UDBC Server with the command:

```
/opt/isCOBOL_UDBC2023R1/bin/vudbcserver
```

A correct startup will produce a similar output:

```
isCOBOL UDBC 2023 R1 Server console
Starting service:
 /opt/isCOBOL_UDBC2023R1/bin/vsqld  6789
 on port 6789 ... Succesful, Mainpid is: 0
```

2. Run the command-line SQL tool by issuing the command:

```
/opt/isCOBOL_UDBC2023R1/bin/visql -cstring jdbc:veryant:127.0.0.1:6789:VERYSAMPLE -u
   admin
```

Where VERYSAMPLE is the name of the database.

You will be prompted for a password, type ‘admin’.

Type a query to select data from one of the archives available in the database:

```
select * from file1
GO
```

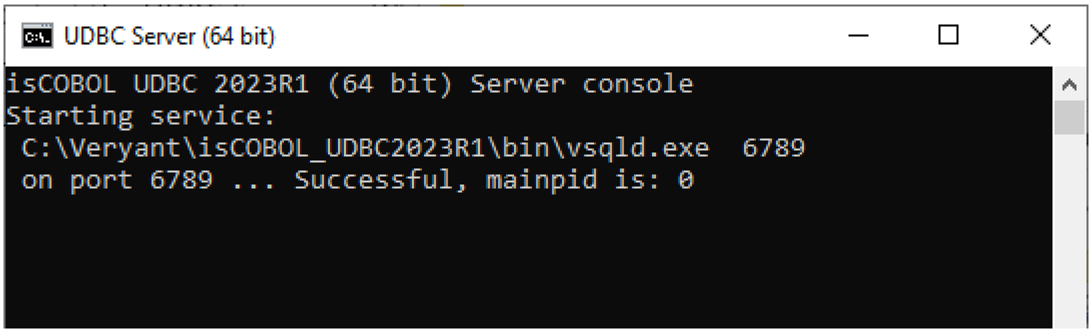
Check the result.

UDBC SQL Server Engine

The UDBC Server is responsible for managing connections from SQL clients and communicating with the isCOBOL File Server in order to access data files. The UDBC Server retrieve necessary information from the Database Configuration.

To start the UDBC Server on Windows select:

```
Start -> Programs -> isCOBOL UDBC 2023R1 -> UDBC Server
```



```
isCOBOL UDBC 2023R1 (64 bit) Server console
Starting service:
 C:\Veryant\isCOBOL_UDBC2023R1\bin\vsqld.exe  6789
 on port 6789 ... Successful, mainpid is: 0
```

To start the UDBC Server on Linux/Unix, run:

```
$VUDBC_HOME/bin/vudbcserver
```

Database Configuration

An isCOBOL UDBC database is a resource that allows you to perform SQL queries on COBOL data files through a File Server service.

You can have one or more databases pointing to different File Servers or to the same File Server with different settings.

On Windows machines databases are described in the Windows registry, in the key *HKEY_CURRENT_USER\Software\Veryant\UDBC*. Each subkey identifies a database and subkey values are the database entries.

On Unix machines instead databases are described in an ini file that is the first applicable found in:

1. /etc/udbc.ini
2. $home/.udbc.ini

The ini file uses the following syntax:

```
[DATABASE-NAME]
entry=value
...
entry=value
```

Note - for your convenience the graphical UDBC Configuration Tool is provided.

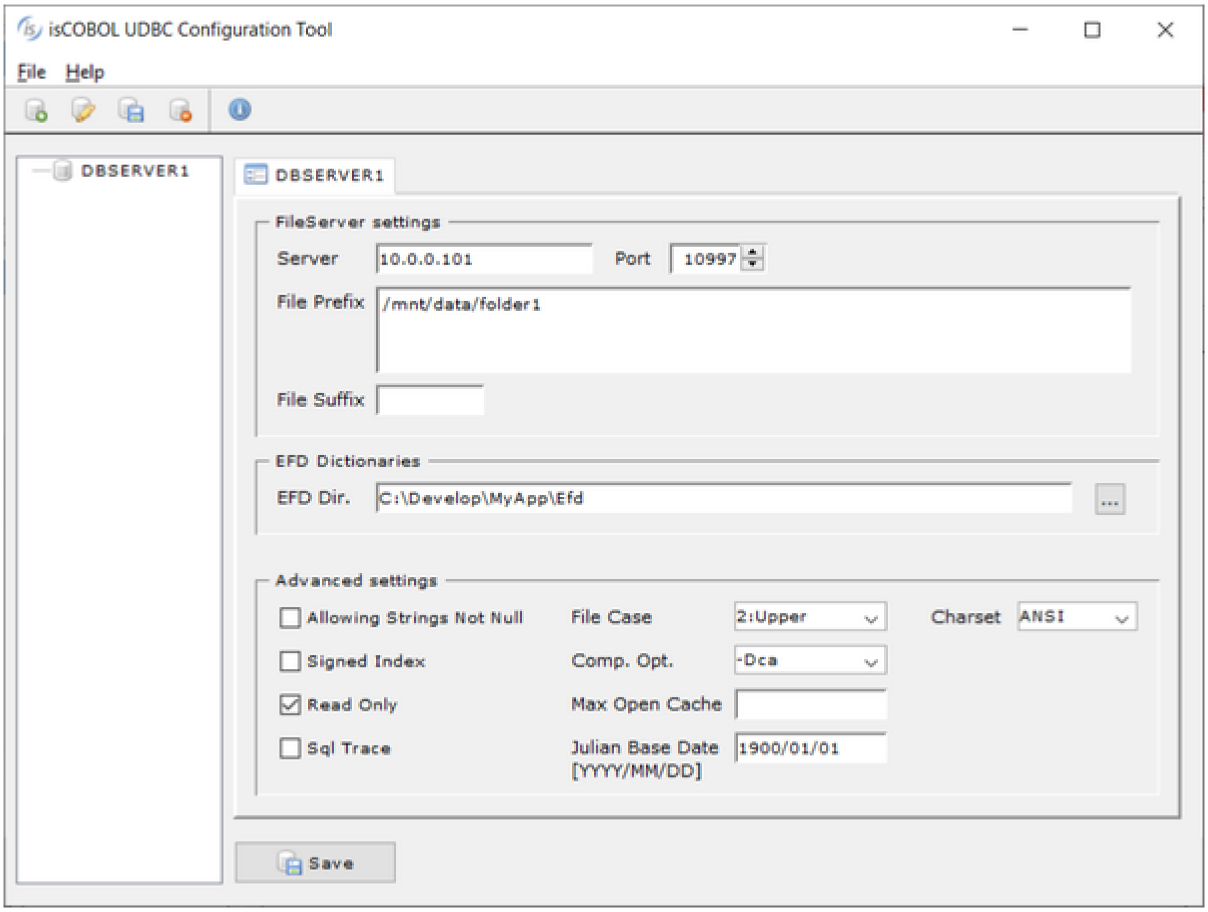
The following table describes all the available entries:

Entry	Meaning	Default if not set
Host	IP or name of the server where the isCOBOL File Server is listening	127.0.0.1
Port	port where the listening isCOBOL File Server	10997
EfdDirectory	folder with EFD dictionaries on the server where UDBC Server is started. Use quotes to delimit paths with spaces.	
FilePrefix	list of directories where files are located on the server where the isCOBOL File Server is listening. Use the server path separator to separate multiple paths. Use quotes to delimit paths with spaces.	
FileSuffix	suffix of data files, if used.	
FileCase	case of file names on the server, set to one of the following values: 0 (unchanged) 1 (lower case) 2 (upper case)	0
CblFlags	data compatibility flags, set to one of the following values: -Dca (Acucobol) -Dci (IBM) -Dcm (Micro Focus) -Dcn (same as -Dci, but the positive COMP-3 items use x0B as the positive sign value instead of x0C)	-Dca
CharSet	server charset, set to one of the following values: ANSI OEM	ANSI
JulianBaseDate	base date to calculate julian dates (a julian date is the number of days between the base date and the current date)	1900/01/01
SignedIndex	support signed data-items in keys. Possible values are: Yes No negative values in signed indexes might cause wrong data ordering so you should set this field to Yes only if you have signed data-items in file keys but there are no negative values	No
StringsNotNull	convert spaces and low values to NULL and vice versa. Possible values are: Yes No	No
MaxOpenCache	number of file definitions cached by UDBC. If the file definition is cached, then UDBC doesn't need to read the EFD dictionary again. It speeds up performance but avoids modifications to the EFD to be active until the next reboot of the UDBC Server	0
ReadOnly	avoid data modification through SQL. Possible values are: Yes No	Yes

SqlTrace	log SQL activity to the file *veryantUDBC.log* in the temp directory. On Linux/Unix */tmp* is used. On Windows, the GetTempPath function is called; this function checks for the existence of environment variables in the following order and uses the first path found: The path specified by the TMP environment variable. The path specified by the TEMP environment variable. The path specified by the USERPROFILE environment variable. The Windows directory. Possible values are: Yes No	No

UDBC Configuration Tool

isCOBOL UDBC is provided with a graphical utility that allows you to manage databases.



Server: 10.0.0.101
Port: 10997
File Prefix: /mnt/data/folder1
File Suffix:
EFD Dir.: C:\Develop\MyApp\Efd
Allowing Strings Not Null: ☐
Signed Index: ☐
Read Only: ☑
Sql Trace: ☐
File Case: 2:Upper
Comp. Opt.: -Dca
Max Open Cache:
Julian Base Date [YYYY/MM/DD]: 1900/01/01
Charset: ANSI

- Click on the *New Database* button in order to create a new database. Input a name of your choice and then compile the fields on the right providing File Server coordinates and the EFDs dictionaries location
- Click on the *Rename Database* button or double click on database name in the tree in order to rename the selected database.
- Click on the *Delete Database* button or press DEL in order to remove the selected database.

- Click on the *Save* button to save the latest changes you made on database settings.

The utility updates the Windows Registry or *udbc.ini* depending on the system when you run it.

VISQL

isCOBOL UDBC is provided with a command-line SQL tool. You can start it with the command:

```
visql -cstring jdbc:veryant:<udbc-server-ip>:<udbc-server-port>:<database> -u <user>
    -p <password>
```

Where *udbc-server-ip* and *udbc-server-port* are the coordinates to reach a listening UDBC Server on the network. By default UDBC Server listens for connections on port 6789.

user and *password* are checked only if the File Server was started having *iscobol.as.authentication=2* in the configuration.

Once connected, you can input queries and see the result. Each query must be closed by the GO command, issued on a separate line, for example:

```
SELECT * FROM file1
GO
```

Command line options

VISQL provides the following command line options:

-c	Specifies the command terminator. The default is "go".
-cstring	Specifies the connection string to use.
-debug	Prints to stdout (System.out) debugging information.
-delimiter	Specifies the delimiter. The default is " | ".
-input	Specifies a file name to read commands from.
-left	Left justify the output.
-noheader	Do not print any header columns.
-nonull	Print the empty string instead of the word "NULL" for null value.
-password / -p	Specifies the user name to log into a database server with.
-pf	Specifies the name of a file that contains the password to log into a database server with. The first line of file should contain the password and nothing else.
-query	Specifies an optional single query to run instead of interacting with the command line or a file. Note that the command must include a command terminator or the command will hang.
-spacer	Changes the spacer between columns from a single space to the first character of the argument.
-trim	Trim the data output. This is useful when specifying a delimiter.

-user / -u	Specifies a user name to log into a database server with.
-w	Specifies the maximum field width for a column. The default is to output the full width of the column.

Options and their values are separated by a space. For example, the following command runs a query directly telling that the command terminator is ";" instead of "GO".

```
visql -cstring "jdbc:veryant:localhost:6789:DBSERVER1" -u admin -p admin -c ";" -query
"select * from file1;"
```

ODBC Driver for UDBC SQL Server

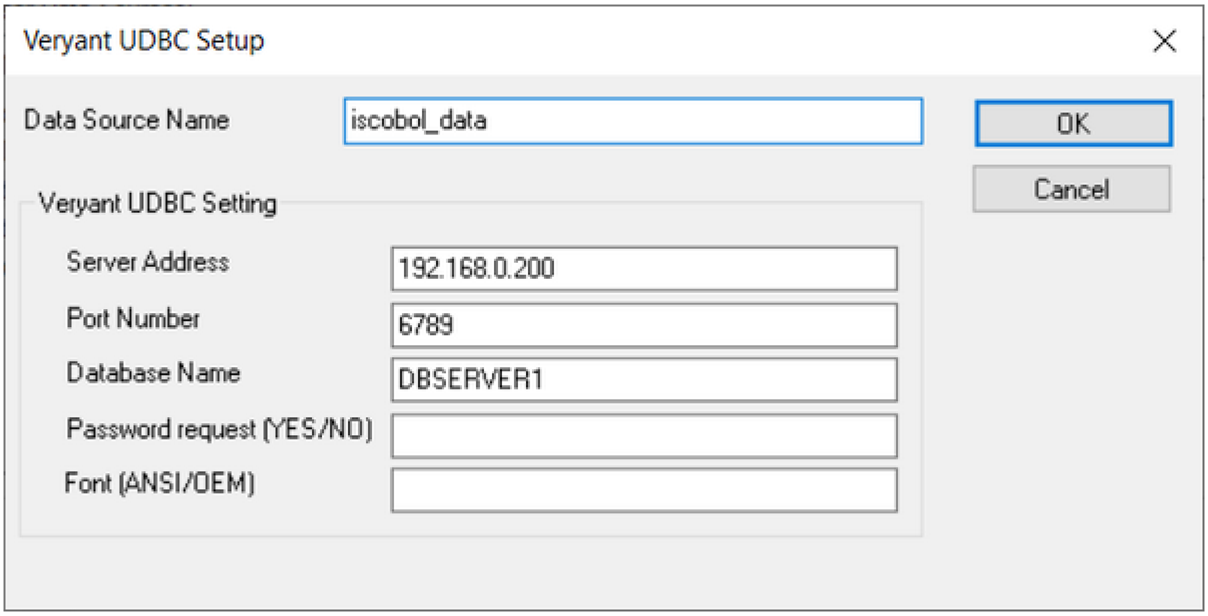
isCOBOL UDBC is provided with an ODBC driver that is installed along with the product on the Windows platform

After the installation, you can create a DSN by following these steps:

1. Open Control Panel, Administrative Tools, ODBC Data Sources
2. Switch between User and System pages depending on the type of DSN you wish to create.
3. Click on the "Add" button
4. Choose "Veryant UDBC Driver" from the list of drivers

Note - if you're running the 32 bit UDBC on a 64 bit Windows, you have to open the 32 bit ODBC manager.

The following panel will appear



Data Source Name: iscobol_data

Server Address: 192.168.0.200

Port Number: 6789

Database Name: DBSERVER1

Password request (YES/NO):

Font (ANSI/OEM):

5. Compile the fields as follows:

Data Source Name	a name of your choice, it will be used by ODBC clients to connect
Server Address	IP or name of the computer where UDBC Server is started and listening

Port Number	port where UDBC Server is listening (default is 6789)
Database Name	name of the UDBC database you want to connect
Password request	reserved for future use, can be left empty
Font	charset. Possible values are "ANSI" and "OEM". If empty, "ANSI" is assumed

JDBC Driver for UDBC SQL Server

isCOBOL UDBC is provided with a JDBC driver included in the library *vjdbc.jar*.

The library must be added to the Java Class Path along with the isCOBOL runtime library (*iscobol.jar*).

The following snippet shows how to connect to a UDBC database through JDBC. The code allows you to connect to a database called "VERYSAMPLE" on the localhost on the default port with user "admin" and password "admin":

```
Class.forName ("com.veryant.jdbc.VerySQLDriver");
Connection conn = DriverManager.getConnection
    ("jdbc:veryant:127.0.0.1:6789:VERYSAMPLE", "admin", "admin");
```

The VISQL utility accesses isCOBOL UDBC via JDBC.

Troubleshooting

The following table describes the most common errors you might encounter using isCOBOL UDBC.

Error Code	**SQL State**	**Description**	**Cause**
0	08005	IIOP: Communication problem(s) [java.net.ConnectException: Connection refused: connect]	The UDBC Server is down or unreachable
67	S0002	[Veryant][VeryantUDBC driver]Base table or view not found:<*tablename*>	The EFD dictionary of one or more tables listed in the SQL query has not been found by the UDBC Server
81	HY000	[Veryant][VeryantUDBC driver]General errorS1000: Error writing table (errno =100)	Unique key violation
84	HY000	[Veryant][VeryantUDBC driver]General errorS1000: Error processing table (errno =101)	The physical file has not been found by the isCOBOL File Server on the server
144	08001	[Veryant][VeryantUDBC driver]Unable to establish connection: Cannot connect to isCOBOL File Server: err #147. Error on connection 10061,10061	The isCOBOL File Server is down or unreachable
147	S1000	[Veryant][VeryantUDBC driver]Error processing table	Mismatch between physical file logical-info and the info included in the EFD dictionary