



Trusted RUBIX™

Version 6

ODBC Guide

Revision 7

RELATIONAL DATABASE MANAGEMENT SYSTEM

Infosystems Technology, Inc.

4 Professional Dr - Suite 118

Gaithersburg, MD 20879

TEL +1-202-412-0152

© 1981, 2014 Infosystems Technology, Inc. (ITI). All rights reserved. Unpublished work. Commercial computer software and software documentation: Government users are subject to ITI's standard license agreement per DFARS 227.7203-3 or, in non-DoD agencies where such protection is unavailable, to "restricted rights" under applicable FAR System clauses.

Infosystems Technology, Inc.
4 Professional Dr - Suite 118
Gaithersburg, MD 20879

THIS DOCUMENTATION CONTAINS CONFIDENTIAL INFORMATION AND TRADE SECRETS OF INFOSYSTEMS TECHNOLOGY, INC. USE, DISCLOSURE, OR REPRODUCTION IS PROHIBITED WITHOUT THE PRIOR EXPRESS WRITTEN PERMISSION OF INFOSYSTEMS TECHNOLOGY, INC. FOR FULL DETAILS OF THE TERMS AND CONDITIONS FOR USING THE SOFTWARE, PLEASE REFER TO THE ITI-TRUSTED RUBIX USER LICENSE AGREEMENT.

The information in this document is subject to change without notice and should not be construed as a commitment by ITI.

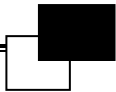
Infosystems Technology, Inc. assumes no responsibility for any errors that may appear in this document.

RUBIX® is a trademark of Infosystems Technology, Inc.

UNIX® is a trademark of The Open Group.

Microsoft® is a trademark of the Microsoft Corporation.

Printed in U.S.A.



OVERVIEW 1

INSTALLATION 1

 LINUX/UNIX.....1

ODBC Manager Installation1

Trusted RUBIX ODBC Driver Installation2

 MICROSOFT WINDOWS.....2

ODBC Manager Installation2

Microsoft Visual C++ 2008 Redistributable Package Installation3

Trusted RUBIX ODBC Driver Installation3

CONFIGURATION OF THE ODBC MANAGER 3

 LINUX/UNIX.....3

Driver Configuration3

Data Source Configuration4

 MICROSOFT WINDOWS.....5

Data Source Configuration5

EXECUTING APPLICATIONS USING THE ODBC MANAGER..... 7

BUILDING APPLICATIONS THAT USE THE ODBC MANAGER 8

 LINUX/UNIX.....8

BUILDING STAND-ALONE ODBC APPLICATIONS 8

 LINUX/UNIX.....8

ODBC SPECIFICATION COMPLIANCE..... 9

Overview

This document describes the installation, configuration, and use of the Trusted RUBIX ODBC driver. It applies to all UNIX/Linux platforms for which a Trusted RUBIX ODBC driver exists.

Open Database Connectivity (ODBC) applications are custom programs that may access standards compliant DBMS's. ODBC is Microsoft's adaptation of the X Open CLI standard. The following website gives details of its use, including the API specification:

[http://msdn.microsoft.com/en-us/library/ms710252\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms710252(VS.85).aspx)

An ODBC application, ODBC driver, and ODBC manager interoperate to allow a single ODBC compliant application to access database services through any ODBC compliant driver without recompiling. In general, this is accomplished by the application linking with libraries provided by the ODBC manager. These libraries present the C language, ODBC standard API to the application. At runtime the application specifies which specific driver it needs to use. The ODBC manager then dynamically loads the specified driver libraries and maps each API call to the API provided by the specific ODBC driver. The ODBC driver then maps the API calls into interactions with the DBMS backend, in our case the Trusted RUBIX Server.

Because ODBC driver libraries are required on each machine that serves as an ODBC client and ODBC is a C language standard (i.e., compiled into machine code), there must be a set of drivers available and installed for each particular client machine. ODBC drivers are generally provided by the provider of the DBMS. The ODBC manager is generally provided by the provider of the OS or a third party.

A specific driver and a configuration (e.g., which database, user, and host) is pre-configured by the system administrator and given a string based name. This name is called the Data Source Name (DSN). The DSN is passed into the ODBC manager by the application during runtime to specify which database service it needs to use. The DSN is then used to retrieve other configuration and connection arguments.

Installation

Linux/Unix

ODBC MANAGER INSTALLATION

Trusted RUBIX is designed and tested to interoperate with the unixODBC ODBC manager. This is a popular, open source ODBC manager available on most UNIX/Linux systems. Your system may or may not have unixODBC installed by default. More information may be found on unixODBC at:

<http://www.unixodbc.org/>

The *yum install* operation may be used to download and install a package, as shown below. Note that using the *yum install* operation will ensure that all dependent packages are downloaded and installed as well as the target package.

```
yum install PACKAGE_NAME
```

Information about a package, including its installation status, may be found using the following command:

```
yum info PACKAGE_NAME
```

For each machine that will be used as a Trusted RUBIX ODBC client, install all of the following packages. Depending on the configuration of your operating system installation, some of these packages may already be installed; if so, you will receive an appropriate message during the package installation operation.

→ unixODBC

→ unixODBC-devel

TRUSTED RUBIX ODBC DRIVER INSTALLATION

For each platform that will be used **only** as a Trusted RUBIX ODBC client, install the ODBC client package from the *Clients* directory of your Trusted RUBIX installation disk as follows:

```
cd RPM_PACKAGE_FILE_DIR
```

```
rpm -ivh ./rubixdbms-odbc-6.0.x-x.xxx.xxx.rpm
```

Take note to install the ODBC client package for your particular client machine. Client packages are located in the *Clients* directory of your installation disk. Packages with the *fcXX* designation are for Fedora XX, packages with the *el5* designation are for RHEL5, and packages with the *el6* designation are for RHEL6. Note that the *rubixdbms-devel* package includes the ODBC libraries for a server machine, so only install the *rubixdbms-odbc* package on a server machine if the *rubixdbms-devel* package is not installed.

Microsoft Windows

ODBC MANAGER INSTALLATION

The Microsoft Windows operating systems has an ODBC Manager installed by default. It may be found in the Control Panel under "Administrative Tools" and is called "Data Sources (ODBC)." The Trusted RUBIX ODBC driver is a 32 bit driver. Therefore, you must use the 32 bit version of the ODBC Manager. On 32 bit versions of Microsoft Windows, this is default. On 64 bit versions of Microsoft Windows you must explicitly choose the 32 bit ODBC Manager. It may be found in the "C:\Windows\syswow64\odbcad32.exe" directory.

MICROSOFT VISUAL C++ 2008 REDISTRIBUTABLE PACKAGE INSTALLATION

The Microsoft Visual C++ 2008 Redistributable Package must be installed on any platform that will host the Trusted RUBIX ODBC Driver. Installation instructions and the package may be found at:

<http://www.microsoft.com/Downloads/details.aspx?familyid=A5C84275-3B97-4AB7-A40D-3802B2AF5FC2&displaylang=en>

TRUSTED RUBIX ODBC DRIVER INSTALLATION

To install the Trusted RUBIX ODBC driver first copy the installation file to your computer. The install file is named "TrustedRubixWinClient-x.x.x.msi". As the Administrator execute the file and follow the instructions presented. Once the install file has finished you must execute the driver configuration script. To do this navigate to the "Trusted RUBIX DBMS Client" in your start menu. Then, place the mouse over the "Configure Driver" item, right click and choose "Run as Administrator."

Configuration of the ODBC Manager

Linux/Unix

The ODBC manager needs configuration to allow it to use the Trusted RUBIX ODBC driver. First, the driver needs to be registered with the ODBC manager by editing the */etc/odbcinst.ini* file. Second, specific data sources must be configured by editing the */etc/odbc.ini* file (called a System DSN, available to all users) or the *HOMEDIR/odbc.ini* file (called a User DSN, available only to the user whose home directory is used).

DRIVER CONFIGURATION

To allow the ODBC manager to use the Trusted RUBIX ODBC driver, it must first be registered in the */etc/odbcinst.ini* file. To register the Trusted RUBIX ODBC driver, become the *root* user and edit the file using any text editor. Add the following lines to the *odbcinst.ini* file:

```
[Trusted RUBIX]
Description      = Trusted RUBIX ODBC driver for Linux
Driver           = /usr/lib/librxodbc.so
UsageCount      = 1
```

DATA SOURCE CONFIGURATION

An ODBC data source specifies an ODBC driver, log/tracing arguments, and all of the connection arguments needed to access the DBMS backend. The set of arguments recognized by Trusted RUBIX ODBC are:

- **Driver:** the name of the ODBC driver the data source will use. This must correspond to the section heading in the *odbcinst.ini* file for the driver. In our case this will always be *Trusted RUBIX*.
- **Database:** the database name the data source will access. Defaults to *default_database*.
- **Catalog:** The catalog name that will be set as current upon database connection. Defaults to *default_catalog*.
- **Host:** name or numerical IP of the remote database host. Defaults to *localhost*.
- **Port:** port number of the remote database host. Defaults to 4156.
- **User:** Linux user name or numerical UID from the database host machine to be used for the database session. For local connections only, this defaults to the current process's user name. For remote database hosts this must always be specified.
- **Group:** Linux group name or numerical GID from the database host machine to be used for the database session. Defaults to the default GID (as configured on the database host machine) for the user specified or implied by the User argument.
- **Password:** Password associated with the Linux user name as configured on the database host machine. For security considerations it is recommended that passwords be passed into the ODBC Connect functions (e.g. as part of the connection string to *SQLDriverConnect*) as opposed to storing them in DSN configuration files. If passwords are stored in DSN configuration files the file should be carefully isolated with the DAC/MAC security protections of the operating system as needed.
- **TraceFile:** Full path to a file used to log/trace all calls to the Trusted RUBIX ODBC driver. If the file does not exist it is created. Defaults to */tmp/rxodbc.log* file.
- **TraceMode:** Numerical trace mode used to determine how much information is logged. The value may be between zero and four inclusive. A value of zero will turn logging off. A value of one will log entry and exiting of ODBC interface functions. A value of two will log the entry and exiting of ODBC interface functions along with the parameters. A value of three will log the entry and exiting of ODBC interface functions, the parameters, and row data. A value of four will log the entry and exiting of ODBC interface functions, the parameters, row data, and debugging error messages. Defaults to zero.

The *minimum* set of DSN arguments required by Trusted RUBIX to make a local connection (client and server are on the same machine) are:

- Driver

The *minimum* set of DSN arguments required by Trusted RUBIX to make a remote connection (client and server are on different machines) are:

- Driver
- Host
- User
- Password

An ODBC data source is created by editing either the */etc/odbc.ini* or *HOMEDIR/.odbc.ini* file. The

former case is called a System DSN and is available to all users on the system. The latter is called a User DSN and is only available to the user whose home directory contains the *.odbc.ini* file. Multiple data sources may be created and modified as requirements dictate. Each data source must have a unique name.

The entry in the *odbc.ini* or *.odbc.ini* file contains a section heading that contains the data source name (DSN) and lines that contain the ODBC configuration and connection arguments. For example, the following *odbc.ini* entry contains all of the possible Trusted RUBIX ODBC arguments:

```
[mysdn]
Driver      = Trusted RUBIX
Host        = my.rxhost.com
Port        = 4156
Database    = MyDB
Catalog     = MyCatalog
User        = user1
Group       = group1
Password    = MyPassword
TraceFile   = /tmp/myrxodbc.log
TraceMode   = 2
```

In the above example the data source name (DSN) is *mysdn* and this value would be passed into an application. The DSN would then be used by the ODBC manager and the ODBC driver to extract the remaining arguments.

At a *minimum*, an entry in an *odbc.ini* or *.odbc.ini* file needs to have a DSN name and a Driver argument. In this case any additional required arguments must be passed into the application and from there to the driver.

Microsoft Windows

The Microsoft Windows operating systems ODBC Manager may be found in the Control Panel under "Administrative Tools" and is called "Data Sources (ODBC)." The Trusted RUBIX ODBC driver is a 32 bit driver. Therefore, you must use the 32 bit version of the ODBC Manager. On 32 bit versions of Microsoft Windows, this is default. On 64 bit versions of Microsoft Windows you must explicitly choose the 32 bit ODBC Manager. It may be found in the "C:\Windows\syswow64\odbcad32.exe" directory.

DATA SOURCE CONFIGURATION

An ODBC data source specifies an ODBC driver, log/tracing arguments, and all of the connection arguments needed to access the DBMS backend. The set of arguments recognized by Trusted RUBIX ODBC Windows driver are:

- **Database:** The database name the data source will access. Defaults to *default_database*.
- **Catalog:** The catalog name that will be set as current upon database connection. Defaults to *default_catalog*.
- **Host:** Name or numerical IP of the remote database host. This must always be specified.
- **Port:** Port number of the remote database host. Defaults to 4156.
- **User:** Linux user name or numerical UID from the database host machine to be used for the database session. This must always be specified.
- **Group:** Linux group name or numerical GID from the database host machine to be used for the

database session. Defaults to the default GID (as configured on the database host machine) for the user specified by the User argument.

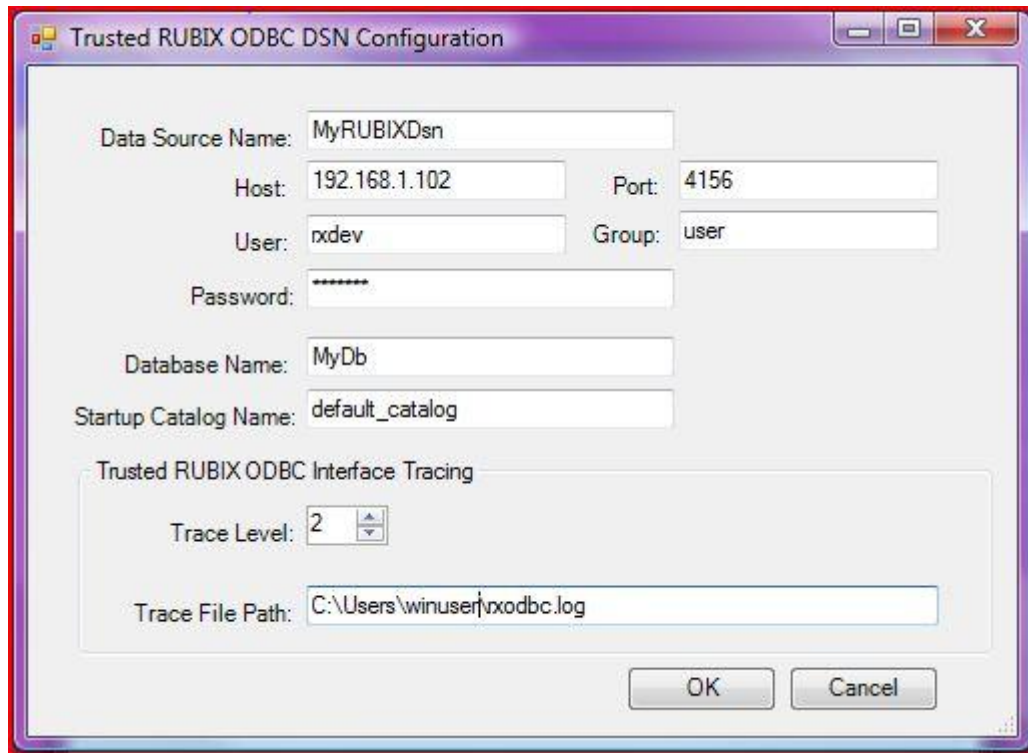
- **Password:** Password associated with the Linux user name as configured on the database host machine. This must always be specified.
- **TraceFile:** Full path to a file used to log/trace all calls to the Trusted RUBIX ODBC driver. If the file does not exist it is created. Defaults to *C:\rxodbc.log*. If the user is not operating with Administrator privileges the log file may not be able to be created in the *C:* directory. In this case the *TraceFile* path should be explicitly set to a directory that is writable by the user.
- **TraceMode:** Numerical trace mode used to determine how much information is logged. The value may be between zero and four inclusive. A value of zero will turn logging off. A value of one will log entry and exiting of ODBC interface functions. A value of two will log the entry and exiting of ODBC interface functions along with the parameters. A value of three will log the entry and exiting of ODBC interface functions, the parameters, and row data. A value of four will log the entry and exiting of ODBC interface functions, the parameters, row data, and debugging error messages. Defaults to zero.

The *minimum* set of DSN arguments required by the Trusted RUBIX ODBC Windows driver to make a connection are (note that all connections from a Windows platform is remote):

- Driver
- Host
- User
- Password

An ODBC data source is created by utilizing the Windows ODBC Manager. Once started choose either "User DSN" or "System DSN" and then "Add." Then, choose the "Trusted RUBIX DBMS" driver. A configuration window for you to enter your DSN information. The DSN may be modified through the ODBC manager by choosing a created DSN and selecting "Configure." Lastly, it may be removed by choosing "Remove."

A typical DSN configuration window is shown below:



Executing Applications Using the ODBC Manager

In general an ODBC application will have some method to specify, at a minimum, a DSN name. It should also have some method to specify a password. How this is accomplished is dependent upon the application. For instance, this may be done through a configuration file, a runtime pop-up window, or a command line argument. Please consult the documentation of your ODBC compliant application for details.

Depending on the particular ODBC API function the application uses to connect to a database backend, it may be capable of passing in extended arguments. The older API function allows only the DSN name, user name, and password to be specified. The newer API function allows any combinations of arguments to be specified. In the latter case a configuration string may be used that looks like:

```
DSN=dsnname;UID=username;PWD=password;Database=dbname;Host=hostname;
```

Any valid Trusted RUBIX ODBC connection argument may be used in this configuration string. All arguments passed into the application take precedence over identical arguments in the *odbc.ini* or *.odbc.ini* files. The UID argument used by the new API function (as used in the argument string above) is identical to the User argument in an *odbc.ini* file. That is, they both specify the user name for the DBMS session. If two or more identical arguments are specified in the same configuration string, the first occurrence is chosen.

Building Applications that Use the ODBC Manager

Linux/Unix

To build an ODBC application that will use the ODBC manager to connect to a Trusted RUBIX database, simply link your application with the *unixODBC* libraries as follows:

```
gcc MyODBCApp.c -o MyODBCApp -lodbcc
```

Your application will be capable of connection to any ODBC compliant DBMS, including Trusted RUBIX, by passing in the appropriate DSN name during the connection phase.

Building Stand-Alone ODBC Applications

Linux/Unix

To build an ODBC application that will directly use the Trusted RUBIX ODBC driver and bypass the ODBC manager, simply link your application with the Trusted RUBIX libraries as follows:

```
gcc MyODBCApp.c -o MyODBCApp -lrxodbc
```

Note that the behavior may be different than if you linked into the ODBC manager. This is because the ODBC manager performs certain functions of its own and may modify API calls before passing them on to the underlying driver. Your application will only be capable of connecting to Trusted RUBIX DSN's. Note that you will still need to install the *unixODBC* package as Trusted RUBIX will use it to search the *odbc.ini* and *.odbc.ini* files.

ODBC Specification Compliance

The following table provides the ODBC functions implemented by the Trusted RUBIX ODBC driver.

SQLAllocHandle	SQLBindCol	SQLBindParameter
SQLBrowseConnect	SQLCancel	SQLCloseCursor
SQLColAttribute	SQLColumnPrivileges	SQLColumns
SQLConnect	SQLCopyDesc	SQLDescribeCol
SQLDescribeParam	SQLDisconnect	SQLDriverConnect
SQLEndTran	SQLExecDirect	SQLExecute
SQLFetch	SQLFetchScroll	SQLForeignKeys
SQLFreeHandle	SQLFreeStmt	SQLGetConnectAttr
SQLGetCursorName	SQLGetData	SQLGetDescField
SQLGetDescRec	SQLGetDiagField	SQLGetDiagRec
SQLGetEnvAttr	SQLGetFunctions	SQLGetInfo
SQLGetStmtAttr	SQLGetTypeInfo	SQLNativeSql
SQLNumParams	SQLNumResultCols	SQLParamData
SQLPrepare	SQLPrimaryKeys	SQLProcedureColumns
SQLProcedures	SQLPutData	SQLRowCount
SQLSetConnectAttr	SQLSetCursorName	SQLSetDescField
SQLSetDescRec	SQLSetEnvAttr	SQLSetPos
SQLSetStmtAttr	SQLSpecialColumns	SQLStatistics
SQLTablePrivileges	SQLTables	

- The Trusted RUBIX Driver supports *Forward-Only* and *Static* cursor types.
- Bookmarks and asynchronous operations are not supported at this time.
- Multi-threaded concurrency (synchronized by Connection Handles) is supported.
- Bulk operations are supported only by binding arrays of parameters to statements. For instance, binding an array of parameters to an INSERT statement will result in one execution of the INSERT statement for each element in the parameter array. The entire operation including the parameter array will be sent to the server and executed as one unit and will receive one response from the server, resulting in a significant performance gain over executing individual INSERT statements.
- All ODBC SQL data types are supported except SQL_WCHAR, SQL_WVARCHAR, SQL_WLONGVARCHAR, SQL_BIGINT, SQL_TYPE_UTCDATETIME, SQL_TYPE_UTCTIME, and SQL_GUID.
- All ODBC C data types are supported except SQL_C_WCHAR, SQL_C_SBIGINT, SQL_C_UBIGINT, SQL_C_BOOKMARK, SQL_C_VARBOOKMARK, and SQL_C_GUID.