

Crystal Reports

Open Database Connectivity (ODBC)

Overview

This document focuses on Open Database Connectivity (ODBC) technology and how it interacts with Crystal Reports (CR). This document is mostly theory but it is important to understand the concepts covered as they apply to all ODBC data sources used with Crystal Reports. If you are using Crystal Reports XI and later, troubleshooting suggestions are available [here](#).

Contents

INTRODUCTION	3
WHAT IS ODBC?	3
NATIVE VS ODBC DRIVERS	3
NATIVE VS ODBC ACCESS	4
Advantages	4
Disadvantages	4
MDAC	5
ODBC ADMINISTRATOR	5
User DSN	6
System DSN	6
File DSN	6
Drivers	6
Tracing	6
Connection Pooling	6
About	6
ODBC REQUIREMENTS FOR CRYSTAL REPORTS	8
FIVE LAYERS OF ODBC CONNECTIVITY	8
Crystal Reports Layer	8
ODBC Translation Layer	8
ODBC Layer	9
DBMS Translation (ODBC data source) Layer	9
The Database Layer	10
P2SODBC.DLL	12
ODBC BACK END DRIVERS	13
CLIENT/SERVER COMMUNICATION LAYER	13

INSTALLING ODBC DRIVERS 14
ODBC CONNECTION MODEL FOR CLIENT/SERVER DATABASES 14
FINDING MORE INFORMATION 15
CONTACTING CRYSTAL DECISIONS FOR TECHNICAL SUPPORT 15

Introduction

This document will explore the architecture of open database connectivity (ODBC) technology and how it interacts with Crystal Reports.

What is ODBC?

Open Database Connectivity (ODBC) is a standard developed by the Microsoft Corporation through which many different types of data can be accessed by a single application. An application need only communicate with one set of files (ODBC) to be able to work with any source of data that can be accessed by ODBC.

Native vs ODBC Drivers

There are hundreds of Database Management Systems (DBMS) available for personal computers. Normally a company that designs an application that accesses data, such as Crystal Reports, must develop a means for the application to communicate with each type of data that a customer might want to use. Crystal Reports does this in three ways:

- Crystal Reports ships with database drivers written for specific database types. These are called “native” drivers. They are used to connect directly to their specific database type, without going through the ODBC layer. When you create a new report in Crystal Reports 8 or later, native drivers are used when you choose a data source from either **Database Files** or **More Data Sources** in the **Data Explorer**. Native drivers are generally named “P2*.DLL”.
- Crystal Reports also ships with a driver that will access ODBC drivers. It is called “P2SODBC.DLL”. As long as a database installs an ODBC driver to access its data, Crystal Reports can connect to that database using P2SODBC.DLL.
- Crystal Reports also ships ODBC drivers written for specific database types. These drivers were designed specifically for Crystal Reports and the specified database. They will have a “CR” prefix in the ODBC Administrator. These ODBC drivers are generally named “CR*.DLL”.

Native vs ODBC Access

Advantages

Perhaps the biggest advantage to accessing data through ODBC is the ability to access a wide range of data with just one interface. Since most popular Database Management Systems now offer ODBC drivers, with more appearing every day, Crystal Reports can use any type of data you have. Because of the extreme flexibility built into ODBC, you can use the same report file with different ODBC data sources. For example, you might design a report using an Oracle data source, and later, if your company switches to Microsoft SQL Server, you can simply change the ODBC data source used by your report. The only requirement is that the new data source must have the same structure (tables and fields) that the original data source had (although table names can be different). See *How to change the ODBC data source accessed by a report* in the Crystal Reports on-line help.

Experienced SQL (Structured Query Language) programmers also benefit from the ODBC standard. Since Crystal Reports uses SQL to communicate with ODBC, SQL programmers and Database Administrators can view and edit the SQL statements sent to ODBC, controlling exactly how data is retrieved from the data source.

Finally, by using SQL pass-through technology to send an SQL statement to ODBC and retrieve an initial set of data, Crystal Reports offloads much of the data retrieval and sorting work on to the server system, freeing up local memory and resources for more important tasks. In addition, only the data specified by the SQL statement is returned to Crystal Reports, reducing network traffic and the use of network resources. By working more efficiently with the original data, Crystal Reports saves you time and effort, and lets you concentrate on the design process and other more important work.

Disadvantages

Because of the many layers involved in passing data through ODBC from a database to an application, ODBC data sources often take more time to return data than other directly accessed data sources. First, Crystal Reports must request some data. The request must be translated by the ODBC translation layer into a format that ODBC understands (a SQL statement). ODBC must determine where the requested data exists and pass the request on to the ODBC data source. For more information, refer to the section: *DBMS Translation (ODBC data source) Layer*. The data source must analyze the request and translate it again into a format that can be understood by the DBMS. This complex process not only takes time, but it can also fail at any of several possible levels.

In addition, ODBC data sources must be correctly configured and set up on each computer in the ODBC.INI and ODBCINST.INI files before they can be used. If you create a report on one system and try to open it on

another system that does not have the same ODBC data source set up, Crystal Reports will not be able to connect to the data.

When working with ODBC, you should also be aware that the SQL language used by ODBC is based on the standards set for the SQL language by the American National Standards Institute (ANSI). Some SQL-based DBMS applications, however, provide additional features to the SQL language that are specific to that DBMS. If your data uses features unique to your DBMS, ODBC will not be able to translate those features (though in many cases it will still retrieve most of the data).

MDAC

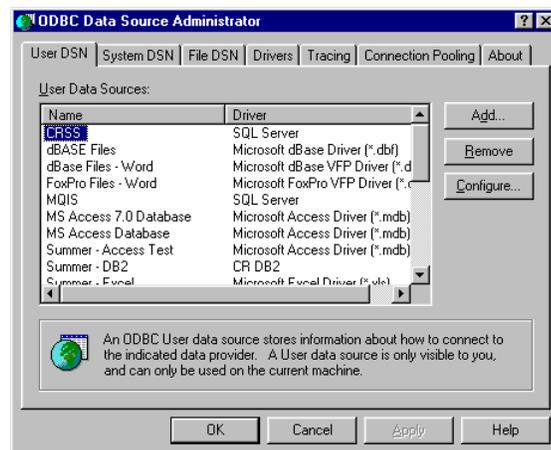
The Microsoft Data Access Components (MDAC) is a package of components integrated to provide a complete data access solution. By installing MDAC, you are effectively installing the ODBC layer and several Microsoft ODBC drivers.

For more information about the MDAC components, refer to the Microsoft site at: <http://www.microsoft.com/data/whatcom.htm>

ODBC Administrator

Once MDAC is installed, an ODBC Administrator will appear in your Windows Control Panel. You will use this administrator to create and configure your ODBC Data Source Names (DSN). An ODBC DSN must be configured for each ODBC driver that will be used to access data. ODBC DSNs must be configured on each computer that will be accessing data through ODBC. For certain drivers, a DSN must be configured for each database that will be accessed through ODBC.

When you launch the ODBC Administrator, you will see the following:



User DSN

This tab lists all User DSNs on the local computer. A User DSN is accessible by the current user logged on to the computer. That means if you create a User DSN and then log on to the computer as a different user, you will not see that User DSN.

System DSN

This tab lists all System DSNs on the local computer. A System DSN is accessible by all users logged on to the computer.

File DSN

This tab lists all File DSNs on the local computer. It allows connection information to be used repeatedly by a single user or shared among several users.

Drivers

This tab lists all ODBC drivers currently registered on the local computer. The list of drivers is retrieved from the following registry subkey:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBCINST.INI\ODBC Drivers
```

Tracing

This tab allows you to perform an ODBC Trace.

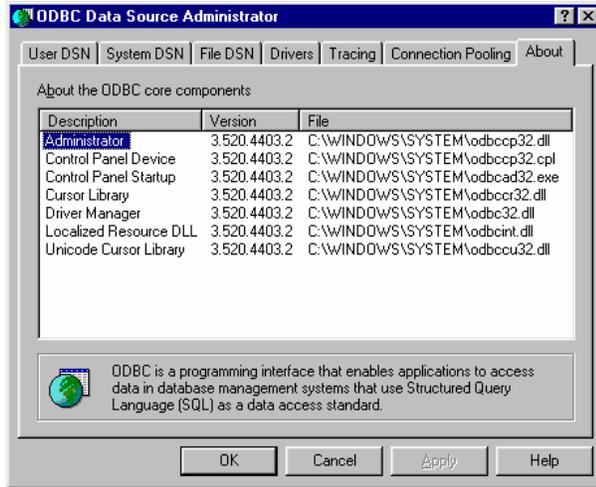
Connection Pooling

This tab is beyond the scope of this document.

About

This tab lists the files being loaded for the ODBC layer. Always ensure that all files being loaded are the same version and are loading from the same location. For a list of the ODBC component files, please read Knowledge Base article [c2001040](#). Quite often, you must reinstall the ODBC layer if there are several versions of ODBC on a computer. MDAC will reinstall the core component files of the ODBC layer.

A correct ODBC configuration would look similar to the following. Notice that all files are the same version and are being loaded from the same directory:



ODBC Requirements for Crystal Reports

In order for Crystal Reports to connect to a database through ODBC, there are some requirements that must be met:

1. The database client software must be installed on the client computer.
2. The client computer must be able to connect to the database server from its client software.
3. The client software's working directory (i.e. *C:\Orant\Bin* for Oracle) must be in the Windows search path.
4. The ODBC driver must be installed on the local computer.
5. A DSN for the ODBC driver must be configured in the ODBC Administrator.

Five Layers of ODBC Connectivity

The process by which Crystal Reports accesses data from an ODBC data source consists of five layers:

1. Crystal Reports Layer
2. ODBC Translation Layer
3. ODBC Layer
4. DBMS Translation (ODBC data source) Layer
5. Database Layer

By using the Structured Query Language (SQL), all five layers can conveniently pass data from the database to your report.

Crystal Reports Layer

When working with ODBC data, Crystal Reports generates an SQL statement that requests the appropriate data from ODBC. The powerful SQL generator built into Crystal Reports is designed to create an SQL statement that will let the ODBC data source or ODBC itself do as much of the report generation as possible, returning only the data needed to produce the report.

The SQL query that a report generates can be viewed in the Report Designer on the **Database** menu by clicking **Show SQL Query**.

ODBC Translation Layer

Crystal Reports uses the Dynamic Link Library P2SODBC.DLL (32 bit) or PDSODBC.DLL (16 bit) to communicate with ODBC. This file is unique to the Crystal Reports environment and provides your report with access

to any ODBC data source. This is the driver that actually passes data to and from ODBC.

ODBC Layer

ODBC is a set of several DLLs and INI files built into the Windows environment that act as a gateway through which database requests and data can pass. Any database file or format that is to be used via ODBC must be set up as an ODBC data source.

In 32-bit Windows, the core ODBC DLLs are:

- ODBC32.DLL
- ODBCCP32.DLL
- ODBCINT.DLL

ODBC uses the Windows Registry database to retrieve information about individual data sources.

ODBC uses the SQL language for all transactions between Crystal Reports and ODBC. Even if the database does not normally use SQL to create and work with tables, the ODBC driver provided by the database (the DBMS translation layer) must communicate with ODBC using SQL. For most users, this feature of ODBC is transparent, but more advanced users often take advantage of the features of the SQL language used by ODBC.

DBMS Translation (ODBC data source) Layer

This layer consists of one or more drivers provided by a DBMS that allow ODBC to communicate with the database. Crystal Reports ships with several ODBC drivers for many of the most common database formats.

Most DBMS applications that run on a Windows-based platform offer an ODBC driver. **Crystal Reports can connect to any database whatsoever if that database has an ODBC driver.**

When the DBMS provides an ODBC driver, it must register the driver with ODBC on the computer on which it has been installed. It does this by assigning a name to the driver and recording the filename in the ODBCINST.INI file. Usually, this step is handled automatically when the DBMS application is installed on the system. However, your network system or DBMS application may require that you register the ODBC driver manually with the ODBC Setup application.

Once an ODBC driver is registered, you need to establish an ODBC data source using that driver. You create the ODBC Data Source Name (DSN) in the ODBC Administrator. The ODBC data source is the object that you connect to when accessing data from Crystal Reports through ODBC. Data sources are recorded in the ODBC.INI file. The data source

keeps track of the DBMS translation files (ODBC drivers) and sometimes the database itself. An ODBC data source can specify just a database format, such as Oracle, Gupta, Sybase SQL, or Microsoft SQL Server. Some users, however, prefer to actually specify a particular database. In this case, the ODBC data source extends across both the DBMS translation layer and the database layer.

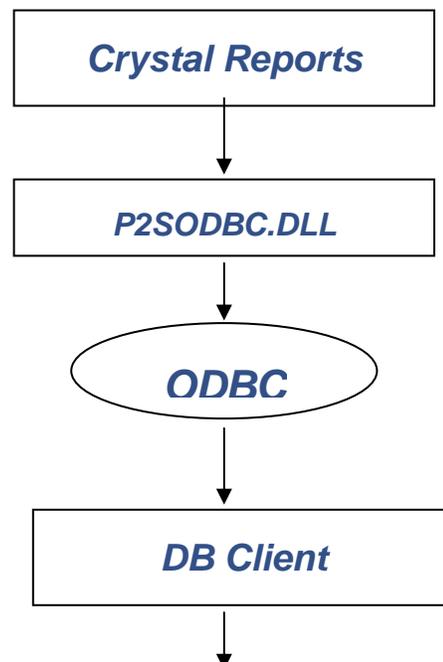
If you are using a client/server database, such as a SQL Server, the ODBC drivers communicate with the database server through the Database Communication Layer, the same layer that your database client uses to communicate with the database server.

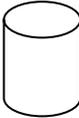
NOTE	IMPORTANT: ODBC drivers find their specific DBMS client files on the local computer mainly through key directories that the DBMS client software has installed in the search path. The important thing to remember is that a workstation client on a local PC must be able to connect to its server successfully.
-------------	--

For Crystal Reports and the ODBC drivers, it does not matter what kind of platform (i.e. UNIX, or Windows NT, or so on) your database server exists on. It is the DBMS client software that connects and communicates with the server; Crystal Reports and the drivers need only communicate with the DBMS client.

The Database Layer

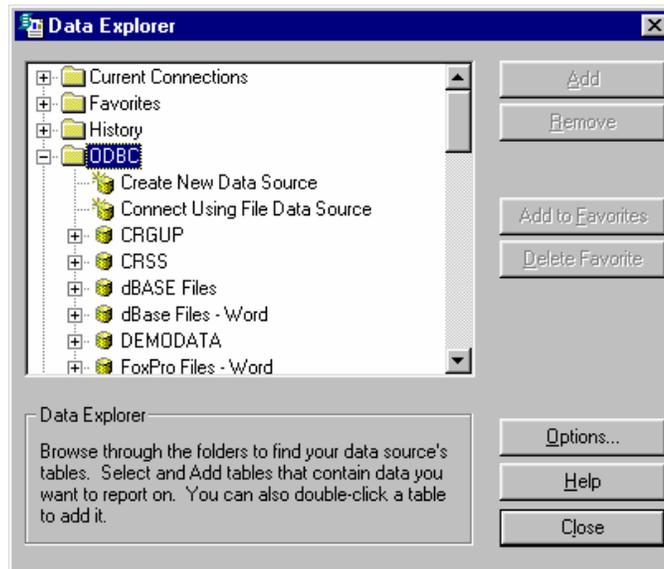
The database file which the ODBC data source refers to can be located anywhere on a system. Once the ODBC data source is set up, Crystal Reports does not need to know the actual location and format of the data. Thus, the database can have any format and can be located anywhere on a network, as long as ODBC can communicate with it through the ODBC data source.





P2SODBC.DLL

When you log on to an ODBC data source in Crystal Reports (on the **File** menu, click **Log On Server**), P2SODBC.DLL gets loaded. This is the front-end driver that ships with Crystal Reports. P2SODBC.DLL calls ODBC32.DLL, ODBCINT.DLL, and ODBC32.DLL to retrieve all available data sources that have been defined through Microsoft ODBC Administrator. All the ODBC data sources configured on the computer will be displayed in the Data Explorer window in Crystal Reports within the ODBC folder.



There are four Registry subkeys that reference the ODBC drivers and the ODBC data sources configured on your computer:

First subkey

HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBCINST.INI\ODBC Drivers

Lists all ODBC drivers installed on the computer

Second subkey

HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBC.INI

HKEY_CURRENT_USER\Software\ODBC\ODBC.INI

Lists all data sources configured on the computer

NOTE	System DSNs are available to all users on a computer and will appear in the HKEY_LOCAL_MACHINE registry path. User DSNs are available only to the user logged
-------------	---

	onto the computer and will appear in the HKEY_CURRENT_USER registry path. Each user may have different User DSNs defined.
--	---

Third subkey

HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBCINST.INI

Lists the ODBC driver being used by each data source and its location

Fourth subkey

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\Current Version\SharedDlls

References the directory in which the ODBC driver resides.

ODBC Back End Drivers

Once you select a data source from the list, an API call is made to the registry to find the back-end ODBC driver the data source is pointing to. That driver gets loaded only if it finds the required components provided by the DBMS client software. The back-end ODBC driver then takes control of the database connectivity to your server.

Next, a logon screen appears that allows you to log on to your server. This logon window is provided by the back-end ODBC driver (and not by Crystal Reports). The driver sends the logon parameters you have entered to the server for verification.

Client/Server Communication Layer

Through the client portion of your client/server database, which resides on your local workstation, back-end ODBC drivers communicate with the database server. ODBC drivers find their specific DBMS client files on the local computer mainly through key directories that the DBMS client has installed in the search path. The important thing is that a workstation client must be able to connect to its server successfully. For Crystal Reports or the back-end ODBC drivers, it does not matter what platform your database server exists on, since it is the DBMS client that connects and communicates with the server. The connection is successful once the server responds to the ODBC drivers. The back-end driver notifies P2SODBC.DLL.

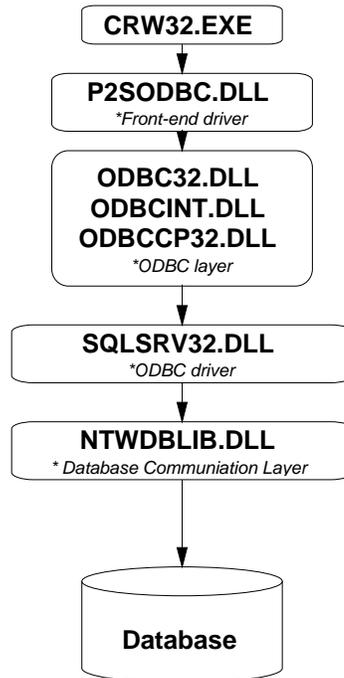
Installing ODBC Drivers

You can configure ODBC drivers that ship with Crystal Reports using the Microsoft ODBC Setup application. ODBC data sources can be created using the Microsoft ODBC Administrator. Crystal Reports Professional ships and installs ODBC drivers for the following databases:

- Microsoft Access
- DB2
- Dbase
- Excel
- FoxPro
- Informix 7
- Informix 9
- Oracle 7
- Oracle 8
- Paradox
- SQLBase
- Microsoft SQL Server
- Sybase
- Text\ASCII
- Visual FoxPro

ODBC Connection Model for Client/Server Databases

This model demonstrates the overall concept of connecting to Client/Server databases via ODBC. For this example, Microsoft SQL Server is used to demonstrate the layout. The ODBC driver and the Database Communication Layer vary, depending on the database being used.



Finding More Information

For further information, please refer to the Crystal Reports User Guide.

Contacting Crystal Decisions for Technical Support

We recommend that you refer to the product documentation and that you visit our Technical Support web site for more resources.

Self-serve Support:

<http://support.crystaldecisions.com/>

Email Support:

<http://support.crystaldecisions.com/support/answers.asp>

Telephone Support:

<http://www.crystaldecisions.com/contact/support.asp>

► www.businessobjects.com

No part of the computer software or this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage and retrieval system, without permission in writing from Business Objects.

The information in this document is subject to change without notice. Business Objects does not warrant that this document is error free.

This software and documentation is commercial computer software under Federal Acquisition regulations, and is provided only under the Restricted Rights of the Federal Acquisition Regulations applicable to commercial computer software provided at private expense. The use, duplication, or disclosure by the U.S. Government is subject to restrictions set forth in subdivision (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at 252.227-7013.

The Business Objects product and technology are protected by US patent numbers 5,555,403; 6,247,008; 6,578,027; 6,490,593; and 6,289,352. The Business Objects logo, the Business Objects tagline, BusinessObjects, BusinessObjects Broadcast Agent, BusinessQuery, Crystal Analysis, Crystal Analysis Holos, Crystal Applications, Crystal Enterprise, Crystal Info, Crystal Reports, Rapid Mart, and WebIntelligence are trademarks or registered trademarks of Business Objects SA in the United States and/or other countries. Various product and service names referenced herein may be trademarks of Business Objects SA. All other company, product, or brand names mentioned herein, may be the trademarks of their respective owners. Specifications subject to change without notice. Not responsible for errors or omissions.

Copyright © 2007 Business Objects SA. All rights reserved.