

# ODBC

Open Database Conectivity

Adrián Barrio Andrés

## Índices

---

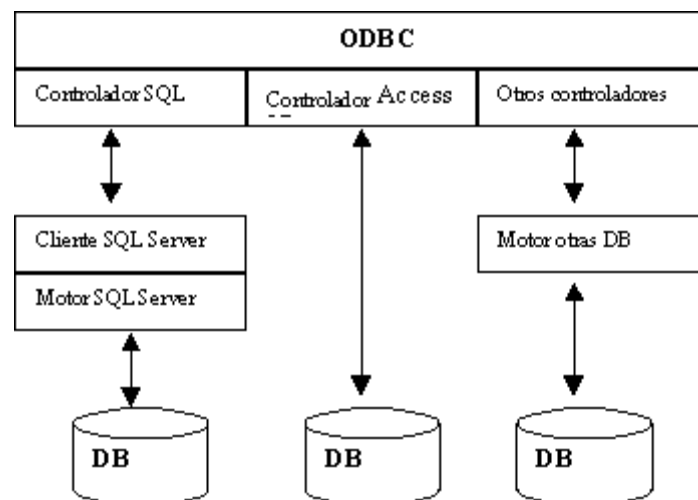
Resumen .....	3
Más información .....	3
¿Cómo interactúan estos tres componentes?.....	4
¿Qué es tan importante acerca de ODBC? .....	4
Ejemplos de ODBC en uso.....	4
Respuestas y preguntas básicas .....	4
Cómo conectarse a partir de ODBC .....	5
Funciones importantes.....	6
Manejadores.....	7

## Resumen

Conectividad abierta de bases de datos (ODBC) es la interfaz estratégica de Microsoft para obtener acceso a datos en un entorno heterogéneo de sistemas relacionales y no-relacionales de administración de base de datos. Basado en la especificación de interfaz de nivel de llamada del grupo de acceso SQL, ODBC proporciona una manera abierta, independiente del proveedor de acceso a datos almacenados en una variedad de bases de datos de mainframe, minicomputadora y propietaria PC.

ODBC alivia la necesidad de aprender múltiples interfaces de programación de aplicaciones para desarrolladores corporativos y fabricantes independientes de software. ODBC proporciona ahora una interfaz de acceso a datos universal. Con ODBC, los desarrolladores de aplicaciones pueden permitir que una aplicación simultáneamente acceso, ver y modificar los datos de varias bases de datos diferentes.

ODBC es un componente básico de la arquitectura de servicios abiertos de Microsoft Windows. Apple ha respaldado ODBC como clave de habilitación de la tecnología anunciando soporte en System 7 en el futuro. Con el apoyo de la industria creciente, ODBC está rápidamente emergiendo como un sector importante estándar de acceso a datos para aplicaciones Windows y Macintosh.



## Más información

ODBC es una especificación a la que los desarrolladores escribir:

- Una aplicación habilitada para ODBC "front-end" o "cliente" desktop, también conocido como un "cliente ODBC." Esta es la aplicación que el usuario del equipo ve en la pantalla.

- o -

- Un controlador ODBC para un "fondo" o "servidor" DBMS (sistema de administración de bases de datos). Esta es la aplicación de DBMS que reside en un equipo que se utiliza para almacenar los datos de acceso varios usuarios. Esta aplicación no es lo que se carga en el equipo del usuario final. Esta aplicación de servidor es normalmente más eficaz (más rápido, con seguridad centralizada y copias de seguridad de datos etc.) que la aplicación cliente. El controlador de ODBC reside entre el cliente de ODBC y el DBMS; Sin embargo, se carga en el equipo cliente.

Para utilizar ODBC, se requieren los tres componentes siguientes:

Ejemplos de cliente - un ODBC front-end habilitado (también llamado cliente ODBC) - Ejemplos ODBC: Microsoft Access, una aplicación creada con Access, una aplicación creada con Microsoft Visual Basic, una aplicación creada con C + Win SDK + ODBC SDK o habilitados para ODBC, las aplicaciones de otros fabricantes (como Lotus).

### ¿Cómo interactúan estos tres componentes?

El cliente ODBC utiliza un lenguaje o vocabulario de comandos (que se conoce como "ODBC") para solicitar datos de o para enviar datos en el back-end o servidor DBMS. Sin embargo, el DBMS no comprende la solicitud del cliente ODBC hasta que el comando se pasa a través del controlador ODBC para DBMS en concreto. Este controlador ODBC es software que reside en el front-end. El controlador ODBC traduce el comando en un formato que puedan entender el servidor ODBC. El servidor ODBC envía la respuesta al controlador de ODBC, que traduce la respuesta en un formato que puedan entender el cliente ODBC.

### ¿Qué es tan importante acerca de ODBC?

En primer lugar, los desarrolladores de aplicaciones no es necesario modificar sus aplicaciones para que puedan tener acceso a datos desde varios back-ends. Siempre que hay un controlador ODBC para un determinado fondo, una habilitada para ODBC front-end puede acceder a él. En segundo lugar, un controlador de ODBC para un DBMS concreto permite que cualquier aplicación habilitada para ODBC ser un cliente ODBC.

### Ejemplos de ODBC en uso

- Acceder al acceso de datos front-end desde un back-end de Oracle utilizando el controlador Oracle ODBC, que se distribuye con Access 1.1.
- Visual Basic front-end acceso a los datos desde un fondo dBASE utiliza el controlador ODBC, que forma parte del paquete de controladores de MS ODBC base de datos de dBASE.
- Aplicación de C escrita con C + ODBC SDK + Win SDK acceso a datos desde una / 400 mediante el AS / 400 disponible de Rochester Software de controlador ODBC.

### Respuestas y preguntas básicas

- ¿Qué es el SDK de ODBC?

El SDK de ODBC se distribuyen en discos y contiene alrededor de 150 páginas de texto que permite al desarrollador escribir un cliente habilitado para ODBC o un controlador ODBC. El SDK de ODBC debe utilizarse junto con un compilador de C y el Kit de desarrolladores de Software (SDK) de Windows.

- ¿Cuál es la base de datos ODBC Driver Pack?

La base de datos ODBC Driver Pack es una colección de los controladores ODBC que se vende por Microsoft.

- ¿Qué es el catálogo de controladores ODBC?

El catálogo de controladores ODBC contiene descripciones de los controladores ODBC desarrollados por Microsoft y otras compañías de software. Contiene una explicación de ODBC y muestra el soporte para el estándar de la industria.

- ¿Controladores ODBC están disponibles actualmente?

Decenas de controladores ODBC están disponibles actualmente, incluyendo aquellos para Microsoft SQL Server, XLS, texto, xBase, Paradox, ODS, Oracle, RedISAM, FoxBase, Btrieve, Integra SQL, DEC Rdb, SQLBase, Informix, Ingres, DB2, SQL/DS, Teradata y SQL/400.

- ¿Cómo se puede escribir un controlador ODBC?

Para escribir un controlador ODBC, utilizar C + Win SDK + ODBC SDK.

- ¿Qué clientes ODBC están disponibles actualmente?

De Microsoft: Access, Visual Basic, Kit de Foxpro + conectividad, Word 6.0 para Windows, Excel 5.0, herramienta de MS Query.

De otros proveedores: PowerSoft, descripción de la propiedad, Lotus 1-2-3, Lotus Approach, Lotus Notes, Q+E y otros.

- ¿Cómo uno escribe un cliente de ODBC?

Mediante C + Win SDK + ODBC SDK, o mediante Access, o utilizando Visual Basic, o utilizando WinFox + Kit de conectividad o utilizando una de las herramientas de otros proveedores diseñados para ello.

## Cómo conectarse a partir de ODBC

Nuestro objeto aquí no es explicar la forma de explotarla sino cómo establecer una conexión entre la base de datos, almacenada en cualquier lugar del disco duro y nuestra página web alojada también en cualquier parte y reconocida por nuestro servidor personal a partir del directorio virtual.

Para crear este vínculo, nos servimos de los conectores ODBC (Open DataBase Connectivity) los cuales establecen el enlace con la base de datos.



El primer paso para crear esta conexión es ir al panel de configuración y abrir el icono ODBC 32bits. Dentro de él, deberemos crear un DSN (Data Source Name) de tipo sistema o usuario. Para ello nos colocamos en la solapa correspondiente (DSN sistema o DSN usuario) y seleccionamos "Añadir". A continuación se nos pedirá seleccionar los controladores de la aplicación que hemos utilizado para crear la base de datos, el nombre que le queremos asignar (aquel que empleemos en nuestros scripts) y el camino para encontrarla en el disco duro.

Esta DSN permite en realidad definir la base de datos que será interrogada sin necesidad de pasar por la aplicación que hayamos utilizado para construirla, es decir, con simples llamadas y órdenes desde nuestros archivos ASP podremos obtener los datos que buscamos sin necesidad de ejecutar el Access o el MySQL los cuales, evidentemente, no tendrán por qué encontrarse en el servidor donde trabajemos.



## Funciones importantes

Tarea	Nombre de la función	Conformidad	Propósito
Conexión a una fuente de datos	<a href="#">SQLAllocHandle</a>	ISO 92	Obtiene un entorno, de conexión, declaración o manejador de descriptor.
La obtención de información acerca de una fuente del controlador y los datos	<a href="#">SQLDataSources</a> <a href="#">SQLDrivers</a>	ISO 92 ODBC	Devuelve la lista de fuentes de datos disponibles. Devuelve la lista de controladores instalados y sus atributos.
Establecer y recuperar los atributos del controlador	<a href="#">SQLSetConnectAttr</a> <a href="#">SQLGetConnectAttr</a>	ISO 92 ISO 92	Establece un atributo de conexión. Devuelve el valor de un atributo de conexión.
Establecer y recuperar campos del descriptor	<a href="#">SQLGetDescField</a> <a href="#">SQLGetDescRec</a>	ISO 92 ISO 92	Devuelve el valor de un solo campo de descriptores. Devuelve los valores de varios campos del descriptor.
Preparación de las peticiones SQL	<a href="#">SQLPrepare</a>	ISO 92	Prepara una sentencia SQL para su ejecución posterior.
la presentación de las solicitudes	<a href="#">SQLExecute</a> <a href="#">SQLExecDirect</a>	ISO 92 ISO 92	Ejecuta una declaración preparada. Ejecuta un comunicado.
Recuperación de resultados e información sobre los resultados	<a href="#">SQLRowCount</a> <a href="#">SQLNumResultCols</a>	ISO 92 ISO 92	Devuelve el número de filas afectadas por una inserción, actualización o eliminación petición. Devuelve el número de columnas del conjunto de resultados.
La obtención de información sobre las tablas del sistema de la fuente de datos (funciones de catálogo)	<a href="#">SQLColumnPrivileges</a> <a href="#">SQLColumns</a>	ODBC Open Group	Devuelve una lista de las columnas y los privilegios asociados de una o más tablas. Devuelve la lista de los nombres de las columnas en las tablas especificadas.

Terminación de una declaración	<a href="#">SQLFreeStmt</a>	ISO 92	Finaliza el procesamiento de estados, descarta resultados pendientes, y, opcionalmente, libera todos los recursos asociados con el identificador de instrucción.
Terminar una conexión	<a href="#">SQLDisconnect</a> <a href="#">SQLFreeHandle</a>	ISO 92 ISO 92	Cierra la conexión. Libera un entorno, de conexión, declaración o manejador de descriptor.

## Manejadores

Una conexión se compone de un conductor y una fuente de datos. Un identificador de conexión identifica cada conexión. El identificador de conexión define no sólo el controlador que debe utilizar, pero que la fuente de datos para utilizar con dicho controlador. Dentro de un segmento de código que implementa ODBC (el Administrador de controladores o un controlador), el identificador de conexión identifica una estructura que contiene la información de conexión, tales como los siguientes:

- El estado de la conexión
- Los diagnósticos actuales a nivel de conexión
- Las asas de las declaraciones y los descriptores actualmente asignados a la conexión
- La configuración actual de cada atributo de conexión

ODBC no impide múltiples conexiones simultáneas, si el controlador compatible con ellas. Por lo tanto, en un entorno particular, ODBC, múltiples asas de conexión podrían apuntar a una variedad de conductores y fuentes de datos, para el mismo conductor y una variedad de fuentes de datos, o incluso a múltiples conexiones a un mismo controlador y la fuente de datos. Algunos controladores de limitar el número de conexiones activas que apoyan; la opción `SQL_MAX_DRIVER_CONNECTIONS` en `SQLGetInfo` especifica el número de conexiones activas es compatible con un controlador particular.

Manejadores de conexión se utilizan sobre todo cuando se conecta a la fuente de datos ( `SQLConnect` , `SQLDriverConnect` o `SQLBrowseConnect` ), desconectarse de la fuente de datos ( `SQLDisconnect` ), obteniendo información sobre el conductor y los datos de origen ( `SQLGetInfo` ), la recuperación de diagnóstico ( `SQLGetDiagField` y `SQLGetDiagRec` ), y las transacciones que realizan ( `SQLEndTran` ). También se utilizan cuando establecer y obtener los atributos de conexión ( `SQLSetConnectAttr` y `SQLGetConnectAttr` ) y al conseguir el formato nativo de una instrucción SQL ( `SQLNativeSql` ).

Manejadores de conexión se asignan con `SQLAllocHandle` y liberaron con `SQLFreeHandle`.