

ODBC Driver for 4D Server

Reference Manual



ODBC Driver for 4D Server Reference Manual

Copyright© 2000 4D SA / 4D, Inc.
All rights reserved

The Software described in this manual is governed by the grant of license in the 4D Product Line License Agreement provided with the Software in this package. The Software, this manual, and all documentation included with the Software are copyrighted and may not be reproduced in whole or in part except for in accordance with the 4D Product Line License Agreement.

Microsoft, Windows and Windows NT are registered trademarks and ODBC is a trademark of Microsoft Corporation in the United States and other countries.

SIMBA is a registered trademark and SimbaEngine is a trademark of Simba Technologies Incorporated.

4th Dimension, 4D, the 4D logo, 4D Server, 4D Runtime, 4D Compiler, 4D Write, 4D Draw, 4D, and the 4D logo are registered trademarks of 4D SA.

Apple, Macintosh, Power Macintosh, and LaserWriter are trademarks or registered trademarks of Apple Computer Inc.

All other referenced trade names are trademarks, registered trademarks, or copyrights of their respective holders.

IMPORTANT LICENSE INFORMATION

Use of this Software is subject to the 4D Product Line License Agreement, which is provided in electronic form with the Software. Please read the 4D Product Line License Agreement carefully before completely installing or using the Software.

Table of Contents

Chapter 1	General Architecture5
	ODBC Architecture	5
	ODBC Driver.	6
Chapter 2	Installation and Setup7
	Requirements	7
	Client machine	7
	Server machine	7
	Installation.	8
Chapter 3	Creating and Modifying a Data Source.	11
	ODBC Administrator Version 2.5	12
	Creating a User DSN	12
	Creating a System DSN	14
	ODBC Administrator Version 3.x	14
	Creating a User DSN	15
	Creating a non- sharable File DSN	17
	Creating a System DSN	21
Chapter 4	Specifications	23
	Allowing Access to 4D Server	23
	Structural Limitations	24
	SQL Statement Limitations	24
	Null Values.	25
	Selecting null values	25
	Updating null values	25
	Comparison of Floats	25
	Primary Key	25

Chapter 5	ODBC Conformance Levels	27
	SQL Conformance Level	27
	Support of minimum SQL grammar	27
	Support of core SQL grammar	28
	Support of extended SQL grammar	28
	API Conformance Level	29
	Scalar Functions	29
Chapter 6	Mapping Data Types	31
	Conversion from 4D Server to ODBC SQL data types	31
	Conversion from ODBC SQL to 4D Server data types	32
Appendix A	SQL Samples	33
	Examples	33
	Simple SELECT statement	33
	SELECT statement with a WHERE clause	34
	SELECT statement with the BETWEEN predicate	34
	SELECT statement with a “begins with” condition	34
	SELECT statement with a sort on an alphanumeric column	35
	SELECT statements with string functions	35
	SELECT statements with numeric functions	36
	SELECT statement with a grouping order	37
	SELECT statement with an inner join	38
	SELECT statement with UNION on two distinct tables	38
	SELECT statement with an IN predicate	40
	Syntax considerations	40
Appendix B	API Conformance Level	41
	More Information on Specific Calls	43

1

General Architecture

ODBC stands for Open DataBase Connectivity. It is an interface that allows applications to query data sources, using Structured Query Language (SQL) as a standard. These data sources can be of any type, from a text file to the most sophisticated DBMS.

The ODBC driver for 4D Server allows any PC (Windows) application to communicate with 4D Server through ODBC calls and SQL queries. Based on existing 4D components (Server API and Network Components), this driver is a fully integrated element of our open architecture. With only a client installation, this solution can be deployed without any changes to an existing database in production.

This product follows the standards of the Open DataBase Connectivity interface defined by Microsoft. It supports the SQL syntax based on the X/Open and SQL Access Group (SAG) SQL CAE specification (1992).

ODBC Architecture

The ODBC architecture using the ODBC Driver for 4D Server consists of three parts:

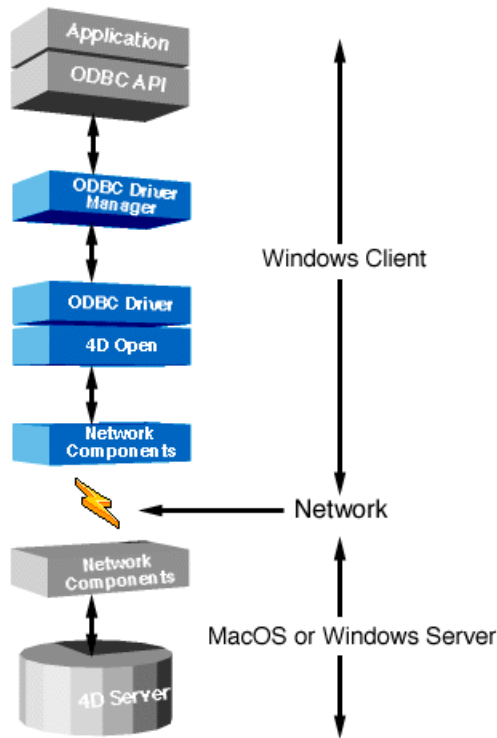
- **Application.** The application calls functions defined in the ODBC API to access a data source.
- **Driver Manager.** The Driver Manager implements the ODBC API, loads the 4D Server driver dynamically, dispatches and manages the ODBC calls.
- **ODBC Driver for 4D Server.** The driver allows you to connect your ODBC-compliant application to a 4D Server database.

ODBC Driver

The ODBC Driver for 4D Server consists of four components:

- **ODBC Driver.** The ODBC driver for 4D Server itself. The driver processes ODBC function calls and manages exchanges between the application and the data source.
- **4D Open Library.** The client native part of 4D Server.
- **Network Components.** The networking part of 4D Server.
- **Data Source.** 4D Server is the data source, which contains data that an application needs to access.

The following figure describes an ODBC architecture using the ODBC Driver for 4D Server:



The ODBC Driver for 4D Server is only available for Windows. The 4D Server application can be located on a Macintosh or a PC, but the client application and the driver must be installed on a Windows machine.

2

Installation and Setup

This section describes the hardware and software requirements for ODBC Driver for 4D Server and how to install it from the CD-ROM.

Requirements

Client machine

Hardware

The client machine should be a PC with one of the following 32-bit systems installed: Windows 95/98, Windows NT 3.5.1 or Windows NT 4.

Software

Any ODBC-compliant application should be able to operate with the ODBC driver.

Server machine

Hardware

The server machine can be one of the following:

- PC running a 32-bit system (Windows 95/98, Windows NT 3.5.1 or Windows NT 4).
- Power Macintosh running Mac OS System 7.x or higher.

Software

The data source should be 4D Server version 6.0.5 or higher.

Installation

This section describes the installation of the SETUP program, which is automatically launched during the installation of the ODBC Driver for 4D Server.

The SETUP.exe program performs all the tasks necessary for installing the ODBC driver for 4D Server. This program correctly configures all the dependent files, such as ODBC.INI, ODBCINST.INI and the registry database.

- To install the ODBC Driver for 4D Server:

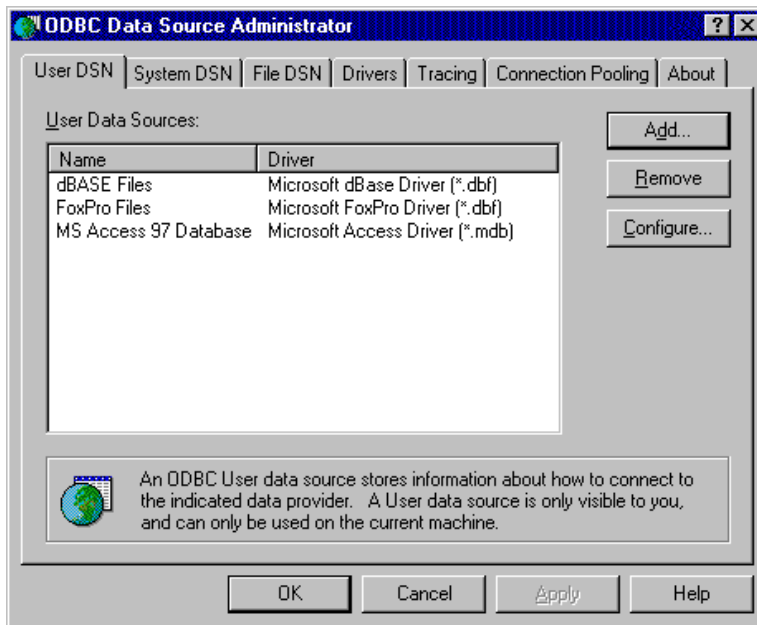
- 1 Double-click the Setup icon.**

The Welcome dialog box appears.



- 2 In the Welcome dialog box, click Continue to proceed with the installation.**

The ODBC Data Source Administrator window appears:



If you want to configure the data source now, refer to chapter “Creating and Modifying a Data Source” on page 11. Otherwise, just click the **Cancel** button.

Eventually, a message indicates that the installation was successful.



3

Creating and Modifying a Data Source

The ODBC architecture is based on applications that access one or several data sources. These data sources must be defined and qualified on the client in order to gain access to the right one. The Data Source Names (DSN) can be defined at three levels:

- User DSN
- File DSN
- System DSN

Each level corresponds to a specific application and is used in a unique way. Be careful, because client software does not support all the configurations. For example, “MS Query” from Microsoft Office 97 only works with the File DSN.

Chapters 3 and 4 describe how to configure a data source using the Driver Manager with ODBC Administrator versions 2.5 or 3.x.

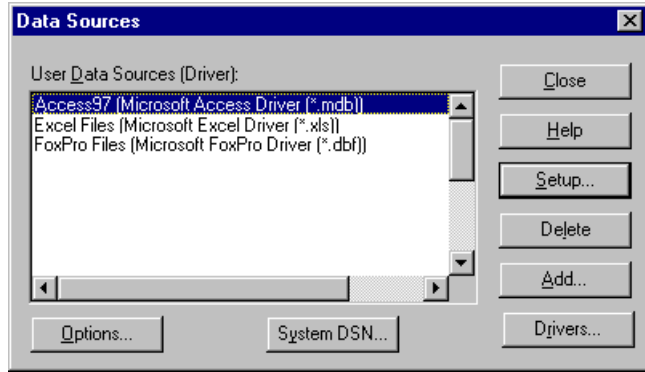
Refer to the documentation of your ODBC-compliant application to determine which version of the Driver Manager you should configure, and at which level.

ODBC Administrator Version 2.5

Creating a User DSN

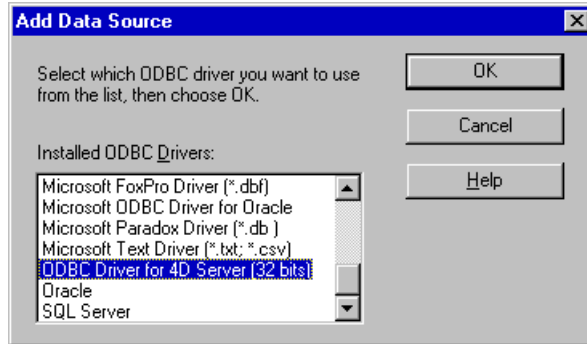
- ▶ To create a User DSN:
 - 1 **Double-click the ODBC Administrator icon in the Windows Control Panel to start the ODBC Administrator.**

The Data Sources dialog box appears:



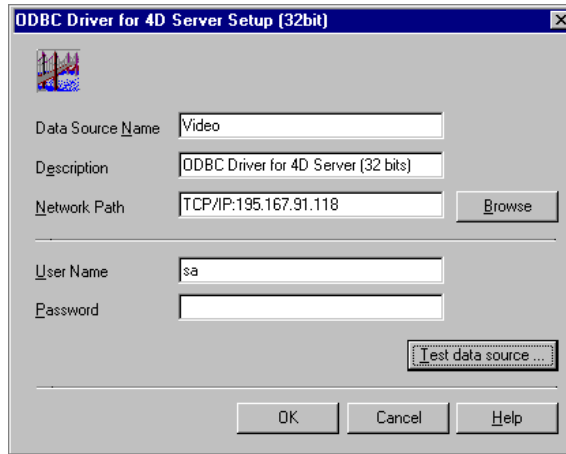
- 2 **To add a Data Source, click the Add button.**

The Add Data Source dialog box appears, displaying a list of the ODBC drivers installed on your machine.



3 Select the ODBC Driver for 4D Server, then click OK.

The ODBC Driver for 4D Server Setup (32bit) dialog box appears:



4 In the Data Source Name text box, enter the name of the data source you want to access.

The data source name can be any name.

5 In the Description text box, enter the description of the driver.

By default, the description is “ODBC Driver for 4D Server (32bit)”, but it can be any string you want.

6 In the Network Path text box, enter the location and the protocol used to access your 4D Server data source. This should be typed according to the following rules:

The Network Path depends on the network protocol used on our network:

- Appletalk protocol: Enter “Appletalk:DatabaseName@ZoneName”.
- TCP protocol: Enter “TCP/IP:Address,{PortNumber}”.
- IPX protocol: Enter “IPX:Address”.

You can also choose your 4D Server data source by using the Browse button. In this case, after you have selected the 4D Server data source, do not modify the Network Path as it appears in the Network Path text box.

For more information, please refer to the *4D Server Reference* and *Network Components for 4D Server* documentation.

- 7 You can test the new data source by clicking on the button “Test data source...”.

If the connection failed, a message displays the number of the error and the description of the problem. Otherwise, a successful message is displayed.

- 8 Click the OK button.

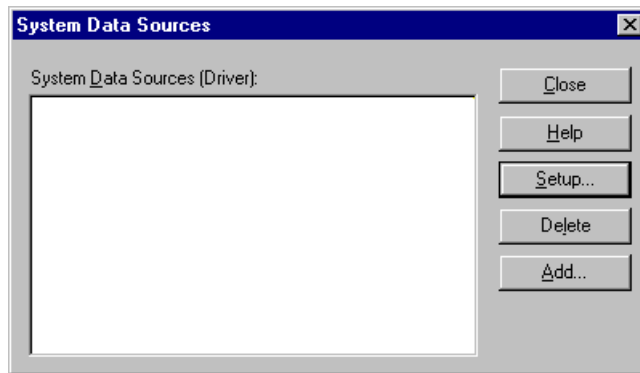
The User DSN will be added to the list.

Creating a System DSN

- To create a System DSN:

- 1 In the Data Sources dialog box, click the System DSN button.

The System Data Sources window appears:

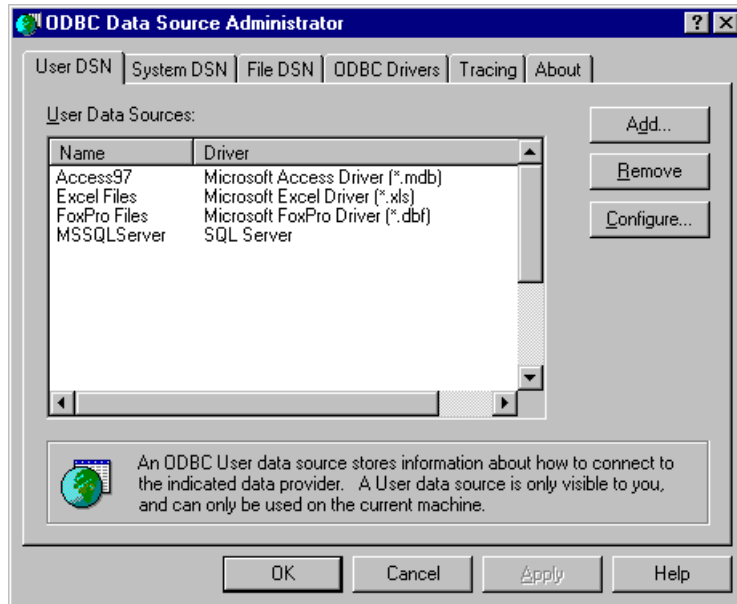


- 2 Click the Add button to add a System Data Source, then follow steps 2 through 7 in the “Creating a User DSN” section.

ODBC Administrator Version 3.x

- ▶ To start the ODBC Administrator:
- **Double-click the ODBC Administrator icon in the Windows Control Panel.**

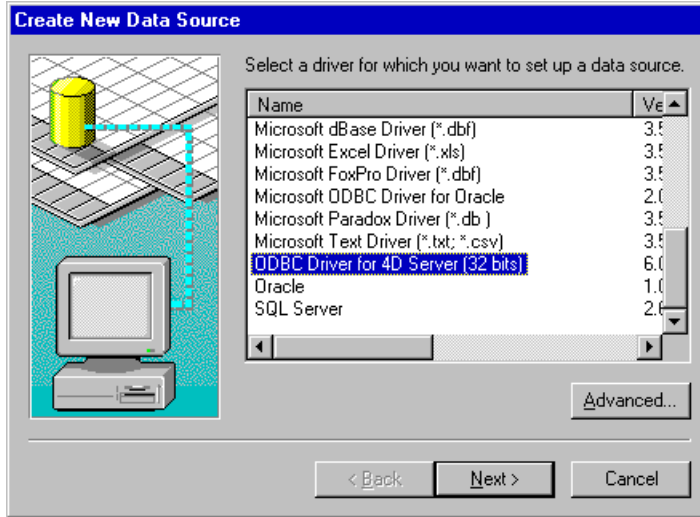
The ODBC Data Source Administrator window appears:



Creating a User DSN

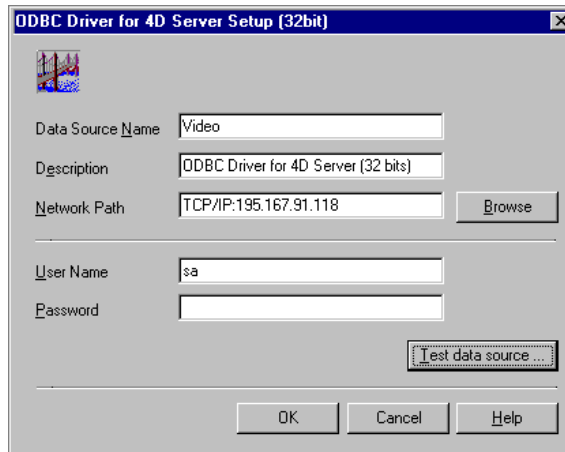
- ▶ To create a User DSN:
 - 1 **Click the Add button.**

The Create New Data Source dialog box appears, displaying a list of drivers:



- 2 Select the ODBC Driver for 4D Server (32 bits), then click Advanced.

The ODBC Driver for 4D Server Setup (32bit) dialog box appears.



- 3 In the Data Source Name text box, enter the name of the data source that you want to access.

The data source name can be any name you enter.

- 4 In the Description text box, enter the description of the driver.

By default, the description is “ODBC Driver for 4D Server (32 bit)”, but it can be any string that you choose.

5 In the Network Path text box, enter the location and the protocol used to access your 4D Server data source. This should be typed according to the following rules:

The Network Path depends on the network protocol used on your network:

- Appletalk protocol: Enter “Appletalk:DatabaseName@ZoneName”.
- TCP protocol: Enter “TCP/IP:Address,{PortNumber}”.
- IPX protocol: Enter “IPX:Address”.

You can also choose your 4D Server data source by using the Browse button. In this case, after you have selected the 4D Server data source, do not modify the Network Path as it appears in the “ODBC Driver for 4D Server Setup (32bit) ” dialog box.

For more information, please refer to the *4D Server Reference* and *Network Components for 4D Server* documentation.

6 You can test the new data source by clicking on the button “Test data source...”.

If the connection failed, a message displays the number of the error and the description of the problem. Otherwise, a successful message is displayed.

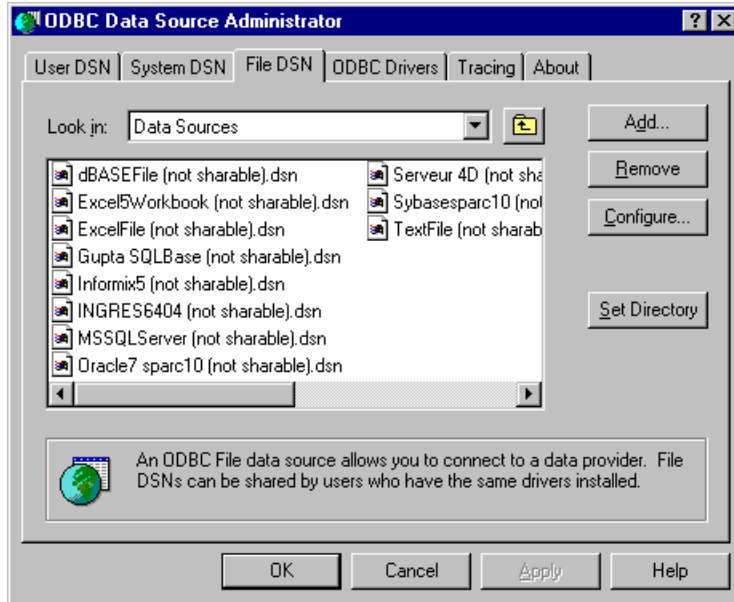
7 Click the OK button.

The User DSN will be added to the list.

Creating a non-sharable File DSN

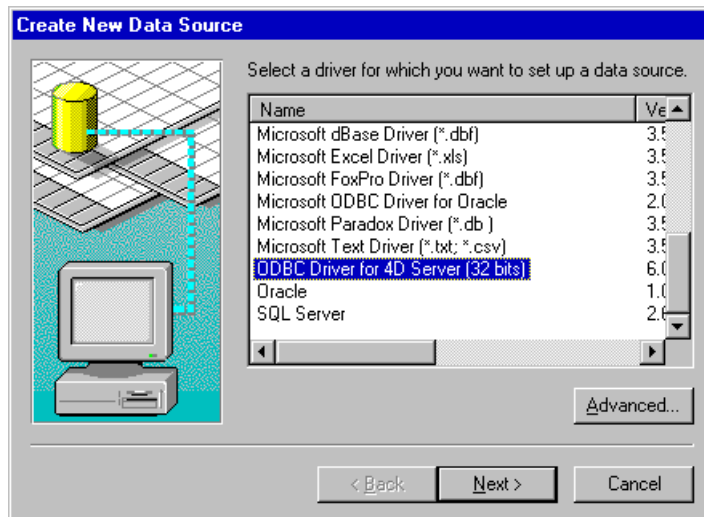
First, you should refer to the previous section, “Creating a User DSN”, since the non-sharable File DSN is a pointer for this definition.

- ▶ To create a non-sharable File DSN:
 - 1 Click the File DSN tab in the ODBC Data Source Administrator window.



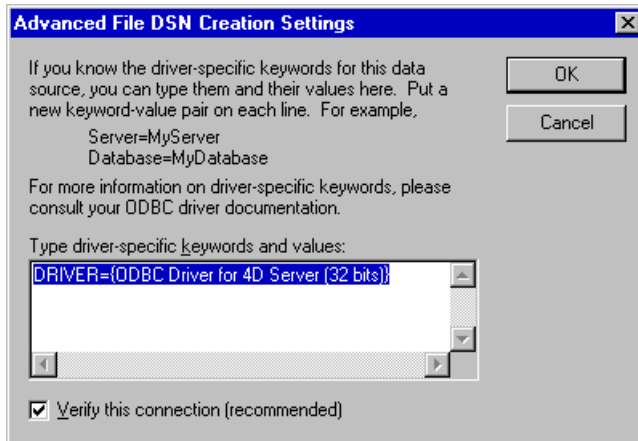
- 2 Click the Add button.

The Create New Data Source dialog box appears:



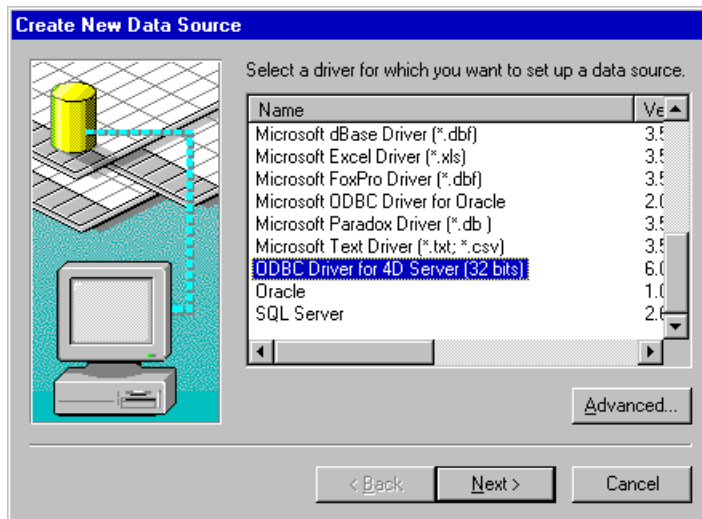
3 Select the ODBC Driver for 4D Server (32bit), then click Advanced.

The Advanced File DSN Creation Settings dialog box appears:



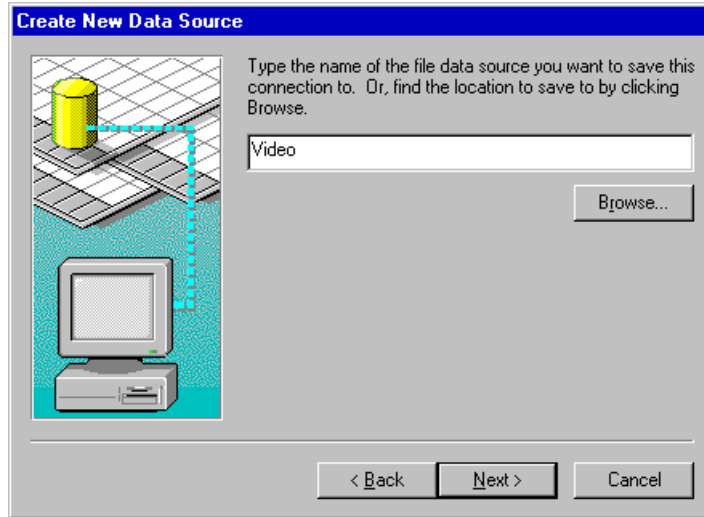
4 Deselect the "Verify this connection" check box, then click OK.

5 In the Create New Data Source dialog box, click Next.



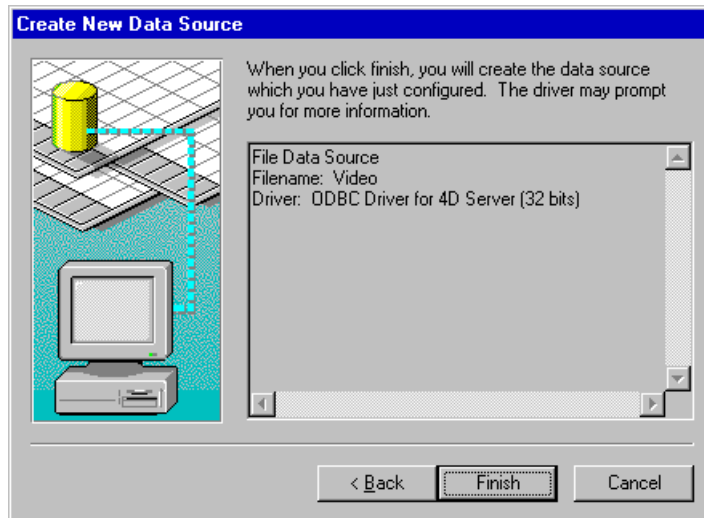
Another dialog box appears, in which you can enter the name of the data source.

- 6 In the file name text box, enter the name of the data source (in our example, Video).



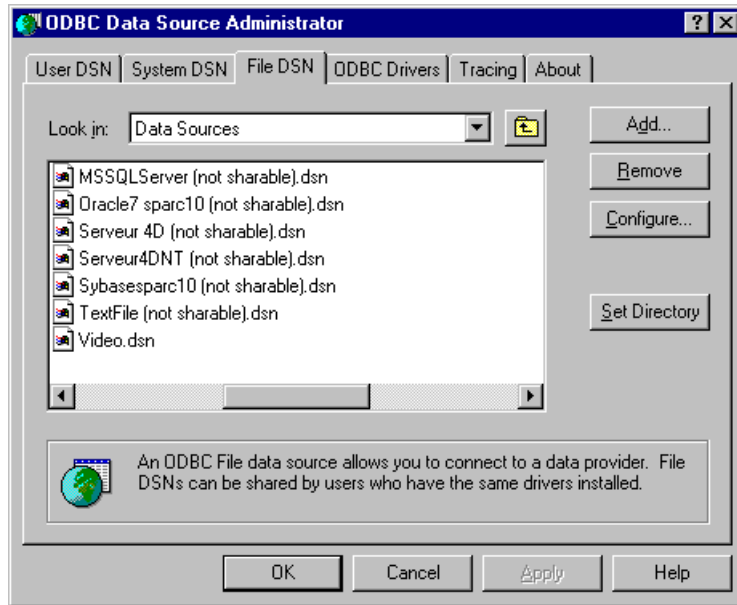
- 7 Click Next.

An information dialog box displays a summary of the definition of your File Data Source.



8 After checking all the information, click Finish.

The File DSN is added to the list:

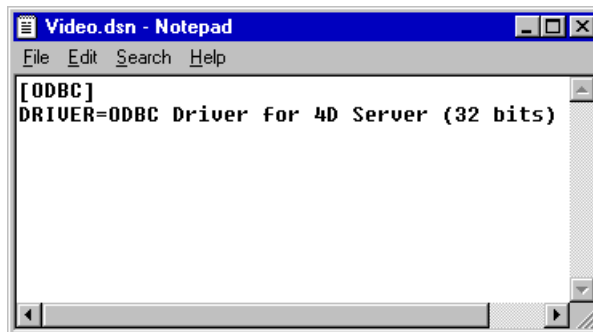


9 Quit ODBC Administrator 3.x by clicking OK.

10 Open the video.dsn file using the Notepad text editor (Notepad.exe).

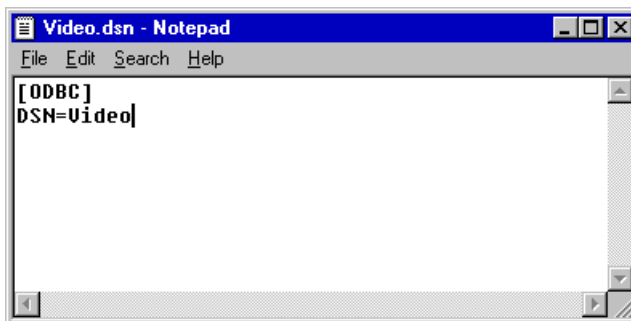
The video.dsn file is located in your system folder in the “System folder\ODBC\Data sources” path.

The file content appears as follows:



- 11 Replace the existing text “DRIVER= ODBC Driver for 4D Server (32 bits)” with “DSN=Name of the file data source”.

In our example, we enter: “DSN=Video”.

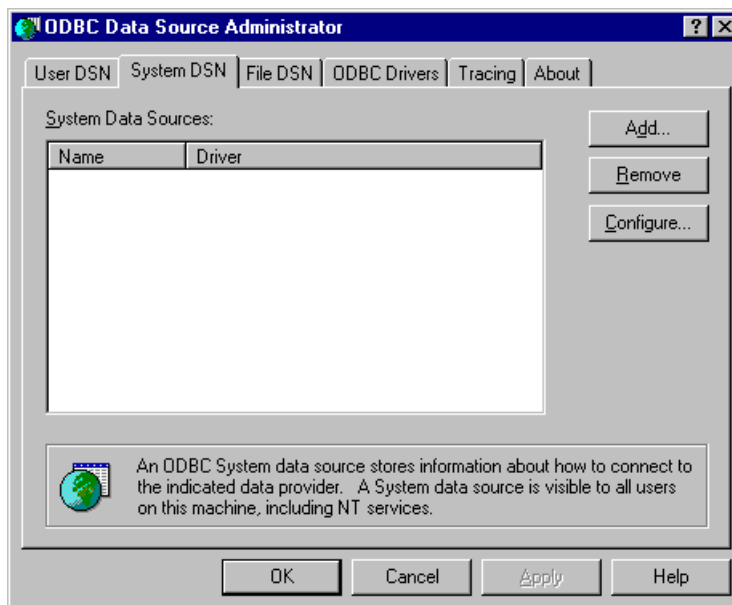


- 12 Save the File DSN and quit the Notepad application.

Creating a System DSN

► To create a system DSN:

- 1 Click the System DSN tab in the ODBC Data Source Administrator window.



- 2 Click the Add button to add a System Data Source, then follow steps 2 through 6 of the section “Creating a User DSN”.

4

Specifications

This chapter covers several specifications concerning the use of the ODBC Driver for 4D Server:

Allowing Access to 4D Server

The ODBC Driver for 4D Server is based on 4D Open. 4D Open is the API (Application Programming Interface) that allows non-4D Client applications to connect to 4D Server.

To access 4D Server using the ODBC Driver, you must enable the 4D Open connection.

- ▶ To do so:
 - 1 **Open your structure file in the Design environment, using 4D or 4D Client.**
 - 2 **In the File menu, choose Database Properties.**
 - 3 **Select the “Data Control and Access” page.**
 - 4 **Check the “Allow 4D Open Connections” option.**

Structural Limitations

Description	Limit
Number of tables in a database	255
Number of columns per table	511
Number of rows per table	16 million
Number of select list columns in a query	300
Size of column	2 gigabytes
Maximum table name length	31 characters
Maximum column name length	31 characters
Maximum username length	30 characters
Number of connections	limited by your 4D Server
SQL statement length	16K
Number of statements per connection	limited by memory
Table qualifier	limited by the OS
Number of joined tables	limited by memory

For more information on the 4D specifications, please refer to the *4th Dimension Language Reference* manual.

SQL Statement Limitations

- Maximum of 1000 characters in a literal.
- Maximum of 300 ANDed predicates.
- A character in a character string literal may be any ANSI character between 1 and 255 decimal. A single-quote (') must be represented as two consecutive single-quotes (").

Null Values

Selecting null values A 4D date field equal to !00/00/00! is considered a null value.

An empty 4D text or alphanumeric field is considered a null value.

Updating null values A null SQL_DATE or SQL_TIMESTAMP is updated as a 4D Date field equal to !00/00/00! in 4D Server.

A null SQL_CHAR, SQL_VARCHAR or SQL_LONGVARCHAR is updated as a 4D alphanumeric or text field equal to empty.

- ▼ For example, in the following code, the SELECT statement returns 2 and 3:

```
CLIENTS.IDCLIENTS.String
1  MyString
2  NULL
3  ""
SELECT CLIENTS.ID
FROM CLIENTS
WHERE CLIENTS.String IS NULL
```

Comparison of Floats

The ODBC-compliant applications used to access 4D Server can store data in different ways than does 4D Server. This difference may produce rounding approximation and different result sets in a query containing comparisons.

Primary Key

The ODBC driver for 4D Server only considers the first (by creation order) unique, mandatory and indexed column as the primary key.

5

ODBC Conformance Levels

SQL Conformance Level

The main characteristic of the ODBC driver is its SQL conformance level, in other words, its capacity to support and address a certain SQL grammar. The ODBC version 2.5 specification provides three levels of SQL grammar conformance: Minimum, Core and Extended. Each higher level provides more fully-implemented data definition and data manipulation language support than the previous one.

The description of each level explains what is supported in the ODBC Driver for 4D Server.

Support of minimum SQL grammar

ODBC Driver for 4D Server partially supports Minimum SQL Grammar as it is defined in Appendix C of the Microsoft *ODBC Programmer's Reference*.

The following is a list of supported elements:

- Supported Data Manipulation Language (DML): simple SELECT, INSERT, searched UPDATE, and DELETE statements
- Supported expressions: simple
- Supported data types: CHAR, VARCHAR, LONG VARCHAR

Support of core SQL grammar ODBC Driver for 4D Server partially supports Core SQL Grammar as it is defined in Appendix C of the *ODBC Programmer's Reference*.

The following is a list of supported elements:

- Supported Data Manipulation Language (DML): full SELECT.
- Supported expressions: Subqueries, set-functions (SUM, MAX, MIN, AVG, and COUNT)
- Approximate-numeric-literal
- Between-predicate: BETWEEN
- Correlation-name
- Exact-numeric literal
- In-predicate: IN
- Supported data types: DECIMAL, SMALLINT, INTEGER, REAL, DOUBLE PRECISION

Support of extended SQL grammar

ODBC Driver for 4D Server partially supports Extended SQL Grammar as documented in Appendix C of the *ODBC Programmer's Reference*.

The following is a list of supported elements:

- DML: outer joins, UNION and UNION ALL
- Expressions: scalar functions, date and time literals
- Date arithmetic
- Extended predicates
- Data types: BIT, TINYINT, DATE, TIME, TIMESTAMP, LONG VARBINARY

For examples about SQL statements, please refer to Appendix A of this manual.

API Conformance Level

An ODBC driver is accessed by a set of specific ODBC calls. Each driver implements more or fewer of these calls as needed to support the standard. Most of the time, you will not have to pay attention to this level of programming because of the transparency at the application level.

The application you are using will implement these calls to access the data source, and you will have only to control the SQL query sent through this API.

If you are directly programming the ODBC API in your application, you can refer to Appendix B of this manual for a complete list of the calls implemented by our driver.

Scalar Functions

The ODBC Driver for 4D Server supports ODBC scalar functions which may be included in an SQL statement as a primary expression. For a description of the SQL syntax for these functions, please refer to Appendix G of the *ODBC Programmer's Reference*.

The following is a list of the scalar functions supported by the ODBC Driver for 4D Server.

- String functions: CONCAT, LENGTH, RIGHT, UCASE, LCASE, LOCATE, RTRIM, LEFT, LTRIM, SUBSTRING
- Numeric functions: MOD
- Date functions: DAYOFMONTH, MONTH, DAYOFWEEK, YEAR
- System functions: IF, NULL
- Conversion functions: CONVERT

6

Mapping Data Types

4D Server supports a set of data types. The ODBC Driver for 4D Server maps these data types to their appropriate ODBC SQL data types. The following conversion tables list the 4D Server data types and their corresponding ODBC SQL data types.

Conversion from 4D Server to ODBC SQL data types

4D Server Data Types	ODBC SQL Data Types
Alpha	SQL_VARCHAR
Text	SQL_LONGVARCHAR
Real	SQL_DOUBLE
Integer	SQL_SMALLINT
Long Integer	SQL_INTEGER
Date	SQL_DATE
Time	SQL_TIME
Boolean	SQL_BIT
Picture	SQL_LONGVARBINARY
Subtable	N/A
BLOB	N/A

Conversion from ODBC SQL to 4D Server data types

ODBC SQL Data Types	4D Server Data Types
SQL_VARCHAR	Alpha
SQL_CHAR	Alpha
SQL_LONGVARCHAR	Text
SQL_REAL	Real
SQL_DOUBLE	Real
SQL_DECIMAL	Real
SQL_SMALLINT	Integer
SQL_TINYINT	Integer
SQL_INTEGER	Long Integer
SQL_LONGVARBINARY	Picture
SQL_DATE	Date
SQL_TIMESTAMP	Date
SQL_TIME	Time
SQL_BIT	Boolean

A

SQL Samples

To illustrate the SQL grammar (described in Chapter 5) and to help those who are discovering the SQL language, we provide several queries with their results, based on the “VIDEO” database distributed with the driver.

Examples

Simple SELECT statement

This example uses a simple SELECT statement. This query downloads the contents of the FirstName fields of all the customers in the CUSTOMERS table.

```
▼ SELECT  
CUSTOMERS.FirstName  
FROM  
CUSTOMERS
```

The result is:

```
Cary  
John  
Jon  
Jeff  
Perry  
Fred  
Gus  
Dave  
Jeanette  
Rich  
Kate  
Cindi  
Jerry  
Thom  
Tico  
Janet
```

SELECT statement with a WHERE clause This example uses a SELECT statement with a WHERE clause. This query downloads the video titles and rental prices from the VIDEOS table if the rental price is greater than 21.1.

```
▼ SELECT
  VIDEOS.Title, VIDEOS.RentingPrice
FROM
  VIDEOS
WHERE
  (VIDEOS.RentingPrice>21.1)
```

The result is:

Demoiselles de Rochefort (Les)	24.5
Missing in Action I	24
Jane Fonda's 10min Workout	50
Slippery When Wet	26.3
Living on a Prayer	45.2
Wyle Traps the Road Runner!	26.3

SELECT statement with the BETWEEN predicate This example uses a SELECT statement with the BETWEEN predicate. This query downloads the video titles and the rental prices from the VIDEOS table if the rental price is greater than or equal to 21.1 or less than or equal to 24.5.

```
▼ SELECT
  VIDEOS.Title, VIDEOS.RentingPrice
FROM
  VIDEOS
WHERE
  (VIDEOS.RentingPrice BETWEEN 21.1 And 24.5)
```

The result is:

Demoiselles de Rochefort (Les)	24.5
Missing in Action I	24
Boyz in New Jersey	21.1

SELECT statement with a "begins with" condition This example uses a SELECT statement with a "begins with" condition. This query downloads the video titles and the rental prices from the VIDEOS table if the title begins with the letter 'M' (case-sensitive).

```

▼ SELECT
  VIDEOS.Title, VIDEOS.RentingPrice
FROM
  VIDEOS
WHERE
  (VIDEOS.Title Like 'M%')

```

The result is:

Missing in Action I	24
---------------------	----

SELECT statement with a sort on an alphanumeric column

This example uses a SELECT statement with a sort on an alphanumeric column. This query downloads the video titles and the rental prices from the VIDEOS table if the rental price is lower than 21.1. The video titles are sorted alphabetically.

```

▼ SELECT
  VIDEOS.Title, VIDEOS.RentingPrice
FROM
  VIDEOS
WHERE
  (VIDEOS.RentingPrice<21.1)
ORDER BY
  VIDEOS.Title

```

The result is:

Blaze of Glory	18.8
Chip & Jeff's Excellent Adventure	19.5
Night of the Living Noni's	15.5
Prédateurs (Les)	15.5
Return of Cujo	13.5

SELECT statements with string functions

These two examples use SELECT statements combined with string functions.

- ▼ Example 1: This query downloads the video titles from the VIDEOS table if the rental price is lower than 21.1, with the first ten letters of the title converted to uppercase.

```

SELECT
  left(ucase(VIDEOS.Title),10)
FROM
  VIDEOS
WHERE
  (VIDEOS.RentingPrice>21.1)

```

The result is:

```
DEMOISELLE
MISSING IN
JANE FONDA
SLIPPERY W
LIVING ON
WYLE TRAPS
```

- ▼ Example 2: This query locates the position of the letter 'a' (case-sensitive) in the video titles from the VIDEOS table.

```
SELECT
VIDEOS.Title,
locate('a',VIDEOS.Title)
FROM
VIDEOS
```

The result is:

Demoiselles de Rochefort (Les)	0
Prédateurs (Les)	5
Missing in Action I	0
Chip & Jeff's Excellent Adventure	0
Jane Fonda's 10min Workout	2
Return of Cujo	0
Slippery When Wet	0
Blaze of Glory	3
Night of the Living Noni's	0
Boyz in New Jersey	0
Runaway	4
Living on a Prayer	11
Wyle Traps the Road Runner!	8

SELECT statements with numeric functions

These three examples use SELECT statements combined with numeric functions.

- ▼ Example 1: This query returns the average rental price based on the titles in the VIDEOS table.

```
SELECT
Avg(VIDEOS.RentingPrice)
FROM
VIDEOS
```

The result is:

24.5153846153846

- ▼ Example 2: This query returns the lowest rental price of the titles in the VIDEOS table.

```
SELECT
min(VIDEOS.RentingPrice)
FROM VIDEOS
```

The result is:

13.5

- ▼ Example 3: This query returns the highest rental price of the titles in the VIDEOS table.

```
SELECT
max(VIDEOS.RentingPrice)
FROM VIDEOS
```

The result is:

50

SELECT statement with a grouping order

This example uses a SELECT statement combined with a grouping order. This query uses the Count function and the GROUP BY option to find the number of videos in each category (classic, comedy, etc.).

- ▼ **SELECT**
Count(*),
VIDEOS.Category
FROM
VIDEOS VIDEOS
GROUP BY
VIDEOS.Category

The result is:

1	Classic
2	Comedy
2	Drama
1	Fantastic
1	Health
1	Martial Arts
2	Thriller
2	War
1	Western

SELECT statement with an inner join

This example uses a SELECT statement combined with an inner join on the CLIENTS and TRACKING tables. This query shows how to select data from more than one table at once, using the joins:

```
▼ SELECT
CUSTOMERS.FirstName, CUSTOMERS.LastName,
TRACKING.BorrowDate,
VIDEOS.Title,
VIDEOS.Category
FROM
CUSTOMERS, TRACKING, VIDEOS
WHERE
TRACKING.Video_ID = VIDEOS.Video_ID
AND
CUSTOMERS.Customer_ID = TRACKING.Client_ID
ORDER BY CLIENTS.LastName
```

The result is:

Jon	Bon Jovi	1997-08-02	Demoiselles de Rochefort (Les)	Comedy
Jeff	Davis	1992-12-26	Prédateurs (Les)	Fantastic
Perry	Fitch	1992-08-12	Missing in Action I	War
Fred	Forire	1992-01-17	Chip & Jeff's Excellent Adventure	Comedy
Cary	Grant	1997-08-02	Demoiselles de Rochefort (Les)	Comedy
Gus	Guilbert	1992-01-17	Jane Fonda's 10min Workout	Health
John	Huston	1992-12-26	Prédateurs (Les)	Fantastic
Dave	Popielarz	1992-12-15	Return of Cujo	Martial Arts

SELECT statement with UNION on two distinct tables

This example uses a SELECT statement with UNION on two distinct tables. This query retrieves data from two tables using the UNION capabilities.

Note Note that, in this example, the union of the video categories and the customer names is nonsense, but it is just to be considered as an example!

```
▼ SELECT  
VIDEOS.Category  
FROM  
VIDEOS  
UNION  
SELECT  
CUSTOMERS.FirstName  
FROM  
CUSTOMERS
```

The result is:

Cary
Cindi
Classic
Comedy
Dave
Drama
Fantastic
Fred
Gus
Health
Janet
Jeanette
Jeff
Jerry
John
Jon
Kate
Martial Arts
Perry
Rich
Thom
Thriller
Tico
War
Western

SELECT statement with an IN predicate

This example uses a SELECT statement combined with an IN predicate. This query retrieves the videos that should have been brought back by the customers. The query is performed on the VIDEOS and TRACKING tables. The first part of the query searches out the Video_IDs in the TRACKING table using the Tracking."Days late" criteria. If this criteria is greater than zero, the video is selected. These IDs are used to select the VIDEOS within an IN predicate.

```
▼ SELECT VIDEOS.Title
FROM VIDEOS
WHERE VIDEOS.Video_ID
IN
(
SELECT TRACKING.Video_ID
FROM TRACKING
WHERE (TRACKING."Days Late">0)
)
```

The result is:

```
Demoiselles de Rochefort (Les)
Prédateurs (Les)
Chip & Jeff's Excellent Adventure
Jane Fonda's 10min Workout
Living on a Prayer
Return of Cujo
```

Syntax considerations

If a 4D column contains a space, its name must be written between quotes in order to avoid a syntax error.

```
▼ Example:
SELECT
TRACKING. "Days Late"
FROM
TRACKING
```


B

API Conformance Level

The following table lists the ODBC functions and their API conformance levels.

ODBC function and level	ID	Supported by the driver
<i>Core Level</i>		
SQL_API_SQLALLOCCONNECT	1	Yes
SQL_API_SQLALLOCENV	2	Yes
SQL_API_SQLALLOCSTMT	3	Yes
SQL_API_SQLBINDCOL	4	Yes
SQL_API_SQLCANCEL	5	Yes
SQL_API_SQLCOLATTRIBUTES	6	Yes
SQL_API_SQLCONNECT	7	Yes
SQL_API_SQLDESCRIBECOL	8	Yes
SQL_API_SQLDISCONNECT	9	Yes
SQL_API_SQLERROR	10	Yes
SQL_API_SQLEXECDIRECT	11	Yes
SQL_API_SQLEXECUTE	12	Yes
SQL_API_SQLFETCH	13	Yes
SQL_API_SQLFREECONNECT	14	Yes
SQL_API_SQLFREEENV	15	Yes
SQL_API_SQLFREESTMT	16	Yes
SQL_API_SQLGETCURSORNAME	17	Yes
SQL_API_SQLNUMRESULTCOLS	18	Yes
SQL_API_SQLPREPARE	19	Yes
SQL_API_SQLROWCOUNT	20	Yes
SQL_API_SQLSETCURSORNAME	21	Yes
SQL_API_SQLSETPARAM	22	Yes
SQL_API_SQLTRANSACT	23	Yes

ODBC function and level	ID	Supported by the driver
<i>Level 1</i>		
SQL_API_SQLCOLUMNS	40	Yes
SQL_API_SQLDRIVERCONNECT	41	Yes
SQL_API_SQLGETCONNECTOPTION	42	Yes
SQL_API_SQLGETDATA	43	Yes
SQL_API_SQLGETFUNCTIONS	44	Yes
SQL_API_SQLGETINFO	45	Yes
SQL_API_SQLGETSTMTOPTION	46	Yes
SQL_API_SQLGETTYPEINFO	47	Yes
SQL_API_SQLPARAMDATA	48	Yes
SQL_API_SQLPUTDATA	49	Yes
SQL_API_SQLSETCONNECTOPTION	50	Yes
SQL_API_SQLSETSTMTOPTION	51	Yes
SQL_API_SQLSPECIALCOLUMNS	52	Yes
SQL_API_SQLSTATISTICS	53	Yes
SQL_API_SQLTABLES	54	Yes
<i>Level 2</i>		
SQL_API_SQLBROWSECONNECT	55	Yes
SQL_API_SQLCOLUMNPRIVILEGES	56	Yes
SQL_API_SQLDATASOURCES	57	Yes
SQL_API_SQLDESCRIBEPARAM	58	Yes
SQL_API_SQLEXTENDEDFETCH	59	No
SQL_API_SQLFOREIGNKEYS	60	Yes
SQL_API_SQLMORERESULTS	61	Yes
SQL_API_SQLNATIVESQL	62	Yes
SQL_API_SQLNUMPARAMS	63	Yes
SQL_API_SQLPARAMOPTIONS	64	No
SQL_API_SQLPRIMARYKEYS	65	Yes
SQL_API_SQLPROCEDURECOLUMNS	66	Yes
SQL_API_SQLPROCEDURES	67	Yes
SQL_API_SQLSETPOS	68	No
SQL_API_SQLSETSCROLLOPTIONS	69	No
SQL_API_SQLTABLEPRIVILEGES	70	Yes
<i>SDK 2.0 Additions</i>		
SQL_API_SQLDRIVERS	71	Yes
SQL_API_SQLBINDPARAMETER	72	Yes

More Information on Specific Calls

SQLSetStmtOption / SQLGetStmtOption

Options supported: 3, 2, 1, 1151, and 1153.

SQLSetConnectOption / SQLGetConnectOption

Options supported: 101, 102, 109.

Note SQLMoreResults always returns SQL_NO_DATA_FOUND. The function is supported with its return value, due to the requirements of Microsoft Access.
