

ODBC Driver for 4D Server

リファレンス
Windows[®] and Mac[™]OS



OBDC Driver for 4D Serverリファレンス Windows® and Mac™ OS

Copyright© 1993 - 2000 4D SA

All rights reserved.

このマニュアルに記載されている事項は、将来予告なしに変更されることがあり、いかなる変更に関しても4D SAは一切の責任を負いかねます。このマニュアルで説明されるソフトウェアは、本製品に同梱のLicense Agreement (使用許諾契約書)のもとでのみ使用することができます。

ソフトウェアおよびマニュアルの一部または全部を、ライセンス保持者がこの契約条件を許諾した上での個人使用目的以外に、いかなる目的であれ、電子的、機械的、またどのような形であっても、無断で複製、配布することはできません。

4th Dimension、4D Server、4D、4Dロゴ、4Dロゴ、およびその他の4D製品の名称は、4D SAの商標または登録商標です。

Microsoft とWindows はMicrosoft Corporation 社の登録商標です。

Apple, Macintosh, Mac, Power Macintosh, Laser Writer, Image Writer, ResEdit, QuickTimeはApple Computer Inc.の登録商標または商標です。

その他、記載されている会社名、製品名は、各社の登録商標または商標です。

注意

このソフトウェアの使用に際し、本製品に同梱のLicense Agreement (使用許諾契約書) に同意する必要があります。ソフトウェアを使用する前に、License Agreementを注意深くお読みください。

第1章	アーキテクチャ全般	5
	ODBCアーキテクチャ	5
	ODBCドライバ	6
第2章	インストールとセットアップ	7
	必要条件	7
	クライアントマシン	7
	サーバマシン	7
	インストール	8
第3章	データソースの作成と変更	11
	ODBC Administratorバージョン2.5	12
	ユーザDSNの作成	12
	システムDSNの作成	14
	ODBC Administratorバージョン3.x	14
	ユーザDSNの作成	15
	non-sharableファイルDSNの作成	17
	システムDSNの作成	21
第4章	仕様について	23
	4D Serverへのアクセスの許可	23
	ストラクチャの制限値	24
	SQL文の制限値	24
	ヌル値	25
	ヌル値の選択	25
	ヌル値の更新	25
	浮動小数点の比較	25
	プライマリキー	25
第5章	ODBC の準拠レベル	27
	SQL準拠レベル	27
	Minimum SQL文法のサポート	27
	Core SQL文法のサポート	28

Extended SQL文法のサポート	28
API準拠レベル	29
スカラ関数	29
第6章 データ型の割り当て	31
4D ServerからODBC SQLデータ型への変換	31
ODBC SQLから4D Serverデータ型への変換	32
付録A SQL例題	33
例	33
単純なSELECT文	33
WHERE句の付いたSELECT文	34
BETWEEN述語の付いたSELECT文	34
前方一致条件の付いたSELECT文	35
文字フィールドでソートしたSELECT文	35
文字列関数の付いたSELECT文	36
数値関数の付いたSELECT文	37
GROUP BY句の付いたSELECT文	38
内部結合の付いたSELECT文	38
2つの異なるテーブルに対するUNIONの付いたSELECT文	39
IN述語の付いたSELECT文	41
構文についての注意	41
付録B API準拠レベル	43
API準拠レベル	43
特定の呼び出しについての詳細	45

ODBCは、Open DataBase Connectivityの略です。これは、SQL（構造化クエリ言語）を標準として使用し、アプリケーションがデータソースにクエリを行えるようにするインタフェースです。データソースとして、テキストファイルから非常に高度なDBMSまで、さまざまなタイプを使用できます。

ODBC Driver for 4D Serverでは、PC（Windows）アプリケーションから、ODBC呼び出しとSQLクエリによって、4D Serverと通信を行うことができます。このドライバは既存の4Dコンポーネント（Server APIおよびNetwork Component）を基礎にしており、当社のオープンアーキテクチャに完全に統合されています。このソリューションは、クライアントをアプリケーション側にインストールするだけで、既存の稼働中のデータベースに変更を加えることなく配備することができます。

この製品は、Microsoftが定義するOpen DataBase Connectivityインタフェースの規格に準拠しています。これは、X/OpenおよびSQL Access Group(SAG) SQL CAE仕様(1992)に基づくSQL構文をサポートします。

ODBCアーキテクチャ

4DのODBC Driver for 4D Serverを使用するODBCアーキテクチャは、以下の3つの部分から構成されます。

アプリケーション：アプリケーションはODBC APIで定義される関数を呼び出して、データソースにアクセスします。

ドライバマネージャ：ドライバマネージャはODBC APIを実装し、4D Serverドライバを動的にロードし、ODBC呼び出しの振り分けと管理を行います。

ODBC Driver for 4D Server：このドライバにより、ODBC準拠のアプリケーションを4D Serverデータベースに接続します。

ODBC ドライバ

ODBC Driver for 4D Serverは、4つのコンポーネントから構成されます。

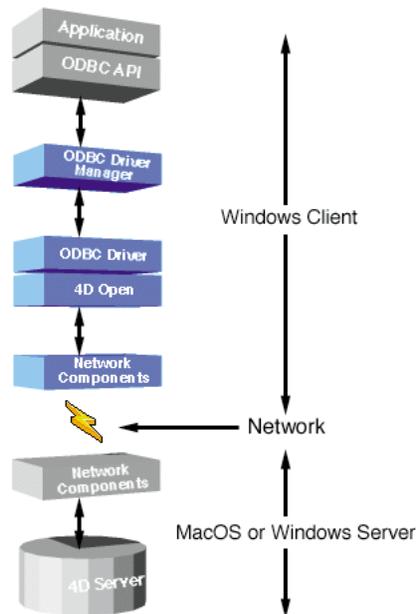
ODBCドライバ：ODBC Driver for 4D Serverそのもの。ドライバは、ODBC関数呼び出しを処理し、アプリケーションとデータソース間でデータのやりとりを管理します。

4D Open Library：4D Serverのクライアントのネイティブ部分です。

ネットワークコンポーネント：4D Serverのネットワーク部分です。

データソース：4D Serverがデータソースになります。つまり、アプリケーションがアクセスする必要があるデータソースです。

以下の図は、ODBC Driver for 4D ServerのODBCアーキテクチャを表わしています。



注：ODBC Driver for 4D Serverは、Windowsでのみ利用可能です。4D Serverアプリケーションは、MacintoshまたはWindowsのいずれにもインストールできますが、クライアントアプリケーションおよびドライバは、Windowsマシンにインストールする必要があります。

この章では、ODBC Driver for 4D Serverのハードウェアとソフトウェアの必要条件、およびCD-ROMからインストールする方法を説明します。

必要条件

クライアントマシン

ハードウェア

クライアントマシンは、Windows95/98、WindowsNT3.5.1またはWindowsNT4.0のいずれかの32ビットOSがインストールされたPCである必要があります。

ソフトウェア

ODBC準拠のアプリケーションは、ODBCドライバを操作できる必要があります。

サーバマシン

ハードウェア

サーバマシンは次のいずれかです。

32ビットシステム（Windows95/98、WindowsNT3.5.1またはWindowsNT4.0）が稼動するPC。

MacOS System7.x以上が稼動するPower Macintosh。

ソフトウェア データソースとして、4D Serverバージョン6.0.5以上である必要があります。

インストール

この節では、ODBC Driver for 4D Serverのインストール時に自動的に起動される“SETUP”プログラムについて説明します。

“SETUP.exe”プログラムは、ODBC Driver for 4D Serverのインストールに必要なすべてのタスクを実行します。このプログラムは、ODBC.INI、ODBCINST.INIおよびレジストリデータベースといったファイルも同時に設定します。

ODBC Driver for 4D Serverをインストールするには、次の手順に従ってください。

- 1 “セットアップアイコン”をダブルクリックする。
「Welcome」ダイアログボックスが表示されます。



- 2 「Welcome」ダイアログボックスで「Continue」をクリックして、インストールを進める。
「ODBCデータソースアドミニストレータ」ウィンドウが表示されます。



データソースを今ここで定義する場合は、第3章「データソースの作成と変更」を参照してください。後で行う場合は、「キャンセル」ボタンをクリックします。インストールが正常に終了したことを示すメッセージが表示されます。



ODBCアーキテクチャは、アプリケーションが1つまたは複数のデータソースにアクセスすることをベースとしています。データソースは、各々のクライアント上で定義し、設定する必要があります。データソース名(DSN)は、以下の3つのレベルで定義することができます。

ユーザDSN

システムDSN

ファイルDSN

各レベルはそれぞれ特定のアプリケーションに対応しており、独自の方法で使用されます。クライアントソフトウェアは必ずしもすべてのレベルをサポートするわけではないので、注意が必要です。例えば、Microsoft Office97の“MS Query”は、ファイルDSNでのみ機能します。

第3章と第4章では、ドライバマネージャ及びODBC Administratorバージョン2.5または3.xを使用して、データソースを作成する方法を説明します。

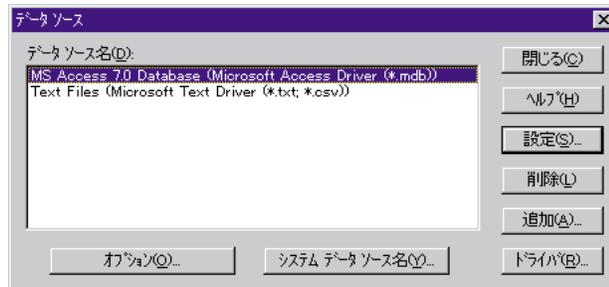
データソースを作成する時、どのバージョンのドライバマネージャを使用するか、またどのレベルのデータソースが必要なかを判断するには、それぞれのODBC対応アプリケーションのマニュアルを参照してください。

ODBC Administrator バージョン 2.5

ユーザDSNの作成

ユーザDSNを作成するには、次の手順に従ってください。

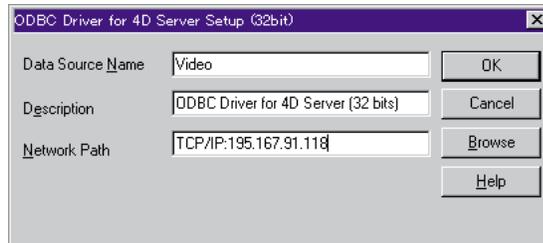
- 1 コントロールパネルでODBC Administratorアイコンをダブルクリックして、ODBC Administratorを起動する。
「データソース」ダイアログボックスが表示されます。



- 2 データソースを追加するには、「追加」ボタンをクリックする。
「データソースの追加」ダイアログボックスが表示され、マシンにインストールされているODBCドライバの一覧が示されます。



- 3 ODBC Driver for 4D Serverを選択して、「OK」をクリックする。
「ODBC Driver for 4D Server Setup(32bit)」ダイアログボックスが表示されます。

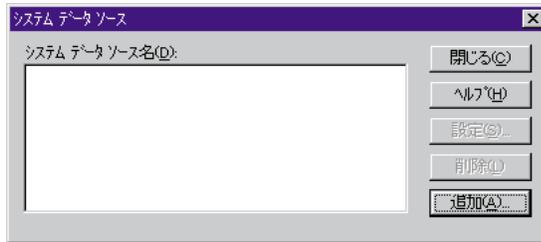


- 4 「Data Source Name」テキストボックスに、データソース名を入力する。
データソース名は任意の名前です。
- 5 「Description」テキストボックスに、ドライバの説明を入力する。
デフォルトでは、「4D ODBC Driver for 4D Server (32bit)」となっていますが、これは好きな文字列を指定することができます。
- 6 「Network Path」テキストボックスに、4D Serverデータソースへのアクセスに使用する場所とプロトコルを入力する。これは以下の規則に従って入力する必要がある。
ネットワークパスは、ネットワークで使用されているネットワークプロトコルに依存します。
Appletalkプロトコル : "Appletalk:DatabaseName@ZoneName"と入力します。
TCPプロトコル : "TCP/IP:Address,{PortNumber}"と入力します。
IPXプロトコル : "IPX:Address"と入力します。
「Browse」ボタンを使って、4D Serverのデータソースを選択することもできます。その場合は、4D Serverデータソースを選択した後、「Network Path」テキストボックスに表示されているネットワークパスを変更しないでください。
詳細については、『4D Serverリファレンス』および『4D Serverネットワークコンポーネント』を参照してください。
- 7 「OK」ボタンをクリックする。
ユーザDSNが一覧に追加されます。

システムDSNの作成

システムDSNを作成するには、次の手順に従ってください。

- 1 「データソース」ダイアログボックスで、「システムデータソース名」ボタンをクリックする。
「システムデータソース名」ウインドウが表示されます。



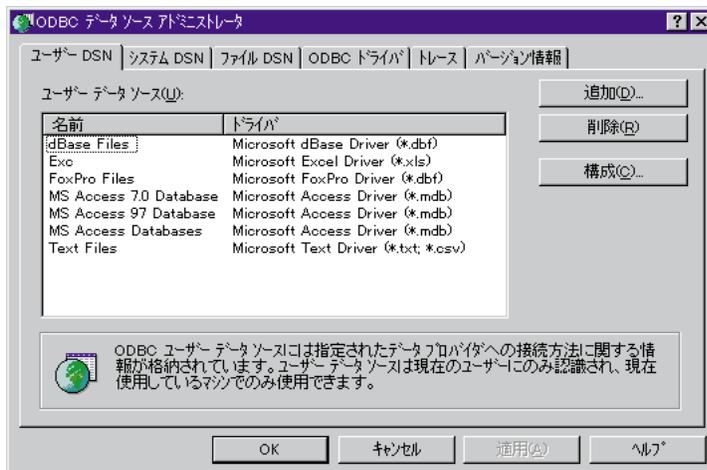
- 2 「追加」ボタンをクリックして、「ユーザDSNの作成」のステップ2から7に従い、システムデータソースを追加する。

ODBC Administrator バージョン 3.x

ODBC Administratorを開始するには、次の手順に従ってください。

コントロールパネルでODBCアイコンをダブルクリックする。

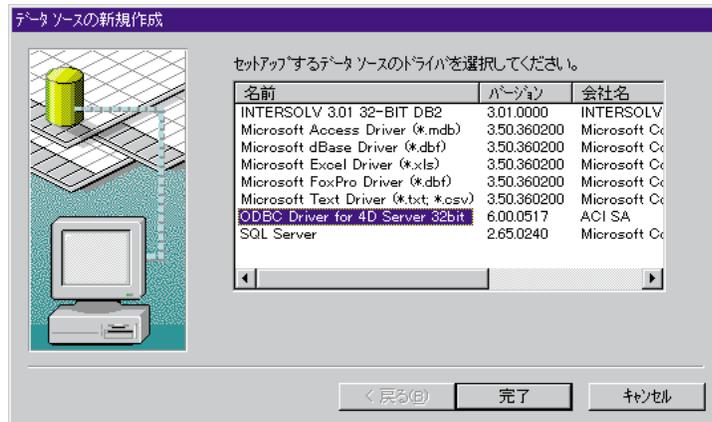
「ODBC データソースアドミニストレータ」ウインドウが表示されます。



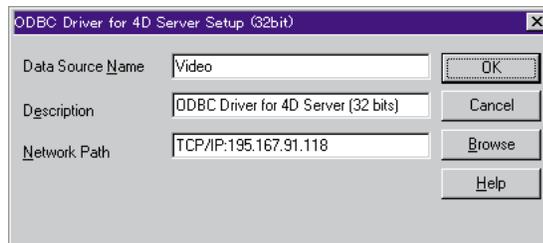
ユーザDSNの作成

ユーザDSNを作成するには、次の手順に従ってください。

- 1 「追加」 ボタンをクリックする。
「データソースの新規作成」ダイアログボックスが表示され、ドライバの一覧が示されます。



- 2 ODBC Driver for 4D Server(32 bits)を選択してから、「完了」をクリックする。
「ODBC Driver for 4D Server Setup(32bit)」ダイアログボックスが表示されます。



- 3 「Data Source Name」テキストボックスに、データソース名を入力する。
データソース名は任意の名前です。
- 4 「Description」テキストボックスに、ドライバの説明を入力する。
デフォルトでは、「4D ODBC Driver for 4D Server(32bit)」となっていますが、これは好きな文字列を指定することができます。
- 5 「Network Path」テキストボックスに、4D Serverデータソースへのアクセスに使用する場所とプロトコルを入力する。これは以下の規則に従って入力する必要がある。
ネットワークパスは、ネットワークで使用されているネットワークプロトコルに依存します。

Appletalkプロトコル : "Appletalk:DatabaseName@ZoneName"と入力します。

TCPプロトコル : "TCP/IP:Address,{PortNumber}"と入力します。

IPXプロトコル : "IPX:Address"と入力します。

「Browse」ボタンを使って、4D Serverのデータソースを選択することもできます。その場合は、4D Serverデータソースを選択した後、「ODBC Driver for 4D ServerSetup (32bit)」に表示されるNetworkPathを変更しないでください。

詳細については、『4D Serverリファレンス』および『4D Serverネットワークコンポーネント』を参照してください。

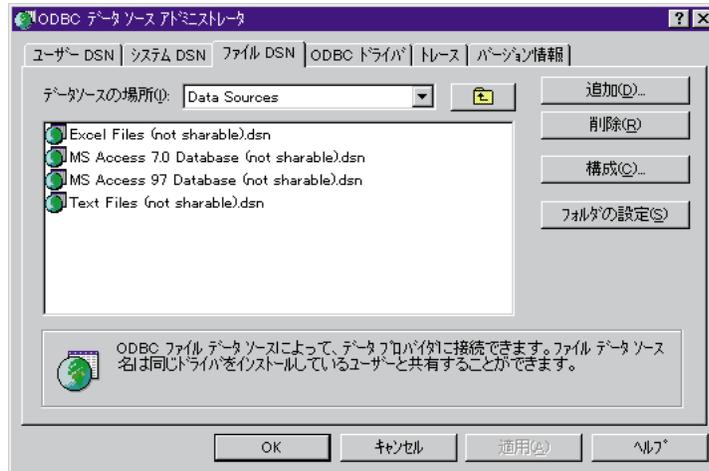
- 6 「OK」ボタンをクリックする。
ユーザDSNが一覧に追加されます。

non-sharableファイルDSNの作成

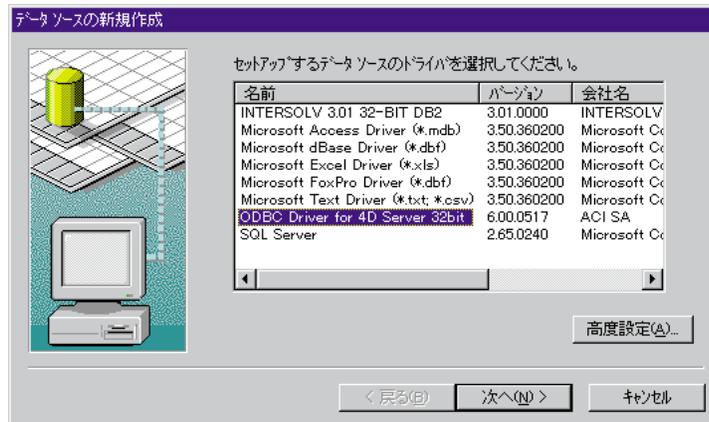
最初に、前節の「ユーザDSNの作成」を参照してください。non sharableファイルDSNはユーザDSNへのポインタです。

non-sharableファイルDSNを作成するには、次の手順に従ってください。

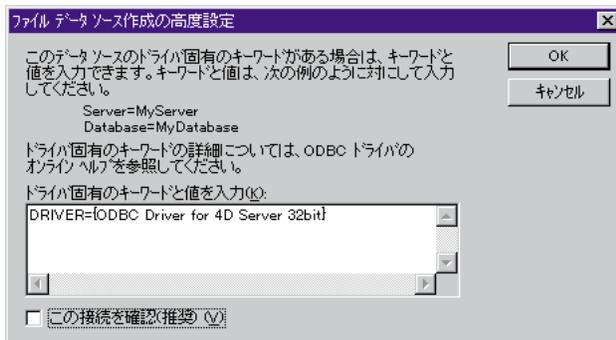
- 1 「ODBCデータソースアドミニストレータ」ウインドウで「ファイルDSN」タブをクリックする。



- 2 「追加」ボタンをクリックする。
「データソースの新規作成」ダイアログボックスが表示されます。



- 3 ODBC Driver for 4D Server(32 bits)を選択してから、「詳細設定」をクリックする。
「ファイルデータソース作成の詳細設定」ダイアログボックスが表示されます。

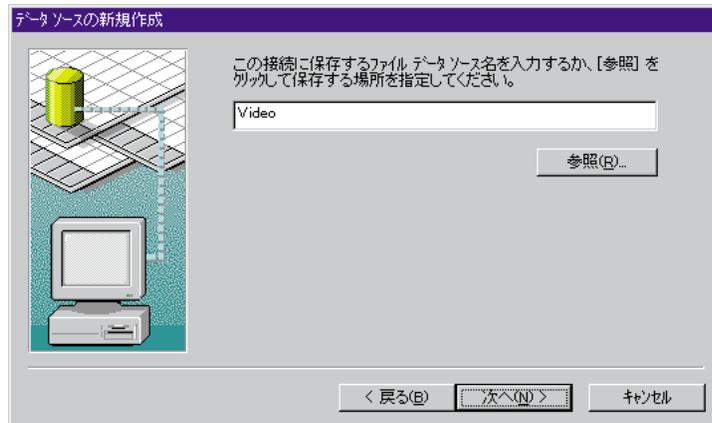


- 4 「この接続を確認(推奨)」チェックボックスの選択を解除し、「OK」をクリックする。
- 5 「データソースの新規作成」ダイアログボックスで「次へ」をクリックする。



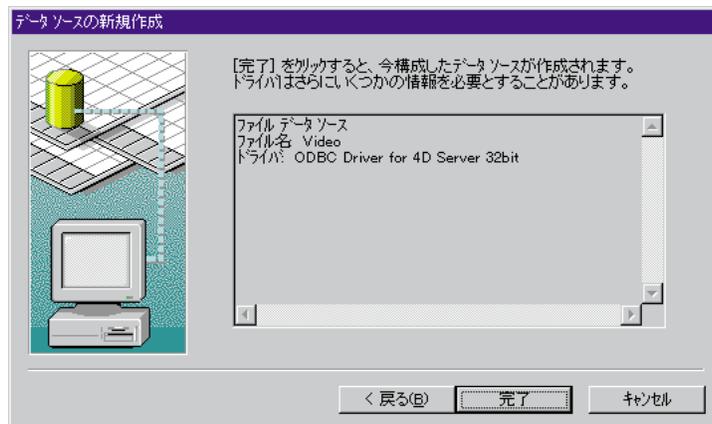
データソースの名前を入力できるダイアログボックスが表示されます。

- 6 テキストボックスに、データソースの名前（この例ではVideo）を入力する。

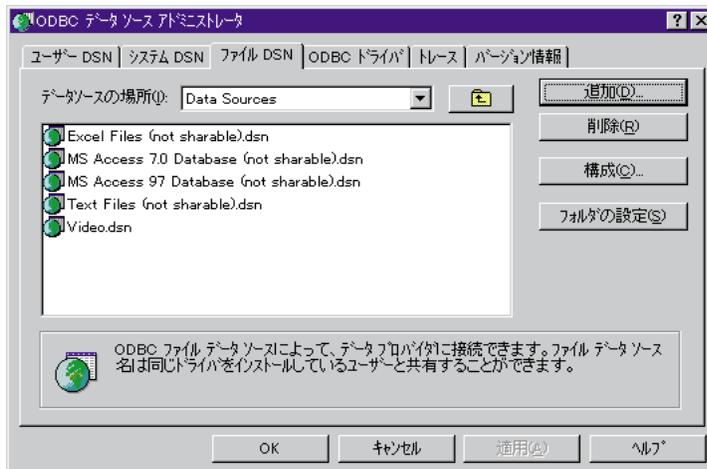


- 7 「次へ」をクリックする。

情報ダイアログボックスに、ファイルデータソースの定義の要約が表示されます。



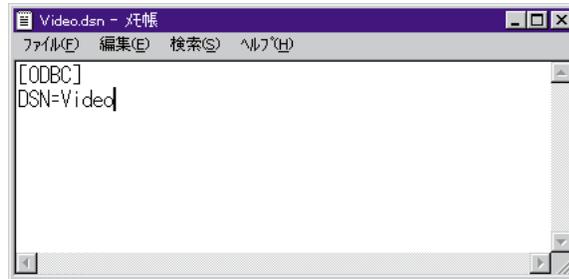
- 8 情報をすべて確認したら、「完了」をクリックする。
ファイルDSNが一覧に追加されます。



- 9 「OK」をクリックして、ODBC Administrator3.xを終了する。
- 10 メモ帳 (Notepad.exe) を使用して、video.dsnファイルを開く。
“video.dsn”ファイルは、“System folder¥ODBC¥Data sources”パスのシステムフォルダにあります。
ファイルの内容は、次のようになります。



- 11 既存の “ DRIVER=ODBC Driver for 4D Server(32bits) ” というテキストを “ DSN=ファイルデータソース名 ” に置き換える。
この例では、 “ DSN=Video ” を入力します。

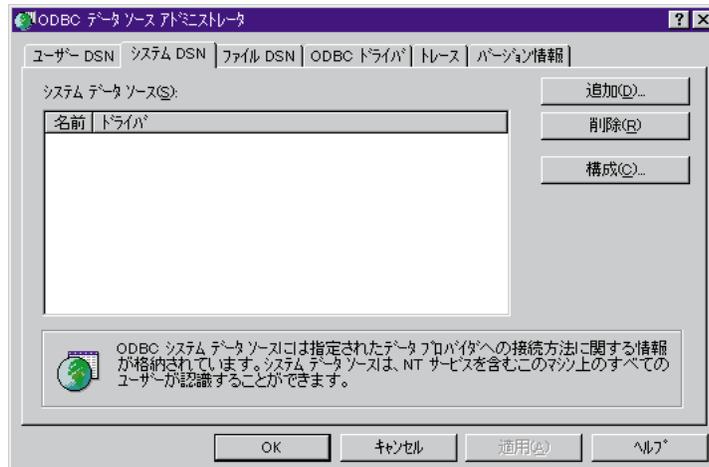


- 12 ファイルDSNを保存し、メモ帳を終了する。

システムDSNの作成

システムDSNを作成するには、次の手順に従ってください。

- 1 「ODBCデータソースアドミニストレータ」ウインドウで「システムDSN」タブをクリックする。



- 2 「追加」ボタンをクリックして、「ユーザDSNの作成」のステップ2から6に従い、システムデータソースを追加する。

この章では、ODBC Driver for 4D Serverの使用方法に関する仕様を説明します。

4D Serverへのアクセスの許可

ODBC Driver for 4D Serverは、4D Openに基づいています。4D Openは、4D Client以外のアプリケーションが4D Serverに接続できるようにするAPI (Application Programming Interface) です。

ODBCドライバを使用して4D Serverにアクセスするには、4D Open接続を有効にする必要があります。

次の手順に従ってください。

- 1 4Dまたは4D Clientでストラクチャファイルを開き、「デザイン」モードに移行する。
- 2 「ファイル」メニューで「データベースプロパティ」を選択する。
- 3 「データ制御&アクセス権」タグを選択する。
- 4 「4D Openの接続を許可する」チェックボックスをONにする。

ストラクチャの制限値

説明制限値	制限値
データベース当たりテーブル数	255
テーブル当たりの列数	511
テーブル当たりの行数	1600万
単一のクエリで取得できる列数	300
列のサイズ	2ギガバイト
テーブル名の最大長	31バイト
列名の最大長	31バイト
ユーザ名の最大長	30バイト
接続数	4D Serverのクライアント トライセンスに依存
SQL文の長さ	16K
接続当たりのステートメント数	メモリに依存
テーブルクオリファイア	OSに依存
結合できるテーブルの数	メモリに依存

4Dに関する詳細は、『4th Dimensionランゲージリファレンス』を参照してください。

SQL文の制限値

リテラルで最大1000文字

ANDされた述部は最大300

文字列中の文字は、10進数で1～255の間の任意のANSI文字です。シングルクォーテーション (') は、2つの連続するシングルクォーテーション (") で表現する必要があります。

ヌル値

ヌル値の選択

!00/00/00!という4Dの日付フィールドは、ヌル値として扱われます。
空の4Dテキストフィールドまたは文字フィールドは、ヌル値として扱われます。

ヌル値の更新

ヌルのSQL_DATEまたはSQL_TIMESTAMPは、4D Serverでは、!00/00/00!に等しい4D日付フィールドとして更新されます。

ヌルのSQL_CHAR、SQL_VARCHARまたはSQL_LONGVARCHARは、空の4D文字フィールドまたはテキストフィールドとして更新されます。

例えば、以下のコードでは、SELECT文が2および3を返します。

```
CLIENTS.ID      CLIENTS.String
    1              MyString
    2              NULL
    3              "(2つの連続するシングルクォーテーション)"
SELECT CLIENTS.ID
FROM CLIENTS
WHERE CLIENTS.String IS NULL
```

浮動小数点の比較

4D Serverへのアクセスに使用するODBC準拠のアプリケーションは、4D Serverとは違う方法でデータを格納することができます。この相違により、比較を含むクエリにおいて、近似値や異なる結果セットが生じることがあります。

プライマリキー

ODBC driver for 4D Serverは、(作成順で)最初の一意の、必須かつインデックスの付いた列だけをプライマリキーと見なします。

SQL準拠レベル

ODBCドライバの主な特性を表すものとして、準拠レベルがあります。サポートする文法により、いくつかのレベルがあります。ODBCバージョン2.5の仕様にはSQL文法の準拠レベルとして、Minimum、Core、Extendedの3つがあります。上位レベルは下位レベルに比べ、データ定義及びデータ操作言語が高度にインプリメントされています。

各レベルの説明により、ODBC driver for 4D Serverで何がサポートされているかがわかります。

Minimum SQL文法のサポート

ODBC driver for 4D Serverは、『Microsoft ODBC Programmer's Reference』の付録Cに定義されている、Minimum SQL文法を一部サポートします。

次は、サポートされている項目の一覧です。

サポートされるデータ操作言語（DML）：単純なSELECT、INSERT、検索を伴ったUPDATE、およびDELETE文

サポートされる式：Simple

サポートされるデータ型：CHAR、VARCHAR、LONG VARCHAR

Core SQL文法のサポート

ODBC driver for 4D Serverは、『Microsoft ODBC Programmer's Reference』の付録Cに定義されている、Core SQL文法を一部サポートします。

次は、サポートされている項目の一覧です。

サポートされるデータ操作言語 (DML) : 全SELECT文
サポートされる式 : サブクエリ、埋め込み関数 (SUM、MAX、MIN、AVG、COUNT)
Approximate-numeric-literal
Between述語 : BETWEEN
相関名
Exact-numeric literal
In述語 : IN
サポートされるデータ型 : DECIMAL、SMALLINT、INTEGER、REAL、DOUBLEPRECISION

Extended SQL文法のサポート

ODBC driver for 4D Serverは、『Microsoft ODBC Programmer's Reference』の付録Cに定義されている、Extended SQL文法を一部サポートします。

次は、サポートされている項目の一覧です。

DML : 外部結合、UNIONおよびUNION ALL
式 : スカラ関数、日付および時間文字
日付計算
拡張された述語
データ型 : BIT、TINYINT、DATE、TIME、TIMESTAMP、LONG VARBINARY
SQL文の例については、本書の「付録A」を参照してください。

API準拠レベル

ODBCドライバは、一連のODBC呼び出しによってアクセスされます。各ドライバは、標準をサポートする必要性に応じて、これらの呼び出しの実装レベルを変えています。多くの場合、アプリケーションレベルではこのプログラミングのレベルに注意する必要はありません。

使用するアプリケーションは、これらの呼び出しを実装してデータソースにアクセスするので、ユーザはこのAPIを経由して送信されるSQLクエリを制御するだけです。

アプリケーションの中でODBC APIを直接プログラミングする場合は、本書の「付録B」に、このドライバで実装されているコールの完全なリストがあります。

スカラ関数

ODBC Driver for 4D Serverは、SQL文中のODBCスカラ関数をサポートします。これらの関数のSQL構文については、『Microsoft ODBC Programmer's Reference』の付録Gを参照してください。

次は、ODBC Driver for 4D Serverによってサポートされるスカラ関数の一覧です。

文字列関数：CONCAT、LENGTH、RIGHT、UCASE、LCASE、LOCATE、RTRIM、LEFT、LTRIM、SUBSTRING

数値関数：MOD

日付関数：DAYOFMONTH、MONTH、DAYOFWEEK、YEAR

システム関数：IF、NULL

変換関数：CONVERT

4D Serverは、データ型のセットをサポートします。ODBC Driver for 4D Serverは、これらのデータ型を適切なODBC SQLデータ型に割り当てます。以下の表は、4D Serverのデータ型とそれに対応するODBC SQLデータ型を示します。

4D ServerからODBC SQLデータ型への変換

4D Serverのデータ型	ODBC SQLのデータ型
文字	SQL_VARCHAR
テキスト	SQL_LONGVARCHAR
実数	SQL_DOUBLE
整数	SQL_SMALLINT
倍長整数	SQL_INTEGER
日付	SQL_DATE
時間	SQL_TIME
ブール	SQL_BIT
ピクチャ	SQL_LONGVARBINARY
サブテーブル	なし
BLOB	なし

ODBC SQLから4D Serverデータ型への変換

ODBC SQLのデータ型	4D Serverのデータ型
SQL_VARCHAR	文字
SQL_CHAR	文字
SQL_LONGVARCHAR	テキスト
SQL_REAL	実数
SQL_DOUBLE	実数
SQL_DECIMAL	実数
SQL_SMALLINT	整数
SQL_TINYINT	整数
SQL_INTEGER	倍長整数
SQL_LONGVARBINARY	ピクチャ
SQL_DATE	日付
SQL_TIMESTAMP	日付
SQL_TIME	時間
SQL_BIT	ブール

SQL言語の学習者に役立つようにSQLの文法（第5章を参照してください）を説明し、いくつかのクエリの例とその結果を示します。例として付属の“VIDEO”データベースを使用します。

例

単純なSELECT文

この例は、単純なSELECT文を使用しています。このクエリは、CUSTOMERSテーブルのすべてのFirstnameフィールドの内容を返します。

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

結果は次の通りです。

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

WHERE句の付いたSELECT文

この例は、SELECT文をWHERE句とともに使用しています。このクエリは、レンタル料が1500を超えるレコードを検索し、VIDEOSテーブルからビデオタイトルとレンタル料を返します。

```
SELECT
VIDEOS.Title, VIDEOS.RentingPrice
FROM
VIDEOS
WHERE
(VIDEOS.RentingPrice>1500)
```

結果は次の通りです。

Demoiselles de Rochefort (Les)	2400
Missing in Action I	2300
Jane Fonda's 10min Workout	5000
Slippery When Wet	2800
Living on a Prayer	4200
Wyle Traps the Road Runner!	2600

BETWEEN述語の付いたSELECT文

この例は、SELECT文をBETWEEN述語とともに使用しています。このクエリは、レンタル料が1500以上で2500以下のレコードを検索し、VIDEOSテーブルからビデオタイトルとレンタル料を返します。

```
SELECT
VIDEOS.Title, VIDEOS.RentingPrice
FROM
VIDEOS
WHERE
(VIDEOS.RentingPrice BETWEEN 1500 And 2500)
```

結果は次の通りです。

Demoiselles de Rochefort (Les)	2400
Missing in Action I	2300
Boyz in New Jersey	1800

前方一致条件の付いたSELECT文

この例は、SELECT文を前方一致条件とともに使用しています。このクエリは、ビデオタイトルが大文字の“M”で始まるレコードを検索し、VIDEOSテーブルからビデオタイトルとレンタル料を返します。

```
SELECT
VIDEOS.Title, VIDEOS.RentingPrice
FROM
VIDEOS
WHERE
```

```
(VIDEOS.Title Like 'M%')
```

結果は次の通りです。

Missing in Action I	2300
---------------------	------

文字フィールドでソートしたSELECT文

この例は、SELECT文を文字フィールドでのソートとともに使用しています。このクエリは、レンタル料が1500未満のレコードを検索し、VIDEOSテーブルからビデオタイトルとレンタル料を、ビデオタイトルのアルファベット順に並べてから返します。

```
SELECT
VIDEOS.Title, VIDEOS.RentingPrice
FROM
VIDEOS
WHERE
```

```
(VIDEOS.RentingPrice<1500)
```

```
ORDER BY
```

```
VIDEOS.Title
```

結果は次の通りです。

Blaze of Glory	1300
Chip & Jeff's Excellent Adventure	1400
Night of the Living Noni3s	1200
Preateurs (Les)	1200
Return of Cujo	1000

文字列関数の付いたSELECT文

次の2つの例は、SELECT文を文字列関数と組み合わせて使用しています。

例1：このクエリは、レンタル料が21.1を超えるレコードを検索します。そしてビデオタイトルの内容を大文字に変換し、その最初の10文字を返します。

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

結果は次の通りです。

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

例2：このクエリは、VIDEOSテーブルからビデオタイトルおよび、タイトル中に含まれる小文字の“a”の先頭からの位置を返します。

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

結果は次の通りです。

Demoiselles de Rochefort (Les)	0
Privateurs (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 Nones	0
Boyz in New Jersey	0
Runaway	4
Living on a Prayer	11
Wyle Traps the Road Runner!	8

数値関数の付いたSELECT文

次の3つの例は、SELECT文を数値関数と組み合わせて使用しています。

例1：このクエリは、VIDEOSテーブルより全レコードの平均のレンタル料を返します。

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

結果は次の通りです。

```
2300
```

例2：このクエリは、VIDEOS テーブルの中で一番安いレンタル料を返します。

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

結果は次の通りです。

```
1000
```

例3：このクエリは、VIDEOSテーブルの中で一番高いレンタル料を返します。

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

結果は次の通りです。

```
5000
```

GROUP BY句の付いたSELECT文

この例は、SELECT文をGROUP BY句と組み合わせて使用しています。このクエリは、Count関数とGROUP BY句を使って、VIDEOSテーブルより各カテゴリー（クラシック、コメディなど）ごとのビデオの数を返します。

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

結果は次の通りです。

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

内部結合の付いたSELECT文

この例は、CLIENTおよびTRACKINGテーブルに対して、SELECT文を内部結合と組み合わせて使用します。このクエリは、結合を使って複数のテーブルから一度にデータを選択する方法を示しています。

```
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
```

結果は次の通りです。

Jon	Bon Jovi	1997-08-02	Demoiselles de Rochefort (Les)	Comedy
Jeff	Davis	1992-12-26	Predateurs (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	Predateurs (Les)	Fantastic
Dave	Popielarz	1992-12-15	Return of Cujo	Martial Arts

2つの異なるテーブルに対するUNIONの付いたSELECT文

この例は、2つの異なるテーブルに対して、SELECT文をUNIONとともに使用しています。このクエリは、UNION機能を使用して、2つのテーブルからデータを取得しています。

注：この例では、ビデオカテゴリーと顧客名のUNIONを作成しても意味がありませんが、単なる例だと考えてください。

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

結果は次の通りです。

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

IN述語の付いたSELECT文

この例は、SELECT文をIN述語と組み合わせて使用しています。このクエリは、顧客から返ってきているはずのビデオを取得します。このクエリは、VIDEOSとTRACKINGテーブルを使用して実行されます。まず初めに、TRACKINGテーブルからDays lateフィールドが0より大きいレコードを検索します。そして検索されたTRACKINGテーブルのVide_IDと一致するVideo_IDを今度はVIDEOSテーブルより検索します。条件を満たすレコードのVIDEOS.TitleがVIDEOSテーブルより返されます。

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

結果は次の通りです。

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

構文についての注意

4Dのフィールド名にスペースが含まれる場合、構文エラーを回避するために、フィールド名を引用符で囲む必要があります。

例

```
SELECT
TRACKING. "Days Late"
FROM
TRACKING
```


API準拠レベル

次に、ODBC関数とそのAPI準拠レベルを示します。

ODBC関数およびレベル	ID	ドライバによるサポート
コアレベル		
SQL_API_SQLALLOCCONNECT	1	はい
SQL_API_SQLALLOCENV	2	はい
SQL_API_SQLALLOCSTMT	3	はい
SQL_API_SQLBINDCOL	4	はい
SQL_API_SQLCANCEL	5	はい
SQL_API_SQLCOLATTRIBUTES	6	はい
SQL_API_SQLCONNECT	7	はい
SQL_API_SQLDESCRIBECOL	8	はい
SQL_API_SQLDISCONNECT	9	はい
SQL_API_SQLERROR	10	はい
SQL_API_SQLEXECDIRECT	11	はい
SQL_API_SQLEXECUTE	12	はい
SQL_API_SQLFETCH	13	はい
SQL_API_SQLFREECONNECT	14	はい
SQL_API_SQLFREEENV	15	はい
SQL_API_SQLFREESTMT	16	はい
SQL_API_SQLGETCURSORNAME	17	はい
SQL_API_SQLNUMRESULTCOLS	18	はい
SQL_API_SQLPREPARE	19	はい
SQL_API_SQLROWCOUNT	20	はい
SQL_API_SQLSETCURSORNAME	21	はい
SQL_API_SQLSETPARAM	22	はい
SQL_API_SQLTRANSACT	23	はい

ODBC関数およびレベル	ID	ドライバによるサポート
レベル1		
SQL_API_SQLCOLUMNS	40	はい
SQL_API_SQLDRIVERCONNECT	41	はい
SQL_API_SQLGETCONNECTOPTION	42	はい
SQL_API_SQLGETDATA	43	はい
SQL_API_SQLGETFUNCTIONS	44	はい
SQL_API_SQLGETINFO	45	はい
SQL_API_SQLGETSTMTOPTION	46	はい
SQL_API_SQLGETTYPEINFO	47	はい
SQL_API_SQLPARAMDATA	48	はい
SQL_API_SQLPUTDATA	49	はい
SQL_API_SQLSETCONNECTOPTION	50	はい
SQL_API_SQLSETSTMTOPTION	51	はい
SQL_API_SQLSPECIALCOLUMNS	52	はい
SQL_API_SQLSTATISTICS	53	はい
SQL_API_SQLTABLES	54	はい
レベル2		
SQL_API_SQLBROWSECONNECT	55	はい
SQL_API_SQLCOLUMNPRIVILEGES	56	はい
SQL_API_SQLDATASOURCES	57	はい
SQL_API_SQLDESCRIBEPARAM	58	はい
SQL_API_SQLEXTENDEDFETCH	59	いいえ
SQL_API_SQLFOREIGNKEYS	60	はい
SQL_API_SQLMORERESULTS	61	はい
SQL_API_SQLNATIVESQL	62	はい
SQL_API_SQLNUMPARAMS	63	はい
SQL_API_SQLPARAMOPTIONS	64	いいえ
SQL_API_SQLPRIMARYKEYS	65	はい
SQL_API_SQLPROCEDURECOLUMNS	66	はい
SQL_API_SQLPROCEDURES	67	はい
SQL_API_SQLSETPOS	68	いいえ
SQL_API_SQLSETSCROLLOPTIONS	69	いいえ
SQL_API_SQLTABLEPRIVILEGES	70	はい

SDK2.0追加		
SQL_API_SQLDRIVERS	71	はい
SQL_API_SQLBINDPARAMETER	72	はい

特定の呼び出しについての詳細

SQLSetStmtOption/SQLGetStmtOption

サポートされるオプション : 3、2、1、1151、および1153

SQLSetConnectOption / SQLGetConnectOption

サポートされるオプション : 101、102、109

注 : SQLMoreResultsは常にSQL_NO_DATA_FOUNDを返します。この関数は、Microsoft Accessの必要条件のために、その戻り値とともにサポートされています。

