Developing ADO-based Applications for D3

D3 Version:	D3 Unix 7.1 and higher D3 NT 7.1 and higher
Doc Version:	January 11, 2000

Developing ADO-based Applications for D^3

Introduction

This document provides D^3-specific information for the developer creating ADO-based applications that will access a D^3 database. The document contains the following information:

- Background about Microsoft's transition from ODBC to OLE DB and ADO.
- D^3 components needed to use ADO.
- Information about ADO installation.
- Available methods and properties for ADO objects.
- Syntax for connecting to a D^3 database.

For additional information about ADO and OLE DB, go to Microsoft's Universal Data Access Web site (www.microsoft.com/data).

Background

Over the years, ODBC has become the industry-standard interface for "connecting" client applications to databases. In recent years, however, Microsoft revamped it's data access strategy and delivered *OLE DB* as the successor to ODBC. In addition, a high-level programming interface called *ADO* (ActiveX Data Objects) has been introduced to work on top of OLE DB, just as DAO and RDO work with ODBC.

To assist in the transition from ODBC-based development to OLE DB-based development, Microsoft has provided an OLE DB provider that allows ADO to connect to any ODBC data source. This provider is called Microsoft's *OLE DB Provider for ODBC*, also known as *MSDASQL*.

Required D3 Components

The following D3 components are required to develop ADO-based applications that access a D3 database:

- Any D3 Unix or D3 NT server product, version 7.1 or above.
- D3 ODBC client software, version 7.1.C17 or above.
- D3 ODBC server software, version 7.1.S9 or above.

Installing ADO

ADO is automatically installed and registered with components of Visual Studio 6.0 (Visual Basic, Visual C++, and Visual Interdev). Users of Visual Studio 5.0 components can download for free the *Microsoft Data Access Components (MDAC) 2.0 SDK*, which contains ADO. This download is available at www.microsoft.com/data.

NOTE—Before using ADO, you must first install the D3 ODBC client software. See the Pick Systems *D3 ODBC and D3 SQL User's Manual* for how to install the D3 ODBC client software.

Available Methods and Properties of ADO Objects

The ADO programming model has seven objects:

- Connection
- Command
- Parameter
- Recordset
- Field
- Property
- Error

The following tables list the methods and properties of each ADO object and which methods and properties are available to developers using the *OLE DB Provider for ODBC* with D^3 ODBC.

Connection Object

Methods

The Connection object has the eight methods shown in the following table. Seven of these methods are available to developers using the *OLE DB Provider for ODBC* with D^3 ODBC.

Connection Methods	Available
BeginTrans	Yes
Cancel	No
Close	Yes
CommitTrans	Yes
Execute	Yes*
Open	Yes

Connection Methods (cont.)	Available
OpenSchema	Yes**
RollbackTrans	Yes

*Available options – adCmdText, adCmdTable, adExecuteNoRecords
**Available options – adSchemaIndexes, adSchemaProviderTypes, adSchemaTables

Properties

The Connection object has the eleven properties shown in the following table. Eight of these properties are available to developers using the *OLE DB Provider for ODBC* with D^3 ODBC.

Connection Properties	Available
Attributes	Yes
CommandTimeout	No
ConnectionString	Yes
ConnectionTimeout	No
CursorLocation	Yes*
DefaultDatabase	No
IsolationLevel	Yes**
Mode	Yes
Provider	Yes
State	Yes
Version	Yes

*Available options – adUseServer, adUseClient
**Available options – adXactReadUncomitted, adXactReadCommitted, adXactRepeatableRead, adXactSerializable

Command Object

The Command object is not typically needed because stored procedures and queries are not available in D^3 and the Connection object allows execution of commands. Nonetheless, it can be used to execute SQL statements in the same way as the Execute method of the Connection object.

Methods

The Command object has the three methods shown in the following table. One of these methods is available to developers using the *OLE DB Provider for ODBC* with D^3 ODBC.

Command Methods	Available
Cancel	No
CreateParameter	No
Execute	Yes*

*Available options – adCmdText, adCmdTable

Properties

The Command object has the seven properties shown in the following table. Three of these properties are available to developers using the *OLE DB Provider for ODBC* with D^3 ODBC.

Command Properties	Available
ActiveConnection	Yes
CommandText	Yes
CommandTimeout	No
CommandType	Yes*

Command Properties (cont.)	Available
Name	No
Prepared	No
State	No

*Available options – adCmdText, adCmdTable, adExecuteNoRecords

Parameter Object

The Parameter object is not used because stored procedures and queries are not available in D^3.

Recordset Object

Methods

The Recordset object has the twenty-four methods shown in the following table. Twenty-one of these methods are available to developers using the *OLE DB Provider for ODBC* with D^3 ODBC.

Recordset Methods	Available
AddNew	Yes
Cancel	No
CancelBatch	Yes
CancelUpdate	Yes
Clone	Yes
Close	Yes
CompareBookmarks	Yes
Delete	Yes
Find	Yes
GetRows	Yes

Recordset Methods (cont.)	Available
GetString	Yes
Move	Yes
MoveFirst	Yes
MoveLast	Yes
MoveNext	Yes
MovePrevious	Yes
NextRecordset	No
Open	Yes
Requery	Yes
Resync	No
Save	Yes
Supports	Yes
Update	Yes
UpdateBatch	Yes

Properties

The Recordset object has the twenty-two properties shown in the following table. Thirteen of these properties are available to developers using the *OLE DB Provider for ODBC* with D^3 ODBC.

Recordset Properties	Available
AbsolutePage	No
AbsolutePosition	No
ActiveConnection	Yes
BOF	Yes
Bookmark	Yes

Recordset Properties (cont.)	Available
CacheSize	Yes
CusorLocation	Yes*
CursorType	Yes**
EditMode	Yes
EOF	Yes
Filter	Yes
LockType	Yes***
MarshalOptions	No
MaxRecords	No
PageCount	No
PageSize	No
RecordCount	No
Sort	No
Source	Yes
State	Yes****
Status	Yes
StayInSync	No

*Available options – adUseServer, adUseClient
**Available options – adOpenForwardOnly, adOpenStatic
***Available options – adLockReadOnly, adLockOptimistic, adLockBatchOptimistic
****Available options – adStateOpen, adStateClosed

Field Object

Methods

The Field object has the two methods shown in the following table. None of these methods are available to developers using the *OLE DB Provider for ODBC* with D^3 ODBC.

Field Methods	Available
AppendChunk	No
GetChunk	No

Properties

The Field object has the ten properties shown in the following table. Nine of these properties are available to developers using the *OLE DB Provider for ODBC* with D^3 ODBC.

Field Properties	Available
ActualSize	Yes
Attributes	Yes*
DefinedSize	Yes
Name	Yes
NumericScale	Yes
OriginalValue	Yes
Precision	Yes
Type	Yes**
UnderlyingValue	No
Value	Yes

*Available options – adFldFixed, adFldIsNullable, adFldMayBeNull, adFldUpdateable

**Available options – adInteger, adSmallInt, asDBDate, adVarChar, adNumeric, adTime, adTimestamp

Property Object

Methods

The Property object has no methods.

Properties

The Property object has the three properties shown in the following table. All three of these properties are available to developers using the *OLE DB Provider for ODBC* with D^3 ODBC.

Property Properties	Available
Attributes	Yes
Name	Yes
Type	Yes

Error Object

Methods

The Error object has no methods.

Properties

The Error object has the seven properties shown in the following table. Four of these properties are available to developers using the *OLE DB Provider for ODBC* with D^3 ODBC.

Error Properties	Available
Description	Yes
HelpContext	No
HelpFile	No
NativeError	No
Number	Yes
Source	Yes
SQLState	Yes

Considerations for the D^3 Developer Using ADO

Using the Current and Future Providers

Currently, the *OLE DB Provider for ODBC* is the only means by which an ADO-based application can connect to a D^3 database. In the future, a native *OLE DB Provider for D^3* will be available.

Applications developed using the *OLE DB Provider for ODBC* will also run with the native provider with only minor modification to the application.

Connecting to a D^3 Database

There are two types of connection strings available for making a connection to a D^3 database:

- a string that specifies an existing ODBC DSN
- a string that specifies an ODBC driver

Both of these options are described below.

NOTE—The second option, using a string that specifies an ODBC driver, is more efficient, since the use of a DSN requires a search of the registry.

Existing DSN

To use an existing ODBC DSN, include the string “DSN=” and the DSN name in the ConnectionString property of the ADO Connection object. For example, a line of code in Visual Basic might look like this:

```
oConn.ConnectionString =
        “PROVIDER=MSDASQL;DSN=sqldemo”
```

The DSN, sqldemo, contains all the necessary information to connect to the D^3 database.

To specify an ODBC file DSN, use the keyword, FILEDSN, such as,

```
oConn.ConnectionString =
        “PROVIDER=MSDASQL;FILEDSN=fsqldemo”
```

ODBC Driver

Instead of using an ODBC DSN, all the connection information can be included in the connection string itself. To connect to a D^3 database, the connection string must include the keywords below:

Required Keywords	Description
DRIVER	Use “{D3 ODBC Driver}”.
SERVER	Host name of the machine where the D^3 ODBC server is running.
VIRTUALMACHINE	Name of the D^3 virtual machine or Unix servers, this is typically `pick0` or Windows servers, this is the name of the machine where the primary VME resides.

Required Keywords	Description
PORTNUMBER	TCP port number where the D3 ODBC server is “listening”; by default, `1603.`
UID	D3 user
ACCOUNT	D3 account
D3VERSION	Use `710`

For example, a line of code in C++ might look like this:

```
pConn->Open("PROVIDER=MSDASQL;DRIVER={D3 ODBC
        Driver};SERVER=prod;
        VIRTUALMACHINE=pick0;PORTNUMBER=1603;U
        ID=dm;ACCOUNT=sqldemo;
        D3VERION=710;");
```

Optional keywords for use in the connection string are described below:

Optional Keywords	Description
PWD	D3 user password
ACPASSWORD	D3 account password
CONNECTDIALOG	“Yes” to force a dialog box at connection time.

For example, to prompt for passwords at connection time, include the CONNECTDIALOG keyword as shown below:

```
pConn->Open(“PROVIDER=MSDASQL;DRIVER={D3 ODBC
        Driver};SERVER=prod;
        VIRTUALMACHINE=pick0;PORTNUMBER=1603;U
        ID=dm;ACCOUNT=sqldemo;
        D3VERSION=710;CONNECTDIALOG=YES“);
```

Cursors

As with D^3 ODBC, only forward-only and static cursors are available in ADO-based applications. If a dynamic or keyset cursor is requested, a static cursor is returned. The location of the cursor should typically be set to adUseServer.

Locking

Pessimistic locking is not available. If the LockType is set to adLockPessimistic, the option is set but ignored. Query-based updates use an optimistic, by-value scheme.

Synchronous Execution

Since asynchronous execution is not supported in D^3 ODBC, applications developed with ADO run with synchronous query execution.

Known Limitations

- If a recordset is opened with a static cursor and the SQL statement is not successfully executed, the error returned is –2147217887. "The request properties can not be supported by this ODBC Driver," rather than the expected error –21472179000 "The command contains one or more errors."
- The REQUERY method should not be used in a program that involves transactions. To update the data in a recordset, the recordset should be closed (using CLOSE) and re-opened (using OPEN).