1

DBD::ADO

Version

Version 0.03.

At the time of this writing, the `DBD::ADO` driver even ADO itself are relatively new. Things are bound to change, so be sure to read the latest documentation.

Author and Contact Details

The driver maintainer is Phlip Plumlee. He can be contacted via the *dbi-users* mailing list.

Supported Database Versions and Options

The `DBD::ADO` module requires Microsoft ADO version 2.1 or later to work reliably. Using NT with Service Pack 4 is recommended. The module is pure Perl, making use of the `Win32::OLE` module to handle the ADO requests.

The `DBD::ADO` module supports SQL statements into anything your raw ADO supports. This can include the Jet data drivers for the various Microsoft Office file formats, any number of ODBC data drivers, or experimental data providers that expose file system folder heirarchies or Internet directory services as data sources.

Each provider system variously supports SQL, either in a native format like MS-SQL Server's Transact SQL or as an emulation layer in the data provider, such as a Jet data driver reading an Excel spreadsheet.

Connect Syntax

The `DBI->connect()` Data Source Name, or *DSN* has the following format:

```
dbi:ADO:DSN
```

Where DSN must be an ODBC Data Source Name registered with the Control Panel ODBC Data Sources applet. If your DBI application runs as a service, or daemon, the DSN must appear on the ‘‘System DSN’’ tab. A CGI application is the obvious example.

There are no driver specific attributes for the `DBI->connect()` method.

Numeric Data Handling

The numeric, string, and date data types depend on the interaction of four forces: What a Perl ‘‘scalar’’ supports, how the `Win32::OLE` layer translates VARIANTs into scalars, the types that VARIANT itself permits, and the types your target provider emits.

A user-programmer must research those forces in their relevant documentation. Rest assured that `DBD::ADO` will then transmit the type correctly.

String Data Handling

See above.

Date Data Handling

See above.

LONG/BLOB Data Handling

See above.

Other Data Handling issues

The `DBD::ADO` driver does not yet support the `type_info()` method.

Transactions, Isolation and Locking

`DBD::ADO` reflects the capabilities of the native *Connection* to the user. Transactions, if a provider supports them, are per connection - all statements derived from one connection will ‘‘see’’ updates to that data that await a `COMMIT` statement. Other connections, if a provider supports them, will not see these updates.

ADO supports 4 locking levels. Databases can be very sensitive to the lock typed picked. If you pick the wrong type and then BEGIN a large TRANSACTION, the locking type can escalate up to lock nearly everything. `DBD::ADO` should expose these levels to the user for tuning, but currently doesn’t.

No-Table Expression Select Syntax

Dependent on the data provider associated with the database handle.

Table Join Syntax

Dependent on the data provider associated with the database handle.

Table and Column Names

Dependent on the data provider associated with the database handle.

Case Sensitivity of LIKE Operator

Dependent on the data provider associated with the database handle.

Row ID

Dependent on the data provider associated with the database handle.

Automatic Key or Sequence Generation

Dependent on the data provider associated with the database handle.

Automatic Row Numbering and Row Count Limiting

Dependent on the data provider associated with the database handle.

Parameter Binding

Parameter binding is not yet supported by `DBD::ADO`.

Unsupported values of the TYPE attribute do not currently generate a warning.

Stored Procedures

Calling stored procedures is not yet supported by `DBD::ADO`.

Table Metadata

`DBD::ADO` does not currently support the `table_info()` method. It awaits the needed slack time and/or other volunteers.

Driver-specific Attributes and Methods

DBD::ADO has no significant driver-specific handle attributes or private methods.

Positioned updates and deletes

DBD::ADO does not support positioned updates or deletes.

Differences from the DBI Specification

DBD::ADO is a very new and currently incomplete driver. It is evolving rapidly though, and since it's written in pure Perl using Win32::OLE, it's easy for people to enhance.

URLs to More Database/Driver Specific Information

Information about ADO can be found at *http://www.microsoft.com/data/ado/*.

Concurrent use of Multiple Handles

DBD::ADO supports an unlimited number of concurrent data source connections to one or more data sources subject to the limitations of those data sources.

It also supports the preparation and execution of a new statement handle while still fetching data from another statement handle associated with the same database handle. This is also subject to the limitations of the data source.