

Advantage™ Database Server .NET Data Provider

Key Features

- Provides native client/server access to the high performance Advantage Database Server RDBMS
- Supports data access via SQL as well as direct table access and server-side control unique to Advantage
- Provides easier conversion to the .NET Framework than other database engines, because Advantage supports both the expected disconnected recordset functionality in ADO.NET as well as direct navigational access to the database
- Includes full server-based transaction processing to eliminate database corruption, drastically minimizing support costs
- Provides complete referential integrity support including primary/foreign key definition and cascaded updates and deletes

The Advantage .NET Data Provider is a managed provider that offers native data access to the Advantage Database Server and Advantage Local Server. The Advantage .NET Data Provider conforms to the ADO.NET data access model and creates a seamless interface between the Advantage Database Server or Advantage Local Server and any .NET development language including Microsoft Visual Studio .NET and Delphi for .NET.

SERVERS

The Advantage .NET Data Provider ties in seamlessly with the Advantage Database Server, a true client/server solution that adds performance and stability to multi-user applications. Advantage Database Server supports Windows, Linux, and NetWare networks for cross-platform distribution and requires no database administrator. The Advantage .NET Data Provider also provides access to the Advantage Local Server. The Advantage Local Server provides non-client/server access to your data, which is perfect for single and small multi-user networked environments, and ideal for customers who may later want the enhanced performance and security capability of a client/server RDBMS with the Advantage Database Server. Distribution of the Advantage .NET Data Provider and the Advantage Local Server is royalty-free for applications deployed in both stand-alone and peer-to-peer environments.

COMPLETE DATA CONTROL UNIQUE TO ADVANTAGE

Whether you're a seasoned .NET developer or are just converting to the framework, Advantage provides robust and powerful functionality previously not available to .NET developers. Advantage Database Server allows developers the flexibility to combine powerful SQL statements and relational data access methods with the performance and control of navigational operations, such as direct table and index access and direct table movement (seeks, skips, etc.). Combining these two powerful data access methods allows the result set returned from an SQL query or from a table opened directly to be navigated via the highly optimized Advantage ISAM database engine. The result is an easy-to-use interface that supports SQL commands and direct-result-set navigation, all in one integrated solution.

There are many additional benefits provided by the ability to perform direct data access and navigational operations with the Advantage .NET Data Provider. Traditional data access through ADO.NET relies heavily on client-side DataSets, and does not allow server-side cursors or pessimistic locking. For developers converting to ADO.NET from other data access models, this can be a cumbersome paradigm shift when designing new applications or converting existing applications. By providing navigational data access operations, Advantage provides all the direct table access methods many developers have learned to rely upon. Developers can use server-side cursors and perform powerful data manipulation tasks directly on the data that resides on the server. No longer will developers with existing applications that rely on direct data access and navigation have to spend valuable time re-developing their applications to use non-navigational data access operations with ADO.NET. ADO.NET developers previously unfamiliar with the power inherent with direct table access and server-side control will experience a whole new realm of functionality. In addition, directly accessing table data through navigational operations can provide performance improvements in certain operations such as reading through a large recordset, or positioning directly to a specific record. Only the Advantage .NET Data Provider provides this type of migration ease and flexibility for developers of all backgrounds.

KEY FEATURES — CONTINUED

- Includes database security functionality and encryption support
- Includes triggers that provide a powerful means to maintain business rules at the database level - independent of the client application
- Designed for use with Microsoft Visual Studio .NET and Delphi for .NET, including support for the Visual Studio .NET component designer

EASE YOUR DEVELOPMENT

There are many situations where data access through direct navigational methods provides exceptional performance benefits or eases application development. For example, consider the scenario of a stadium seating reservation system where it is critical that only one person is allowed to purchase a given seat. To ensure that only one user is reserving a specific seat, a developer can simply rely upon the server-side control provided by the Advantage .NET Data Provider and directly lock the record for that seat until the transaction is completed. Thus, avoiding the possibility of having two users trying to reserve the same seat at the same time. Without server-side control, a non-Advantage .NET developer would only have a disconnected client-side view of the data and, therefore, would not be able to easily or effectively keep others from reserving a desired seat while going through the process to reserve that seat.

FULL SCALABILITY—WRITE ONCE, DEPLOY ANYWHERE

Advantage applications can be deployed in stand-alone, peer-to-peer, client/server, and Internet environments with one set of source code. Advantage does not require a different .NET Data Provider or different set of source code for different network environments. The Advantage .NET Data Provider will automatically determine if the Advantage Database Server is available directly or via the Advantage Internet Server, or whether the Advantage Local Server should be used. Therefore, you only need to develop one application with one version of code for local, peer-to-peer, client/server, or Internet file access.

CLASS SUMMARY

The following classes are available in the Advantage .NET Data Provider.

AdsCommand: Represents an Advantage SQL statement or stored procedure to execute or a table to directly open on the Advantage server.

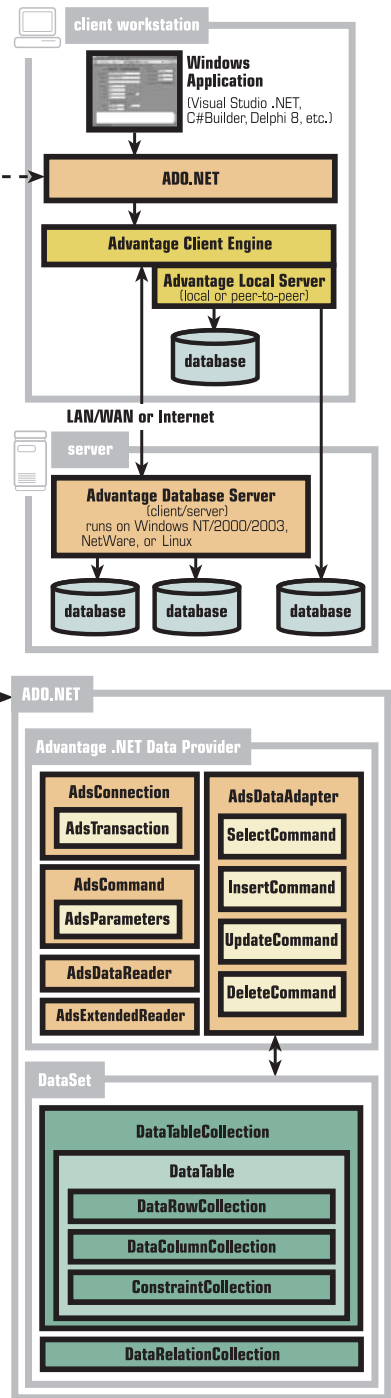
AdsCommandBuilder: Provides the capability to automatically generate the necessary UPDATE, DELETE, and INSERT statements on an AdsDataAdapter object to reconcile differences between a DataSet and the underlying Advantage database table.

AdsConnection: Represents a connection to Advantage Database Server or Advantage Local Server.

AdsConnectionStringBuilder: Provides a simple way to create and manage the contents of connection strings used by the AdsConnection class.

AdsDataAdapter: Represents a set of data commands and a database connection that are used to fill a DataSet and update an Advantage database.

AdsDataReader: Represents an Advantage cursor (result set) or table. Provides the functionality to read through the data rows and columns.



AdsExtendedReader: This class is derived from AdsDataReader. It represents an Advantage server-side cursor (result set) and gives powerful navigational access to the developer. Among other things, it provides the capability to directly access table data, filter result sets through Advantage Optimized Filters, apply ranges (scopes), traverse result sets in natural or index order, seek for records, lock and unlock tables, lock and unlock individual records, and to update records directly on the server.

AdsException: This is the exception that is thrown when the Advantage .NET Data Provider encounters an error condition.

AdsFactory: Provides a set of methods for creating specific instances of Advantage .Net Data Provider objects.

AdsHelper: This class is a .NET component that developers can include in their application to simplify tasks such as creating DataSet and AdsDataReader objects, executing stored procedures, and executing SQL statements. It is an Advantage-specific implementation of the Microsoft Data Access Application Block for .NET version 2.

AdsInfoMessageEventArgs: Provides data for the InfoMessage event in the AdsConnection class. This class provides information for warning conditions.

AdsParameter: Represents a parameter to an AdsCommand, and optionally, its mapping to DataSet columns.

AdsParameterCollection: Collects all parameters relevant to an AdsCommand as well as their respective mappings to DataSet columns.

AdsTransaction: Represents an Advantage transaction.

NAME SPACE

The Advantage .NET Data Provider exists in the name space Advantage.Data.Provider.

SPECIFICATIONS

Server Operating Systems (via the Advantage Database Server)

Novell NetWare 5.x or greater (IP, IPX)
Microsoft Windows x86 (IP, IPX)
Microsoft Windows x86_64 (IP)
Linux x86, x86_64 (IP)

Client Operating Systems

Microsoft Windows x86 and x86_64

SUPPORTED DATA TYPES

Advantage ADT table data types

autoincrement	4-byte read-only integer value from 1 to 4,294,967,296 unique for each record in the table.
binary	Variable length BLOB containing binary data. The size of each field is limited to 4 GB. The binary image data is actually stored in a separate file, called a memo file, to reduce table bloat.
character	Fixed length character field from 1 to 65,530 bytes that is stored entirely in the table.
cicharacter	Case insensitive fixed length character field from 1 to 65,530 bytes that is stored entirely in the table.
curdouble	Currency data stored internally as an IEEE double.
date	4-byte date field.
double	8-byte IEEE floating point value in the range 1.7E +/-308 (15 digits of precision).
image	Variable length BLOB containing image data. The size of each field is limited to 4 GB. The binary image data is actually stored in a separate file, called a memo file, to reduce table bloat.
integer	4-byte long integer values from -2,147,483,647 to 2,147,483,647
logical	1-byte logical (boolean) field.
memo	Variable length memo field containing character data. The size of each field is limited to 4 GB. The memo data is actually stored in a separate file, called a memo file, to reduce table bloat.

mod time	Timestamp field automatically updated when a record is updated
money	Currency data stored internally as a 64-bit integer, with 4 implied decimal digits from - 922,337,203,685,477,5807 to +922,337,203,685,477,5807. The Money data type will not lose precision.
numeric	Fixed length (exact representation) numeric up to 32 bytes.
raw	Fixed length data-typeless raw data field from 1 to 65,530 bytes.
row version	64-bit auto-incrementing integer value
shortinteger	2-byte short integer value from -32,767 to 32,767.
time	4-byte value representing time of day.
timestamp	8-byte value representing date and time of day.
varbinary	Variable length binary data stored entirely in the table.
varchar	Variable length character data stored entirely in the table.

DBF table data types

binary	Variable length BLOB containing binary data. The size of each field is limited to 4 GB. The binary image data is actually stored in a separate file, called a memo file, to reduce table bloat.
character	Fixed length character field from 1 to 65,530 bytes that is stored entirely in the table.
date	8-byte date field.
double	8-byte IEEE floating point value in the range 1.7E+/-308 (15 digits of precision).
image	Variable length BLOB containing image data. The size of each field is limited to 4 GB. The binary image data is actually stored in a separate file, called a memo file, to reduce table bloat.
integer	4 byte-Integer values from -2,147,483,648 to 2,147,483,647.
logical	1-byte logical (boolean) field.
memo	Variable length memo field of up to 65,530 bytes. The size of each field is limited to 4 GB. The memo data is actually stored in a separate file, called a memo file, to reduce table bloat.
money	Currency data stored internally as a 64-bit integer, with 4 implied decimal digits from - 922,337,203,685,477,5807 to +922,337,203,685,477,5807. The Money data type will not lose precision.
numeric	Fixed length (exact representation) numeric up to 32 bytes.
timestamp	8-byte value representing date and time of day.
varbinary	Variable length binary data stored entirely in the table.
varchar	Variable length character data stored entirely in the table.

Database Maximums

Maximum ADT table size

- Windows with NTFS — 16 exabytes (18,446,744,073,709,551,616 bytes)
- Windows with FAT32 — 4 gigabytes (4,294,967,296 bytes)
- NetWare 5 or greater with NSS file systems — 16 exabytes (18,446,744,073,709,551,616 bytes)
- NetWare 5 or greater with traditional file systems — 4 gigabytes (4,294,967,296 bytes)
- Linux pre-2.1.2 — 11 glibc and pre-2.4 kernel — 2 gigabytes (2,147,483,648 bytes)
- Linux glibc 2.1.2 — 11+ with kernel 2.4+ — 8 exabytes (9,223,372,036,854,775,807 bytes)

Maximum DBF table size — Maximum Record Count (2,147,483,648) multiplied by Record Length (depending upon operating system and file system)

Maximum number of records — 2.2 billion

Maximum record length — 65,530 bytes

Maximum field name length — 128 characters for ADT tables, 10 characters for DBF tables

Maximum number of columns per table — 3,500 for ADT tables, 2,035 for DBF tables

iANYWHERE SOLUTIONS, INC.
WORLDWIDE HEADQUARTERS
ONE SYBASE DRIVE
DUBLIN, CA 94568-7902
U.S.A.

North America
Advantageinfo@iAnywhere.com
1 800 801 2069

Germany
AD5-team@iAnywhere.com
+49 (0) 7032 / 798 - 200
United Kingdom
AdvantageUK@iAnywhere.com
+44 (0)117 333 9000