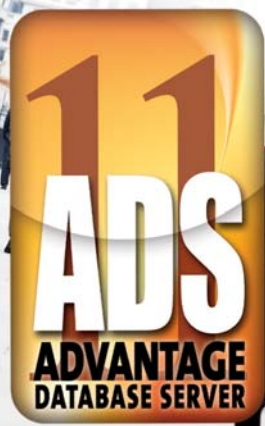




What's New in Advantage Database Server 11

WHITE PAPER



Contents:

Introduction	1
Advantage 11	1
Server Enhancements	1
Online Table Maintenance	1
Advantage Web Platform	2
Stored Procedures with Varying Output	2
Improved Stored Procedure & Trigger Input Parameters	3
Root Dictionary	3
Transaction Logs > 4GB	3
Support for SQL Intermediate Files > 4GB	3
Improved Checks for Corrupt Memo Files	3
Reserve Connections for LAN or Web Platform Usage	3
Connection Pooling	3
Hibernate Support	4
Restore Database NoWarnings Option	4
Security	4
FIPS-Enabled Server AES encryption Enhancements	4
SSL/TLS Enhancements	4
Delphi Enhancements	4
64-bit Delphi Support	4
Delphi XE2 Support	4
Locate Enhancements	4
Delay Load ACE in Delphi	5
ExtraConnectString Support in TAdsConnect for Connection Options	5
SQL Enhancements	5
VO SQL Parameters	5
Explicit Date, Time, and Timestamp Literals	5
Binary Literals	5
New System Variables	5
New System Procedures	6
New Clients Supported	6
Python/Django Support	6
Ruby Driver	6
PHP PDO Driver	6
64-bit Lazarus Support	6
New Utilities	7
Advantage Web Administrator Utility	7
SQL Command Line Utility	7
Advantage Data Architect Enhancements	7
SQL Scripts over 32K (Triggers and Views)	7
Replication Queue Record Data Display	8
Performance Improvements	8

Client Enhancements.....	8
Improved .NET template projects for AEPs and Triggers.....	8
Improved Error Messages.....	8
Replication Enhancements.....	8
Queue Display Order	8
New System Procedures.....	8
Ability to Replicate to Older Servers.....	9
Advantage 10.....	9
Data Handling.....	9
Unicode Support.....	9
Nested Transactions	9
Transaction-Free Tables.....	9
Table Data Caching	9
Temporary Table Caching.....	9
Event IDs.....	10
SQL Enhancements.....	10
Stored Procedures in the FROM Clause.....	10
Boolean SQL Expressions	10
Limiting Query Results.....	10
ROWNUM Support	10
SQL Bitwise Operators	10
SQL Timeout Property.....	10
Query Execution Plan Improvements	11
New Expression Engine Functions	11
New ISOWeek Scalar and Expression Engine Function.....	11
Hex Scalar and Expression Engine Functions	11
Delphi Enhancements	11
Advanced Delphi Property Editors.....	11
New Delphi Methods.....	12
New Delphi Component for Notifications	12
Indexing.....	12
Binary Indexes.....	12
sp_Reindex Procedure.....	12
Performance Improvements	12
Express Queue Support	12
Transaction Processing System.....	13
Advantage Optimized Filters (AOFs).....	13
Record Count Operations	13
Efficient Table Creation	13
Indexing	13
Referential Integrity Cascade Operations.....	13
Appending and Deleting Records	13
Temporary File Handling.....	14
Memo Header Caching	14
Table Open Performance.....	14
Cached File Pool.....	14

Record Locks	14
Advantage Extended Procedures (AEPs).....	14
Worker Threads	14
Rights Checking Behavior.....	15
Memory Management	15
Advantage Data Architect Enhancements	15
Reporting	15
Crystal Reports Settings	15
R&R ReportWorks Support	15
New Clients and Platform Support	16
More 64-bit Clients	16
64-Bit Advantage ADO.NET Improvements	16
Support for the Latest Development Environments	16
Updated Platform Support.....	16
Support for Vulcan.NET.....	16
No Delphi 3, Delphi 4, C++Builder 3, or C++Builder 4 Components.....	16
Installation and Configuration	17
Automatically Configured Worker Thread Count	17
Side-By-Side Server Installations.....	17
New Help File Format.....	17
Server Discovery	17
Miscellaneous	17
New System Variables.....	17
Support Capture Utility	17
Advantage 10.1	18
Support for Latest Development Environments	18
Unicode Full Text Search Support	18
Advantage Data Architect Copy/Paste Support for Rows	18
Strong Encryption and FIPS Compliance	18

Introduction

Advantage Database Server 11 is another leap forward in functionality, usability, and performance. Advantage 11 offers many attractive improvements and enhancements including:

- Client-less access to Advantage data through the Advantage Web Platform OData web service. Now you can easily access data from mobile devices, tablets, Macintosh, or any device, on any platform, with any modern development environment
- Online table maintenance capabilities that allow you to maintain Advantage data like never before; all while users are actively accessing the tables.
- Updated and new clients
- New utilities
- SQL enhancements
- Performance enhancements
- and much more

The following is a list of new features contained in Advantage Database Server 11, in no particular order. How important each item is will depend on the relevance to your current project – and could quite easily change with the next.

Advantage 11

Server Enhancements

Online Table Maintenance

Added the capability to perform the following maintenance operations while a table is in use by one or more clients. Prior to ADS 11, it was necessary to open a table exclusively (thus disallowing any other use of the table) during the operation.

- Online Pack - remove deleted records from a table while it is in use
- Online Reindex - rebuild a table's indexes while it is in use
- Online Alter - change a table's structure while it is in use (Add, delete, modify fields)

See Online Table Maintenance in the Help File for details.

Advantage Web Platform

The Advantage Web Platform is a web service that allows “client-less” access to Advantage data from any device, platform, or development environment that can make Internet (HTTP) calls. This includes but is not limited to devices such as desktop/laptops, mobile phones, PDAs, and tablets. It allows access from any current operating system including previously unsupported systems such as Mac OS and Android because no Advantage client is required. The Advantage Web Platform extends your application development opportunities and opens up endless possibilities to access Advantage from many different types of architectures, hardware, and business scenarios.

The Advantage Web Platform is a web service built on the standard OData web protocol to access data (www.odata.org). The OData standard, follows RESTful principles, and allows you to build web applications that can run in limitless scenarios. You can write one application that can run equally well on an iPhone, Android, or a Windows or Mac desktop machine.

Some web platform features include:

- OData Protocol Support
- Supports OData Clients including .NET, Objective-C (iOS), Android, Silverlight, PHP, Java, and more
- XML (AtomPub), JSON, and JSONP Support
- CRUD Operations
- Connection and Statement Caching
- Batch support
- Filter support
- OrderBy query support
- Metadata support
- Unicode support
- Referential Integrity support
- Off-line data storage synchronization through OData reference caching (delta support)
- Stored Procedure support via service operations
- Chunked HTTP requests
- Tunneled HTTP requests
- MIME encoding of binary updates
- Full SQL query support (optional)
- Pass-through query support

See Advantage Web Platform in the Help File for details.

Stored Procedures with Varying Output

SQL script stored procedures were enhanced with the capability to return the last SELECT or EXECUTE PROCEDURE statement in the script as the output of the stored procedure. This allows a stored procedure to return cursors (including live cursors) having a varying structure from call to call.

See Create Procedure in the Help File for details.

Improved Stored Procedure & Trigger Input Parameters

- Stored procedure input parameters can be referenced directly through `_param` notation.
- Trigger `__new` cursors are automatically declared and can be accessed directly without a `FETCH`.

See `Create Procedure` and `Create Trigger` in the Help File for details.

Root Dictionary

Advantage now supports a “root” dictionary. The root dictionary allows better control over security when using the Advantage Web Administrator Utility. System procedures through the Advantage Web Platform can only be run when connected to the root dictionary. Some of the additions include:

- `sp_mgKillUser` now requires `DB:Admin` or `SERVER:Admin` membership. Also, you must be connected to a dictionary to kill users on that dictionary.
- `sp_mgKillUser` allows a wild card (using an asterisk) to be given as the user name when run by a `SERVER:Admin` member on the root dictionary.
- `sp_GetSQLStatements` now requires `DB:Admin` membership and with `remote server only` retrieves statements for the current dictionary.

See `Root Dictionary` in the Help File for details.

Transaction Logs > 4GB

The transaction processing system has been updated to support transactions that result in writing more than 4GB of data to the transaction log file.

Support for SQL Intermediate Files > 4GB

SQL statements that produce intermediate files (for sorting, grouping, etc.) exceeding 4GB are now supported.

Improved Checks for Corrupt Memo Files

Improved the Memo and BLOB handling logic for the Advantage proprietary memo format (`.adm`) to more reliably detect memo corruption.

Reserve Connections for LAN or Web Platform Usage

Users connecting to Advantage through the Advantage Web Platform are counted as separate users. To prevent the possibility of all Advantage user connections being taken up by traditional fully connected applications and leaving none available for web applications, Advantage now allows the total user count to be partitioned into two categories. The user count can be divided between users for traditional connected applications and Advantage Web Platform applications.

See `Web Platform Users` in the Help File for details.

Connection Pooling

Connection pooling has been added to the Advantage Client Engine. The Advantage Web Platform uses pooling to provide high performance connection capability. This functionality is also available to your own applications through the `AdsConnect101` API. See `AdsConnect101` in the Help File for details.

Hibernate Support

Added dialect support to the JDBC driver to support Hibernate. See *Hibernate* in the Help File for details.

Restore Database NoWarnings Option

The system procedure `sp_RestoreDatabase` now recognizes the option "NoWarnings". This option turns off the logging of messages to the error log when tables are automatically created during an online restore operation. Without this option, informational entries (5168, 7041, and system code 2) are logged when a restore operation creates a table.

See *Backup and Restore Options* in the Help File for details.

Security

FIPS-Enabled Server AES encryption Enhancements

Improved error codes when old clients connect to FIPS-enabled servers.

Adsbackup utility updated to support FIPS and SSL settings and be able to use connection string options. See *The adsbackup Utility* in the Help File for details.

Client updates to support FIPS and SSL settings. See the documentation for specific clients in the Help File for details.

SSL/TLS Enhancements

Simplified specification of SSL certificate where no path is necessary if certificate is in same folder as ADS.

See *TLS Key File* in the Help File for details.

Delphi Enhancements

64-bit Delphi Support

Advantage now supports Delphi's 64-bit cross-compiler. This means that Delphi can now be used to build and debug 64-bit binaries. Moreover, this also allows AEPs and Triggers written in Delphi to be used with 64-bit versions of Advantage Database Server.

Delphi XE2 Support

Added support for Delphi XE2, including support for the Windows 64-bit cross-compiler.

Locate Enhancements

Updated the `TAdsQuery` `Locate` method to give precedence to the index implicitly set by an `ORDER BY` clause if one exists. See `AdsTableOptions.AdsLocateBehavior` in the Help File for details.

Delay Load ACE in Delphi

Updated the Advantage Delphi Components to use delayed loading of the Advantage Client Engine. This means that an Advantage-enabled application can load and run without loading any Advantage DLLs until it first uses them. This provides the opportunity to run initialization code before loading the Advantage DLLs or to run in certain scenarios (such as non-Advantage functionality) without ever loading or needing the DLLs. See Delay-Loading the Advantage Client Engine in the Help File for details.

ExtraConnectionString Support in TAdsConnection for Connection Options

The TAdsConnection component now supports an ExtraConnectionString property which, when set, will be appended to the end of the automatically-generated Connection String and passed to AdsConnect101. This allows the user to specify settings like the DateFormat, Decimals, and Epoch on a per-connection basis. This can reduce the need to rely on global options set via the AdsSettings component. See TAdsConnection.ExtraConnectionString in the Help File for details.

SQL Enhancements

VO SQL Parameters

Added support to use parameters with AdsSQLServer objects in VO. Allows specifying parameters in initial query execution as well as changing parameters and refreshing an open AdsSQLServer object. See AXSQL Parameter Support in the Help File for details.

Explicit Date, Time, and Timestamp Literals

Added support for explicit date, time and timestamp literals.

Binary Literals

Support for binary literals. See SQL Literals in the Help File for details.

New System Variables

- **::conn.OSUserLoginName** – Retrieve the OS user login name
- **::conn.NetworkAddress** – Retrieve the IP address of the connected user for networked connections
- **::conn.TerminalClientAddress** – Retrieve the IP address of the Terminal Services client if the connection is made from a Terminal Server
- **::conn.ClientHostName** – Retrieve the computer name of the user
- **::conn.IsRoot** – Determine if the current connection is to the root dictionary

See System Variables in the Help File for details.

New System Procedures

- **sp_PackTableOnline** – Pack a table while it is online
- **sp_PackAllTablesOnline** – Pack all tables while they are online
- **sp_ReindexOnline** – Re-index table while it is online
- **sp_GetQueryLoggingResults** – Retrieves the contents of the active query logging table
- **sp_mgGetCrashDumpInfo** – Retrieve a listing of ADSDump files located on the server
- **sp_GetLinks** – Retrieve the links in the current dictionary
- **sp_mgSetConfigValue** – Update a server configuration value
- **sp_ChangeCurrentUserPassword** – Change the current connected user's data dictionary password
- **sp_mgGetErrorLog** – Retrieve error log entries from the ads_err.adt and/or ads_err.dbf error log files

See System Procedures in the Help File for details.

New Clients Supported

Python/Django Support

Added the Advantage Python Driver to provide database access from the Python environment to Advantage Database Server. In addition, added the Advantage Django Backend, which is a Python module designed to provide access from the Django environment to Advantage Database Server. See Advantage Python Driver in the Help File for details.

Ruby Driver

Added Support for Ruby. There are two different Ruby APIs supported by Advantage.

First is the Advantage Ruby API. This API provides a Ruby wrapping over the interface exposed by the ACE API. This package allows Ruby code to interface with Advantage.

Second, there is support for ActiveRecord, an object-relational mapper popularized by being part of the Ruby on Rails web development framework. This package uses (and has a dependency) on the Advantage Ruby API. See Ruby in the Help File for details.

PHP PDO Driver

Added the Advantage PDO (PHP Data Objects) Driver to provide another option for database access from PHP to Advantage Database Server. See PHP in the Help File for details

64-bit Lazarus Support

Improved TDataSet descendant support for Lazarus to be able to build 64-bit binaries for Linux and Windows. Added support for Lazarus 0.9.30. See Lazarus in the Help File for details.

New Utilities

Advantage Web Administrator Utility

The Advantage Web Administrator is an exciting new utility that provides the ability to manage one or more databases from anywhere, including mobile devices and tablets. It connects to Advantage through the Advantage Web Platform. The Web Administrator is built using JavaScript and jQuery and can run on nearly any browser on platforms including Windows, Linux, iOS, and Android allowing you to manage ADS wherever you go.

Some available functionality includes:

- View basic database information (number of users, connections, etc.)
- View installation information (serial number, version, etc.)
- View users connected to a database.
- Disconnect (“kill”) users and disable logins for a database
- View open tables and indexes
- Enable and view query logging information for a database
- Execute ad hoc queries against a database
- View crash dump information that is available on the server
- View error log entries
- Modify some Advantage Database Server configuration values

See Web Administrator in the Help File for details.

SQL Command Line Utility

The new standalone SQL command line utility supports running SQL statements and scripts. This provides the capability to run SQL statements in an ad hoc fashion from a command prompt or from a script (e.g., batch file). Capabilities include:

- Execute an SQL statement from the command line (DOS prompt)
- Execute an SQL statement or script (or multiple scripts) contained in a file from the command line
- Run SQL statements in an interactive fashion (in a session) at the command line

See SQL Command Line Utility in the Help File for details.

Advantage Data Architect Enhancements

SQL Scripts over 32K (Triggers and Views)

Advantage Data Architect now handles SQL scripts over 32K in length for views, triggers, stored procedures, and user defined functions.

Replication Queue Record Data Display

Added the ability to display the record data for a replication queue entry. See `sp_GetReplicationEntryDetails` in the Help File for details.

Performance Improvements

Improved performance of expression parsing, which will help with table and index opens, SQL parsing, filtering, etc. This can have a significant effect on some platforms such as Windows 2008 R2 or when the server is under a large concurrent load.

Improved performance of rollbacks to savepoints.

Optimized the function used to sort keys when building indexes.

Improved pack table performance.

Improved the efficiency of server-side initialization code required for many types of client requests.

Client Enhancements

Improved .NET template projects for AEPs and Triggers

Updated the Visual Studio Advantage Extended Procedure (AEP) and Trigger template projects for .NET to include code to dispose of `IDbCommand` objects prior to exiting the external function.

Improved Error Messages

Updated many error messages to include the column name that is involved in the particular error.

Replication Enhancements

Queue Display Order

The Replication queue is now displayed in logical order in Advantage Data Architect.

New System Procedures

A system procedure has been added so replication queue entries involved in a transaction can be safely deleted. See `sp_DeleteReplicationEntry` in the Help File for details.

A system procedure has been added to generate INSERT, MERGE, UPDATE, and DELETE entries based off a queue entry to help fix replication problems. The procedure will also generate a SELECT statement to display records involved in a replication queue entry. See `sp_GetReplicationEntryDetails` in the Help File for details.

A system procedure has been added to make a test connection to the target of the replication server. See `sp_TestReplicationConnection` in the Help File for details.

Ability to Replicate to Older Servers

Added the ability for a newer Advantage server to replicate to an older server version. For example, it is possible to replicate from a v11.x server to v9.x or v10.x. See *Replicating to Older Servers* in the Help File for details.

Advantage 10

Data Handling

Unicode Support

Processing of Unicode character text is supported by the Advantage Windows and Linux servers and all Advantage clients. Unicode character data can be stored in three new field types, NCHAR, NVARCHAR and NMEMO. These new field types are available in all table types supported by Advantage. Advantage 10 includes new APIs within the Advantage Client Engine. With UTF16 encoding, these APIs read and write Unicode text directly. Unicode characters can also be supplied directly in SQL statements and filter expressions. Unicode columns may be sorted or indexed using various collation locales. See *Unicode Support* in the Help File for details.

Nested Transactions

Transactions can now be nested within other transactions. Normally, this transaction nesting occurs as stored procedures or triggers that contain begin/commit pairs call each other. See *Nesting Transactions* in the Help File for details.

Transaction-Free Tables

There are some cases where it may be desirable to update a table within a transaction, yet have those updates remain outside of the transaction (audit tables, debug log tables, key-generation tables, etc). While this is possible using a secondary connection for such updates, this is not always feasible (for example, when the table is modified in a stored procedure or trigger). Advantage now provides a mechanism to specify a table as a transaction-free table. See *Transaction-Free Tables* in the Help File for details.

Table Data Caching

Table Data Caching is a feature that enables the caching of table data in the Advantage caching system. This feature is intended for use with tables that contain static data that is used often and shared among multiple users. This feature can be used with tables that are backed up on a regular basis, or tables that contain static or read-only data such as zip code lookup tables, insurance code lookup tables, etc. See *Table Data Caching* in the Help File for details.

Temporary Table Caching

Advantage now fully caches temporary tables in memory when possible. Only when Advantage cannot fit temporary table data in its cache (or when it is configured to not cache any data) will it create a physical file or write table data to disk. See *Temporary Tables* in the Help File for details.

Event IDs

Events can now be signaled with a user-defined data string that will be returned when the signal is received. The typical use of this string data is to provide a method of locating the record or table for which a signal is sent, however any string data can be used, providing a flexible mechanism to deliver per-event context to threads receiving the signals. See Events (Notifications) in the Help File for details.

SQL Enhancements

Stored Procedures in the FROM Clause

The Advantage query engine now supports using stored procedure results in place of a table or view reference in the FROM clause. For example:

```
SELECT * FROM (EXECUTE PROCEDURE sp_mgGetConnectedUsers()) connectedUsers
```

Boolean SQL Expressions

The SQL engine now supports Boolean value expressions. For example, the following statement is now valid:

```
"SELECT ( flag = FALSE ), (val = 1) FROM table1 WHERE fld1 OR fld2"
```

Limiting Query Results

The Advantage query engine now supports the START AT clause when using a SELECT TOP statement. START AT can be used to provide paging functionality. The following query will return the first 10 employees:

```
SELECT TOP 10 * FROM employees
```

To return the next 10 employees in the table, the following syntax is now supported:

```
SELECT TOP 10 START AT 11 * FROM employees
```

See Limiting Query Results in the Help File for details.

ROWNUM Support

The ROWNUM scalar function is now supported. ROWNUM can be used to generate integer numbers starting at 1 for each row in the result of a query. The ROWNUM function is primarily intended for use in the select list and can be used to provide a numbering of rows in the result set. The number associated with a row is determined when the row is selected for inclusion in the result set. See ROWNUM in the Help File for details.

SQL Bitwise Operators

The Advantage query engine now supports six bitwise operators: & (AND), | (OR), ^ (XOR), ~ (NOT), >> (SHIFT RIGHT), << (SHIFT LEFT). See Operators in SQL in the Help File for details.

SQL Timeout Property

Added support for an optional SQL timeout value for a given connection or statement handle. The timeout setting will independently apply to the initial query execution, and to any operation that supports Advantage callback functionality. This new functionality is exposed via a new ACE API `AdsSetSQLTimeout` and the new Delphi

properties TAdsConnection.SQLTimeout and TAdsQuery.SQLTimeout. See Callback Functionality in the Help File for details.

Query Execution Plan Improvements

The SQL execution plan has been improved to include more detailed information about the indexes that are used to optimized each specific segment of the WHERE clause. The information includes the order in which the segments are evaluated, the estimated key count that the server uses to select the index for the optimization if applicable, and the actual number of keys that are returned for the specific segment if it is evaluated.

New Expression Engine Functions

The following new expression engine functions are supported by Advantage. These new functions can be used to create indexes that Advantage will use to optimize SQL queries that reference their corresponding scalar functions. See Indexes with Expressions and Indexes and SQL Performance in the Help File for details.

- WEEK
- QUARTER
- DAYOFYEAR
- DAYOFWEEK
- HOUR
- MINUTE
- SECOND
- DAYNAME
- MONTHNAME

New ISOWeek Scalar and Expression Engine Function

A new ISOWEEK expression engine and scalar engine function has been created that returns the ISO 8601 week number of a given date value. See ISOWEEK for details on the expression engine function (which can be used to create an index for filter and query optimization). See supported DATE/TIME scalar functions in the Help File for details.

Hex Scalar and Expression Engine Functions

CHAR2HEX and HEX2CHAR have been added to facilitate hexadecimal conversions. The function CHAR2HEX can be used to convert character data containing hexadecimal characters to a binary value. Two hexadecimal characters will be converted to one byte. The function HEX2CHAR converts a binary value to a character value. Each byte of the binary value is represented as two hexadecimal characters. See Functions to Convert Hexadecimal Values in the Help File for details.

Delphi Enhancements

Advanced Delphi Property Editors

The SQL Utility available in the Advantage Data Architect is now used as the TAdsQuery.SQL property editor in Delphi and C++Builder. This editor provides many additional features including syntax highlighting, code

templates, find/replace functionality, ability to run and preview query results, ability to verify query syntax, ability to debug SQL scripts, etc. In addition, you can now create new tables from inside the Delphi IDE by right clicking on a TAdsTable or TAdsQuery instance and selecting "Create New Table". See Advanced Property Editors in the Help File for details.

New Delphi Methods

The existing `sp_SetApplicationID` and `sp_GetApplicationID` canned procedures have been exposed in the Advantage TDataSet Descendant via the new `TAdsConnection.ApplicationID` property and `TAdsConnection.GetApplicationID` method. See `ApplicationID` in the Help File for details.

The `TAdsConnection` component has a new constructor called `CreateFromHandle` which can be used to clone a connection using an existing Advantage Client Engine (ACE) handle. See `CreateFromHandle` in the Help File for details.

New Delphi Component for Notifications

A new Delphi/C++Builder component called `TAdsEvent` can be used to listen for and handle notifications. This component automatically handles the creation of a background thread and an Advantage connection, allowing the developer to handle asynchronous events with ease.

Indexing

Binary Indexes

Advantage now supports binary indexes for logical expressions. These are especially useful for building indexes of deleted records for faster filtering and traversal of records on tables with large numbers of deleted records. When a binary index with the `DELETED()` expression exists, Advantage can use it for optimizing the filtering of deleted records when traversing record data in natural record order and when creating Advantage Optimized Filters (AOFs). This optimization helps with both DBF tables (when filtering deleted records) and with ADT tables. See `Binary Indexes` in the Help File for details.

sp_Reindex Procedure

Added a new system procedure called `sp_Reindex` to provide reindexing functionality in SQL.

Performance Improvements

Express Queue Support

Advantage Database Server now supports a dynamic queuing of client requests based on the historical cost of a connection's requests. Connections that are predicted to be under a dynamically computed threshold may be given preference in the request queue when the system is busy. This can make interactive applications that are making inexpensive requests more responsive when used in a busy system. The developer can also change a connection's request priority with the new system procedure `sp_SetRequestPriority`. It is not necessary for the developer to make any application changes in order to take advantage of this functionality. See `Express Queue` in the Help File for details.

Transaction Processing System

A number of improvements have been made to the Advantage Transaction Processing System resulting in significant performance improvements. Many internal optimizations to lock lists and visibility lists have increased the performance of transactions with a large number of operations. In addition, the performance of shorter repeated transactions has also been improved via a new temporary file handle pool that is now used instead of dynamically creating and deleting TPS log files with every transaction.

Advantage Optimized Filters (AOFs)

Improved the Advantage Optimized Filter (AOF) multi-segment index algorithm to consider more index tags when optimizing an AOF, not just the first index found. This modification increases the possibility of fully optimizing a multi-segmented AOF. In addition, it may reduce the number of index scans required to resolve the filter.

Enhanced Advantage Optimized Filter (AOF) cost estimations for improved ordering of filter segments combined with AND operators. This enhancement can improve performance for both navigational operations that set AOFs directly and for SQL statements, which automatically use AOFs for optimization. Advantage is now able to more accurately estimate the cost of evaluating each segment and can order them appropriately and can make better decisions on when to short-circuit the index scans.

Improved Advantage Optimized Filter (AOF) optimization and record traversal when a binary DELETED() index exists. This enhancement provides a fix for an issue where large numbers of deleted records at the beginning of a table had to be traversed at the server in order to read the first record in the table. See Binary Indexes in the Help File for details.

Record Count Operations

Improved record count operations on DBF tables when a binary DELETED() index exists. See Binary Indexes in the Help File for details.

Efficient Table Creation

Improved table creation efficiency. The parsing of field definitions is faster and now uses fewer memory allocations. This affects temporary table creation, trigger execution, stored procedure parameter passing, and various other operations that either explicitly or implicitly involves table creation.

Indexing

Performance improvements have been made to Advantage's low level indexing code. These improvements increase the speed of most index operations including updates, inserts, deletes, and seeks.

Referential Integrity Cascade Operations

Improved the performance of referential integrity cascade operations.

Appending and Deleting Records

Improved the performance of appending records and deleting records. Modifications were made to optimize table header writes required with each update.

Temporary File Handling

Improved cache usage with intermediate files (temporary files used by the SQL engine). In previous versions of Advantage, it was possible for the cache to be overrun with intermediate files. The lazy cleanup of intermediate files was replaced by active deletion, which can provide much better throughput especially on a busy system.

Improved the performance of temporary tables by storing their data in memory when possible.

Added functionality to reuse temporary files. When Advantage does not have enough cache memory to hold a result file from a static cursor, a temporary table, or intermediate query files, it uses a physical temporary file on disk to store the information. In version 10, these files are stored for a short time in a temporary file pool for reuse. Avoiding physical file creation and deletion can improve performance in a busy system.

Memo Header Caching

Improved caching of ADM and FPT memo headers. Advantage no longer writes the physical header to disk on every update. Page recycle information is maintained in the cached header with a safe version of the header residing on disk in case of a power outage. Now only the first 4 bytes of the root are written and only if the file length changes (when new blocks are written to the file).

Table Open Performance

Removed a retry loop for server-side table opens. This modification makes attempts to open a table that does not exist much faster.

Cached File Pool

Improved searching of the cached file pool. In a very busy system, the number of closed intermediate files could be somewhat large and it is a global pool. Reducing the search time relaxes a potential global bottleneck.

Record Locks

Improved the storage algorithm used for record locks, increasing performance when Advantage is managing a large number of record locks (for example, during a long transaction).

Advantage Extended Procedures (AEPs)

Improved performance of stored procedures and Advantage Extended Procedures (AEPs) by using in-memory tables for the virtual __input and __output tables.

Worker Threads

Improved the efficiency of signaling worker threads when client requests are ready. An inefficient pattern was identified that potentially required worker threads to immediately stop and wait for a sync object after being signaled to run. This modification can improve a busy system that is handling a large number of small requests.

Rights Checking Behavior

The default behavior for rights checking has been changed. The new default behavior is to ignore the rights checking setting for table opens and creations and always ignore the client rights check. Free table opens in most clients would previously default to do rights checking; the client would do an existence check for a table before attempting to open it. For most applications, this unnecessary and potentially expensive check could result in long timeouts on the client. See Effects of Upgrading to Version 10 in the Help File for details.

Memory Management

Optimized ACE objects to avoid allocating a large number of relatively small portions of the heap. This modification avoids heap fragmentation and increases performance.

Advantage Data Architect Enhancements

Added support for Unicode files in the SQL Utility.

Added a new Collation property to connections in order to facilitate specifying a Unicode collation.

Added an ARC setting to control the font size in data grids.

ARC now highlights DBF deleted records in data grids when using the SHOW DELETED setting.

Added a protocol type setting to the remote management utility which allows users to test both UDP and TCP settings.

Reporting

Crystal Reports Settings

Added per-alias Crystal Reports settings instead of only providing global settings (for options like Collation, LockingMode, ShowDeleted, etc).

R&R ReportWorks Support

Many Advantage users have a repository of reports that were built with the R&R ReportWorks XBase edition from Liveware Publishing. Traditionally these reports used direct file access and could not utilize the security and performance features of the Advantage Database Server. Starting with Advantage version 10, R&R ReportWorks files using DBF/CDX tables can now be accessed via the Advantage Client Engine. See Advantage with R&R ReportWorks in the Help File for details.

New Clients and Platform Support

More 64-bit Clients

64-bit versions of the following clients have been added in Advantage version 10:

- ODBC
- OLE DB
- Linux PHP Driver
- adsbackup utility for Windows and Linux
- Advantage Local Server

to compliment these existing Advantage 9 64-bit clients:

- Advantage Client Engine (ACE) for Windows and Linux
- Advantage ADO.NET Provider

64-Bit Advantage ADO.NET Improvements

The Advantage ADO.NET provider can now detect the platform type at runtime and correctly load either ACE32.DLL or ACE64.DLL as appropriate. This means .NET applications using the Advantage ADO.NET provider no longer need to specify a platform target of x86 in order to work on 64-bit operating systems. The platform target can now remain at its default setting (Any CPU).

Support for the Latest Development Environments

- RAD Studio/Delphi 2009
- RAD Studio/Delphi 2010
- Visual Studio 2008, .NET 3.5, Entity .NET Framework and LINQ to Entities
- Visual Studio 2010, .NET 4.0, Entity .NET Framework and LINQ to Entities

Updated Platform Support

- Windows 7
- Windows Server 2008 SP2
- Windows Server 2008 R2

Support for Vulcan.NET

Official release of the Advantage driver for Vulcan.NET. Vulcan.NET is the next generation of the xBase family of languages.

No Delphi 3, Delphi 4, C++Builder 3, or C++Builder 4 Components

Support was dropped for these development environments in Advantage version 9, but we continued to ship the components as a courtesy. They no longer build with some product improvements we have made, and therefore will not be provided in Advantage version 10.

Installation and Configuration

Automatically Configured Worker Thread Count

The server will now automatically configure the number of worker threads when it starts up. If the configured worker thread count is not specified or is zero, Advantage Database Server will calculate the number of worker threads based on the number of processors on the system. See the worker thread configuration setting in the Help File for details.

Side-By-Side Server Installations

In some cases, it is useful to install multiple versions of Advantage on a single physical server. Typically, this is done when multiple Advantage-enabled applications are using the same physical server, but are shipped using different versions of Advantage. Starting with Advantage version 10, additional instances of Advantage can now be installed on the same physical server. See *Installing Multiple Instances* in the Help File for detailed installation instructions and additional details.

New Help File Format

All Advantage help files have been combined into a single HTML Help 1.0 (CHM) help file. In addition, many of the Advantage Tech Tips from the Developer's Zone have been included in the help file and will now show up in help file search results.

Server Discovery

A new API `AdsFindServers` has been implemented. It can be used to retrieve a list of instances of Advantage Database Server on a network. This API can be used in combination with a server-side alias to eliminate the need for end users to choose a database server and connection path.

Miscellaneous

New System Variables

All trigger metadata information is now available in SQL script triggers via new system variables, see *System Variables* in the Help File for details.

Support Capture Utility

The Advantage Support Capture Utility is now installed with the server, and can be used to easily bundle relevant files when working on issues with the Advantage Technical Services team.

Advantage 10.1

Support for Latest Development Environments

The Advantage Delphi Components now include support for Delphi XE.

Advantage now supports Lazarus 0.9.28.2 or greater (cross-platform class libraries for Free Pascal that emulate Delphi). The Windows installer must still be used to get the Lazarus packages and source code, which can then be used on the Windows platform, or copied to a Linux image (Macintosh is not supported, as our components still use the Advantage Client Engine, which only supports Windows and Linux). For details, see Getting Started with Lazarus in the help file.

Unicode Full Text Search Support

The Advantage Database Server now supports Full Text Search (FTS) on Unicode data.

- The Contains() scalar function can now be used with Unicode data as input in the filter expressions, Advantage Optimized Filter (AOF) expressions and SQL engine expressions.
- The Score() and ScoreDistinct() scalar functions in SQL engine now supports Unicode data as input.
- FTS indexes may be built on NChar, NVarChar, and NMemo Field type to speed up the searches using the Contains() scalar in AOF. The indexes will also improve performance of evaluating the Contains(), Score() and ScoreDistinct() expressions in the SQL engine.
- FTS with Unicode data is always case and diacritical insensitive.

Advantage Data Architect Copy/Paste Support for Rows

Advantage Data Architect now supports Copy and Pasting records into and out of the Table Browser and the SQL Utility. These records can be pasted to another grid or SQL Utility, into Microsoft Word, Excel, or directly into HTML email.

Support for BLOB fields is included, copy and pasting from tables with different fields can also be accomplished via the Field Mapping Utility. To access the copy and pasting functionality a new context menu button has been added. See Field Mapping Utility and Table Browser in the help file for more information.

Strong Encryption and FIPS Compliance

Advantage Database Server now supports strong cryptographic functionality that can be used in Federal Information Processing Standard (FIPS) 140-2 compliant products. The cryptographic functionality in versions prior to 10.1 is based on RC4, which is not a FIPS-compliant encryption algorithm. Beginning with v10.1, new encryption and communications support are available through libraries from The OpenSSL project.

The new cryptographic functionality is not available by default in Advantage products. It must be purchased separately with the FIPS Encryption Security Option Add-on. Please contact your Advantage sales representative or visit <http://www.sybase.com/products/databasemanagement/advantagedatabaseserver/encryption> for additional licensing information.

Note that enabling and using FIPS-compliant cryptography in Advantage Database Server does not make an application conform to FIPS 140-2; all parts of the application must be examined and possibly updated for FIPS-compliance.

The following summarizes the new cryptographic functionality:

- Added support for Transport Layer Security (TLS) v1.0 communications. TLS operates over TCP/IP and uses RSA for the key exchange, Advanced Encryption Standard (AES), 128-bit or 256-bit, for encryption and SHA-1 (Secure Hash Algorithm) for message authentication. These cipher suites are referred to as AES128-SHA and AES256-SHA.
- In addition, the cipher suite RC4-MD5, which uses RSA for the key exchange, RC4 for encryption and MD5 for message authentication, is also available. This cipher suite is not FIPS-compliant.
- Added support for data (table) encryption using 128-bit and 256-bit AES.
- Enhanced data encryption using 64-bit message numbers for each piece of encrypted data (e.g., record, memo, index page, etc.) to ensure unique initialization vectors (and, therefore, unique cipher streams) across a database. Each time a record is updated, a new message number is generated for it.
- Improved key strength by salting and hashing keys. This makes attacks via password dictionaries (rainbow tables) infeasible and makes brute force attacks much more expensive.
- Added the capability to encrypt the data dictionary files (.add, .am, .ai) with an externally provided password.
- Added the capability to run Advantage Database Server and the Advantage client in FIPS mode. This ensures that it is not possible to use any cryptographic functionality that is not FIPS approved. For example, if Advantage Database Server is running in FIPS mode, it is not possible to open data dictionaries that support RC4 or tables encrypted with RC4.
- Added system procedures sp_DecryptTable and sp_EncryptTable that can be used to change table structures to support AES encryption.
- Added system procedure sp_ChangeDDEncryptionType to convert data dictionaries to support AES encryption.
- Added system procedure sp_GetTableEncryptionType to retrieve the type of encryption used on a table.
- Added system procedure sp_GetSecurityInfo to retrieve information such as a connection's default encryption type, the communication type, the communication encryption type, and dictionary encryption state.