

# ODBC 3.0: What's New

This document describes what is new in ODBC 3.0 and why it is being done. Each section briefly describes a change and refers to the chapter in the ODBC 3.0 Programmer's Reference that fully explains the new material.

## General Goals

ODBC 3.0 is designed to achieve the following goals:

- Conformance to the X/Open CAE Specification for SQL Call Level Interface
- Increased performance
- Migration path for OLE-DB

## Descriptors

One of the major new features incorporated from the X/Open CAE Specification is the use of descriptors. A descriptor is a data structure that holds information about either columns in a result set or dynamic parameters in an SQL statement. Descriptors streamline many application operations, providing a direct and uniform way to access column or parameter data. Many of the enhancements achieved in ODBC 3.0 are a result of using descriptors.

Both column and parameter data is described in two separate descriptors, one that conceptually describes the data or parameters in the application buffers, the other that describes that data or parameters in the driver buffers.

For a general discussion of descriptors, see Chapter 13, "Descriptors." For a specific discussion of descriptors fields, and the manipulation of descriptors, see the sections on **SQLSetDescField** and **SQLSetDescRec** in Chapter 21, "ODBC Function Reference."

## Binding and Fetching Enhancements

ODBC 3.0 includes the following enhancements to binding and fetching column data:

- Quick rebinding is supported, in which an offset can be added to all addresses in the current bindings.
- Long data can be packed into a single, "out-of-line" piece of application memory.
- An application can fetch a row in the result set based on predicates.
- Result sets can be nested, providing for summary data to be mixed with the rest of the result set.

Filename: in.doc Project: \*Insert existing text here and delete this text. Do not remove the following paragraph.

Template: Author: . Last Saved By: .

Revision #: 2 Page: 1 of 6 Printed: 09/28/95 09:26 AM!Unexpected End of Expression

- Row status is kept both as input values and output values. An application can set the input row status to indicate that the row should be ignored when doing a bulk operation via **SQLSetPos**.

For more information, see the **SQLBindCol**, **SQLFetchScroll**, and **SQLSetPos** sections in Chapter 21, "ODBC Function Reference."

## Parameter Enhancements

The use of the descriptors allows the following enhancements to binding dynamic parameters:

- Row-wise binding of parameters is supported.
- An application can determine which parameter of an SQL statement caused **SQLExecDirect** or **SQLExecute** to return an error.
- Quick rebinding is supported, in which an offset can be added to all addresses in the current bindings.

For more information, see the **SQLBindParameter** and **SQLBindParam** sections in Chapter 21, "ODBC Function Reference."

## Diagnostics

In ODBC 3.0, information about the outcome of functions calls is included in a diagnostic data structure. Each environment, connection, statement, and descriptor handles has a diagnostic data structure. The structure's header fields return general function execution information; its record fields describe the latest errors or warnings return by an ODBC function associated with the handle. ODBC 3.0 includes the following enhancements on diagnostic information:

- The diagnostic data structure is extensible.
- Reading data from the structure is non-destructive.
- Status records occur according to their rank or severity.
- When an error occurs with one or more columns in error, but not the entire row, the column (or columns) that caused the error is indicated.

For more information, see Chapter 15, "Errors," and the **SQLGetDiagField** and **SQLGetDiagRec** sections in Chapter 21, "ODBC Function Reference."

## New Attributes

The connection and statement options in ODBC 2.0 have become attributes in ODBC 3.0, and the functions used to retrieve and set them have been changed as shown in the **Deprecated Functions** table. The environment attribute, and the

Filename: in.doc Project: \*Insert existing text here and delete this text. Do not remove the following paragraph.  
Template: Author: . Last Saved By: .  
Revision #: 2 Page: 2 of 6 Printed: 09/28/95 09:26 AM!Unexpected End of Expression

functions used to retrieve and set it, are new for ODBC 3.0.

The following environment, connection, and statement attributes are new. Statement attributes used solely as a convenience to set fields in descriptor headers are not listed here.

New Attribute	Attribute Type
SQL_ATTR_OUTPUT_NTS	Environment
SQL_ATTR_ASYNC_ENABLE	Connection
SQL_ATTR_AUTO_IPD	Connection
SQL_ATTR_USE_LOCATORS	Connection
SQL_ATTR_APP_ROW_DESC	Statement
SQL_ATTR_APP_PARAM_DESC	Statement
SQL_ATTR_IMP_ROW_DESC	Statement
SQL_ATTR_IMP_PARAM_DESC	Statement
SQL_ATTR_METADATA_ID	Statement

For more information, see the individual function sections in Chapter 21, “ODBC Function Reference.”

## Bookmark Support

The following enhancements to bookmark use have been added in ODBC 3.0.

- Variable-length bookmarks are supported, which means that a primary key or a unique index associated with a table can be used as a bookmark.
- An application can call **SQLSetPos** to update or delete a row by bookmark.
- Two bookmarks that point to rows in the same result set can be compared for equality or inequality.

## Asynchronous Processing

ODBC 3.0 allows drivers to support asynchronous processing on a statement-by-statement basis or on a connection-by-connection basis. When asynchronous processing is supported on a connection-by-connection basis, all statements in a connection must be processed either synchronously or asynchronously; a mixture of synchronous and asynchronous statements is not allowed.

## Locator Support

ODBC 3.0 provides support for large object (LOB) data. An application can

Filename: in.doc Project: \*Insert existing text here and delete this text. Do not remove the following paragraph.  
 Template: Author: . Last Saved By: .  
 Revision #: 2 Page: 3 of 6 Printed: 09/28/95 09:26 AM!Unexpected End of Expression

manipulate an LOB in an efficient, random access fashion by using a four-byte locator to represent the data on the server. This is based on the concept of locators in SQL 3.

The following functions have been added in ODBC 3.0 to work with locators:

**SQLGetLength**

Retrieves the length of a string value represented by a locator.

**SQLGetPosition**

Retrieves the starting position of a string value within another string value that is represented by a locator.

**SQLGetSubString**

Retrieves a portion of a string value represented by a locator; or creates a new large object and retrieves a locator to that value.

**SQLLocatedUpdate**

Appends, replaces, or deletes all or part of the large object represented by a locator.

**SQLLocator**

Specifies whether a locator is held or freed.

For more information, see the individual function sections in Chapter 21, “ODBC Function Reference.”

## Unicode Support

ODBC 3.0 provides support for Unicode applications with Unicode-ANSI mapping in the Driver Manager, and Unicode function passthrough to Unicode drivers. ODBC 3.0 supports Unicode and ANSI versions of all functions that accept strings as parameters. New SQL data types are added as part of the extended set of SQL data types.

For more information, see the “Unicode” section in Chapter 17, “Programming Considerations.”

## New and Modified Data Types

The following changes have been made to data types in ODBC 3.0:

- Interval data types describing the difference between two dates and times have been added to the extended SQL and C data types.
- Binary and character large object data types have been added.
- Unicode data types have been added.
- The SQL and C datetime data types have been changed.

For more information, see Appendix D, “Data Types.”

## Deprecated Functions

The functions listed in the left column of the table below are deprecated in ODBC 3.0. Calls to these functions by ODBC 2.0 applications are mapped by the ODBC 3.0 Driver Manager to the functions in the right column.

Deprecated Functions	ODBC 3.0 Functions
SQLAllocEnv	SQLAllocHandle
SQLAllocStmt	SQLAllocHandle
SQLAllocConnect	SQLAllocHandle
SQLColAttributes	SQLColAttribute
SQLError	SQLGetDiagRec/SQLGetDiagField
SQLFreeConnect	SQLFreeHandle
SQLFreeEnv	SQLFreeHandle
SQLGetConnectOption	SQLGetConnectAttr
SQLGetStmtOption	SQLGetStmtAttr
SQLSetConnectOption	SQLSetConnectAttr
SQLSetStmtOption	SQLSetStmtAttr
SQLTransact	SQLEndTran

For more information on the mapping, see the “Deprecated Functions and their Mappings” section in Chapter 17, “Programming Considerations,” and the individual function sections in Chapter 21, “ODBC Function Reference.”

In addition, the **SQLInstallODBC** function in the installer DLL will not be supported in ODBC 3.0.

## Programmer’s Reference Changes

Aside from the technical changes mandated by the changes listed above, the ODBC Programmer’s Reference has significantly changed for ODBC 3.0.

Parts 1 through 3 have been completely rewritten. Whereas the ODBC 2.0 Programmer’s Reference contained separate parts for developing applications (Part 2) and developing drivers (Part 3), this information has been rolled into a single, cohesive part: Part 2, “Developing Applications and Drivers.” The scope of the material in Part 2 has been considerably expanded. Note that many ODBC 3.0 changes have not yet been included in Parts 1 through 3.

Part 4 has been updated with the changes made to the ODBC functions. The reference sections in Part 4 may change a great deal before the final release of

Filename: in.doc Project: \*Insert existing text here and delete this text. Do not remove the following paragraph.  
 Template: Author: . Last Saved By: .  
 Revision #: 2 Page: 5 of 6 Printed: 09/28/95 09:26 AM!Unexpected End of Expression

ODBC 3.0.

Appendixes A and D have been changed for version 3.0. Appendixes B, C, E, F, and G have not as yet been updated.

## ODBC 3.0: What's New

---