Related Functions

For information about	See
Returning the setting of a connection option	**SQLGetConnectOption** (extension)
Determining if a driver supports a function	**SQLGetFunctions** (extension)
Returning the setting of a statement option	**SQLGetStmtOption** (extension)
Returning information about a data source's data types	**SQLGetTypeInfo** (extension)

SQLGetStmtOption

ODBC 1.0

Extension Level 1 **SQLGetStmtOption** returns the current setting of a statement option.

Syntax RETCODE **SQLGetStmtOption**(*hstmt*, *fOption*, *pvParam*)

The **SQLGetStmtOption** function accepts the following arguments:

Type	Argument	Use	Description
HSTMT	*hstmt*	Input	Statement handle.
UWORD	*fOption*	Input	Option to retrieve.
PTR	*pvParam*	Output	Value associated with *fOption*. Depending on the value of *fOption*, a 32-bit integer value or a pointer to a null-terminated character string will be returned in *pvParam*.

Returns SQL_SUCCESS, SQL_SUCCESS_WITH_INFO, SQL_ERROR, or SQL_INVALID_HANDLE.

Diagnostics When **SQLGetStmtOption** returns SQL_ERROR or SQL_SUCCESS_WITH_INFO, an associated SQLSTATE value may be obtained by calling **SQLError**. The following table lists the SQLSTATE values commonly returned by **SQLGetStmtOption** and explains each one in the context of this function; the notation "(DM)" precedes the descriptions of SQLSTATEs returned by the Driver Manager. The return code associated with each SQLSTATE value is SQL_ERROR, unless noted otherwise.

SQLSTATE	Error	Description
01000	General warning	Driver-specific informational message. (Function returns SQL_SUCCESS_WITH_INFO.)
24000	Invalid cursor state	The argument *fOption* was SQL_ROW_NUMBER or SQL_GET_BOOKMARK and the cursor was not open, or the cursor was positioned before the start of the result set or after the end of the result set.
IM001	Driver does not support this function	(DM) The driver corresponding to the *hstmt* does not support the function.

SQLSTATE	Error	Description
S1000	General error	An error occurred for which there was no specific SQLSTATE and for which no implementation-specific SQLSTATE was defined. The error message returned by **SQLError** in the argument *szErrorMsg* describes the error and its cause.
S1001	Memory allocation failure	The driver was unable to allocate memory required to support execution or completion of the function.
S1010	Function sequence error	(DM) An asynchronously executing function was called for the *hstmt* and was still executing when this function was called. (DM) **SQLExecute**, **SQLExecDirect**, or **SQLSetPos** was called for the *hstmt* and returned SQL_NEED_DATA. This function was called before data was sent for all data-at-execution parameters or columns.
S1011	Operation invalid at this time	The *fOption* argument was SQL_GET_BOOKMARK and the value of the SQL_USE_BOOKMARKS statement option was SQL_UB_OFF.
S1092	Option type out of range	(DM) The value specified for the argument *fOption* was in the block of numbers reserved for ODBC connection and statement options, but was not valid for the version of ODBC supported by the driver.
S1109	Invalid cursor position	The *fOption* argument was SQL_GET_BOOKMARK or SQL_ROW_NUMBER and the value in the *rgfRowStatus* array in **SQLExtendedFetch** for the current row was SQL_ROW_DELETED or SQL_ROW_ERROR.
S1C00	Driver not capable	The value specified for the argument *fOption* was a valid ODBC statement option for the version of ODBC supported by the driver, but was not supported by the driver. The value specified for the argument *fOption* was in the block of numbers reserved for driver-specific connection and statement options, but was not supported by the driver.

Comments

The following table lists statement options for which corresponding values can be returned, but not set. The table also lists the version of ODBC in which they were introduced. For a list of options that can be set and retrieved, see **SQLSetStmtOption**. If *fOption* specifies an option that returns a string, *pvParam* must be a pointer to storage for the string. The maximum length of the string will be SQL_MAX_OPTION_STRING_LENGTH bytes (excluding the null termination byte).

fOption	***pvParam* contents**
SQL_GET_BOOKMARK (ODBC 2.0)	A 32-bit integer value that is the bookmark for the current row. Before using this option, an application must set the SQL_USE_BOOKMARKS statement option to SQL_UB_ON, create a result set, and call **SQLExtendedFetch**. To return to the rowset starting with the row marked by this bookmark, an application calls **SQLExtendedFetch** with the SQL_FETCH_BOOKMARK fetch type and *irow* set to this value. Bookmarks are also returned as column 0 of the result set.
SQL_ROW_NUMBER (ODBC 2.0)	A 32-bit integer value that specifies the number of the current row in the entire result set. If the number of the current row cannot be determined or there is no current row, the driver returns 0.

Related Functions

For information about	**See**
Returning the setting of a connection option	**SQLGetConnectOption** (extension)
Setting a connection option	**SQLSetConnectOption** (extension)
Setting a statement option	**SQLSetStmtOption** (extension)

SQLGetTypeInfo

ODBC 1.0

Extension Level 1 **SQLGetTypeInfo** returns information about data types supported by the data source. The driver returns the information in the form of an SQL result set.

Important Applications must use the type names returned in the TYPE_NAME column in **ALTER TABLE** and **CREATE TABLE** statements; they must not use the sample type names listed in Appendix C, "SQL Grammar." **SQLGetTypeInfo** may return more than one row with the same value in the DATA_TYPE column.

Syntax RETCODE **SQLGetTypeInfo**(*hstmt*, *fSqlType*)

The **SQLGetTypeInfo** function accepts the following arguments:

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.

WHAT WILL YOU BUILD? | sales@docketalarm.com | 1-866-77-FASTCASE

fastcase® Smarter legal research.