

QryMethod Function

Description

Executes a query method; a query corresponds to a statement handle for a Connection Object.

Syntax

flag = **QryMethod** (*hQry*, *method*, *arg1*, *arg2*, *arg3*, *arg4*, *arg5*)

Parameters

The QryMethod function has the following parameters:

Parameter	Description
<i>hQry</i>	Handle to the Query.
<i>method</i>	See methods below.
<i>arg1...arg5</i>	Method specific; see methods below.

Method	Description														
QRY_CANCEL\$	<div> <div> <div>Cancels an active query.</div> <table><tr><th>Value</th><th>Description</th></tr><tr><td>HQry [in]</td><td>Query handle.</td></tr></table> </div> </div>	Value	Description	HQry [in]	Query handle.										
Value	Description														
HQry [in]	Query handle.														
QRY_DESTROY\$	<div> <div> <div>Releases a Query handle.</div> <table><tr><th>Value</th><th>Description</th></tr><tr><td>HQry [in]</td><td>Query handle.</td></tr></table> </div> <div>Error returned for an invalid handle or if an error occurred destroying the query.</div> </div>	Value	Description	HQry [in]	Query handle.										
Value	Description														
HQry [in]	Query handle.														
QRY_EXECUTE\$	<div> <div> <div>Executes a query and checks for results.</div> <table><tr><th>Value</th><th>Description</th></tr><tr><td>hQry [in]</td><td>Query handle.</td></tr><tr><td>arg1 [in]</td><td>Script</td></tr><tr><td>arg2 [in]</td><td>Execution type (defaults to "standard result set").</td></tr><tr><td>arg3 [out]</td><td>Set to TRUE\$ if there are results, FALSE\$ otherwise.</td></tr></table> </div> </div>	Value	Description	hQry [in]	Query handle.	arg1 [in]	Script	arg2 [in]	Execution type (defaults to "standard result set").	arg3 [out]	Set to TRUE\$ if there are results, FALSE\$ otherwise.				
Value	Description														
hQry [in]	Query handle.														
arg1 [in]	Script														
arg2 [in]	Execution type (defaults to "standard result set").														
arg3 [out]	Set to TRUE\$ if there are results, FALSE\$ otherwise.														
QRY_GETERROR\$	<div> <div> <div>Retrieves all pending query errors from the DS/XO API.</div> <table><tr><th>Value</th><th>Description</th></tr><tr><td>hQry [in]</td><td>Query handle.</td></tr><tr><td>arg1 [out]</td><td>Local error list (@VM-delimited).</td></tr><tr><td>arg2 [out]</td><td>Local severity list.</td></tr><tr><td>arg3 [out]</td><td>Native error list.</td></tr><tr><td>arg4 [out]</td><td>Native severity list.</td></tr><tr><td>arg5 [out]</td><td>Error text list.</td></tr></table> </div> <div>Error returned if no errors were pending.</div> </div>	Value	Description	hQry [in]	Query handle.	arg1 [out]	Local error list (@VM-delimited).	arg2 [out]	Local severity list.	arg3 [out]	Native error list.	arg4 [out]	Native severity list.	arg5 [out]	Error text list.
Value	Description														
hQry [in]	Query handle.														
arg1 [out]	Local error list (@VM-delimited).														
arg2 [out]	Local severity list.														
arg3 [out]	Native error list.														
arg4 [out]	Native severity list.														
arg5 [out]	Error text list.														

QRY_GETROW\$	Retrieves a row from the result set.											
	Value	Description										
	hQry [in]	Query handle.										
	arg1 [out]	Result row (@FM-delimited).										
	arg2 [in]	Fetch direction (defaults to FD_FORWARDS\$).										
	arg3 [in]	Conversion (defaults to CONV_OI\$).										
QRY_LISTCOLUMN\$	Creates a result set (as if a script had been executed with QRY_EXECUTES\$) of the columns in a database object. The result set contains the columns COLUMN NAME, DATA TYPE, PRECISION, SCALE, NULLABLE, TYPE NAME, LENGTH, RADIX, and REMARKS.											
	Column	Description										
	COLUMN NAME	The name of the column.										
	DATA TYPE	The data type of the column. Data types are defined in the insert DSXO_API and are prefixed with "DS_".										
	PRECISION	The column's precision. The precision of a column is its length, if it is a character or binary type, the amount of detail it holds for time and date/time types, or the total number of digits for the decimal type. For date/time, the precision is the number of significant characters in the string "yyyy-mm-dd hh:mm:ss.ffff". For time, the precision is the number of significant characters in the string "hh:mm:ss.ffff".										
	SCALE	The column's scale. The scale of a column is the number of digits to the right of the decimal point for decimal data types or the number of digits of precision for fractions of a second for time and date/time data types.										
	NULLABLE	True if the column can contain null values.										
	TYPE NAME	The database-specific name of the column's type.										
	LENGTH	The length used by the database to store the column.										
	RADIX	For numeric types, the base of the number. Binary numbers are base-2, decimal numbers are base-10.										
	REMARKS	Miscellaneous information that the database can choose to provide.										
		<table><tr><td>Value</td><td>Description</td></tr><tr><td>hQry [in]</td><td>Query handle.</td></tr><tr><td>arg1 [in]</td><td>Database object name.</td></tr><tr><td>arg2 [in]</td><td>An optional owner name; defaults to null.</td></tr><tr><td>arg3 [in]</td><td>An optional qualifier name; defaults to null.</td></tr></table>	Value	Description	hQry [in]	Query handle.	arg1 [in]	Database object name.	arg2 [in]	An optional owner name; defaults to null.	arg3 [in]	An optional qualifier name; defaults to null.
	Value	Description										
hQry [in]	Query handle.											
arg1 [in]	Database object name.											
arg2 [in]	An optional owner name; defaults to null.											
arg3 [in]	An optional qualifier name; defaults to null.											

QRY_LIS TTABLES\$	<p>Creates a result set (as if a script had been executed with QRY_EXECUTE\$) of the tables in the connected database. The result set contains the columns QUALIFIER, OWNER, NAME, TYPE, and REMARKS.</p> <table><tr><th>Column</th><th>Description</th></tr><tr><td>QUALIFIER</td><td>Often corresponds to the DOS path for local databases or the database name for SQL Server databases. It is datasource dependent.</td></tr><tr><td>OWNER</td><td>For datasources that support owners for objects, this specifies the owner of the object.</td></tr><tr><td>NAME</td><td>The name of the object.</td></tr><tr><td>TYPE</td><td>The type of the object. This will usually be one of the following: SYSTEM TABLE, TABLE, GLOBAL TEMPORARY, LOCAL TEMPORARY, ALIAS, SYNONYM, VIEW, PROCEDURE, RULE, DEFAULT, or TRIGGER.</td></tr><tr><td>REMARKS</td><td>Miscellaneous information.<table><tr><th>Value</th><th>Description</th></tr><tr><td><i>hQry [in]</i></td><td>Query handle.</td></tr><tr><td><i>arg1 [in]</i></td><td>Object types. (Also defined in the insert DSXO_API.)</td></tr></table><table><tr><th>Symbol</th><th>Value</th><th>Description</th></tr><tr><td>OBJ_DATATABLES\$</td><td>1</td><td>Data tables</td></tr><tr><td>OBJ_SYSTABLES\$</td><td>2</td><td>System tables</td></tr><tr><td>OBJ_TEMPTABLES\$</td><td>4</td><td>Temporary tables</td></tr><tr><td>OBJ_VIEWSS\$</td><td>8</td><td>Views</td></tr><tr><td>OBJ_ALIASES\$</td><td>16</td><td>Aliases</td></tr><tr><td>OBJ_PROCS\$</td><td>32</td><td>Stored Procedures</td></tr><tr><td>OBJ_FUNCSS\$</td><td>64</td><td>Database functions</td></tr></table></td></tr></table>	Column	Description	QUALIFIER	Often corresponds to the DOS path for local databases or the database name for SQL Server databases. It is datasource dependent.	OWNER	For datasources that support owners for objects, this specifies the owner of the object.	NAME	The name of the object.	TYPE	The type of the object. This will usually be one of the following: SYSTEM TABLE, TABLE, GLOBAL TEMPORARY, LOCAL TEMPORARY, ALIAS, SYNONYM, VIEW, PROCEDURE, RULE, DEFAULT, or TRIGGER.	REMARKS	Miscellaneous information. <table><tr><th>Value</th><th>Description</th></tr><tr><td><i>hQry [in]</i></td><td>Query handle.</td></tr><tr><td><i>arg1 [in]</i></td><td>Object types. (Also defined in the insert DSXO_API.)</td></tr></table> <table><tr><th>Symbol</th><th>Value</th><th>Description</th></tr><tr><td>OBJ_DATATABLES\$</td><td>1</td><td>Data tables</td></tr><tr><td>OBJ_SYSTABLES\$</td><td>2</td><td>System tables</td></tr><tr><td>OBJ_TEMPTABLES\$</td><td>4</td><td>Temporary tables</td></tr><tr><td>OBJ_VIEWSS\$</td><td>8</td><td>Views</td></tr><tr><td>OBJ_ALIASES\$</td><td>16</td><td>Aliases</td></tr><tr><td>OBJ_PROCS\$</td><td>32</td><td>Stored Procedures</td></tr><tr><td>OBJ_FUNCSS\$</td><td>64</td><td>Database functions</td></tr></table>	Value	Description	<i>hQry [in]</i>	Query handle.	<i>arg1 [in]</i>	Object types. (Also defined in the insert DSXO_API.)	Symbol	Value	Description	OBJ_DATATABLES\$	1	Data tables	OBJ_SYSTABLES\$	2	System tables	OBJ_TEMPTABLES\$	4	Temporary tables	OBJ_VIEWSS\$	8	Views	OBJ_ALIASES\$	16	Aliases	OBJ_PROCS\$	32	Stored Procedures	OBJ_FUNCSS\$	64	Database functions
Column	Description																																										
QUALIFIER	Often corresponds to the DOS path for local databases or the database name for SQL Server databases. It is datasource dependent.																																										
OWNER	For datasources that support owners for objects, this specifies the owner of the object.																																										
NAME	The name of the object.																																										
TYPE	The type of the object. This will usually be one of the following: SYSTEM TABLE, TABLE, GLOBAL TEMPORARY, LOCAL TEMPORARY, ALIAS, SYNONYM, VIEW, PROCEDURE, RULE, DEFAULT, or TRIGGER.																																										
REMARKS	Miscellaneous information. <table><tr><th>Value</th><th>Description</th></tr><tr><td><i>hQry [in]</i></td><td>Query handle.</td></tr><tr><td><i>arg1 [in]</i></td><td>Object types. (Also defined in the insert DSXO_API.)</td></tr></table> <table><tr><th>Symbol</th><th>Value</th><th>Description</th></tr><tr><td>OBJ_DATATABLES\$</td><td>1</td><td>Data tables</td></tr><tr><td>OBJ_SYSTABLES\$</td><td>2</td><td>System tables</td></tr><tr><td>OBJ_TEMPTABLES\$</td><td>4</td><td>Temporary tables</td></tr><tr><td>OBJ_VIEWSS\$</td><td>8</td><td>Views</td></tr><tr><td>OBJ_ALIASES\$</td><td>16</td><td>Aliases</td></tr><tr><td>OBJ_PROCS\$</td><td>32</td><td>Stored Procedures</td></tr><tr><td>OBJ_FUNCSS\$</td><td>64</td><td>Database functions</td></tr></table>	Value	Description	<i>hQry [in]</i>	Query handle.	<i>arg1 [in]</i>	Object types. (Also defined in the insert DSXO_API.)	Symbol	Value	Description	OBJ_DATATABLES\$	1	Data tables	OBJ_SYSTABLES\$	2	System tables	OBJ_TEMPTABLES\$	4	Temporary tables	OBJ_VIEWSS\$	8	Views	OBJ_ALIASES\$	16	Aliases	OBJ_PROCS\$	32	Stored Procedures	OBJ_FUNCSS\$	64	Database functions												
Value	Description																																										
<i>hQry [in]</i>	Query handle.																																										
<i>arg1 [in]</i>	Object types. (Also defined in the insert DSXO_API.)																																										
Symbol	Value	Description																																									
OBJ_DATATABLES\$	1	Data tables																																									
OBJ_SYSTABLES\$	2	System tables																																									
OBJ_TEMPTABLES\$	4	Temporary tables																																									
OBJ_VIEWSS\$	8	Views																																									
OBJ_ALIASES\$	16	Aliases																																									
OBJ_PROCS\$	32	Stored Procedures																																									
OBJ_FUNCSS\$	64	Database functions																																									
QRY_TR ANSLAT EFLAG\$	<p>Translates the bit-masked flag returned from the DS/XO API into TRUE\$ for success and FALSE\$ for failure; success includes both success and success with information (meaning possible pending messages) and failure includes an error (meaning possible pending messages), an invalid handle, and no more data.</p> <table><tr><th>Value</th><th>Description</th></tr><tr><td><i>arg1 [in]</i></td><td>The flag value returned from the DS/XO API.</td></tr></table>	Value	Description	<i>arg1 [in]</i>	The flag value returned from the DS/XO API.																																						
Value	Description																																										
<i>arg1 [in]</i>	The flag value returned from the DS/XO API.																																										

Returns

TRUE for successful execution or **FALSE** for failure.

See also

[XOGetProperty\(\)-XO_TABLEDESCRIPT\\$](#)

Remarks

```

* example for QryMethod:
function ExecuteScript(Script)
$insert XO_Equates
* returns @rm/@fm delimited results
Results = ""
* create connection (this could take params if you wanted,
* but calling it with no params lets the user choose what
* connection, etc. this is how the query window calls it:
hXO = XOInstance()
hQry = 0
if hXO then
    * create the query handle for the connection handle
    hQry = QryInstance(hXO)
    if hQry then
        * execute a script
        Flag = QryMethod(hQry, QRY_EXECUTE$, Script)
        if Flag then
            /* retrieve results and stick them in an @rm
            delimited array */
            loop
                Flag = QryMethod(hQry, QRY_GETROW$, Row)
            while Flag
                Results := @rm: Row
            repeat
                Results [1,1] = ""
            * cancel script
            Flag = QryMethod(hQry, QRY_CANCEL$)
        end else
            gosub error
        end
        * close the query handle
        Flag = QryMethod(hQry, QRY_DESTROY$)
    end else
        gosub error
    end
    * close the connection
    Flag = XOMethod(hXO, XO_DESTROY$)
end else
    gosub error
end
return Results
* Error handling
Error:
    if hQry then
        Flag = QryMethod(hQry, QRY_GETERROR$, "", "", "", |
            "",Text)
    end else
        Flag = XOMethod(hXO, QRY_GETERROR$, "", "", "", |
            "",Text)
    end
    convert @vm to @tm in Text
    Def = ""
    Def<MTEXT$> = Text
    Def<MCAPTION$> = "ExecuteScript Error"
    Def<MICON$> = "!"
    Msg(@window, Text)
    Results = ""
return

```