

XO_EQUATES \$Insert Record

```
Compile Insert XO_Equates
* Basic+ functions
declare function XOGetCompDef          ;* body = (DataSourceName, [DataSourceType])
declare function XOInstance            ;* hXO = (DataSourceName, [DataSourceType, LoginID, Password, Timeout,
Options, Scope])
declare function XOMethod              ;* flag = (hXO, MethodName [, Param1, ...])
declare function XOGetProperty         ;* flag = (hXO, PropertyName, RetValue [, Qualifier])
declare function XOSetProperty         ;* flag = (hXO, PropertyName, Value, [, Qualifier])
declare function QryInstance          ;* hQry = (hXO)
declare function QryMethod            ;* flag = (hQry, MethodName [, Param1, ...])
declare function QryGetProperty       ;* flag = (hQry, PropertyName, RetValue [, Qualifier])
declare function QrySetProperty       ;* flag = (hQry, PropertyName, Value, [, Qualifier])
* by declaring the Connection functions as subroutines, the return value can be ignored
declare subroutine XOGetProperty , XOSetProperty , XOMethod
declare subroutine QryGetProperty, QrySetProperty, QryMethod
* Repository constants
equ XO_TABLE$          to "SYSREPOSDATASOURCES"
equ XO_TYPE$           to "DATASOURCE"
equ XO_CLASS$          to "CONNECTION"
* XO parameter information
equ XO_PARM_FILE$      to "SYSENV"
equ XO_PARM_KEY$       to "CFG_CONNECTION"
equ XO_DATASOURCETYPES$ to 1      ;* @vm delimited list of all supported Data Source Types
equ XO_ATTRIBUTES$     to 2      ;* for each Data Source Type, @svm delimited list of attribute names
equ XO_ATTRIBDESC$     to 3      ;* a short description for each attribute name
equ XO_DATASOURCETYPESUB$ to 4    ;* a function that performs datasource specific tasks
* options for XOInstance()
equ XO_USETRANS$       to 0      ;* use transactions (default)
equ XO_NOTRANS$        to 1      ;* do not use transactions
equ XO_NOSHARE$        to 0      ;* do not share XO (default)
equ XO_SCOPEDSHARE$    to 2      ;* share only within the specified scope
equ XO_GLOBALSHARE$    to 4      ;* share globally (anyone can use it)
equ XO_LOGINDIALOG$    to 0      ;* if login fails, allow entry of login ID and password (default)
equ XO_NODIALOG$       to 8      ;* if login fails, just return error
equ XO_ONLYDIALOG$     to 16     ;* don't attempt to log in without allowing the user to enter connection
data
equ XO_RESOLVE$        to 0      ;* attempt to fill in blank connection parms (default)
equ XO_NORESOLVE$     to 32     ;* do not attempt to fill in blank connection parms
* connection methods
equ XO_DESTROY$        to 1      ;* destroy passed XO
equ XO_TRANSLATEFLAG$  to 2      ;* Param1 is a DS/XO API FLAG, returns TRUE$=success or FALSE$=failure
equ XO_GETERROR$       to 3      ;* Retrieves pending errors for the Connection Object
equ XO_COMMITTRAN$     to 4      ;* commits the current transaction
equ XO_ROLLBACKTRAN$   to 5      ;* rolls back the current transaction
equ XO_ADDREF$         to 6      ;* add reference to Connection Object handle
equ XO_METHOD_MIN$     to XO_DESTROY$
equ XO_METHOD_MAX$     to XO_ADDREF$
* connection properties
equ XO_VALID$          to 1      ;* TRUE$ if Connection Object handle is valid
equ XO_QRYLIST$        to 2      ;* @vm delimited list of Query handles for the Connection Object handle
equ XO_DSLLIST$        to 3      ;* @vm delimited list of DataSet handles for the Connection Object handle
equ XO_SOURCENAME$     to 4      ;* data source name
equ XO_SOURCETYPE$     to 5      ;* data source type
equ XO_TABLELIST$      to 6      ;* list of data source tables (Arg=TableTypes)
equ XO_TABLEDESCRPT$   to 7      ;* returns column descriptions for the specified table (Arg=TableName)
equ XO_TRANSACTIONS$   to 8      ;* returns boolean (TRUE=transactions specified)
equ XO_OPTIONS$        to 9      ;* returns bitmasked options as passed to XOInstance()
equ XO_SCOPE$          to 10     ;* returns scope of connection as passed to XOInstance()
equ XO_INTERNALHANDLE$ to 11     ;* returns internal handle used by connection object
equ XO_TYPEQUOTED$     to 12     ;* returns information on which types are quoted; @fm-delim'd array
indexed by DT_... values (see DSXO_API insert)
equ XO_QUOTECHAR$      to 13     ;* returns the character used to quote data, eg. '
equ XO_QUOTEDQUOTE$    to 14     ;* returns the quote character as it appears if it is quoted, eg. '' or \'
equ XO_NULLVALUE$      to 15     ;* returns value used to specify null value, eg. NULL
equ XO_FORMATDATETIME$ to 16     ;* returns formats for date (field 1), time (field 2), and datetime (field
3) using strftime() notation (see DSXO_API)
equ XO_PROPERTY_MIN$   to XO_VALID$
```

```

equ XO_PROPERTY_MAX$          to XO_FORMATDATETIME$
* query methods
equ QRY_DESTROY$              to 1      /* destroy passed query
equ QRY_TRANSLATEFLAG$        to 2      /* Param1 is a DS/XO API FLAG, returns TRUE$=success or FALSE$=failure
equ QRY_GETERROR$              to 3      /* Retrieves pending errors for the query
equ QRY_EXECUTE$              to 4      /* executes a script
equ QRY_GETROW$                to 5      /* gets the next result row
equ QRY_CANCEL$                to 6      /* cancels the query
equ QRY_LISTTABLES$            to 7      /* create result set of tables (Arg1=TableTypes)
equ QRY_LISTCOLUMNS$          to 8      /* create result set of columns (Arg1=Table)
equ QRY_METHOD_MIN$            to QRY_DESTROY$
equ QRY_METHOD_MAX$            to QRY_LISTCOLUMNS$
* query properties
equ QRY_VALID$                 to 1      /* returns TRUE$ if handle is valid
equ QRY_CONNECTION$            to 2      /* connection handle for the query
equ QRY_ROWCOUNT$             to 3      /* number of rows in query
equ QRY_COLCOUNT$             to 4      /* number of columns in query
equ QRY_COLDESCRIPT$           to 5      /* column name, type, precision, scale, nullable, and OI type (Arg=iCol)
equ QRY_COLNAME$               to 6      /* column name (Arg=iCol)
equ QRY_COLTYPE$               to 7      /* column type (Arg=iCol)
equ QRY_COLPRECISION$          to 8      /* column precision (Arg=iCol)
equ QRY_COLSCALE$              to 9      /* column scale (Arg=iCol)
equ QRY_COLNULLABLE$           to 10     /* column nullable (Arg=iCol)
equ QRY_COLOITYPE$             to 11     /* column OpenInsight type (Arg=iCol)
equ QRY_TIMEOUT$               to 12     /* timeout in seconds
equ QRY_PROPERTY_MIN$          to QRY_VALID$
equ QRY_PROPERTY_MAX$          to QRY_TIMEOUT$
* datasource type functions
* flag = fn(Instruction, In, Out, Arg )
declare function ODBC_Sub
declare function SQLServer_Sub
* instructions for datasource type function
equ DST_HASOPTIONS$ to 1      /* in = xo_attribute, out = true if it has options
equ DST_CHOOSEOPTION$ to 2     /* in = xo_attribute, out = chosen option, arg = current setting
equ DST_INSERTFROMSELECT$ to 3 /* take a SELECT script and return an INSERT script
equ DST_UPDATEFROMSELECT$ to 4 /* take a SELECT script and return an UPDATE script
equ DST_DELETEFROMSELECT$ to 5 /* take a SELECT script and return an DELETE script

```