Explanation of the Code

Through BASIC+ calls, OpenInsight communicates with the Access database through use of objects. Two objects are used:

- The **Connection object**, which manages the connection between OpenInsight and Access. The functions that use the Connection object have the prefix **XO** (XOGetProperty(), XOInstance(), XOMethod(), and XOSetProperty()). BASIC+ communicates with the Northwind database through the **NWIND** connection, defined previously in the Client/Server workspace.
- The **Query object**, which sends the query to Access using the Connection object and receives the results. The functions that use the Query object have the prefix **Qry** (QryGetProperty(), QryInstance(), QryMethod(), and QrySetProperty()). Once the **NWIND** connection object has been established ("instantiated" is the fancy object-oriented term), the Query object can be used to run database queries using that connection.

The XO_Equates $Insert record contains declarations and equates for arguments and functions that are used with the Connection object and the Query object.

The logic flow is as follows:

1. **Create ("instantiate") a Connection object**. To instantiate the object, call XOInstance(), passing the connection name. If successful, the connection "handle", a nonzero number, is returned in **hXO**. Calling XOInstance() is the equivalent, in code, of the **NWIND** connection object we created in the Client/Server workspace. The code is as follows:

```
hXO = XOInstance('NWIND')
```

2. **Instantiate a Query object**. After we have a handle to the Connection object, we can use it to create a Query object. This is done using QryInstance(). If successful, the query handle will be a nonzero number, returned in **hQry**. The code is as follows:

```
hQry = QryInstance(hXO)
```

3. **Execute the Query**. The query handle can be used to execute the query. To execute a "method" (function), use QryMethod(). The **QRY_EXECUTE$** code (equated to 4 in XO_Equates) is used to execute the query. The third parameter is the actual query to execute. The code is shown below:

```
flag = QryMethod(hQry, QRY_EXECUTE$, "select companyname from customers order by companyname")
```

4. **Process the Results**. After the query has successfully executed, the rows are available to BASIC+. The **QRY_GETROW$** code (equated to 5 in XO_Equates) retrieves the next row from Access. QryMethod() is called with this code, and the results variable is built through a loop.

```
row = ""
results = ""
loop
  flag = QryMethod(hQry, QRY_GETROW$, row)
  while flag
    results<-1> = row
  repeat
```

5. **Destroy the Query and Connection Objects, and populate the combo box**. To prevent memory problems, the query and connection objects must be destroyed after use. The **QRY_DESTROY$** code (equated to 1 in XO_Equates) passed to QryMethod() instructs the server to destroy the Query object. The **XO_DESTROY$** code (also equated to 1 in XO_Equates) passed to XOMethod() instructs the server to destroy the Connection object, closing the connection. The Set_Property() call sets the LIST property of the combo box, populating it with the results of the query.

```
QryMethod(hQry, QRY_DESTROY$)
end

rv = Set_Property (@window : '.COMPANIES', 'LIST', results)
XOMethod(hXO, XO_DESTROY$)

end
```