How the Data Set Object Was Used

Note the following differences:

- **The** DS_EQUATES **$Insert Record**. This record contains a list of codes that can be passed to the DSGetProperty(), DSSetProperty(), and DSMethod() to control in detail the communication between the BASIC+ program and the external database.
- **Creating a Data Set instance**. After the connection has been established, the original program created an instance of a query object using QryInstance(). In this example, we will create an instance of a data set object using DSInstance(). Both functions require a valid connection handle. After creating a data set instance, its properties and methods are available, just as the query object's properties and methods were available before. The code is:

```
hDS = DSInstance('CUSTOMERS_NWIND', hXO)
```

- **Executing the SQL Select**. The code below executes the data set object's default script (the SQL SELECT script, which is what we want). There is no need to build the script in code because it is stored in the Data Set object.

```
flag = DSMethod(hDS, DS_EXECUTE$)
```

- **Retrieving the Results**. Instead of a loop, one line of code retrieves all of the results. The DSGetProperty() call, passing the DS_RECORD$ argument, returns all the records in a record mark delimited variable. This is the line of code:

```
rv = DSGetProperty(hDS, DS_RECORD$, results)
```

- **Destroying the instance**. Just as the query instance needed to be destroyed, the data set instance also must be destroyed, to preserve memory. A DSMethod() call exists for this purpose. The code is:

```
DSMethod(hDS, DS_DESTROY$)
```

- **Converting record mark delimiters to field marks**. This is needed because the combo box requires a field mark delimited variable. This is the code:

```
convert @rm to @fm in results
```