SQL Parameters Contracts

Note that some of these contracts are used by the Data form wizard and should not be modified.

Support for parameterized queries is one of the exciting database support enhancements in this release of Drumbeat. The SQL Parameters contracts allow you to harness the power of this new feature.

The contracts essentially provide similar functionality to the Database Filter contracts, but by a different mechanism. These contracts supply parameters to an underlying query or stored procedure and modify the SQL source for the records before the recordset is created. As a result, the recordset is filtered when it is created. Database Filter contracts, on the other hand, create an unfiltered recordset first, then apply the filter in a subsequent step. As a result, you get better performance via the SQL parameters, which can be dramatic for large recordsets. A requirement for using these contracts is a recordset or command object whose content is a parameterized query or stored procedure. Because parameterized queries were not supported in previous versions of Drumbeat, you may find it useful to review the User's Guide for information in how to create parameterized queries in the Drumbeat.

For more detailed information on individual contracts, see the individual contract descriptions, which are found on the 'Description' tab of the Contract Properties Dialog. The Contract Properties dialog can be accessed from the Drumbeat Tools menu.

Using the SQL Parameters Contracts

Most of the contracts are interactions between specified form elements and a recordset or command object. When applied, the value of the form element is passed as parameter to the query. You can set up your query to accept multiple parameters linked with SQL comparison operators to generate complex WHERE clauses. Multiple form elements can then be used in combination, each passing a value to a different parameter in the query.

Several common parameters must be set when the contracts are applied (we are talking contract parameters as opposed to query parameters in this case). The **Enclosing Token** parameter lets you set a delimiter to wrap the value being passed to the query according to the data-type requirements of the underlying database. Typically, text and date fields require the single quote character, while numbers do not require any delimiter. This may vary across different database platforms. The **SQL Parameter** parameter specifies the parameter in the query or stored procedure to which the value is being passed.