

Firebird Embedded C# Code Example

5-Dec-2019

```
//-----
static public bool boFindValueOrValueRecords (string stDbPathFilename,
                                              string stUserID,
                                              string stPassword,
                                              string stTableName,
                                              string stFieldNamesWithKeyIDNameCommaDelimited,
                                              string stFieldNameToSearch,
                                              List<string> liststFindValues,
                                              List<List<string>> listliststFindValueRecordResults)
{
    bool boErrorFlag = false;
    string stConn = new FbConnectionStringBuilder {
        Database = stDbPathFilename,
        UserID = stUserID,
        Password = stPassword,
        ServerType = FbServerType.Embedded,
        ClientLibrary = @"(path)\fbclient.dll"
    }.ToString();
    try {
        using (FbConnection fbConn = new FbConnection(stConn)) {
            fbConn.Open();
            //The "lower" makes the search "case insensitive"
            string stLowerFieldNamesWithKeyIDName = "lower(" + stFieldNameToSearch + ")";
            string stCmd = "SELECT " + stFieldNamesWithKeyIDNameCommaDelimited +
                " FROM " + stTableName +
                " WHERE " + stLowerFieldNamesWithKeyIDName +
                " LIKE lower('%' || @p0 || '%)";
            //This allows for search of more than one value
            for (int iii = 1; iii < liststFindValues.Count; iii++)
                stCmd += " OR " + stLowerFieldNamesWithKeyIDName +
                    " LIKE lower('%' || @p" + iii.ToString() + " || '%)";
            using (FbCommand fbCmd = new FbCommand(stCmd, fbConn)) {
                for (int iii = 0; iii < liststFindValues.Count; iii++) {
                    string stPlaceHolder = "@p" + (iii).ToString();
                    string stValue = liststFindValues[iii];
                    fbCmd.Parameters.AddWithValueValue(stPlaceHolder, stValue);
                }
                FbDataReader fbReader = fbCmd.ExecuteReader();
                while (fbReader.Read()) {
                    ConvertFbReaderToListlistst(fbReader, listliststFindValueRecordResults);
                }
                fbReader.Close();
            }
        }
    }
    catch (Exception ex) {
        boErrorFlag = true;
        MessageBox.Show(ex.Message);
    }
}
```

```

    return boErrorFlag;
} //boFindLikeValueRecords
//-----
//-----
static private void ConvertFbReaderToListlistst(FbDataReader fbReader, List<List<string>> listliststRecords)
{
    List<string> liststOneRecord = new List<string>();
    try {
        int iFieldCount = 0;
        object[] objFieldValues = objRemoveNulls(fbReader, ref iFieldCount);
        if (iFieldCount > 0) {
            liststOneRecord.Add(objFieldValues[0].ToString());
            for (int iii = 1; iii < iFieldCount; iii++) {
                liststOneRecord.Add(objFieldValues[iii].ToString());
            }
            listliststRecords.Add(liststOneRecord);
        }
    }
    catch (Exception ex) {
        MessageBox.Show(ex.Message);
    }
} //ConvertFbReaderToListlistst
//-----
//-----
static private object[] objRemoveNulls(FbDataReader fbReader, ref int iFieldCount)
{
    iFieldCount = fbReader.FieldCount;
    object[] objFieldValues = new object[iFieldCount];
    fbReader.GetValues(objFieldValues);
    for (int iii = 0; iii < iFieldCount; iii++) {
        if (Convert.IsDBNull(objFieldValues[iii]))
            objFieldValues[iii] = " ";
    }
    return objFieldValues;
} //objRemoveNulls
//-----

```