Implementing database technology within optical thin film software
Fred Goldstein
FTG Software Associates
P.O. Box 579 Princeton, NJ 08542

Development Environment

The FilmStar Development Environment currently includes the FilmStar BASIC language, the FilmStar Workbook spreadsheet, the FSPlot graphics module, and the Report Generator. These modules work together to provide RAD (rapid application development) tools for company-specific optical thin film requirements. The development environment automates calculations and designs thin films meeting complex specifications. In addition, it simplifies processes and procedures for coating personnel and QC technicians.

Database Engine

Extending the capabilities of the development environment, we have implemented Microsoft’s Jet Engine database technology within FilmStar DESIGN, MONITOR, CRYSTAL and MEASURE. The Jet Engine library, the foundation of Microsoft Access, interfaces with a variety of industry-standard databases (Access, dBase, FoxPro, Paradox, and Btrieve).

Figure 1. Database spectral1.dbf in DESIGN or MEASURE

A database table is similar to a spreadsheet. Each row is a record and each column is a field. When viewed in a spreadsheet such as Microsoft Excel the first row contains the name of each field. The above table shows records which might be required for spectral measurements. The FilmStar Database supports up to ten user-definable fields plus date and comments. For security, field properties are specified in a separate file and cannot be changed from FilmStar menus. Two database tables (thin film designs and measured spectra) can be displayed and accessed simultaneously.

Figure 2. Same database viewed in Microsoft Excel
FilmStar databases can be read-only. This prevents users from inadvertently altering databases but still allows queries and analyses. Read-only mode is appropriate when using DESIGN to reverse-engineer spectra obtained with FilmStar MEASURE or when opening thin film designs in MONITOR.

Databases are useful because records can be selected and sorted according to various user-specified criteria. Suppose we want to select records corresponding to a coating technician. We could use the Database Find dialog shown at the right. This is equivalent to the following SQL (structured query language) statement

```
SELECT * FROM spectra1 WHERE 
  LCASE(technician) LIKE '*clinton*'
```

( 
  denotes that a statement is on one line.)

More complex sorting and selecting is available through an SQL query editor:

```
SELECT * FROM spectra1 WHERE (fdate = #10/15/1996# AND comment LIKE '*FAILS*') 
  ORDER BY technician
```

SQL provides enormous power and flexibility; only a few SQL commands are actually required. Since statements are assigned to macro keys or included in FilmStar BASIC, technicians do not need to understand SQL. Once the correct record is located, clicking the Open button loads the spectrum or thin film design and activates the FilmStar main menu. The database remains open and the user can go back and forth by clicking. Since the database is continuously updated, users are protected against system failure. Databases may be defined as read-only to prevent inadvertent changes.

**An Example - Comparing Coating Performance**

While updating, selecting and sorting records are useful, the FilmStar Database goes much further through interaction with other FilmStar development modules: BASIC, the Workbook, Report Generator, etc. This provides an environment where users can automatically perform calculations over sets of selected records.

Suppose that a coating facility needs to compare the performance of coating chambers. The spectrum for each coating must be located, analyzed, and tallied. Since each spectrum was stored in FilmStar database spectra1.dbf, it's easy to compute pass-fail criteria over selected data files. Calculations are performed under control of FilmStar BASIC. QC technicians can run BASIC programs without viewing the editor and, for security against unwanted alterations, the editor can be made completely inaccessible. In fact, the end-user need not be aware that there is an editor.

The simulated spectra in this example were actually created in FilmStar DESIGN. This enables engineers to introduce random and systematic coating errors and determine whether the errors can later be deduced from the resulting spectra. The BASIC simulation program is included with FilmStar.
The FilmStar BASIC program is run and after a few minutes the screen shown in Figure 5 appears. A listing is included in the Appendix. Similar programs would compare technicians, determine yields vs. time, etc. Programmers will note how standard BASIC syntax is supplemented with FilmStar-specific functions like `dbSqlExec`. Actual pass-fail calculations are performed in Workbook model QCBBAR.XLS rather than in BASIC. While the calculations could also be performed in BASIC, the Workbook is particularly well suited to spectral analysis.

It is important to note that these operations can all be reduced to simple end-user menus and dialogs. End-users require no knowledge of the Workbook (shown below) which is opened and run automatically by the controlling FilmStar BASIC program.

With hundreds of designs and thousands of measured spectra, coating manufacturers need database management tools. Integrating these tools into FilmStar provides the economy and ease-of-use required in practice.

Optical coating engineers need to implement and simplify computer-related tasks for technicians who may require step-by-step instructions. Using its e-mail network, FTG Software readily helps users develop and debug BASIC and Workbook code examples. (Very often this leads us to add new capabilities and commands to the underlying FilmStar modules.) There is no additional cost for this service. We include a sample FilmStar BASIC program in the appendix, so it can be appreciated that only modest programming skills are required to implement custom database solutions.

FilmStar database tables can be set up and maintained in Microsoft Excel as .dbf (dBase) files. While this is adequate for many users, others might prefer the additional power of Microsoft Access or another Windows database application. This leads to the question, "Why not use Access together with FilmStar MEASURE and transfer results by DDE or OLE? Why use the FilmStar Database?" After all, Access and Excel can actually run FilmStar!

While it's very easy to link FilmStar with other Windows software, the built-in FilmStar database is far simpler for non-experts on a routine basis. (Open up Microsoft Access and imagine training coating operators and technicians in its use!) While some understanding of database concepts is required to set up and maintain a FilmStar database, virtually no knowledge is required for its daily use in coating or QC departments. On the other hand, companies who employ database experts will be pleased to know that FilmStar utilizes industry-standard database formats.

In cases where no one is available with the time or expertise to implement FilmStar software procedures, FTG Software's Custom Applications Division develops and supports specific solutions on a consulting basis. Programs are normally available at fixed-cost and include free follow-up support.
Appendix - BASIC Code Example

'DBTECH2.BAS for FilmStar DESIGN or MEASURE, ©1966 FTG Software Associates
'This sample program compares the performance of coating chambers by determining whether
spectra meet specifications. DBTECH1.BAS shows another way to do this using dbListFiles

DefInt I-N: DefSng A-H, O-Z
Dim report$, test$
Const path$ = "c:\winfilm\dbase\"

Sub Main
    RunTest "AR-450-750" ' product type
End Sub

Sub RunTest(ctype$)
    wbCloseAll   ' close any open Workbook files
    Select Case ctype$
        Case "AR-450-750"
            test$ = "qcbbar.xls"
        Case Else
            ' add other coating tests here
    End Select
    If Not wbOpen(path$ & test$) Then End ' get FilmStar Workbook test file
    Yield "Eddy-2", ctype$
    Yield "Leybold-1", ctype$
    Yield "Leybold-2", ctype$
    Display report$, "Coating Yield for " & ctype$, 5000, 2500, "Arial", 11, 1
    WbExit  ' close the Workbook
    k = macro("Busy 0;"
End Sub

Sub Yield(chamber$, ctype$)
    If Not dbActivate("dbSpectra") Then Exit Sub  ' can't open database? then quit
    If dbSqlExec("SELECT * FROM spectra1 WHERE field2 LIKE '" & ctype$ & "' AND field3 LIKE '" & chamber$ & "'"") Then
        If Instr(Ucase$(dbGetText(0)), "NO MATCHING") = 0 Then
            Do
                ktot = ktot + 1    ' total record count
                k = dbActivate("dbSpectra")
                s$ = dbGetText(13)  ' spectrum file name
                MainActivate
                k = macro("DataOpen " & s$ & ";AxesDraw;DataPlot;"
                wbActivate  ' paste data into Workbook and evaluate pass/fail
                If wbGetText("Passed") = "TRUE" Then ' get pass/fail from Workbook
                    kFlg = 1: kpass = kpass + 1  ' number that pass test
                Else
                    kFlg = 0
                End If
            Loop
            report$ = report$ & "Chamber " & Ucase$(chamber$) & ": " & format$(100 * kpass/ktot,"0.00") & "%&Chr$(13)&Chr$(10)
        Else
            End ' error in SQL statement...so end the proceedings
    End If
End Sub

Function LastRecord()
    s$ = Mid$(dbGetText(0), 8)  ' read record number label
    k = Instr(s$, "/"): i1$ = Mid$(s$, 1, k - 1): i2$ = Mid$(s$, k + 1)
    LastRecord = (i1$ = i2$)  ' 50/50 indicates last record
End Function