

# Migration von OpenVMS Fortran Programmen mit Oracle RDB-Anbindung zu PC/Windows

## Das Problem / die Aufgabe

Auf DEC Alpha unter **OpenVMS** waren **Fortran**-Programme im Einsatz, die **Oracle-RDB Datenbanken** nutzten. Der Zugriff auf eine Datenbank erfolgte mittels **SQL** (SQL = Structured Query Language). Damit wurde in sogenannten **SQLMOD**-Dateien Prozeduren definiert, die schließlich von einem Preprozessor in Routinen übersetzt wurden, die ein Fortran Programm rufen kann, um eben die gewünschten SQL Befehle auszuführen. Zudem wurden in den OpenVMS Fortran-Programmen Betriebssystemfunktionen genutzt, die auf **PC/Windows** nicht aufrufkompatibel zur Verfügung stehen.

Für PC unter Windows sind zwar sehr leistungsfähige Fortran Compilersysteme erhältlich, jedoch aber kein Preprozessor, der die vorhandenen **SQLMOD**-Dateien in entsprechenden Code übersetzt, der dann von einem Fortran-Programm gerufen werden kann.

## Die Lösung

Der in Fortran programmierte Zugriff auf Datenbanken mittels SQL unter Windows ist mithilfe von **ForDBC** möglich, sofern für die Datenbank ein **ODBC** Treiber vorliegt. Dies war und ist der Fall für die vom Kunden für den PC-Einsatz gewählte Oracle Datenbank. ForDBC ist eine Bibliothek, die ODBC Funktionen Fortran Programmierern unter Windows einfach zugänglich macht.

Als Compilersystem wurde **Intel Visual Fortran v10.1** (kurz **IVF**) mit **Microsoft Visual Studio 2008** (kurz **VS**) gewählt, da sich dies als zuverlässig und auch für große Projekte als geeignet erwiesen hatte. Außerdem bietet der IVF leichten Zugriff auf die Windows Betriebssystemfunktionen (**WinAPI**), der notwendig ist, um die im Fortran-Programm verwendeten Laufzeitfunktionen des OpenVMS Betriebssystems zu ersetzen. Die Programmierung und die Tests wurden unter Windows XP Professional (SE) durchgeführt.

Für die Lösung wurde zunächst eine Studie erarbeitet, die zeigte, daß der Lösungsweg zum Erfolg führt, da die Umsetzung einiger tausend (!) **SQLMOD** Prozeduren notwendig war. Die wesentlichen Schritte zur Bewältigung der Aufgabe waren:

### Projektumgebung erstellen und fehlende Routinen ermitteln

Nachdem die Fortran-Quellen von OpenVMS auf PC kopiert waren, wurde eine Projektumgebung in VS aufgebaut (vgl. Abb. 1). Diese konnten nach einigen kleineren Korrekturen und Anpassungen ohne Probleme übersetzt werden (mit IVF). Beim Binden (link) wurden die erwarteten Linker-Fehler gemeldet (z.B.: error LNK2019: unresolved external symbol `_LIB$WAIT` referenced in function `_MAIN__`), da sowohl die OpenVMS Betriebssystemfunktionen fehlten, als auch die **SQLMOD**-Prozeduren.

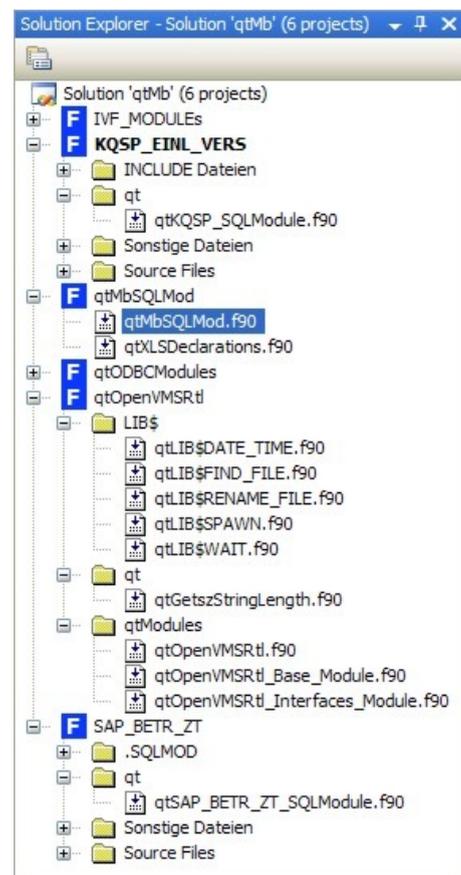


Abb. 1: IVF Projektstruktur der Migrations-Studie (in VS2008)

## ❑ OpenVMS Betriebssystemfunktionen ersetzen

Die fehlenden OpenVMS Betriebssystemfunktionen wurden durch gleichlautende Routinen ersetzt, die ihrerseits entsprechende Funktionen der WinAPI riefen. Zum Beispiel kann die Funktionalität der OpenVMS Funktion LIB\$FIND\_FILE z.T. durch die WinAPI Funktionen FindFirstFile und FindNextFile nachgebildet werden.

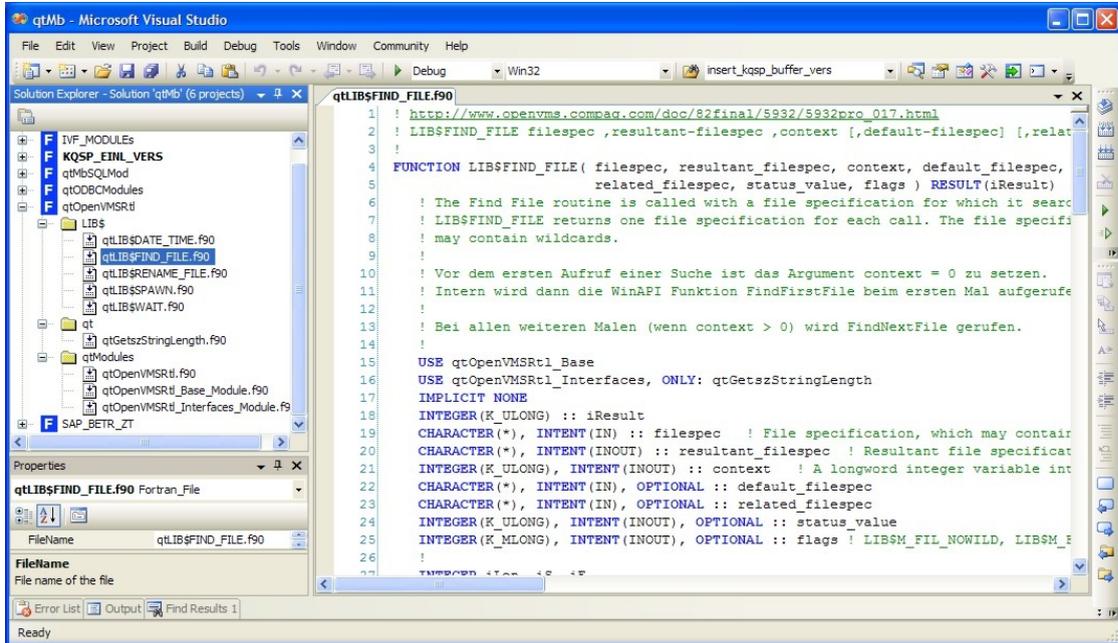


Abb. 2: Der Ersatz der OpenVMS Funktion LIB\$FIND\_FILE unter Windows.

## ❑ SQLMOD Prozedur durch ODBC Funktionen ersetzen

SQLMOD Prozeduren sind SQL Skripten sehr ähnlich. Sie deklarieren zunächst Variablen (Name und Typ), die den Spaltennamen der Tabellen entsprechen. Dann folgt ein SQL Befehl. Beispiel:

```
procedure update_BetrStatus
  sqlcode,
  BetrStatus      char(5),
  BetrZeit        char(16);
  UPDATE ProdStatus
    SET uTime=BetrZeit,uState=BetrStatus
    WHERE proc_N='Charge_K4';
```

Von einigen Besonderheiten abgesehen sind die Deklarationen meist leicht in Fortran abbildbar. Für den SQL Befehl gilt das gleiche. Beispiel:

```
szSQLStmnt = "UPDATE ProdStatus SET uTime=?, uState=" &
  // "WHERE proc_N='Charge_K4'" // CHAR(0)
```

Die eigentliche Aufgabe bestand dann darin, die für die Ausführung des SQL Befehls in ODBC notwendigen Vorbereitungen zu programmieren (Database Connect, Column-/Parameterbinding, Fehlerbehandlung u.a.). Dies war mithilfe von ForDBC machbar (vgl. Abb.3).

## ❑ Test

Nachdem die vom Linker als fehlend gemeldeten Routinen (OpenVMS Betriebssystemfunktionen, SQLMOD Prozeduren) ersetzt waren, konnten Tests durchgeführt werden, die die gleichartige Funktionalität der Programme unter Windows belegten.

```

qtSAP_Betrie...it_SQL.SQLMOD
219 procedure update_BetrStatus
220     sqlcode,
221     BetrStatus      char(5),
222     BetrZeit        char(16);
223
224     UPDATE ProdStatus
225     SET uTime=BetrZeit,uState=BetrStatus
226     WHERE proc_N='Charge_K4';
227

qtSAP_Betrie...SQLModule.f90
52     SUBROUTINE update_BetrStatus( sqlcode, BetrStatus, BetrZeit )
53     ...
61     INTEGER, INTENT(INOUT) :: sqlcode
62     CHARACTER(5), INTENT(IN) :: BetrStatus
63     CHARACTER(16), INTENT(IN) :: BetrZeit
64     !
65     CHARACTER(128) szSQLStmt
66     INTEGER (SQLHANDLE) :: hBef = SQL_NULL_HANDLE
67     INTEGER (SQLRETURN) iRet
68     INTEGER cbLengths(2)
69
70     iRet = SQLAllocStmt( ghConnect, hBef )
71     szSQLStmt = "UPDATE ProdStatus SET uTime=?, uState=? WHERE proc_N='Charge_K4'" //CHAR(0)
72     iRet = SQLPrepare( hBef, szSQLStmt, SQL_NTSL )
73
74     iRet = qtMbSQLBindParameterChar( hBef, 1, SQL_PARAM_INPUT, BetrStatus, cbLengths(1) )
75     iRet = qtMbSQLBindParameterChar( hBef, 2, SQL_PARAM_INPUT, BetrZeit, cbLengths(2) )
76
77     iRet = SQLExecute( hBef )
78     CALL CheckSQLReturnCode( hBef, iRet )
79     sqlcode = iRet
80
81     iRet = SQLFreeHandle ( SQL_HANDLE_STMT, hBef )
82     END SUBROUTINE update_BetrStatus
83

```

Abb. 3: Die Umsetzung der SQLMOD Prozedur update\_BetrStatus in Fortran mittels ForDBC.

### □ Zugriff und Programmierung standardisieren / Generatorentwurf

Die Umsetzung der SQLMOD Prozeduren mittels ForDBC hatte sich als machbar erwiesen. Dabei zeigte sich, daß sich Programmierungen oftmals wiederholten und sehr schematisch erfolgten. Diese Programmierschritte wurden zu eigenen Routinen zusammengefaßt, die halfen den Programmieraufwand zu reduzieren. Zudem wurde ein Generator konzipiert, der die Umsetzung der SQLMOD Dateien in die Fortran Programmierung auf Basis von ForDBC bzw. ODBC automatisieren sollte.

*Kunde:* k.A. (wg. Geheimhaltungsvereinbarung)  
*Projektzeitraum:* 2010  
*Verwendete Software:* Intel Visual Fortran v10.1, Microsoft Visual Studio 2008, ForDBC  
*Stichworte:* OpenVMS, RDB, Datenbank, Fortran, SQLMOD, SQL, ODBC, WinAPI, Windows

**QT software GmbH**  
Konstanzer Str. 10  
D-10707 Berlin  
Germany  
Telefon 030/9290087-0  
Telefax 030/9290087-2  
eMail [info@qtsoftware.de](mailto:info@qtsoftware.de)  
<http://www.qtsoftware.de>  
Shop <http://www.qtsoftware.com>