Open QM

Ladybridge Systems

Case Study—Mellins i·Style

www.mellins.co.za

Mellins i-Style is a South African optometric company with 52 practices throughout the country, founded over 40 years ago. With the complexity of optometric stock, lenses in particular, an off-the-shelf software solution could not be found that met their business needs.

Around 1985, development was started on a system that today handles stock, debtors, creditors, purchase orders, lab orders, EDI medical claims, SMS communication to patients, appointments and patient recalls.

Previously running on D3, the application was ported to QM on Linux in 2011 and now runs 230 concurrent users on an Intel I7 system with 4Gb memory which, in the words of Nico Roets of Pienaar Consulting, “purrs like a cat”.

The main reason cited for adopting QM was the low licence price. The application also uses the device licensing feature to allow multiple connections from the same client PC to share a licence, further reducing costs. In common with most QM users, this site has opted for the ten year extended upgrade entitlement. Although Mellins do not currently use AccuTerm, the provision of this bundled with the QM licence is seen as added value.

Other options were considered, including a total rewrite of the software on MySQL which would lead to lower database costs but significantly longer and hence more expensive development.

Nico Roets who led the migration process, cited as important factors in the decision the greater functionality in QMBasic, object oriented programming, sockets, and much more. The built-in support for CSV data in the query processor and elsewhere was a big plus. And, of course, why throw away many years of experience with MV based systems?

Mellins biggest need was to have a central DBMS server for its 52 (and still growing) practices with users connecting via ssh. Use of a central server instead of distributed systems had the advantage of consistency of the user application throughout the user base. The very small footprint and low overheads of QM allow the application to support a large user community on a relatively low end server.

The security features of QM were significantly easier to use than those in the previous implementation of the application. Historically, the

application has used fixed user number allocations to control security and the ability to maintain this technique in QM helped minimise application changes.

The migration to QM made it possible to keep the user interface exactly the same as it had always been although there are plans to change this in future. The end users have seen no change in their use of the application aside from no longer needing to perform system maintenance tasks such as backup of separate local servers. Having to retrain all of the users for a different application interface on top of a migration would have been an enormous task.

Of course, any migration tends to have drawbacks too. The lack of ODBC connectivity for SQL queries was identified as an issue (though there are third party solutions for this).

Nico Roets said that product stability and good support services are very important. The continuing development of QM, willingness to implement enhancements, rapid resolution of problems, and high quality up to date documentation all show commitment on the part of Ladybridge Systems to the QM product.

The first step in performing the migration was to install a four user developer licence, supplied free of charge. Actual installation took only a few minutes. The developers then spent a little while familiarising themselves with the file structure, user rights, OPTION settings etc. Transfer of the application program source code and the data files from D3 into QM required no more than restore of an account-save.

With over 900 programs and subroutines to migrate, about 30% of these required some changes. Differences in some conversion codes, particularly Pick user exits, required use of different functions in QM (though it is usually possible to emulate most user exits with a user written conversion code). Case sensitivity (largely controllable by QM options) caused a few minor problems. Alternate key index processing and the Information style printing subsystem of QM are different from D3. The ability to select compatibility options both at command language level and in QMBasic resolved some issues of command syntax differences.

The more strict error checking in the QMBasic compiler, particularly the detection of unassigned or unused variables, was viewed as a positive. Also, the availability of I-type dictionary items as an alternative to correlatives was described by Nico as “my prayers have been answered”.

As a final quote from Nico, “Overall this migration has been one of the most successful tasks I've ever accomplished. QM has renewed my love and interest for the MVDBMS. Thanks for great support, documentation and an excellent product!”