IBExpert GmbH White Paper

Firebird White Paper

Firebird Software Interfaces

Holger Klemt, December 2024

How do you create a function extension for Firebird-based software if the software manufacturer is unwilling or unable to help you?

We have an increasing number of support and coaching appointments with customers who use standard software that works with the Firebird database server as a backend, and who want to customise or extend the range of functions.

The motivation is often the mass import or export of data for other systems or even very simple dialogues in the standard software, where additional data can be entered that is not possible in the software itself. However, the masks implemented by the software manufacturer are often too complicated and simple data entry is more complex than necessary.

When it comes to connecting mobile barcode scanners, you often have to rely on isolated solutions that cannot really be integrated into the planned programme sequence, because the software manufacturer either offers considerable budgets for customer-specific projects which are not cost-effective, or due to a shortage of specialists, can only just manage to keep the existing software up to date, and cannot offer extensions or interfaces to other systems due to a lack of expertise.

Even if the database can be opened with suitable tools such as IBExpert and a look at the tables enables basic analyses, write operations are often much more complex and can hardly be mapped in the correct order by external tools without help from the software manufacturer.

Data models of software packages whose foundations were laid 25 or 30 years ago with the first Delphi versions can no longer be described as such by today's standards.

Detecting which tables with which cryptic name in the ERP software correctly map invoice headers, or items or delivery notes with warehouse bookings can quickly lead to failure. The help of the software manufacturer may be well-intentioned, but is usually limited to better basic knowledge in support, because only a few long-standing in-house developers have a complete overview of the most important processes and tables involved. Often, they are already retired or about to retire and there is not enough time to explain all the details to external parties. It is even worse, however, if the software manufacturer's entire programming team only actually makes minimal adjustments or ensures the function because familiarisation with such processes takes a very long time.

The software used, for example, to create an order in ERP software, conclude a new customer contract in insurance broker software or create a new payroll in management software for temporary employment agencies is more than sufficiently familiar with the in-house software interface, but not which table relationships need to be entered correctly into the database in which order and with which data in order to reproduce the processes, so that the data from other sources can also be mapped correctly.

Even seemingly simple processes such as assigning a contact person and their e-mail address to the customer base for individual e-mails are often far more complicated than you might think.

In our workshops we usually start with the IBExpert full version, connect to the database in use and use *Services/Database Monitoring* to determine the attachment IDs from an instance of the client software, which then starts the process to be emulated manually.

Depending on the software, more than one connection may be relevant, e.g. because separate ReadWrite and ReadOnly attachments are active or because the software was simply programmed to be more complicated than you think. Distributed DLLs or other reasons are often the cause. However, as soon as we have remote process IDs that can be read in the client's Task Manager for the IP address, for example, we can note these attachment IDs and add new sessions to the *Services Trace and Audit*, which only ever enters one of these attachment IDs as the Trace Connection ID and immediately see all statements sent by this client in the Trace log, including SQL statements and parameters used. It is important that the maximum length of the SQL statements in the log is increased from 300 to e.g. 8000 and that the maximum number of parameter values in the log is also increased from 30 to e.g. 250. The config can then be saved permanently as the default with *Save Config as Default*. (However, I would leave the Trace Connection ID at 0 so that all connections are logged).

In the next step, we now ask a clerk to execute the process in the application software as usual and, when everything is complete, to give us feedback so that we can now pause the trace session.

All SQL commands with parameters used can now be read in detail in the trace log and, for example, transferred to an IBEBlock script, which imports data from other sources into the system. Although this is often not child's play, it is usually just hard work.

The data from an external source, which has been connected as XML, JSON, CSV or ODBC or whatever, is saved in variables in the script and then entered into the Firebird database with suitable default values. As soon as this has worked with the first process, the application software needs to be checked to ensure that the data is completely visible.

Even complex article price tables can be imported not just only a single time with such a procedure, they can also be directly linked live to web-based data sources, even if the software manufacturer does not want to know anything about it.

As part of our support hotline packages, we are also happy to offer you the tools you need to take the first steps with our help, so that you can then carry out more complex customisations yourself or with further support hotline assistance at a later date.

Even simple processes, which are required for the automatic printing of QR code labels on a thermal transfer printer when certain items appear on delivery notes, can be implemented very reliably using such processes, even if your software manufacturer is not able to print suitable QR codes at all, let alone print them in the right quantity on the right printer where they are needed.

A customised project with your software manufacturer often fails due to the 5-digit € budget and in contrast, with in-house implementation, you are not dependent on the software manufacturer's long implementation times, who then regularly comes up with fresh excuses as to why it still doesn't work. As soon as the basic know-how for such scripts is established in-house, further automation projects can be implemented cost-effectively.

A practical example: for a client we have implemented an automated import of requests for quotations from a very large industrial group, which are submitted as PDFs with a large number of technical documents in ZIP files.

The automatic import provides a quotation header and correctly referenced quotation items in the ERP, so that quotation processing does not start with the typing of well over 100 quotation items spread over 200 pages of PDF, which are extremely error-prone when entered manually.

Even if no item ends up in the current order, dozens of hours are not wasted on pure data entry, and the technically qualified clerk can begin with the ERP quotation compilation and calculation of the required prices a few minutes after receiving the enquiry, and compare similar items in previous requests for quotations, so that they are not offered one way or the other.

Using the Trace API, it was possible to quickly analyse and reproduce the requirements for the automatic import of quotations into the ERP using a manually compiled quotation.

We have already presented the basic techniques for this in the IBExpert and Firebird Online Get-Together videos on YouTube: https://www.youtube.com/playlist?list=PL6V0Rcmrx7muaxeLyJpgZXqEFMJ1S0v01.