

Oracle Forms is stronger than time

Frank Hoffmann, Cologne Data GmbH



This year, Oracle celebrates an impressive double anniversary: 40 years of ORACLE databases and Forms Bill Friend, Oracle's fifth member of staff after Bruce Scott (#1004), had been tasked by Larry Ellison in 1979 with developing the famous software tool. Before Christmas of the same year, Oracle released the first commercial version of Forms together with the first commercial Oracle Database and the first Reports tool two years later.

Mercedes advertised its 1979 G-Class with the claim „Stronger than time”—a slogan that could also be applied to Oracle Forms. The ensuing software evolution has already lasted 40 years. The module code from ASCII to XML. Support for Java versions from 1 to 11, database support for the versions 2.3 to 19 and Windows support from 3.1 to 2016, together with Linux, Solaris and AIX support. The tool has kept pace with these changes every stage along the way, but without users having to re-write their existing code—a huge benefit. Standard 4GL functions such as row locking, binding, and use of PL/SQL in the current database also represent huge advantages over any other PL/SQL tool.

It's a fitting time to look back to the beginnings of Oracle Forms, in order to understand the reasons why it was created. Tips from Michael Ferrante and social media led me to Bill Friend, the developer of the original version, and I had the privilege of interviewing him in detail on the subject. At 60, he is still very lively and can recall numerous details from the early days of Forms. With a 9-hour difference between our time zones, we agreed to talk in the late afternoon (CET).

The story that follows and the interview are condensed from information provided by Bill Friend and an interview conducted at the end of December 2018. It was Bill's first ever interview on the subject of Oracle Forms in 40 years.

Forms sees the light of day

Forms first sees the light of day in 1979. Several interesting and exciting developments appeared in 1979. Quincy Jones gave Michael Jackson his big break with „Off the Wall”. Mercedes developed the first of its G-Class series. In Japan, Namco launched „Galaxian”, its first color arcade-based game. Galaxian was followed by the famous „Galaga” and „Pac-Man” games, long before home computers emerged as a dominant force. It was also a year of major technical developments however, such as TCP/IP, the C programming language, and the sales launches of the Intel 8088 and

Motorola 68000 processors. These developments all laid the foundations for future developments by Microsoft, Apple and Oracle, and the spread of the Internet.

Oracle's first C tools appear

Around this time, a young developer was looking for a job. His name was Bill Friend, and he did what other job-seekers usually did in those days: he grabbed the Yellow Pages, flicked through to „Data Processing”, and looked for suitable companies. Under „S”, his eye fell on „Software Development Labs”. A company with three founding members and a talented developer named Bruce Scott in their employ. Bill Friend's first phone call was answered by Bob Miner. When asked by Bill about a job, Bob replied: „We're creating the first commercial relational database management system, we're doing it on DEC PDP minicomputers in assembly language, but we want to rewrite it in C for portability. We need someone who can write some tools for the thing”.

Bill drove to Sand Hill Circle in Menlo Park for his interview, and was invited to lunch with Bob Miner and Larry Ellison. Once Bob had demonstrated a couple of database queries using UFI (User Friendly Interface—a forerunner of SQL Plus) and Bill had described his programming experience to Bob and Larry, the two sides quickly came to an agreement. The fact that Bill didn't have a university degree wasn't seen as a problem, nor that he had never developed anything in C at that point—that could, after all, be learned. For Larry and Bob, Bill was exactly the right person: „smart” enough for the job. Their instinct was to prove correct. Bill Friend thus became the number 5 at Oracle, the second employee recruited by the company, and the beginning of the company's expansion from 5 employees to 500 in the years that followed.

Bill immediately bought „The C Programming Language”, a book by Brian Kernighan and Dennis Ritchie (K&R book), which had appeared in 1978. He had great fun teaching himself C, and had found the very job he'd always wanted.

A simple task given to Bill by Larry Ellison became ORACLE Forms

Besides familiarizing himself with C, Bill Friend also got to grips with the new relational database system, working closely with Bruce Scott. Within a couple of months he was familiar not only with C, but also with the database technology and the DB APIs. He then had to figure out for himself exactly what his tasks were going to be. In this period, no one at Oracle was supervised, much less told exactly what to do. Everyone in the company did what they did best, and with a passion. Bill however had no specific project of his own, and so he went to Larry and asked for one. Larry responded by saying that the company needed a more user-friendly way of writing data to the database than by means of an „INSERT” command. Bill, he suggested, could perhaps program a „prompt” to be displayed for each database column. And that was how it started. Starting from these very vague instructions, Bill produced a detailed concept that went much further than simply satisfying Larry's request.

Bill's basic concept for Oracle Forms 1979 Bill's intention was to make life easier for the forms developer and forms user, and to achieve a high level of productivity in the development of forms, in line with the 4GL theory:

Source: Wikipedia
„A 4th-generation programming language (4GL) or (procedural language) is any computer programming language that belongs to a class of languages envisioned as advancement upon third-generation programming languages (3GL).”

4GL programming languages are designed to enable functions or complete applications to be written quickly, with the fewest possible lines of code, for a certain area of application.

These were the targets that Bill Friend had set himself:

- The design process of an input form should be interactive and easily modified.
- Users should be guided dynamically through the input form at runtime.
- The runtime engine should perform the transaction to the database dynamically. The developer of the form should be able to perform standard Insert, Update, Locking, Delete and Query transactions without having to program anything. In the event of an error, roll-back should be possible. In addition, all SQL commands should be executed in the correct sequence.
- End users should be able to perform even complex queries themselves by using relational operators such as „<”, „>”, „LIKE”, and in conjunction with other variables.
- The implementation should preserve the logic of the relational model, with transactional integrity.
- User-friendly data input of relationships to other tables should also be possible by means of „lists of values”.
- Pre- and post-logic triggers should be developed for automated SQL transactions, to enable the SQL logic to be extended with procedural functionality.
- The best aspects of two worlds were to be combined: Dynamically generated SQL with transaction management, and facility for use of procedural extensions with runtime triggers for application logic. (SYSTEM-R triggers were an ORACLE vision and were not yet implemented in the database; they appeared first in Forms).
- The tool should be portable to all operating systems.
- A „CRT” form should make the system compatible with a range of monitors (e.g. 24x80 character grid on a cathode ray tube monitor).
- Input was also to be possible on teletypes, since not every workstation was equipped with a monitor at this time.

Using the full range of possibilities offered by C, Bill Friend developed the first version of Oracle Forms within a little over 4 weeks. Larry Ellison named the product IAF (Interactive Application Facility). The name was subsequently changed, first to „FastForms”, then to „SQL Forms”, until it finally became „Oracle Forms and Reports” (from Version 3 onwards).

The birth of Oracle Reports

Larry had promised a customer a reporting tool, and Bill was asked whether he could perhaps quickly produce one. His holiday could be upgraded (to Paris or Hawaii) if he were willing to postpone it in order to complete the task. Bill took an older documentation solution (FMT) produced by Bruce Scott and added functions to it similar to those in Forms. The facility to declare additional variables, select statements with variables, loops, and IF statements were also added. Around two weeks later, RPT appeared and was delivered together with the database. Bill was then able to enjoy a holiday on Hawaii with his girlfriend.

New 4GL tool produces its first success stories

IAF and RPT were a hit with Larry Ellison, and were a factor in expansion of the customer base. Everyone, from the CIA to the Bank of America, major oil and gas companies, down to small IT consulting businesses now developed „forms and reports”. At that time, customers were still tied completely to the database manufacturer and were unable to develop software applications of their own. Data input with UFI (later SQL-PLUS) or the new C-API was not user-friendly.

Now however, customers could both model a database of their own and create it physically with UFI, and develop user-friendly input forms and reports using IAF/RPT. Further tools developed by Bill were added in the form of EXP/IMP, which enabled databases to be exported and migrated. Bill also integrated the „DATE, TIME and TIMESTAMP” data types into the database, since time data is often very important during data acquisition. The result was a cohesive overall package. In 1985 Sohaib Abbasi became responsible for Oracle Forms. He made Forms the tool we know today. He brought PL/SQL into it long before the DATABASE and made it even more productive. Many Forms developers are still working in projects that started in 1999 with a version of his that is still in production. Forms for the Internet was designed in his time too. The idea to have a stateless client as Java Applet and a stateful server process with a solid SQLNET connection is still a powerful engine today and probably the only pure PL/SQL tool which does not need extensions. It is nearly impossible to replace Forms. It was tried many times without real success. For the last 15 years there were many attempts to do it. In the interviews some good answers are given to find answers to this question.

Bill and Sohaib left as VP and SVP Tool division and were probably one main reason for the success of ORACLE in the first 20 years.

Forms now, in 2019

Many Forms customers have switched to the current platform (12.2.1.3), are launching „native” modernizations, and are looking ahead to the new versions, Forms 19 and Forms 20, which have already been announced. Version 19 will primarily be a maintenance release. Further changes are to be introduced with Version 20. As is now standard with ORACLE databases, the version number is now based upon the year of the product's release.

At present, it seems very likely that Version 19 will once again be shipped together with Reports. This will provide all customers with a little more time to evaluate alternatives.

Mission times of 20 to 30 years are by no means uncommon for Oracle Forms projects. Like Oracle itself with its Enterprise Business Suite (EBS), parts of which are still based upon Oracle Forms, customers are currently planning support and further development of their applications up until 2030. Mission

times of this length are somewhat unusual for other development tools.

According to Michael Ferrante, Principal Product Manager responsible for Oracle Forms, the following features are likely for the pending versions (19/20):

- REST call-up functions for external services
- Support for SSO with FSAL
- Identity Cloud Service support
- OAuth support
- UI improvements (frames, colors, custom color scheme)
- Configurable Java versions for FSAL
- Support for Java 11 FSAL (e.g. Java 11+)
- Forms Builder integration with FSAL (currently only http plugin)

Together with the changes already published in Forms 12, Forms developers will then have a range of features at their disposal in Forms 19 with which to give their forms a facelift. Forms can and should become more professional in their layout, without gray shades, with high contrast, and with color palettes from which the users benefit.

A further new option will be the outplacement of Forms development and deployment to the ORACLE Cloud. In a very new development described in a white paper in November 2018, Mike Ferrante presents the new opportunities for Forms development (DEVOPS) in the cloud. (See annex for link.) The author will present a paper on the subject of Oracle Cloud at the next Forms Day.

Let's start giving some thought to the 50th Forms anniversary in 2029.

Additional Information

Interesting version table with release dates of the Oracle databases
https://en.wikipedia.org/wiki/Oracle_Database

White paper on DEVOPS for Oracle Forms in the ORACLE Cloud (November 2018)
<https://www.oracle.com/technetwork/developer-tools/forms/documentation/oracle-forms-in-dcs-5216022.pdf>



Frank Hoffmann
fh@cologne-data.de



Interview with Bill Friend

December 31, 2018

responsible for Forms from 1979 until 1985

FH> What are your favorite all-time features of ORACLE Forms?

I would say automation of the transaction—the dynamic SQL and correct sequencing of triggers based on end-user manipulation of the form data. User guidance controlled by transactions and triggers.

Productivity though (which encompasses many features) is the heart of it. That was something our competition never seemed to match. The approach was basically 4GL, and where the 4GL was no longer sufficient, you could augment the logic with procedural tools.

FH> Oracle Forms turns 40 this year, and its popularity is growing. Did you originally foresee this success?

BF> I'm honored that Oracle Forms, in another incarnation but true to its roots, is still in use and may be for some time to come. I never expected it to survive this long and frankly was hoping something better would have come along by now (I had also wanted to rearchitect it to get rid of some of the limitations). There are not many examples of systems or products with that kind of lifespan in an industry that has changed so much. I think I started to feel like I had built something important and lasting in 1984, when I saw jobs being advertised for Forms and Reports developers.

FH> Someone said once that Forms/Reports and the database were the only pieces of software that Oracle had ever made itself; the rest was sourced externally. Is that true?

BF> At the time we were starting Oracle, Computer Associates was how Oracle is today—buying mostly everything to grow. Today, Oracle looks much more like CA in some respects. Oracle Designer, Oracle Applications (the original ones), many of the data warehouse tools including the one I built in 1998-2000 were done in house.

FH> How influential was Larry Ellison in the development process of Forms over the years?

BF> Larry is the smartest guy I ever met and also a true visionary—not just in technology, but really visionary on how to grow a company. He also has a very self-deprecating sense of humor that I found endearing and made him super-approachable. I really like the guy, but it could be frustrating at times working for him. Likewise, Bob Miner had a huge influence on me and was a great guy. The one thing everyone in the early days at ORACLE had in common was that they were strong-willed,

very smart, very direct, very cutting when it came to critical issues relating to the software. Everyone was absolutely committed to their own area and in continual competition with others to get the best possible result. Not just on work things—no, tennis, fingertip pull-ups in the doorways, poker, you name it. And no one was more competitive than Larry. He wanted to be the best, and for us to achieve the best results in our work. This had an influence on the products that we developed.

FH> What is the impact of a shift from Forms, a 4GL language, to a 3GL language?

BF> About the time I was leaving, there was a real movement towards programming forms in Java, moving away from 4GL. I'm not sure that was a good thing. You lose a lot if your procedural (3GL) tool doesn't retain the power of the „4GL“ aspects: specifically, managing the transaction and all the associated SQL invisibly to the developer. All this has to be done by the developers themselves in 3GL languages.

Non-procedural 4GL languages are incredibly good at expressing certain complicated things in a simple way, which is why developers love 4GL tools. But these tools lack that ability to be tuned precisely to some aspect that the 4GL was not designed to do or is outside its scope. And that's where people start hating them.

Procedural languages give you the ultimate flexibility—which developers love—but then all the „free“ stuff you got with the 4GL disappears.

Forms, which has had automated transaction management and the facility to integrate procedural triggers since its early days, gives you that ability to mix the two paradigms.

FH> Do you remember how big your development unit was for the initial versions of Forms?

BF> One developer. And at the very end there was also Sohaib Abbasi, who took over after I left. He went on to become the SVP of the Tools Division. In hindsight I should have thought about hiring people to help me, but that thought never even crossed my mind as I just enjoyed doing everything myself.

FH> In your opinion, why do we now see so little promotion of Forms by Oracle?

BF> They are busy promoting cloud technology and they have not provided a migration path from Forms to other cloud

technology or updated it sufficiently. Perhaps this is a missed opportunity. People want to push new products that they think could replace something as old as Forms. As I said in my bio, Oracle is a very competitive place. But until and unless the new products capture the essence of what makes Forms powerful—high performance, high productivity—both for non-programmers and programmers, then Forms will likely continue, and thrive.

I sometimes regret now never getting to work on a replacement myself because I have some ideas from my experience in applications and data warehousing that would have built on the core features of Forms and dealt with a lot of the limitations that it runs up against.

FH> Our next Forms Day is on February 21. Would you be willing to attend from America by SKYPE and get to know the German DOAG Forms fan club?

BF> Yes!



Interview with Sohaib Abbasi

January 21, 2019

responsible for Forms from 1985 until 2003

FH> Can you tell us how you came to Oracle and why you took over the tools of Bill Friend?

In 1982, I joined RSI (Relational Software Inc.) to start the sales operations in 13 mid-western states of the U.S. In 1982, RSI did not have any customers, or even in an office, in the midwest. After two years of meeting hundreds of customers to market and sell Oracle, I transferred to product development in 1984. I chose to work on Tools because the # 1 customer requirement was to help improve developer productivity. Tools improved developer productivity.

FH> How popular was Forms when you got into responsibility and what did you think about its 4GL technology at this time?

Almost all Oracle customers used IAF that was renamed SQL*Forms and later renamed Oracle Forms. It was the primary “front-end” for building data entry, or transaction processing, applications for Oracle. As an early 4GL, Forms pioneered several concepts including declarative application development, metadata-based specification, automated metadata-driven code generation and visual development environment.

FH> How important was backwards compatibility all these years for Oracle Forms?

Forms supported compatibility with multiple versions of the Oracle database, operating systems, and GUIs as well as backward compatibility with older Forms. In fact, early Oracle customers who built applications on mainframes with character-mode interfaces migrated these applications to client-server systems with graphical user interfaces (GUI) and later to the Internet with web browser interfaces – all without changes to their code. In other words, using the Forms, the customers automatically adapted to new computing systems.

FH> How many years have you been responsible for Oracle Forms?

From 1985 to 2003. I started as a programmer on Forms in 1984 and managed the team when I retired from Oracle in 2003.

FH> What do you think are the key benefits of Forms, why did Forms become a strategic tool for Oracle and other tools did not?

The main benefit of Forms was developer

productivity: building transaction processing applications faster than using 3GLs or other tools. Another key benefit was database integration: Forms automatically took advantage of the latest Oracle database innovations. Perhaps, Forms' most important unique benefit was portability and adaptability –portable applications that ran across multiple operating systems and GUIs – adaptable applications that even ran on multiple generations of computing – mainframes, client-server and Internet. Portability and adaptability without requiring customers to change their code.

FH> How important was Forms for Oracle?

Till 1994, on mainframe and minicomputers, almost every Oracle customer used Forms. After 1996, on client-server systems, Forms became the preferred tool for Oracle, competing against third party tools on Microsoft Windows. And, in 2001, Forms enabled applications to be access on the Internet, simply by using a Java-powered web browser. Tools product line, including Forms, was important for Oracle but the reason for its importance changed. Till 1994, Tools were important as a significant revenue generating product line. After 1994, another reason for the importance of Oracle Tools was that Oracle e-Business application suite were built and delivering using these.

FH> How important was Oracle Forms for the growing business of Oracle?

Oracle Tools, including Forms, was one of Oracle's top advantages over the traditional database vendors. IBM, Ingres, Informix and Sybase did not offer similar tools. In the late 1980's, Tools generated more than 35% of Oracle's product revenue.

FH> What were the major enhancements you added to Oracle Forms in your responsibility time for Forms?

Developer productivity: Metadata-driven specification stored in the database. Visual development using a “screen painter”.

Scalability and performance: Client-side PL/SQL that supported application partitioning – moving business logic, coded in PL/SQL, between client and server to improve performance.

Portability and Adaptability: allowed mainframe applications to seamlessly migrate to client-server platforms and later to Internet – all without forcing customers to

change their code.

FH> If my records are correct PL/SQL was added to Forms in 1988 with Version 3 – was this difficult?

Forms was the first tool to include PL/SQL. The database included PL/SQL after Forms. Adding PL/SQL was difficult because Forms had to help debug PL/SQL V1. Adding PL/SQL to Forms was also difficult as Forms supported declarative application development and PL/SQL supported procedural development – two different approaches. In addition, both Forms and PL/SQL separately integrated with the Oracle database and it was difficult to share this context. The benefit was that PL/SQL became an extension to write any business logic in Forms, making Forms even more capable.

FH> How big was your development team 1985,1990,1995,2000..?

Tools development, after Bill Friend left, was one person in 1985: it was me. By the time I retired in 2003, Oracle Tools had more than a thousand developers.

FH> What do you think how many customers have developed with Forms in its peak times?

Majority of the Oracle database customers and every Oracle eBusiness Application customer were using Oracle Tools, including Forms. Ten of thousands of enterprises used Forms.

FH> Did you think Oracle Forms would reach the age of 40 years today with the perspective of another 10 years support?

We certainly did not plan for that long. We always focused on the next version and, more importantly, the next computing platform like the Internet.

FH> Why do you think is it so difficult to replace Forms?

Forms offered a unique declarative development approach well integrated with the Oracle database. As the Forms applications continue to function, there was less reasons for customers to consider alternatives.