Hardware

Hospitals and software companies had been experimenting with mobile patient documentation for quite some time but the best option had only been keyboard-equipped laptops, clumsily rolled from room to room on mobile cabinets, hardly a workable solution. The time was not yet ripe for mobile options...until now. Today, with modern touchscreen tablets, devices are smaller, lighter, and much more comfortable and easier to handle, the solution seemed to be at hand.

Software

The software side of technology has developed just as remarkably. Heimbas opted to use BASIS' revolutionary BUI technology for the new solution. "*All alternatives, such as native iPhone or Android apps, were out of the question,*" explains Armin Kehler. "*The reason is that not only did we know BBj inside out after having worked with BASIS tools for more than 20 years, but our customers are used to running our software on their hardware and OS, and we wanted to keep it that way. With BBj and BUI, we remain platform-independent and operating system-agnostic, without the need to re-develop the same solution several times over.*" With their BUI solution ported from the existing software package, it took Heimbas only 30 project days to complete the transformation, "...*just as we had planned!*" says Kehler.

One of Heimbas' main competitors uses .NET. "*We certainly have an edge over them. Using BBj,*" Kehler explains, "*we are operating system independent, and we don't need to worry about any release changes around the products we use.*" And then there is easier deployment. *"We don't have to update at every single desktop terminal, but just once on the customer's server, and that's it.*"

HEIMBAS®
Softwarelösungen
für das Sozialwesen

Summary

The value of the flexibility of BASIS' interpretive language, which can execute code in a JVM or in a JavaScript/HTML 5 capable device without a JVM, such as an iPad, is the technological secret enabler behind the Heimbas success story. Existing GUI code can run on a mobile device without modification through the magic of the interpreter. BASIS, helping you deliver tomorrow's solutions today!

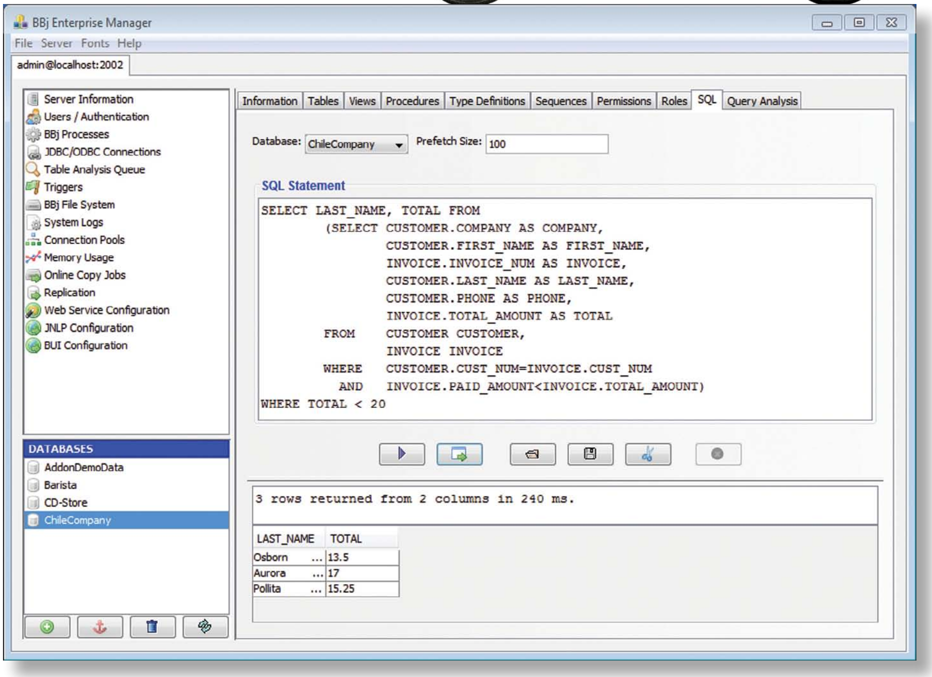
For source code and a sample of incorporating legacy business logic in a mobile application, see the BUI Mortgage Amortization application at links.basis.com/buidemos. ■

DBMS

More Miles per SQL Gallon

BBBj® 11.0 now supports SELECTs from nested SELECTs as well as SELECTs from stored procedures. These enhancements provide developers with additional leverage on those finely-tuned SQL statements and adds functionality to previously written stored procedures. Getting more out of those SQL statements and stored procedures is now more efficient and even easier than ever before, thereby unlocking access to all your production data to modern SQL-based report writing tools such as iReports. > >

By Robert Del Prete
Quality Assurance Engineer



Database: ChileCompany   Prefetch Size: 100

SQL Statement

```
SELECT LAST_NAME, TOTAL FROM
        (SELECT CUSTOMER.COMPANY AS COMPANY,
                CUSTOMER.FIRST_NAME AS FIRST_NAME,
                INVOICE.INVOICE_NUM AS INVOICE,
                CUSTOMER.LAST_NAME AS LAST_NAME,
                CUSTOMER.PHONE AS PHONE,
                INVOICE.TOTAL_AMOUNT AS TOTAL
        FROM    CUSTOMER CUSTOMER,
                INVOICE INVOICE
        WHERE   CUSTOMER.CUST_NUM=INVOICE.CUST_NUM
          AND   INVOICE.PAID_AMOUNT<INVOICE.TOTAL_AMOUNT)
WHERE TOTAL < 20
```

```
3 rows returned from 2 columns in 240 ms.
```

LAST_NAME	TOTAL
Osborn ...	13.5
Aurora ...	17
Pollita ...	15.25

Figure 1. Using the result of a nested SELECT statement as though it were a table

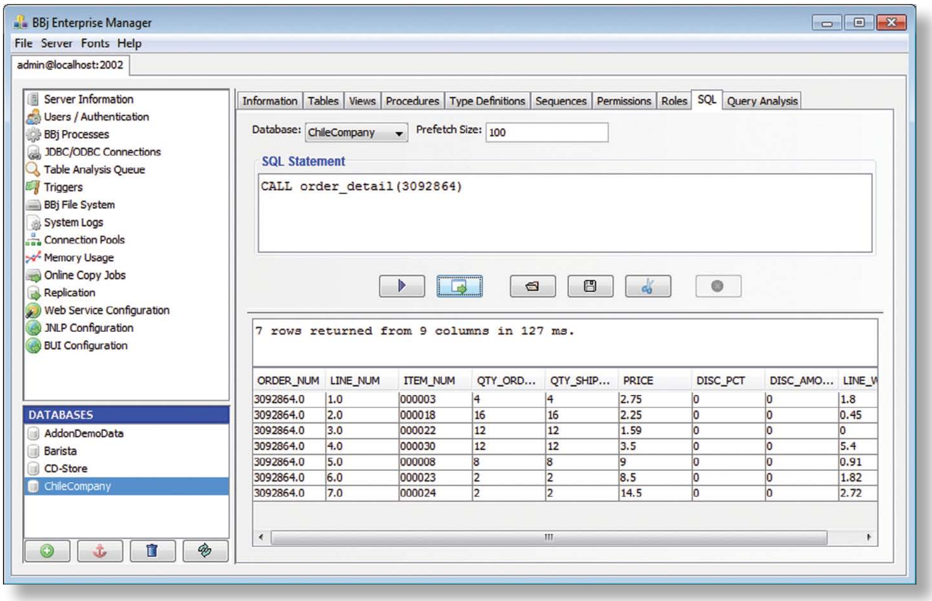
SELECT FROM Nested SELECT

Using the result set of a SELECT statement, developers can now treat that result set the same as any other table in their SELECT statements. Think of it as treating the SELECT statement like a VIEW, although without the requirement to first create the view in the database. This can speed up your development cycle and applications as developers can execute complex queries on the fly without the need to manage and delete temporary views and tables. In addition, they can nest queries to any depth without any limit on the number of nested SELECTs used in any specific query. **Figure 1** shows an SQL query executed in Enterprise Manager that takes advantage of the nested SELECT syntax to extract a smaller result set from an existing query.

SELECT FROM SPROC

A stored procedure (SPROC) consists of a function or procedure written in BBj and embedded in the database. SPROCs can return result sets that are dynamically generated by the code within the procedure. SPROCs can generate results using READ RECORD, leveraging the original high-performing design of the data structures of the legacy production system, and return results usable by SQL. In BBj 11.0, BASIS introduced the ability to perform a SELECT <FIELDS> FROM <MY_CALL_STATEMENT>. This allows developers to treat the result of a SPROC call as a "table" in an SQL statement and provides the ability to further filter, sort, and group SPROC results as well as join those results with other tables in the database. **Figure 2** shows the result of a SPROC using the traditional SQL CALL syntax from the Enterprise Manager's SQL dialog.

A developer can treat this same SPROC's result set as just another table in the database and can use it as a part of a more advanced SQL query. **Figure 3** shows the result of joining the output from the Order Detail SPROC with the Item table as a convenient way to include item descriptions in the result set. This adds new flexibility to pre-existing SPROCs, as their output can be filtered and even augmented without having to change the underlying BBj code that defines the SPROC.



Database: ChileCompany Prefetch Size: 100

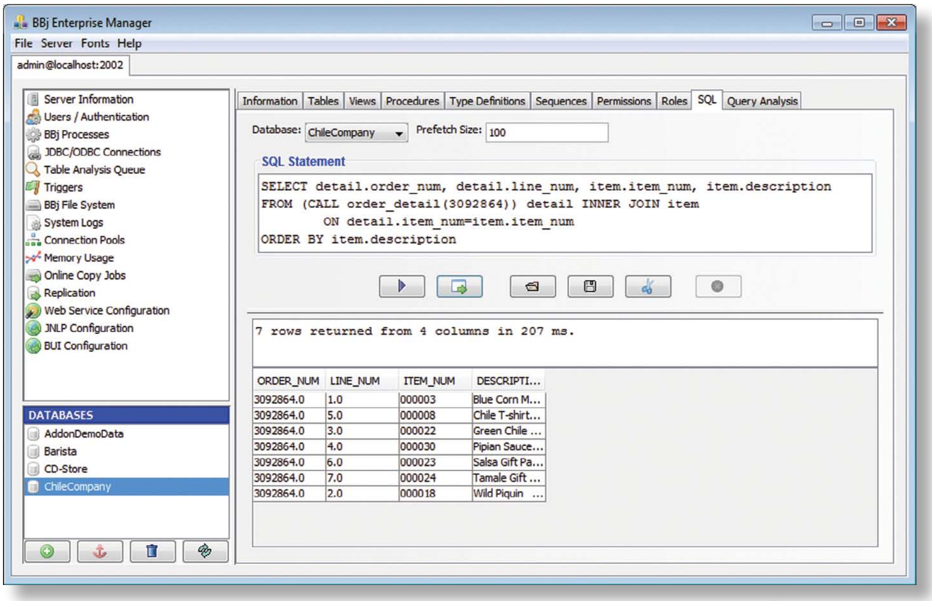
SQL Statement

```
CALL order_detail(3092864)
```

```
7 rows returned from 9 columns in 127 ms.
```

ORDER_NUM	LINE_NUM	ITEM_NUM	QTY_ORD...	QTY_SHIP...	PRICE	DISC_PCT	DISC_AMO...	LINE_W
3092864.0	1.0	000003	4	4	2.75	0	0	1.8
3092864.0	2.0	000018	16	16	2.25	0	0	0.45
3092864.0	3.0	000022	12	12	1.59	0	0	0
3092864.0	4.0	000030	12	12	3.5	0	0	5.4
3092864.0	5.0	000008	8	8	9	0	0	0.91
3092864.0	6.0	000023	2	2	8.5	0	0	1.82
3092864.0	7.0	000024	2	2	14.5	0	0	2.72

Figure 2. A stored procedure that generates current Order Detail information from MKeyed files



Database: ChileCompany Prefetch Size: 100

SQL Statement

```
SELECT detail.order_num, detail.line_num, item.item_num, item.description
FROM (CALL order_detail(3092864)) detail INNER JOIN item
        ON detail.item_num=item.item_num
ORDER BY item.description
```

```
7 rows returned from 4 columns in 207 ms.
```

ORDER_NUM	LINE_NUM	ITEM_NUM	DESCRIPTI...
3092864.0	1.0	000003	Blue Corn M...
3092864.0	5.0	000008	Chile T-shirt...
3092864.0	3.0	000022	Green Chile ...
3092864.0	4.0	000030	Pipian Sauce...
3092864.0	6.0	000023	Salsa Gift Pa...
3092864.0	7.0	000024	Tamale Gift ...
3092864.0	2.0	000018	Wild Piquin ...

Figure 3. Joining the Order Detail SPROC with the Item table

Summary

You can now get additional miles out of your current SQL statements and stored procedures using BBj 11. With added support for SELECTs containing nested SELECTs along with the new functionality of being able to combine stored procedure results with other tables and SELECTs, developers and power end users can go further than ever before! ■

Refer to the BASIS Advantage article *Using Stored Procedures to Add Business Logic to the Database*