



InterBase Releases Server for Novell NetWare 4.11

On March 3, 1998, InterBase Software Corporation announced the availability of its InterBase 4.2.2 Server for Novell NetWare 4.11. InterBase Server is an embeddable database engine that combines business-critical, relational database technology with ease of installation, use and maintenance. InterBase 4.2.2 succeeds the current InterBase 4.0 Server for NetWare 3.12.

“The release of the InterBase 4.2.2 Server underscores the continued commitment to our IT and VAR (value-added-reseller) customers using NetWare,” said Jim Weil, president of InterBase Software Corporation. “This, along with the December 1997 release of InterBase 5.0 Server for Windows NT, Solaris, and HP-UX, illustrates that InterBase Software Corporation is meeting the needs of the embedded database VAR across multiple platforms.”

InterBase 4.2.2 Server for NetWare is already in use by SAGE U.S., Inc., a VAR based in Dallas, Texas that uses InterBase to provide data management in its Carpe Diem software for time tracking of project

development, project management, and billing tasks.

“We are extremely pleased to have the 4.2.2 Server available to us,” said David

“InterBase Software Corporation is meeting the needs of the embedded database VAR across multiple platforms”

Robinson, Carpe Diem development manager, SAGE U.S., Inc. “A large portion of our customers are operating on NetWare 4 platforms, and the 4.2.2 Server will allow us to fulfill the NetWare needs of our growing customer base.”

The upgraded InterBase Server product includes the InterBase 5.1 Client for the 32-bit Microsoft Windows® platform. The enhanced Client is enabled with a 32-bit application interface to the IPX/SPX

Continued on page 2

InterBase Porting to Linux

InterBase Software Corp. is nearing completion of a Linux port of InterBase Server, according to Mike Tossy, Director of Marketing. “We currently expect to be able to offer this port by the end of June,” said Tossy. “The first release will likely be based on InterBase 4.0. We will follow up later in the year with a port of InterBase 5.”

Linux is a freeware workalike of the UNIX operating system. It has gradually grown in popularity on Intel hardware as an affordable alternative to expensive RISC-based UNIX platforms, and resource-hungry Windows NT.

InterBase users have requested a port of the database server to Linux for several years. The response to the announcement on the InterBase list server was enthusiastic and uniformly positive.

InterBase Software Corp. will announce a Beta program for the Linux version in the next few weeks.

Table of Contents

- ▶ **InterBase Releases Server for Novell NetWare 4.11 1**
- ▶ **InterBase Porting to Linux 1**
- ▶ **New Release of Perl Package 2**
- ▶ **Q&A: Chip Handley 2**
- ▶ **Upsizing Paradox Databases 4**
- ▶ **Ask the InterBase Gurus 6**
- ▶ **New Tools 7**

InterBase ports currently available

InterBase 4.0 Server

- Microsoft Windows 3.1x
- SCO OpenServer Release 5.0
- IBM AIX 4.1.x RS/6000
- SGI IRIX V.5.3
- Data General DG/UX R4.11
- SunOS 4.1.4
- DEC Digital UNIX 3.2C

InterBase 4.2.x Server

- NCR UNIX SVR4 2.0.3
- DEC OpenVMS 6.2
- SCO ODT 3.0
- Red Hat Linux 5.0 (6/98)
- Novell NetWare 4.11
- Microsoft Windows NT 3.51

InterBase 5.0 Server

- Microsoft Windows NT 4.0
- Microsoft Windows 95
- Solaris 2.5.1 on SPARC
- HP-UX 10.20

network protocol. This capability is built on Microsoft Winsock,[®] thereby eliminating the third-party client networking component requirement of the previous InterBase 4.0 for NetWare. Both the 32-bit InterBase Client, and the existing 16-bit InterBase Client for Windows are certified to work with the new Server release, using either IPX/SPX or TCP/IP.

Later this year, InterBase Software Corporation plans to follow up with an InterBase 5 Server for NetWare. Thereafter, InterBase projects two releases each year for the Novell platform, starting in the second and fourth quarters of 1999.



Manager of Technical Services Chip Handley enjoys his work.

.....
New Release of Perl Package

BILL KARWIN

An update of **IBPerl**, the popular Perl programmer's interface to the InterBase API is now on the InterBase web site:

<http://www.interbase.com/download/>

Perl is a free interpreted scripting language popular on UNIX platforms. It is well known for its rich collection of contributed modules, especially its CGI programming interface for web server scripts. The home page for Perl is:

<http://www.perl.com/>

IBPerl is a general-purpose, object-oriented interface to the InterBase client API. Any Perl script can directly manipulate data in a local or remote database.

The new version of IBPerl has several enhancements over the previous release:

- Redesigned error handling
- Multiple concurrent connections
- Multiple concurrent transactions
- Multiple concurrent statement cursors
- `fetch()` method to retrieve rows of a query return set directly into Perl lists or hash lists
- Support for fetching Blob data

IBPerl is not a replacement for the InterBase client library; it is a wrapper around the InterBase API, and it functions only when the InterBase client is installed.

IBPerl has been tested only on Solaris, but the Perl port to Win32 is finally stabilizing, so IBPerl will soon work on Wintel.

IBPerl comes with full source code, and it is customizable. I invite Perl hackers to add features and email them to me. Features on the wish list for the next update of IBPerl include:

- Testing to ensure Win32 support
- Compatibility with Tim Bunce's DBI interface
- Support for storing Bob data
- Support for DDL statements
- Support for InterBase array data

IBPerl is a contributed freeware offering, not an officially supported part of the InterBase product. IBPerl comes with no implied liability on the part of Bill Karwin or InterBase Software Corp.

Bill Karwin is the Manager of Technical Publications for InterBase Software Corp., and the author of IBPerl. He can be reached by email at bkarwin@interbase.com.

.....
Q&A: Chip Handley

Each issue of InterCom features an interview with a key InterBase personality. This month, we interview Chip Handley, Manager of InterBase Technical Services.

Q: How long have you worked for InterBase?

I will be celebrating my five-year anniversary with InterBase next month. I have held the titles of Support Engineer, Support Supervisor, Release Manager, and Client Services Manager.

Q: What is it about InterBase that has kept you here for 5 years?

I believe in and am sold on the IB technology. I like the market and the variety of technology I'm exposed to as a part of it—from operating systems and computing environments to our customers' applications.

Q: What makes the IB support team unique?

I'm proud to be a part of the Client Services team because of the way we all work together—we're very much a team. We work in a tiered problem-solving environment: there are 3 tiers, and responsibility for a call is shared among the team. We also work closely with the

international Borland technical services teams, helping them resolve issues with designated contacts and weekly phone calls.

Q: I've noticed you use the term Client Services as well as Technical Services when referring to your team. Why?

I sometimes use the term Client Services because we also deal with customer training, consulting, pre-sales service, and give escalated service levels to our overseas customers. We therefore do more than just technical support, we are here to solve customer needs. Technical Services is the term most people are familiar with, however.

Q: What option is best for a small-volume customer or mid-range VAR?

The best option for a small application developer is to become an InterBase VAR, which offers, among other services, unlimited InterBase Hotline support. We also offer a variety of other options that are

designed to meet different customers needs.

One new option is a low-cost Internet support program, which gives the customer the best value on InterBase technical support. It allows the customer to submit incidents via email and guarantees you a response within two business days.

Q: Does InterBase technical support cover the UNIX versions of InterBase?

UNIX Support Contracts are available, as the majority of our mission critical customers are running on UNIX platforms.

Q: Does InterBase technical support cover InterClient?

InterClient is covered under all the standard contracts at no additional charge.

Q: What is the best way for developers using Borland tools to get technical support?

Basic installation support for InterBase can be obtained through the Borland Technical Support Department with InterBase

providing back line assistance. Once the decision has been made for InterBase as the engine for the application, the developer should use one of the InterBase Support options and work directly with InterBase Software Corp.

Q: Whom should customers contact for support services outside the U.S.?

Outside of the U.S., customers should contact their InterBase Sales Representative for their local service provider.

Q: What does the future hold in store for you team?

My team is working towards using the web to solve problems and proactively get information to our customers. We will be using the web for bug reporting and electronic support services, and we will be publishing a problem resolution database that customers can access. We are also in the process implementing a new call tracking system to better serve our customers needs.

TABLE 1. InterBase technical services offerings

<i>Service</i>	<i>Description</i>	<i>Ordering instruction</i>
Technical Support		
Account Management	Offers a dedicated Support Engineer with guaranteed response time.	Contact your local InterBase Sales representative or call (888)345-2015
InterBase Hotline	Unlimited phone assistance via a toll-free number.	Contact your local InterBase Sales representative or call (888)345-2015
Incidents Support	As needed, pay as you go service. Can be purchased in packs for a volume discount.	Contact your local InterBase Sales representative or call (888)345-2015
On-Line Support	Technical assistance utilizing electronic mail.	Contact your local InterBase Sales representative or call (888)345-2015
Training		
On location	Training offered by a qualified InterBase engineer at the customer site.	Contact your local InterBase Sales representative or call (888)345-2015
Training from MER Systems, Inc.	Five-day instructor lead training course, exploring everything you need to develop and manage a database application.	Contact MER Systems, Inc. (416)410-5166 www.mers.com
Training from DeVries Data Systems	Five-day instructor lead training course, exploring everything you need to develop and manage a database application.	Contact DeVries Data Systems (408)866-8033 www.dvdata.com
Consulting		
Training from experienced partners	Call for more information.	(408)430-1501
Free Technical Information		
Email List Server	An active list server managed off-site by Robert Schieck of MER Systems, Inc.	Send message to listproc@mers.com
Newsgroup		news://forums.borland.com/ borland.public.interbase

Upsizing Paradox Databases

JAMES ARIAS-LA RHEIR

This article describes the issues a database administrator has to consider before migrating a Paradox database to InterBase. This information is presented at a high level so that migrating users see the big picture before diving in. Although this article focuses on migrating Paradox databases to InterBase, the fundamentals of moving from any desktop database (for example, dBASE or Access) to a SQL (Structure Query Language) database server are the same.

Culture Shock!

Changing your Paradox perspective of what the InterBase database might look like it is the first hurdle. Fortunately, a good portion of your current database application design still functions perfectly in the InterBase implementation. The client/server paradigm that InterBase employs entails new concepts that the Paradox developer can (or will learn to) appreciate.

First, the database's structure is the most obvious difference between InterBase and Paradox databases. InterBase databases usually consist of one file (for example, *employee.gdb*) that contains definitions and values for tables, indexes, data validation, triggers, procedures, and so on. Paradox database definitions are distributed among several files, such as:

- A table: *employee.db*
- Indexes for the table: *employee.xg0*
- Validity checks: *employee.val*
- Blobs in the table: *employee.mb*

The next significant difference is the separation of data and code. In Paradox, there is the concept of one or more *working directories* where the data files (Paradox tables), reports, forms, and scripts reside. These reside on either a workstation or dedicated file server. Given the client/server nature of InterBase, only the tables themselves are located on the server machine; all of the other application

files reside on the client. Since you can't specify a remote server's directory as a Paradox working directory, connecting to an InterBase database with a Paradox application requires that you precede the InterBase table name with a Borland Database Engine (BDE) alias.

There is also the concept of *business rules*, sometimes called *integrity constraints*. Paradox and InterBase use similar business rules, although through slightly different implementations. Each DBMS supports referential integrity, primary keys, required fields, and default field values. InterBase has the added functionality of triggers and stored procedures that move the bulk of the data manipulation code off of the client application (as with Paradox) and to the server's database.

Finally, there is the concept of live data. Paradox tables support a pessimistic locking scheme, which forces applications to obtain a record lock before any changes can be attempted. Hence the need for a *pdoxusrs.net* file. InterBase supports an optimistic locking scheme, which enables clients to start changing a record—from a snapshot of the current state of the database—without locking it. This snapshot controls the locking of the data through a transaction stamp, which is managed by the Multi-Generational Architecture of InterBase.

Reasons for Upsizing

Using the InterBase database server provides several advantages over desktop databases:

- You can develop in any software package that can use ODBC drivers or make direct InterBase API calls.
- In all cases, there are increased limits or no limits at all on database and record size, number of columns per table, number of concurrent connections, etc.
- The locking mechanism is more sophisticated; say good-bye to *pdoxusrs.net* problems.

- There are several platforms and operating system versions available as well as portability across those platforms.
- Transactional processing, which provides support for rollback and commit operations, is managed using disk-based writing to provide a more stable environment. Paradox performs all read and write operations in memory in conjunction with the *pdoxusrs.net* file. When a situation occurs, such as a power outage or system crash, that abnormally interrupts Paradox, data loss or corruption may result.
- The Multi-Generational Architecture of InterBase supports a record versioning scheme that reduces data contention. Paradox realizes only one instance of a record and manages it through a lock file (*pdoxusrs.lck*).
- InterBase reliably manages up to 200 concurrent user-connections.
- In InterBase, some database structure changes such as adding and dropping columns can be made on the fly; Paradox requires exclusive access to the tables.
- InterBase requires little *additional* DBA experience. As with application design, your current database knowledge is still valid and easily expands to include the concepts of a database server.
- InterBase maintains a small footprint: 10MB required for server install, 2MB required for client's driver and libraries.

Another advantage InterBase has is the multitude of ways the client application has to communicate with the database server. Following is a brief list:

- Use a BDE alias through an ODBC driver (you get the ODBC driver with the InterBase Server).
- Use a BDE alias through an SQL Links driver (you get the SQL Links driver with one of the Borland Client/Server suites).
- Use a programming language, like C++ Builder to make InterBase API calls. For a web server-based connection, you can

TABLE 2. How Paradox table properties migrate

Table Property	Migration Issues
Datatypes*	Most Paradox datatypes map well to InterBase datatypes. Those that do not <i>transparently</i> translate into InterBase are Auto-increment, separate Time & Date, Logical and BCD.
Picture Statements	Table-based picture statements can be simulated by using check constraints; otherwise, the picture statement can be moved to the data field on the Paradox form.
Table Lookups	CHECK constraints can be used to make a column's entered value conform to another table's column value.
Referential Integrity	By using a combination of CHECK constraints and foreign key designations, you can achieve the same functionality as Paradox referential integrity.
Validity Checks	CHECK constraints can be used to specify requirements of a newly entered value.
Keys*	Keys are called PRIMARY KEYS in InterBase and InterBase does not allow a record to be posted where its primary key value is NULL.
Secondary Indexes*	To improve performance, InterBase indexes have the added option of being made inactive before doing large inserts. Then the index can be re-activated to rebuild the index. InterBase does not support case-insensitive indexes, but you can use collation sequences to achieve similar results.
Password Security	InterBase maintains a security database that is separate from the actual database. A user is given rights to the database using standard SQL GRANT statements (or can be restricted by the REVOKE command). Also, passwords are not 'additive' as in Paradox.
Table Language	In addition to setting the language (character set) for a particular table, InterBase also enables you to specify a single language for the entire database (default) or for a specific column within a table. You can also specify various character orders (COLLATION SEQUENCE).

*These properties are automatically transferred when using the Paradox Copy Utility or Data Pump migration techniques described below.

use the InterClient driver, an all-Java, thin-client JDBC driver. There are InterBase "hooks" to simplify development, which makes for a very fast connection.

- Use a programming language to make ODBC driver API calls, which will in turn make InterBase API calls.

Restructure utility can't modify InterBase metadata, but Paradox's SQL File tool enables you to do such modifications. Fortunately, you can leave most of the front-end application's design intact. You can

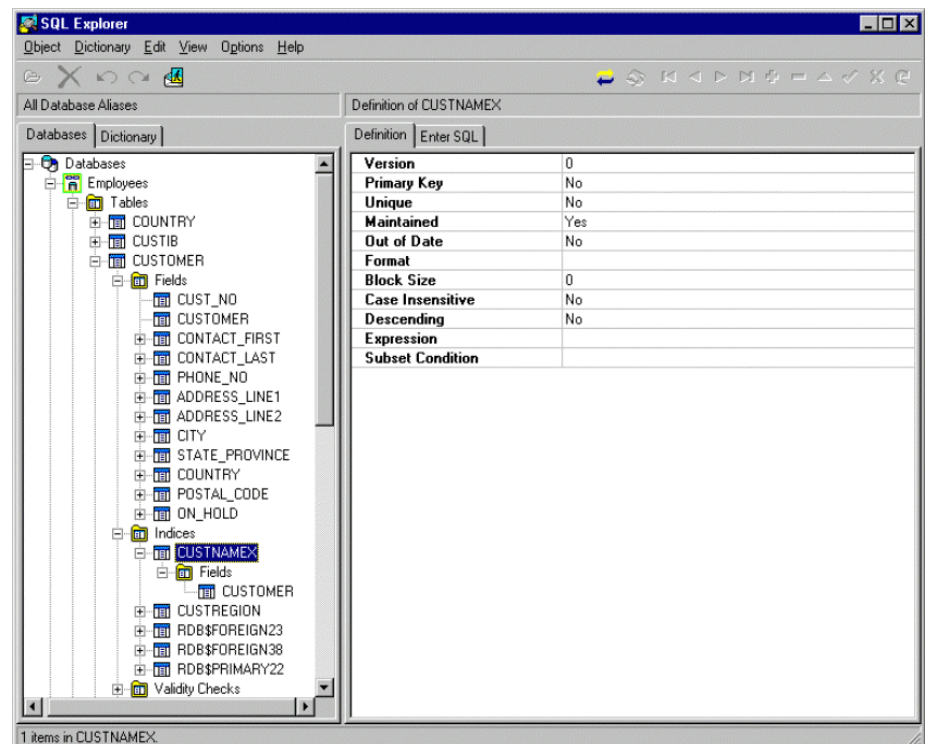
still use the Paradox Data Model to link tables together in forms and reports, and you can leave ObjectPAL code mostly unchanged.

Migrating Paradox tables to an InterBase database

No two database software packages are the same, although the core database design principles such as table relations and the use of indexes still apply. Table 2 lists each property of a Paradox table and its effect in the InterBase scheme. Future articles, will detail the solutions for the problem conversions.

As you've seen so far, depending on the complexity of the tables, migrating Paradox tables to an InterBase database may involve considerable work. For example, you need to learn some basic SQL for database structure (metadata) changes. Database Explorer is a tool that comes with Delphi and C++Builder. It performs most functions—see Figure 1. Paradox's

FIGURE 1. Using Database Explorer to view an InterBase Index Definition.



There are three ways to migrate your Paradox tables into an InterBase database. The following list ranks them from easiest to most difficult:

- Paradox's Copy utility:

The old familiar **Tools | Utilities | Copy** utility translates Paradox datatypes and some table properties to the corresponding InterBase datatypes and properties. An InterBase database must first exist and the aliases must be configured in the BDE.

- Data Pump:

The Data Migration Wizard comes with Delphi and C++Builder. The tool moves data and column structure between different formats. An InterBase database must first exist and the aliases must be configured in the BDE.

- External Files:

InterBase allows you to query/import data from external, fixed-length files, to which Paradox can easily export. Blob data will need to be programmatically extracted to a separate file. This is a data-only migration, and does not translate table structure information, other than column identity.

Because of the SQL foundation InterBase maintains, it is also a good idea to familiarize yourself with SQL and the InterBase extensions to it. There are several good books on SQL; just be sure to purchase one that covers SQL92, as InterBase is fully SQL92 entry-level compliant.

Upcoming articles in InterCom will compose specific solutions to the issues raised here. If you have a specific concern regarding upsizing your desktop database to InterBase and would like it covered in future issues, please send a note describing your concern to jal@interbase.com.

James Arias-La Rbeir is a Client/Server Support Engineer at InterBase Software Corp. He previously served three years as a Senior Support Engineer in the Paradox Technical Support group.

Ask the InterBase Gurus

If you have questions about InterBase that are not covered in the documentation and are beyond the scope of technical support, send them to: intercom@interbase.com.

Pavel Cisar in the Czech Republic asks: "Is there a some kind of cost optimizer in InterBase and if so, what strategy does it use?"

The answer comes from Ann Harrison, one of the early developers of InterBase.

All versions of InterBase have used a cost-based optimizer to select join order. The costs considered are the table cardinality¹ and the index selectivity.² The intention is to minimize the size of the intermediate products.³ Starting in V2.0, we added decision tree pruning,⁴ which reduced the cost of optimizing a 14 way join from 4 hours to 10 seconds, without reducing the performance of the optimized query.

The optimizer first identifies the conjuncts⁵ between tables in the join and indexes that can be used with them. Next, it examines paths through the conjuncts that connect all tables and estimates the size of the intermediate product at each step. Some conjuncts are better than others—equality through a unique index is best; inequality (<>) is so bad it is never used. If there are no indexed paths between tables, the optimizer interpolates a sort/merge join or a natural scan of one table. There's a cost estimate for each of those operations: sort/merge is bad, natural scan is terrible. The optimizer selects the path with the lowest cost.

Finally, the optimizer chooses access strategies for subqueries and applies secondary indexes to queries. InterBase's bitmapped intermediate index structures allow it to use multiple indexes on access to a single table. The V5 optimizer has been enhanced to recognize redundant⁶ and unnecessary⁷ indexes.

In theory, the order of tables in an inner join statement should have no effect on the output of the optimizer. Prior to V5, that

was not the case with joins expressed in SQL92:

```
FROM <blah> JOIN <blah>
ON...
```

rather than the SQL89 syntax:

```
FROM <blah>, <blah>
WHERE...
```

If you find that a query has been badly optimized and the indexes you have defined are appropriate, you can create a PLAN that overrides the optimizer's decisions. If that isn't more than you wanted to know, your boss isn't working you hard enough.

Footnotes

1 Cardinality is the number of rows in the table. InterBase's optimizer estimates cardinality from the number of pointer pages and the size of a row. This number is cheap to maintain and provides a good indication of the I/O necessary to retrieve all rows. However, it does not accurately represent the number of rows in a sparsely populated table. The alternatives we considered would create "hotspots" in the database and degrade update and insert performance. Our decision was that the hotspots would overwhelm any gain in optimization.

2 Selectivity is the ratio of the number of distinct values in the index to the number of entries. The selectivity of an index is set when it is created, reactivated, or when someone requests that it be reset. Keeping more precise statistics on selectivity might improve optimization, but would also reduce update and insert performance.

3 The intermediate product is the number of rows present at any stage of building up the result set. In a simple example, consider the query:

```
SELECT s.name
FROM departments d,
managers m,
WHERE d.manager = m.name
AND d.name = 'NOD'
```

The name column is a unique key in each of the tables and manager is a non-

unique indexed column in departments, meaning there is a 1:n relationship between managers and departments. There are 500 departments and 300 managers.

Starting with the managers produces an initial set of 300. Applying the manager/department conjunct produces 500 intermediate rows. The department/NOD conjunct reduces the final result to one.

Starting with department/NOD, produces one row; applying the department/manager conjunct still produces only one row. That's the join order with the lowest intermediate product.

4 *Decision tree pruning* is throwing out unpromising results without analyzing them completely. The query above can be resolved in one of four ways; see Table 3. Choices 2 and 3 would be discarded after the first conjunct was evaluated because even if the next conjuncts were unique equalities, they'd still have intermediate results as large as or larger than the first case. Cases where NOD and managers are considered together before departments are added aren't considered at all because there is no conjunct between them. The optimizer builds up the longest strings of good choices and then, if none of the strings include all the inputs, it adds sort merge joins for non-indexed conjuncts or natural joins if there are no conjuncts at all.

5 A *conjunct* is a relationship between the data in one table and the data in another. Conjuncts are expressed in the WHERE clause of SQL89 and the ON

clause in SQL92. They can also be derived by 'distributing' the comparison operator. $A.KEY = B.KEY$ describes an explicit conjunct between A and B. $B.KEY = C.KEY$ describes an explicit conjunct between B and C, but also an implicit conjunct between A and C because $A.KEY$ must equal $C.KEY$.

6 *Redundant indexes* are indexes on columns for which other indexes have already been included. Consider a table with a non-unique index on A (descending) and another on B (ascending), plus a unique ascending index on A,B and a unique descending index on B,A—don't snicker, I've seen worse, much worse. A query that includes values for A and B would, prior to V5, use all four indexes, when either of the unique indexes would produce a single value.

7 *Unnecessary indexes* are indexes applied even though they don't restrict the result. Redundant indexes are unnecessary, but not all unnecessary indexes are redundant. Consider a table with a unique indexed column A and a non-unique indexed column B. If your query includes values for both A and B, prior to V5 InterBase would use both indexes, even though the index on A reduces the result set to one row at most.



New Tools

BILL KARWIN

The InterBase community regularly releases tools and development components on the Internet. Here are some recent releases:

IB Objects

Jason Wharton

A suite of shareware components for 32-bit Delphi, providing a visual component layer around the InterBase API. By optimizing these components to the InterBase API rather than conforming to BDE or ODBC, IB Objects provides a superior solution for client/server application programming.

<http://www.ibobjects.com/>

FreeIBComponents

Gregory Deatz

A set of freeware Delphi components, implemented with the TDataSet component, to provide transparent native InterBase access to familiar Delphi data-aware components. *Alpha release.*

www.interbase.com/download/components.html

FreeUDFLib

Gregory Deatz

A library of user-defined functions freely contributed for use with InterBase on Windows platforms.

www.interbase.com/download/udf.html

Marathon

Gimbal Software Services

Marathon is a commercial SQL Tool that allows you to create and manage SQL Database metadata. It is a sophisticated replacement for the "Notepad and ISQL" development methods.

www.sub.net.au/~pokeeffe/marathon.html

Continued on page 8

TABLE 3. Decision tree resolutions

First evaluation	Second evaluation	Third evaluation	Cost analysis
managers	departments	'NOD'	300 / 500 / 1
departments	managers	'NOD'	500 / 500 / 1
departments	'NOD'	managers	500 / 1 / 1
'NOD'	departments	managers	1 / 1 / 1

NEW TOOLS—Continued from page 7

Replication Engine

An example replication solution, contributed by Borland engineers.

www.interbase.dthomas.co.uk/files/ib_repl.zip

The source code is now available:

www.interbase.dthomas.co.uk/files/repl_src.zip

IBPerl

Bill Karwin

A Perl 5 module that provides an object-oriented interface to the InterBase API.

www.interbase.com/download/ibperl.html

Credits

InterCom is a quarterly publication of InterBase Software Corporation. It is mailed to members of the InterBase VAR Program, and also published on the worldwide web.

Bill Karwin, editor
InterBase Software Corporation
1800 Green Hills Road, Suite 150
Scotts Valley, CA 95066
Email: intercom@interbase.com

Contributing authors and editors:

James Arias-La Rheir
Amelia Arnett
Chip Handley
Ann Harrison
Bill Karwin
Rhea Tolman

InterBase® is a registered trademark of InterBase Software Corporation. InterClient™ and InterServer™ are trademarks of InterBase Software Corporation. All other trademarks are property of their respective owners.

© 1997 InterBase Software Corporation. All rights reserved.

InterBase at Borland Conference 1998

This year's Borland Conference in Denver, Colorado has more InterBase presence than ever:

- ▶ InterBase product address
- ▶ Full track of 15 technical presentations
- ▶ Pre-conference tutorial
- ▶ New booth—more space, more product, more people

Register for BorCon 1998: call 1-800-350-4244

Get more information at <http://www.borland.com/borcon98/>



InterBase Software Corporation
1800 Green Hills Road, Suite 150
Scotts Valley, California 95066

