

Borland Corporation

Product Overview

Microsoft and — on a smaller scale — the new Borland Software Corporation have the client/server tools market pretty much to themselves. In Borland's first incarnation, the application development tools space was extremely crowded (remember PowerBuilder?), but the rise of Microsoft left a great many fatalities (or at least stretcher cases). "Borland is on the move again," proclaims Frank Slotman, VP & General Manager of Borland's Software Products operation.

It would have been hard to predict Microsoft's rapid retreat from the Java space. Back in the spring of 2000, Microsoft evangelists were encouraging developers to code Windows applications in Java. But by the end of 2000, Microsoft's Visual J++ product had vanished. C# —

Microsoft's response to Sun's lawsuit — will not kill Java; indeed, in the enterprise space, it is unlikely to even dent it. Visual Cafe — a long time front runner — is now owned by BEA/WebGain, and seems to have lost the independent image it possessed with Symantec. It is therefore not surprising to see Borland's Java products gaining ground: 50% up in revenue from 2Q00 to 3Q00; 250% up from 3Q99 to 3Q00.

Borland has not forgotten its RAD customers, nor have the customers forgotten Borland: Delphi and C++ Builder retain a loyal following and continued attention from Borland. A Linux Journal readers' poll rated Delphi as the most requested port to Linux out of *all* possible applications. The Borland Kylix project is the response to this, and

Company Profile

It's "Borland" again. Inprise, the company formally known as Borland, is Borland once more. Those who admire its technology and wish the company well might breathe a sigh of relief. But history does not repeat itself. Back in the 1980s, Borland's domain — primarily client/server application development tools — was the high ground of the new IT. The landscape looks different now, as does Borland. The constant feature is Borland's commitment to providing the highest quality tools for technologists who implement e-business systems. The new Borland's value proposition and market position might at last have made sense of that.

Borland was founded in 1983 with Turbo Pascal, a compiler that sold at \$50 a copy. By 1992, it had become perhaps the most important client/server tools vendor, with revenue around \$500 million. But the following year the company reported a revenue drop of 35%, and had Microsoft close on its heels.

Borland's response was to move up market — acquiring the Open Environment Corporation in 1996 (the source of the Entera DCE middleware product) and then Visigenic (source of the VisiBroker ORB products) in 1998. The two acquisitions helped the company gain a significant presence in the Unix market, something that has been capitalized on with the more recent Java products.

Dale Fuller took over as CEO from Del Yocam in April 1999, with the company at a low point: the Inprise brand had not delivered a new market to the company, and seemed to have alienated some part of its established customer base. Fuller based the turnaround on the company's core products:

- *Delphi/C++Builder* — High performance RAD toolset for client/server and distributed computing.
- *JBuilder & AppServer* — Java development and deployment for enterprise systems.
- *VisiBroker* — Industry-leading CORBA services implementation.
- *InterBase* — Embedded relational database.

Other products are now growing up around these, but the essential principal remains the same: Borland must concentrate on a limited expression of its core competence — technology for technologists — locating the value-add in quality and strong platform independence. According to the company, Borland is highly focused on assembling and packaging its products into comprehensive platforms for implementing e-business applications, providing development, deployment and management software technologies in a single offering. Its acquisition of Bedouin Inc. last November sets the stage for Borland

to deliver a comprehensive framework of hosted development services which will leverage the software implementation platforms and its sizable user base.

Third-quarter 2000 results, reported October 24, showed revenues at \$47.6 million, up marginally from \$45.7 million in the previous year's Q3. Income was \$8.5 million, up from a loss of \$1.4 million in the previous year's quarter. The company's cash and short-term investments now total around \$250 million. The results were consistent with those for Q2, which showed revenues at \$46.7 million, up from \$40.2 million in Q2 fiscal 1999.

Borland's customers include technology companies such as Compaq, NEC, and National Semiconductor, and government agencies including the UK treasury, US Navy, and NASA. Borland has partnerships with Arthur Andersen, Ernst & Young, KPMG and MCI Systemhouse. Borland employs approximately 900 staff, with all technology development being carried out in the US. Around 50% of revenues are domestic.

Borland Corporation
100 Enterprise Way
Scotts Valley, CA 95066
Tel.: (831) 431-1000
E-mail: info@borland.com
WWW: <http://www.borland.com/>

will shortly deliver the Delphi for Linux product now also branded Kylix.

Borland retains its focus as the applications tools and technology vendor of choice for technologists — recent customer gains such as UUNet in the UK illustrate this. Borland is consolidating its position, but some question remains about whether the position is viable. Many of the other high-profile technology vendors now talk about a shift from nuts and bolts to solutions. Borland is also busy assembling its product components into platforms for implementing e-business systems. So what's the value proposition?

PRODUCTS

RAD PRODUCTS

Delphi and C++ Builder continue to develop quietly. Judging by the industry hype, it would be easy to imagine that client/server RAD has now been overtaken by Java distributed application development or is now confined to Microsoft Visual Studio. But more than half a million downloads of Borland's most recent C++ compiler suggests otherwise.

Delphi 5

Delphi has long been something of a reference standard for 3GL developer productivity and performance of executable code. The Delphi IDE was the model for the JBuilder IDE (though the latter now has a completely different architecture and independent code base). The runtime services now available to Delphi application builder also resemble the Java Enterprise platform.

Delphi 5 retains the three editions: Enterprise, Professional, and Standard. The higher two versions ship with the Borland WebBroker, giving Delphi developers similar deployment options available to Java developers.

Delphi 5 adds multiprocess debugging, with stepping possible across process boundaries, together with support for PVCS and other versioning systems. A big stack of new components — more than 200 in total — are available, these include an HTML client, together with Microsoft Office automation elements. Delphi remains Windows-centric, but is open to the

outside world: thus, it includes Borland's ADOExpress for access to Microsoft Active Data Objects. At the same time, Borland's MIDAS services gives Delphi access to CORBA, COM, and HTTP clients.

MIDAS 3

The Multi-tier Distributed Application Services suite (MIDAS) provides both Delphi and C++ Builder developers with a collection of middleware services. MIDAS covers both COM/Microsoft Transaction Server (MTS), and open technologies over TCP/IP. It supports CORBA, Java, and MTS distributed component clients, HTTP thin clients, and disconnected/roaming clients, in the context of database applications.

MIDAS makes use of the Delphi Remote Data Broker. This provides a general purpose solution for supplying data from an application server to a range of client types. The Remote Data Broker can act as a transaction hub; an alternative Stateless Data Broker can act as an MTS client. Other services include server object pooling, and the ability to propagate database constraints to a client. A Java client component for MIDAS allows any type of pure Java client to interoperate with MIDAS applications.

MIDAS continues to gain new functionality — its functionality now resembles that of the Java Enterprise platform specification. MIDAS is a proprietary solution (and is limited in deployment to Windows platforms), but its open connectivity options enable it to operate within any mixed environment.

InterBase 6.0

The big news is that InterBase has gone open source. InterBase is now available in binary form for Linux, in addition to Solaris and Windows. As of July 2000, Borland has made the source code available. The code can be downloaded from the Borland Web site, and is controlled by the Mozilla Public License — developers can modify the source code, but are required to make the source modifications available.

The InterBase product was first released in 1985. Since then, it has gained plenty of new functionality, but it has retained the original design goal,

attempting to provide enterprise features while minimizing maintenance requirements. InterBase now provides SQL-92 entry-level compliance, together with distributed transaction support: multiple InterBase databases can take part in the same operations, and are controlled by a two-phased commit.

The InterBase license developments are complementary to the Kylix Project, which aims to supply application development tools in an open source environment. However, InterBase will not be the only database supported by Kylix products.

Kylix

The Kylix project began in 1999, coincident with the first Linux World conference. Kylix is in the process of creating native application development tools for Linux: the first of these is the Kylix product itself, which represents the implementation of Delphi on Linux; C++ Builder will follow later. Kylix is intended to be used to create both open source Gnu Public License (GPL) applications, as well as proprietary applications. For the former category to be addressed properly, some of the Kylix tools themselves will be open source.

C++ Builder and Delphi would not be the first application development tools to be ported to Linux: the original Gnu C compiler (GCC)/Emacs editor combination have already been joined by MetroWerks Code Warrior and Cygnus Code Fusion. Borland's position is that all of those tools are best suited to system-level development work, while Delphi is about end-user applications that run both client and server-side.

Linux system services are substantially different to those for Windows, and this will prevent all but the simplest Delphi applications being ported to Linux without modification. However, the required changes are not expected to be substantial. A key aspect of the move to Linux centers around the Borland Visual Component Library (VCL). This is being presented as CLX, a cross-platform library of user interface, database, network, and Internet components. The GUI components can target both Gnome and KDE Linux user interfaces. CLX has been closely modeled on VCL, but is not fully compatible with it. Kylix was announced on January 31 and will ship in 1Q01.

Java PRODUCTS

The evolution of Borland's Java products reflects that of the language itself. What began life as a (much hyped) programming oddity, is now installing itself as a key enterprise integration platform. JBuilder's future has now been mapped out: it includes UML modeling, together with support for Rational products — that probably means that JBuilder will be supported within the Rational Unified Process (RUP) for software life-cycle management.

JBuilder Enterprise edition now comes with support for the BEA WebLogic server, in addition to the Borland AppServer. Integration between TIBCO middleware and the Borland AppServer is now also possible. The next integration target is IBM WebSphere; delivery of this is planned for 1Q01.

JBuilder 4

JBuilder began life as a heavily modified Version of Delphi. Since then, JBuilder has acquired a wholly new design and code base, thus JBuilder 3 and 4 are both coded completely in Java. The final result is excellent — here is a development environment with all the productivity strengths of Delphi, but with its own optimized implementation. It is interesting to observe that the Oracle JDeveloper toolset is also based on JBuilder, but on a much earlier cut of the product. Thus, JDeveloper — although it contains some exclusive Oracle extensions — lacks both the polish and the portability of JBuilder 4.

The latest version of JBuilder retains the 100% Pure Java output and — although a significant amount of development work can be done visually or by coding wizards — JBuilder does not introduce any proprietary markers into the source code. IDE changes over previous versions are minor, and will not be significant to many developers. But debugging and deployment changes are very significant: the toolset can now detect thread deadlock, and race conditions; it also supports Servlet and Java Server Pages (JSP) debugging, with breakpoints inside JSP pages. The Enterprise edition of JBuilder 4 includes an Entity Bean Modeler; this removes much of the complexity in developing Entity JavaBeans (EJBs)

— it includes visual descriptions of relationships between EJBs and database table entity. This edition also ships with the Borland AppServer. Professional and Enterprise editions are also shipped with the Borland JDataStore embedded Java database.

For the first time, JBuilder has acquired some Wireless Application Protocol (WAP) and Wireless Mark-Up Language (WML) support: WML can be generated by the JSPs managed by JBuilder; the Borland AppServer also carries WAP support. The JBuilder Handheld Express add-on extends deployment options to the Java 2 Micro Edition (J2ME). The most important J2ME platform is the Palm operating system.

JDataStore 4

JDataStore was designed to fulfill the common requirements of an embedded database: small footprint, high performance, portability, zero administration, high security, and the ability to synchronize with any JDBC source. JDataStore can also act as a JDBC SQL database in its own right, either in-process with an application, or as a networked server. In addition, it can act as a self-contained portable file system, carrying any files, including serialized Java objects.

Although JDataStore can act as a flexible RDBMS in its own right (particularly for relatively small data sets), Borland presents it as complementary to the mainstream primary data stores: Oracle, IBM DB2, and Microsoft SQL Server. To this end, JDataStore carries its own resolvers that can synchronize modified data with remote JDBC databases.

JDataStore helps to answer the question about the possible performance of Java database engines: particularly for Java applications. Java databases have a number of significant advantages over systems written in C: databases can run in process, sharing the same heap, and removing all networking and data marshalling overheads. The database can also share the same runtime libraries, minimizing the footprint. JDataStore is also able to exploit thread handling optimizations in the Java HotSpot server virtual machines.

SERVER PRODUCTS

The most important two areas of activity for the development of Borland server products are standards implementation, improved integration, and 64-bit support. The former is already obvious, with much improved integration between JBuilder 4 and Borland AppServer 4.5. JBuilder, VisiBroker for Java, and the AppServer can already target the Intel Itanium 64 on Windows 2000 now; Red Hat Linux support for Itanium 64 will be delivered soon.

VisiBroker 4

CORBA has already become the standard arrangement for cross-language integration for enterprise systems; Borland VisiBroker is now the *de facto* standard CORBA implementation — VisiBroker has been adopted by HP, Ericsson, Netscape, and Oracle. “We’re the i in Oracle8i” points out Borland’s Enterprise Products General Manager.

The purpose of a CORBA Object Request Broker (ORB) is to make the interaction between distributed application components, both language-independent and location independent. ORBs such as VisiBroker add enterprise features such as clustering, load balancing, and failover. The VisiBroker product group also includes a CORBA interoperable Naming Service, providing a hierarchical namespace for registered objects.

The core of VisiBroker dates back to the early 1990s with VisiGenic, and the product continues to develop within Borland: the VisiBroker Gatekeeper now provides packaged connectivity facilities for clients outside of a firewall. It enables requests and call-backs to take place over HTTP, including SSL connections. Visigenic originally proposed a Distributed Application Platform. This was essentially a set of design patterns for business services over VisiBroker; the Distributed Application Platform has since been superseded by the Java 2 Enterprise Edition (J2EE) specification and implemented by Borland AppServer.

AppServer 4.5

The Borland AppServer is presented as an extremely high-performance J2EE implementation, in addition to its maximal compliance with published Java standards. Performance is primarily as a result of the underlying VisiBroker CORBA Object Transaction Service (OTS) module — the CORBA layer is not normally visible to the applications

developer, but it is responsible for much of the work behind the scenes.

A number of subtle but important features set AppServer 4.5 apart from competing J2EE implementations

- *Session Beans* — Maintaining session state using a scalable method is now a well-recognized problem for J2EE developers. Borland uses a highly efficient “passivization” technique for session beans, and claims orders of magnitude performance benefits. AppServer 4.5 also provides failover for session state.
- *Container Managed Persistence (CMP)* — CMP is an elegant but potentially inefficient method of implementing EJBs. Borland’s CMP support includes both JDBC standard mapping, and optimized mapping to Oracle, Informix, Sybase, DB2, SQL Server, InterBase, or JDataStore databases. The CMP engine can cache EJB data, and uses a number of optimizations to minimize traffic to the back-end database.
- *Object-Relational Mapping* — Single EJB classes can be mapped on to any number of tables, views, or even multiple databases. Another clear differentiator here is that bean-managed persistence and container-managed persistence can both be used within one EJB.

AppServer 4.5 is itself written entirely in Java 2, and has no native code on any platform. This should finally put an end to the Java performance debate. The AppServer 4.5 also includes a highly optimized single-phased commit JTS/JTA service. The server has its own GUI console — presented both as part of JBuilder 4 Enterprise edition, and accessible independently. AppServer applications can best be managed with the separate AppCenter application management product.

AppCenter 4

AppCenter 4 is an application management toolset, specifically designed to work at the component level. In this respect, AppCenter is complementary to resource or systems management systems such as Tivoli — AppCenter itself can act as a Tivoli managed resource.

AppCenter can manage J2EE and DCOM components, CORBA objects, and their containers. The application models that are developed and used within AppCenter supports the migration and redeployment of components, together with other component aspects such as containment, grouping, dependency and failover — all these relationships are displayed visually, in real time.

AppCenter provides its own visual “cockpit” or design studio, in which the application manager can compose descriptions, performance metrics, exceptions, and events related to application structures and functionality.

Future versions of AppCenter will be able to manage Jini networked devices. AppCenter will also embrace the Java Management eXtension (JMX) and WEB/CIM standards, when these are finalized.

With AppCenter, Borland makes a strong case in favor of component-level management for component-based applications: put simply, structural or platform-based management tools cannot see the application structure, and functional-based management cannot determine where a fault or bottleneck occurs, at least not to a sufficiently fine level of granularity.

SECURITY SERVICE 4.1

The Borland Security Services is integrated with VisiBroker and AppServer — InterBase and JDataStore security is handled differently, and in any case, Borland anticipates that databases will effectively be hidden behind an application server.

The Security Service handles the following issues:

- *EJB Container* — The security aspects of an EJB are described and implemented using the standard CORBA Sec security model.
- *Web Services* — An HTTPS service is built into the Borland AppServer; this is integrated into the Security Service.
- *Perimeter defense* — The Security Service is integrated with the VisiBroker Gatekeeper. This effectively allows the creation of virtual private networks.
- *Transport* — The Secure Sockets Layer (SSL) is used as the transport layer for all Security Service communications.

- *Authentication* — This is based on public key authentication and X.509 certificates. Authentication can use the Unix Network Information System (NIS) or Windows domains.

Borland does not act as a certificate authority, but can work with any of the established public authorities. The Security Service is in part a software product, and in part a service offering — as part of a small but useful collection of developer services.

PLATFORMS

Delphi 5

- *Operating systems:* Microsoft Windows 95, 98, NT 4.0, and 2000

MIDAS

- *Operating systems:* Microsoft Windows 95, 98, and NT 4.0, and 2000

Entera 4. 2

- *Operating systems:* Microsoft Windows NT 4.0, HP-UX, IBM AIX, and Sun Solaris

InterBase

- *Operating systems:* Microsoft Windows NT 4.0 and 2000, Linux, HP-UX, IBM AIX, SCO Unix, Sun Solaris, and Novell NetWare

Kylix

- *Operating systems:* Linux, using KDE, and Gnome

JBuilder & JDataStore

- *Operating systems:* Microsoft Windows 98, NT 4.0, 2000, Sun Solaris, Linux (Red Hat 6, Mandrake 7.0). Note that the Java products are certified for these platforms, but also work on others. Apple Mac OS X will be supported following the Mac OS X general release.

VisiBroker for Java & VisiBroker for C++

- *Operating Systems:* Microsoft Windows 95, 98, NT 4.0, 2000, Red Hat Linux 6, HP-UX, IBM AIX, Sun Solaris, and IBM zSeries.
- *Embedded operating systems:* pSOSystem and VxWorks.

Borland AppServer 4. 5

- *Operating systems:* Microsoft Windows NT 4.0, 2000, Red Hat Linux 6, HP-UX, IBM AIX, and Sun Solaris.
- *Databases:* standard JDBC / SQL-92, Oracle, Informix, Sybase, IBM DB2, Microsoft SQL Server, Borland InterBase, or JDataStore.

Borland AppCenter 4

- *Operating systems:* Microsoft Windows NT 4.0, 2000, Windows 2000, Red Hat 6.0 Linux, HP-UX, IBM AIX, and Sun Solaris.

PRICING

The following are US prices, and should be used as a guide only in other countries:

- *Delphi 5 Enterprise:* \$2,500.
- *MIDAS 3 unlimited server license:* \$300.
- *InterBase Internet license:* \$2,500.
- *JBuilder Enterprise:* \$3,000.
- *JDataStore per seat:* \$100, or per server CPU: \$2,500.

OPINION

Strengths

- *Quality* — Borland products have a habit of doing exactly what they say on the package, with few if any nasty surprises. Software quality for development tools is not a factor that analysts' reports normally cover, but an exception has to be made here.
- *Timeliness* — Both in Java and CORBA domains, Borland has normally been very quick in implementing new standards as they are ratified. This comes in part because the architecture of the products allows for this (in contrast with IBM Visual Age), and in part because Borland takes an active role in specifying the standards.
- *Responsiveness* — Borland has historically been good at listening to developers. It's Community Web site has greatly enhanced this process.
- *Performance* — Right from the beginnings of Delphi, applications written with Borland tools have run fast. Currently, the advice for anyone intending to demonstrate an EJB application has to be: demonstrate it on Borland AppServer.

WEAKNESSES

- *Deployment options* — Even on Version 4, JBuilder still has limited automated EJB deployment options. BEA WebLogic is now fully supported, IBMSphere is necessary, and will be delivered soon. CA Jasmine*ii* and SilverStream would also be useful. But is there a conflict of interest between JBuilder and AppServer?

- *RUP integration* — Microsoft Visual Studio and IBM WebSphere are now both supported by the Rational Unified Process (RUP) environment. Borland is rumored to shortly announce a global partnership with Rational which will permit Borland to bundle various Rational product components, including the RUP. Support for JBuilder would be a big plus for both.
- *Domain specialization* — Unlike most of its competitors, Borland does not have a significant customer-facing services division. This means that it lacks specialized domain knowledge, and has not so far been able to develop much vertical presence. Much of its vertical presence is in high end processing environments such as technology, financial services and telecommunications.

CONCLUSIONS

Borland AppServer has been the performance leader in its class for some time now. More recently— particularly following collaboration with object-relational experts Thought CocoBase — AppServer moves even further ahead of the field. Borland JDataStore confirms the trend: in realistic tests conducted by the author, JDataStore's transactional performance can return up to six-fold speed improvement over the competing Java RDBMS Informix Cloudscape.

Another big benefit for the customer is the almost complete lack of vendor lock-in generated by JBuilder, JDataStore, VisiBroker, and AppServer. That lack of lock-in delivers on the J2EE vision — vendors should be competing on quality of service and breadth of connectivity, with code remaining portable. For ISVs, VARs and other domain specialists involved in coding and selling Java business components, JBuilder and AppServer must be the environment of choice, at least for development. In contrast, the component customers may have their own deployment server choice, to match an existing infrastructure.

Where does this leave Borland? It's no bad thing to be the technical champions in a technical domain. But solving problems before your custom-

ers know they have them is a demonstrably dangerous business strategy. J2EE is creating portability for substantial business components; this is a wholly good thing for developers and customers. But J2EE may also be creating an illusion of sameness across J2EE vendors — and that would be a bad thing for Borland. We can summarize the facts here: AppServer is a more consistent and performant product than IBM WebSphere; AppServer — even without AppCenter — is much easier to manage than BEA WebLogic. Unfortunately for Borland, IBM can present WebSphere as part of a complete “solution” — and both IBM and BEA have excellent customer lists.

So how can Borland be seen to be adding value to its server products? Certainly, high performance and a strong relationship with developers both make a good start. Another part of the proposition comes from careful nurturing of both CORBA and RAD products: the AppServer allows Windows applications to talk to EJBs; that should become a big win over Microsoft Visual Studio products.

Borland can claim to be the *i* in Oracle8; they can also claim to be the *J* in Oracle JDeveloper. (It's unfortunate though, that JDeveloper was based on the core of an early version of Borland JBuilder with few of the advantages of the current release.) JBuilder was also used as a key element of the IBM San Francisco business components project — ironically, IBM VisualAge for Java was not capable of targeting the appropriate JVM. Borland needs to actively pursue this type of relationship.

Borland is likely to score well in the embedded Java development market — JBuilder Handheld Express is an early and important response in this area. Despite claims to the contrary, Borland will end up going head to head with Metrowerks sometime soon, and Borland is likely to win. In the mean time, Borland will have to make do with a most unusual value proposition — quality.