

WHITE PAPER

Sybase PowerBuilder: Back to the Future with Application Development Productivity

Sponsored by: Sybase

Al Hilwa
May 2009

IDC OPINION

Sybase PowerBuilder has remained throughout the years a bastion of application development productivity for a wide class of enterprise applications that combine heavy database orientation with a highly productive graphical user interface (GUI). Revolutionary at its inception for creating and popularizing this specific genre of distributed application development, the PowerBuilder development environment today combines functional maturity with aggressive modernizations that keep it a viable and compelling integrated development environment (IDE) application runtime. In particular, the direction that PowerBuilder has taken to become a seamless Windows .NET environment with the Visual Studio shell incorporated into its infrastructure has meant that the product can now be considered equally for evolving existing systems and for developing new ones for traditional Windows 32 environments and for .NET. In particular, Sybase PowerBuilder provides the following benefits:

- ☒ A solid application development environment that is currently in its fourteenth major release. Since its inception in 1991, PowerBuilder has aggressively evolved to meet customer demands for new architectures and technologies.
- ☒ A highly productive paradigm for rapidly constructing both the back end and the front end of an application. PowerBuilder features the highly acclaimed "DataWindow" metaphor with native support for multiple popular database management systems as well as a graphical approach to painting the user interface screens and components of an application.
- ☒ A powerful, yet easy-to-use scripting language, which has been kept current to support modern language features and techniques that permit the construction of business logic tying the user interface to the enterprise data.
- ☒ Sybase is a financially stable and healthy concern that has provided strong stewardship of the PowerBuilder asset with an intimate customer interaction model and a strategic road map and vision for the product.

TABLE OF CONTENTS

	P
In This White Paper	1
Situation Overview	1
Where Is PowerBuilder Today?	2
A History of Supporting Developers.....	2
The .NET Alignment	4
PowerBuilder 9, 10, and 10.5	4
PowerBuilder 11 and 11.5	4
PowerBuilder 12	5
Modernization of PowerBuilder Applications.....	7
Future Outlook	7
Challenges/Opportunities	7
Conclusion	8

LIST OF TABLES

	P
1 PowerBuilder Release History.....	3

LIST OF FIGURES

	P
1 Supporting .NET in the PowerBuilder Runtime	6

IN THIS WHITE PAPER

In this white paper we discuss the progress of Sybase PowerBuilder and assess the overall strategic options facing many PowerBuilder customers with respect to the evolution of their applications. While the visibility of the PowerBuilder toolset is not where it used to be in the early 1990s, we find that the original value proposition of the product has endured and the strategic direction taken by Sybase to evolve the installed base is sound and well thought through. The strategy of record for PowerBuilder is likely, in due course, to put PowerBuilder applications in almost complete alignment with the Microsoft .NET platform and provide its developers with all the salient aspects of the successful and popular Visual Studio application development environment.

SITUATION OVERVIEW

PowerBuilder is a development tool with a storied history. It is recognized for bringing a rapid application development (RAD) paradigm to the personal computer in the early 1990s that effectively created a revolution in the productivity of custom business application development. PowerBuilder blended scripting capabilities from the established genre of "fourth-generation languages" (4GLs), which emerged in a textual form in the late 1980s with a visual development metaphor that was more commonly associated with not very scalable PC database products of the era. Thus, by combining the dexterity of programming in a 4GL with the visual RAD capabilities of PC databases applied to back-end business database systems, PowerBuilder was a success in short order. From its early years, PowerBuilder managed to gain developer share at a rapid pace over its competition as some of the best-known companies in the world (e.g., American Airlines, Coca-Cola, 3M) understood its advantages and utilized it to build internal business applications. PowerBuilder proceeded on a growth trajectory until the fame and fortune of two-tier client/server architectures began to wane around the turn of the century, as a result of a confluence of factors. The second half of the 1990s was the time when the Internet got its mainstream footing promoting a specific type of Web architecture for consumer applications; it was also a time when disenchantment with the scale and manageability of two-tier client/server architectures gave rise to new entrants such as Forte and Seer that promoted three-tier and multitier architectures for custom applications. Third-generation languages (3GLs) were also being reborn as more programmer-friendly with better-organized frameworks operating at ever higher abstraction levels and richer, more supportive shells (e.g., Visual Basic and Borland Delphi). To some extent, the productive application development tools category began to fragment, culminating in the arrival of the Java platform and language and the soon-to-follow .NET framework and its languages, which created a new architecture for multiple programming languages of varying abstraction levels to reside on top of a common generalized programming infrastructure (supported by an intermediate execution layer with interpretation and/or just-in-time compilation). These full-featured and highly developer-supportive runtime environments took the edge off of some of the most painful aspects of 3GL development (e.g., memory management, type enforcement, error handling) by moving these tasks out of programmers' error-prone hands and into the ever-thickening supplied language runtimes. Finally, along with the changes in programming languages and environments, a new application development architecture based on the World Wide Web began to emerge in the mid-1990s; it slowly

gained traction and now has become one of the most popular targets of new application development tools. Today, much of this evolution has resulted in a situation where the most strategic application development tools are aligned along one of two major ecosystems: the Microsoft community and the Java community. In fact, many application development tools and their runtimes that were developed during the course of the 1990s have found themselves taking this fork in the road to align with one of these two broader ecosystems while also supporting the Web architecture through various extensions. Some attempted to support both to varying degrees, and others oscillated in support between the two large ecosystems as they went through a deep discovery process with their installed base. Sybase has gone through both of these trajectories over its long-running evolution of PowerBuilder, but as of the recent few releases, it has settled on a strategy of choice more decisively.

Where Is PowerBuilder Today?

Despite the fierce shifts in programming languages and application architectures, we are hard-pressed to find an environment as productive to develop in as PowerBuilder. What's more, PowerBuilder has been evolved aggressively as these shifts have taken place. This fact has not been lost on PowerBuilder developers, many — if not most — of whom have continued to use the product as their primary development tool years after they may have run into it for the first time. In our exploration of the PowerBuilder developer community, we have noticed a degree of ardor and fondness that goes over and above the norm in the development language space, which is typically known for a high degree of developer ardor and fondness. This has translated into a level of stickiness in PowerBuilder adoption that also stands out in this market, which again tends to be sticky to begin with. Languages and development tools are sticky in the sense that once developers have adopted them, they tend to stick to their decisions due to invested know-how, accumulated code libraries, and the generally steep learning curve of application development technology in general. Additionally, most programmers have obligations to maintain applications they have developed in the past and thus avoid switching toolsets unnecessarily. In the case of PowerBuilder, Sybase's plan of record is to continue to ensure that developers are able to fully leverage both the Visual Studio IDE and the .NET runtime while retaining their invested skills and the power and productivity of PowerBuilder.

A History of Supporting Developers

It is interesting to follow this trajectory backward to shed some light on how PowerBuilder has managed to keep much of its installed base of developers in the fold. Currently in its fourteenth major production release, PowerBuilder has been evolved aggressively and continuously since its inception, as Table 1 illustrates. In addition to bringing to market strong .NET alignment, the recent releases of PowerBuilder have continued to bring new capabilities to developers. The latest release, 11.5, brings new GUI improvements such as RichText in the DataWindow for displaying columns and adds 3D graphing capabilities (e.g., 3D pie, bar, and other kinds of charts) based on the DirectX runtime support, among other new capabilities. At the back end, new object properties have been added and new native drivers for Oracle 11g and SQL Server 2008 have been introduced.

TABLE 1

PowerBuilder Release History

Release Number	Release Date	Key New Feature Highlights
1	July 1991	Beginning of a historic run
2	June 1992	OO support, Painter
3	May 1993	Bundled DB (Watcom), Version Control
4	November 1995	Reporting, Data Pipeline, Source Control, OLE 2.0
5	July 1996	Machine code compilation, Distributed PowerBuilder three-tier support
6	December 1997	Window ActiveX, CORBA, DataWindow HTML generation, DataWindow synchronization
6.5	August 1998	COM and Java component generators, separate Unix, Macintosh, and Unicode SKUs, internationalization via Translation Toolkit
7	October 1999	New IDE, new layouts and look/feel, Jaguar integration (aka EAServer) — build, create, and deploy COM and MTS components (Component Theme)
8	June 2001	Web Targets, Web DataWindow (integration of PowerSite functionality into PowerBuilder) (Web Development Theme)
9	February 2003	.NET phase 1: Web Services, XML, Java Server Pages
10	July 2004	.NET phase 2: DataWindow .NET, Fully Unicode
10.5	March 2006	Large and small UI and core client/server enhancements, including TreeView DataWindow, RichText Edit Control, .NET-based Web Services engine (Visual Enhancements Theme)
11	July 2007	.NET phase 3: NVOs as .NET assemblies, ASP.NET Web Forms, Windows Forms (.NET Theme, Web Services Theme)
11.5	September 2008	DataWindow enhancements, Oracle 11g and SQL Server 2008 support, improved graphing (Visual Enhancements Theme)
12	<i>Estimated 2010</i>	<i>.NET phase 4 (Fully .NET and WPF Theme)</i>

Source: IDC, 2009

The .NET Alignment

After first trying to align with Java, Sybase discovered that the PowerBuilder user base is more aligned and has considerably more overlap with Microsoft's developer community. As a result, Sybase has chosen to put its lot in with .NET, and the PowerBuilder team is on a trajectory of deepening alignment with .NET technology in a four-phase strategic plan involving four major releases (and two minor ones). To date, Sybase has delivered approximately 70% of this change with the final steps coming in release 12, which is scheduled for the early 2010 time frame.

PowerBuilder 9, 10, and 10.5

Sybase began work on its .NET alignment in 2002 as release 9 of PowerBuilder was being planned. The four-phase strategy was kicked off with release 9, which featured a Web Services enablement necessary to extend the product in such an extensive manner. Release 10 was delivered in 2004, as was a separately packaged DataWindow .NET product. The DataWindow feature was brought to .NET in parallel with and as part of the PowerBuilder 10 release in addition to being available as a separate product. The development of PowerBuilder 11 was reliant on Microsoft technologies that took longer to ship, leading Sybase to ship the 10.5 release as a packaging of all the scheduled improvements that did not rely on the new Microsoft technologies. With release 10.5, ADO.NET support was introduced to allow consistent database access to the major databases from within the managed code environment. Finally, the 10.5 release completed and further refined the Web Services support introduced in the earlier phase of the .NET alignment, namely release 9.

PowerBuilder 11 and 11.5

In release 11, the bulk of the phase 3 .NET alignment was rolled out in the form of the ability to build applications and components in PowerBuilder and deploy them to the .NET Framework version 2.0, including the creation and deployment of .NET Web Forms, Windows Forms, and Non-visual Objects (NVOs) as .NET assemblies and as .NET Web Services. Additionally, the ability to call .NET classes from PowerScript code and debugging support for .NET objects in the native .NET debugger were added to allow PowerBuilder developers to leverage many of the assets in the .NET environment. PowerBuilder 11.5 brought security capabilities by allowing PowerBuilder applications to run in partial trust environments when they are constrained by .NET code access security (CAS) configurations. PowerBuilder lets developers configure CAS zones (sandboxes) for .NET Web Forms, .NET Web Services, .NET Windows Forms, and Smart Client projects to minimize the amount of trust required before application or component code is run by an end user. Finally, .NET assemblies created in PowerBuilder 11.5 are run with the security permissions of the calling application or component instead of the full trust required in prior releases.

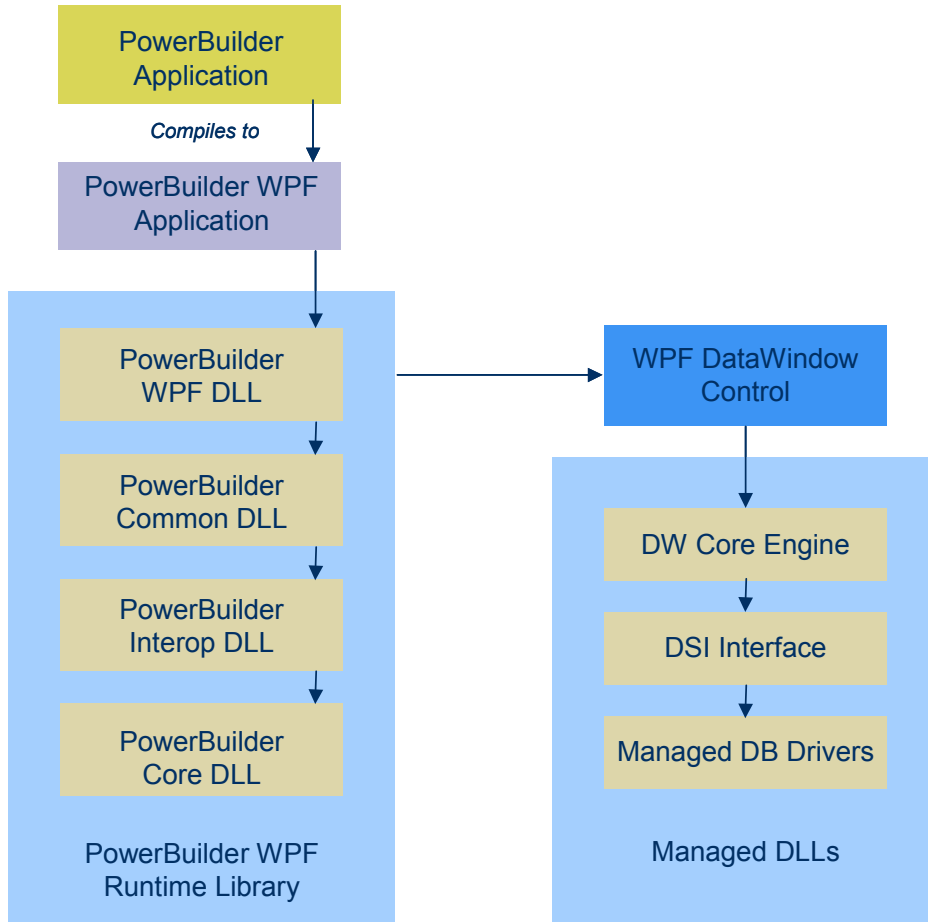
PowerBuilder 12

For release 12, Sybase has entered into a strong partnership position with the Microsoft Visual Studio division to complete the alignment of PowerBuilder with .NET. This effort comprises the following aspects:

- ☒ **PowerScript Features for .NET.** Leading the set of promised additions to PowerBuilder 12 are language modifications that permit many .NET features to be leveraged from within the PowerScript language. This can be preparatory work for snapping PowerScript to the internal MSIL so that PowerScript can be a true .NET language. Examples of the new language features promised include support for unbounded multidimensional arrays, .NET delegates, parameterized constructors, and user-defined enumerations, among others. Other capabilities promised include defining and consuming custom attributes, defining namespaces, and consuming .NET generic types from within PowerScript.
- ☒ **.NET WPF Support.** Another important architectural change that will permit PowerBuilder to snap to .NET technologies is support for the Windows Presentation Framework (WPF), which provides a powerful abstraction for rich vector-based graphics. WPF was introduced with .NET 3.0 and has become the basis of Microsoft's strategic work in the presentation layer, including for the support of rich Internet application (RIA) development through the Silverlight browser plug-in. WPF unifies many disparate services such as 2D, 3D vector graphics, animation, typography, audio, and HD video under a new comprehensive markup-based user interaction framework. Sybase hopes to complete the foundational work on WPF support in PowerBuilder in version 12 in preparation for Silverlight support soon after. WPF relies on an XML-based markup approach internally, which is not necessary for developers to learn, but support for WPF within PowerBuilder is likely to further abridge the learning curve for .NET technologies in general. It is important to note that PowerBuilder will continue to support the Win32 API in addition to WPF.
- ☒ **Managed Code DataWindow.** PowerBuilder 11.x applications today implement the .NET-managed code runtime for virtually all components. The main exception is the DataWindow engine, which runs on native code, creating some restrictions and annoyances such as the specification of DataWindow DLL files and the inability to run in partial trust environments. For PowerBuilder 12, Sybase is rewriting the DataWindow engine in C#, including a new managed code database driver. The overall resulting architecture is represented schematically in Figure 1, which also highlights an important aspect of the new DataWindow architecture, namely the design around separation of concerns where core functions are separate from both user interfaces and data source interfaces.

FIGURE 1

Supporting .NET in the PowerBuilder Runtime



Source: Sybase, 2009

- ☒ **Visual Studio Shell.** The fourth major area of R&D in which Sybase is engaging is a new IDE for PowerBuilder that is based on the UI innovation built into Microsoft Visual Studio. Visual Studio is acclaimed in developer circles as having a powerful yet highly productive IDE, which several million developers have enjoyed. Microsoft's investment in R&D over multiple iterations of Visual Studio will be fully leveraged by Sybase through the adoption of the Visual Studio isolated shell as the core new shell for PowerBuilder developers. Sybase is working closely with Microsoft as a Visual Studio Industry Partner to ensure that this is implemented in a reliable and expeditious way. The isolated shell approach implies that PowerBuilder can provide a high level of customization of the shell to the needs of PowerBuilder developers without bringing unnecessary functionality. It also means that PowerBuilder can run side by side with a Visual Studio installation without any conflict. IDC expects this new capability to make PowerBuilder attractive to existing .NET developers looking to supercharge their productivity in developing business applications.

Modernization of PowerBuilder Applications

In its chosen course of evolving PowerBuilder, Sybase has demonstrated an uncommon level of responsiveness to its customer base. The Sybase PowerBuilder customer advisory program is one of the better-run programs in the industry, with a standing waiting list for membership. Sybase has done considerable due diligence on the direction it planned to evolve PowerBuilder, canvassing and dialoging intensively with its customers prior to settling on its chosen course of evolution. This quality, which appears to be a shared cultural norm inside the overall Sybase organization, has been crucial in Sybase's overall revival as a company and bodes well for the success of PowerBuilder's evolution and road map.

Once fully executed, the overall approach will result in applications that are aligned with .NET and Microsoft's development strategy, yet are in perfect alignment with the historical value proposition of the PowerBuilder toolset and runtime. Thus, all the needed technologies to build enterprise applications in the most productive manner will be offered "in the box," making minimal demands of custom integration on developers. PowerBuilder will continue to have the easiest approach to putting together database-oriented business applications that deliver high-performance database query and reporting through optimized native database drivers. When PowerBuilder 12 is released, existing applications are expected to be migrated in a controlled and relatively pain-free fashion to the .NET runtime, whereby developers will have the added freedom of integrating them transparently with other .NET modules written in C# or any of the .NET languages and leveraging the full .NET framework.

FUTURE OUTLOOK

As Sybase marches forward with the evolution of PowerBuilder, it is keeping an eye on the important trends in application development and deployment software, the IDC market in which Sybase books all of its software revenue.

CHALLENGES/OPPORTUNITIES

There are three fundamental challenges facing Sybase today with PowerBuilder:

- ☒ **Execution.** Sybase must execute with speed on its stated direction for PowerBuilder in its stated time frame and must produce a relatively complete implementation of its stated alignment with .NET. IDC's canvassing of the PowerBuilder user base shows that most continue to see value in their use of the tool but are under significant political pressure to justify the continued decision to evolve their applications along the strategic path outlined by Sybase. The stated direction along the .NET path has been very well-received and has helped in keeping many developers in the fold. There is, however, a palpable anxiety around the time frames promulgated and whether the implementation will be sufficiently complete to provide a viable evolution path for their application.

- ☒ **Alignment.** A very small subset of PowerBuilder developers has historically articulated a preference for a different alignment path for the toolset, such as with the Java ecosystem. This is often largely driven by portability concerns and dissatisfaction with the level of standards compliance and support offered with the .NET platform. Some of these concerns have been attenuated by Microsoft overtures to openness and open source, and yet other concerns are being swept along by the overall shift taking place toward applications architected toward a Web architecture. In this regard, PowerBuilder has to walk the dual path of supporting WPF/Silverlight for Web architectures as well as purer HTML/JavaScript/AJAX approaches and frameworks.

- ☒ **Perception.** Despite the progressive approach taken by Sybase in the evolution of PowerBuilder, some might continue to see it as a development toolset and runtime whose full value has peaked. This is a perception problem insofar as it makes it difficult to attract new developers even if existing developers stay in the fold. IDC sees that the best antidote for this perception issue is a more aggressive marketing strategy in alignment with new trends and technologies that are now pervading the application development space. Thus, finding alignment and integration with Sybase's mobility platform, offering a cloud services version of PowerBuilder, and more aggressively utilizing open source strategies might win Sybase more adoption from an entirely new and more pragmatic crowd of developers that will discover the productivity and leverage of development in PowerBuilder.

This strategic direction taken by Sybase will provide it with the opportunity to bring PowerBuilder into alignment with the modern needs of application developers. If marketed properly, PowerBuilder may emerge to have a renaissance that will take it through the next decade or two with considerable health and a possible expansion of its user base.

CONCLUSION

Trends come and go, but things that work outlive the moniker of legacy. While the competition for tools to build new applications continues to intensify, we find the strategic direction taken by Sybase to evolve the PowerBuilder installed base well thought through and in alignment with the needs of the majority of PowerBuilder developers. Importantly, in due course, it is likely to put PowerBuilder applications in great alignment with the Microsoft .NET platform and provide its developers with all the salient aspects of the successful and popular Visual Studio environment while retaining the productivity of development that is characteristic of PowerBuilder. The strategic direction currently being implemented marries the traditional advantages for which PowerBuilder has gained its fame with a successful and powerful ecosystem of software development tools that is being aggressively evolved on its own merits by Microsoft. This will allow PowerBuilder developers to leverage the power and cost-effectiveness of PowerBuilder with the flexibility and richness available in the .NET ecosystem. Much like the movie whose title we borrow in titling this paper, Sybase's strategic direction with PowerBuilder is on track to marry the past of client/server development happily with its future.

Copyright Notice

External Publication of IDC Information and Data — Any IDC information that is to be used in advertising, press releases, or promotional materials requires prior written approval from the appropriate IDC Vice President or Country Manager. A draft of the proposed document should accompany any such request. IDC reserves the right to deny approval of external usage for any reason.

Copyright 2009 IDC. Reproduction without written permission is completely forbidden.