WHITE PAPER

Sybase PowerBuilder: A Tradition of Productivity Evolved for Modern Developer Ecosystems

Sponsored by: Sybase

Al Hilwa

December 2011

IDC OPINION

Sybase PowerBuilder has remained throughout the years a bastion of application development productivity for a wide class of enterprise applications that combine database access with a highly productive graphical user interface (GUI). Revolutionary at its inception for defining and popularizing this specific genre of distributed application development, the PowerBuilder 12.5 development environment today combines functional maturity with aggressive modernizations that keep it a viable and compelling tool for tackling business applications. The strategic direction taken by the PowerBuilder team has meant that the product can now be considered equally viable for evolving existing applications and for developing new ones. In particular, Sybase PowerBuilder provides the following benefits:

- A mature and highly functional application development environment that is currently in its sixteenth major release. Since its inception in 1991, PowerBuilder has aggressively evolved to meet customer demands for new architectures and technologies.
- A highly productive development 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.
- An articulated strategic direction to snap PowerBuilder to the Web ecosystem through the support of HTML5 technology for GUI development that will make the product appealing to one of the largest and most diverse developer ecosystems.
- The solid backing of Sybase, a financially stable and healthy concern that has provided strong stewardship of the PowerBuilder asset with an intimate customer interaction model. The backing is made even stronger by the acquisition of Sybase by SAP, which, according to IDC, is the fourth largest software company in the world.

Global Headquarters: 5 Speen Street Framingham, MA 01701 USA P.508.872.8200 F.508.935.4015 www.idc.com

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

©2011 IDC

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 PowerBuilder 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 PowerBuilder road map enables PowerBuilder applications to be brought into alignment with the successful Microsoft .NET platform and the familiar Visual Studio IDE, including many third-party controls.

SITUATION OVERVIEW

PowerBuilder is a development tool with a storied history. It is recognized for bringing about a rapid application development (RAD) paradigm shift in the early 1990s that effectively created a revolution in the productivity of business application development. PowerBuilder blended scripting capabilities from the then 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 productive but not very scalable PC database products of that era. 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. PowerBuilder gained 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 give way to Internet architectures. 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 higher levels of abstraction 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 virtual machine–based Java platform and language followed by Microsoft's .NET framework and its languages. The virtual machine–based architectures of Java and .NET allowed them to offer the benefits of both 3GL and 4GL languages. Known as "managed languages," these development models took the edge off of some of the most painful aspects of 3GL development (e.g., memory management, type enforcement, error handling) by moving them out of programmers' error-prone hands and into the 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 and grew to become the most popular target of new application development tools.

Today, most strategic application development tools are aligned along one of three major ecosystems: the Microsoft .NET environment, the Java environment, and generalized Web architectures. Most application development tools and their runtimes, which were developed during the course of the 1990s, have found themselves taking this fork in the road to align with one of these three broader ecosystems. Sybase's strategy primarily aligns PowerBuilder with the Microsoft .NET ecosystem.

Where Is PowerBuilder Today?

As programming languages and application architectures have come and gone, we are hard-pressed to find an environment as productive to develop in as PowerBuilder. What's more, PowerBuilder has been evolved aggressively as the aforementioned 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 truly stands out even in a world where everyone is ardent about their favorite language or tool. This has translated into a level of stickiness in PowerBuilder adoption that also stands out in this market. 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 skill and 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 sixteenth 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. While 11.5 brought new GUI improvements such as RichText in the DataWindow for displaying columns and added 3D graphing capabilities based on the DirectX runtime support, release 12.5 delivers the most comprehensive support for .NET to date, including a DataWindow that was rewritten in C# and natively supports Windows Presentation Framework (WPF) as well as compliance with .NET Common Language Specification.

 ©2011 IDC

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, 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	.NET-based Web Services engine (Visual Enhancements Theme); large and small UI and core client/server enhancements, including TreeView DataWindow, RichText Edit Control
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 graphics enhancements, Oracle 11g and SQL Server 2008 support, improved graphing (Visual Enhancements Theme)
12	April 2010	.NET phase 4: WPF applications, improved DataWindow graphing, integration of Visual Studio isolated shell
12.5	September 2011	WPF DataWindow enhancement, WCF Web Services and PowerBuilder .NET assemblies as targets, Custom Visual User Object target as WPF User Control for use in Visual Studio solutions, REST client support, batch command processing, multithreading, .NET 4.0 support, Visual Studio 2010 isolated shell IDE

Source: IDC, 2011

Aligning with Modern Developer Ecosystems

In its pursuit of an open developer strategy and to embrace the emerging architectural shift toward the Web, Sybase initially built ties into PowerBuilder that aligned it with the Java ecosystem. In time, however, Sybase discovered that the PowerBuilder user base is more aligned and has considerably more overlap with Microsoft's developer community. As a result, Sybase chose to shift its PowerBuilder investment and began a trajectory of deepening alignment with Microsoft's .NET technology stack in a four-phase strategic plan involving four major releases (and two minor ones).

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 a significant 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 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/Smart Client, and .NET Assembly 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 entered into a strong partnership position with the Microsoft Visual Studio division to complete the alignment of PowerBuilder with .NET. Overall, the following major improvement areas are addressed in this release:

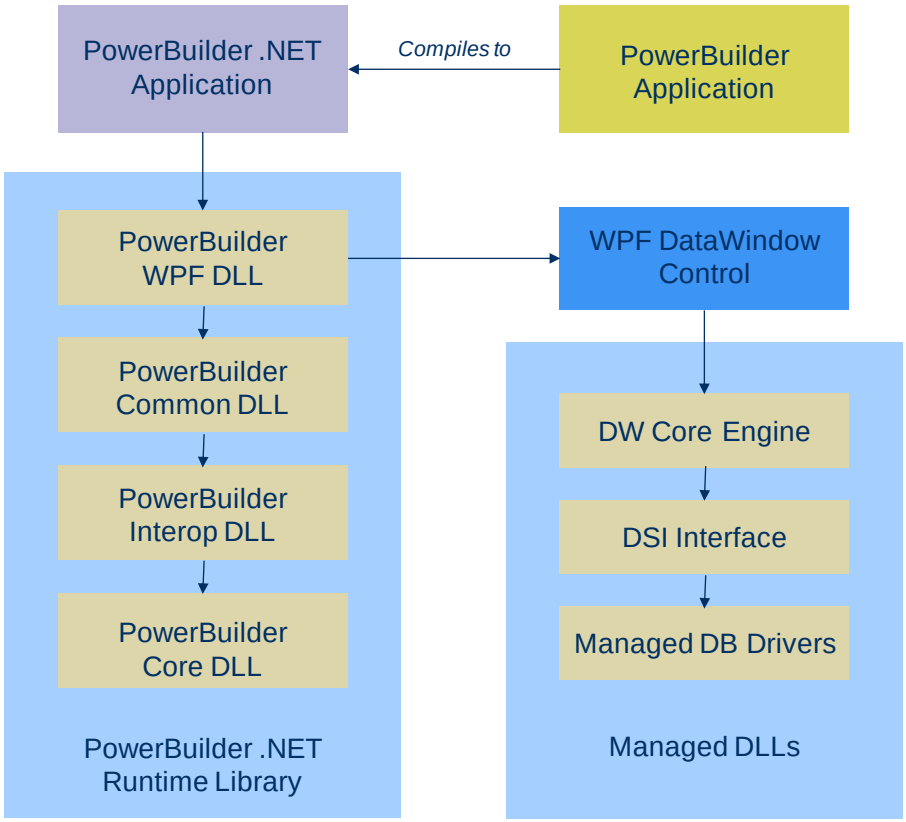
- **PowerScript Features for .NET.** Leading the set of additions to PowerBuilder 12 are language modifications that permit many .NET features to be leveraged from within the PowerScript language. Examples of the new language features include support for unbounded multidimensional arrays, .NET delegates, parameterized constructors, and user-defined enumerations, among others. Other capabilities include defining and consuming custom attributes, defining namespaces, and consuming .NET generic types from within PowerScript.

- **.NET WPF Support.** Another important architectural change that permits PowerBuilder to snap to .NET technologies is support for the Windows Presentation Framework, 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. WPF relies on an XML-based markup (XAML) approach internally, which is not necessary for developers to learn but which allows the presentation tier to be configured in isolation of the rest of the code. It is important to note that PowerBuilder and Microsoft will continue to support the Win32 API in addition to WPF and other new technologies.

- **Managed Code DataWindow.** PowerBuilder 11.x applications implemented the .NET managed code runtime for virtually all components. The main exception was the DataWindow engine, which was written in 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 has rewritten the DataWindow engine in C#, including a new managed code database driver. The 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, 2011

- **Visual Studio Shell.** The fourth major enhancement in PowerBuilder 12 is a new IDE that builds on the UI innovation and R&D 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 is fully leveraged by Sybase through the adoption of the Visual Studio isolated shell as the core new shell for PowerBuilder developers. Sybase has worked closely with Microsoft as a Premier Visual Studio Industry Partner (VSIP) to ensure that support for the shell and the .NET framework overall is implemented in a reliable way. The isolated shell approach implies that PowerBuilder can provide a high level of customization of the shell to suit the needs of PowerBuilder developers without bringing irrelevant functionality.

 ©2011 IDC

PowerBuilder 12.5

For release 12.5, Sybase brought many refinements to build on the architectural underpinnings in release 12, including updating various technologies and delivering many enhancements that the PowerBuilder user community had requested. The main enhancements in this release include:

- **Visual Studio 2010 IDE.** PowerBuilder 12.5 upgrades the IDE in PowerBuilder to the capabilities found in the Visual Studio 2010 Service Pack 1 isolated shell instead of the Visual Studio 2008 IDE used in PowerBuilder 12. This brings multiple monitor support and an overhauled help system to the PowerBuilder developer. PowerBuilder's isolated shell approach implies that PowerBuilder can run side by side with a Visual Studio installation without any conflict, a capability requested by many PowerBuilder developers. IDC expects the use of the Visual Studio IDE to make PowerBuilder attractive to a broader ecosystem of developers, especially those who are interested in integrating PowerBuilder code with code developed in Microsoft tools. In the latest version of the Script Editor, PowerBuilder 12.5 also brings considerably enhanced semantic checking and Find & Replace capabilities.

- **WPF DataWindow.** PowerBuilder 12.5 adds new capabilities to WPF DataWindow such as AutoWidth property, which allows grids to automatically adapt to the text in the columns in a configurable manner; Tab Order and Enabled support for noncolumn properties, which now allows mouse focus for most DataWindow controls; and enhancements to SRD syntax, child windows, and buffer debugging. A new CandleStick graph style is also added to support reporting applications.

- **New target types.** Several new application targets are introduced in PowerBuilder 12.5, including WCF targets, Custom Visual User Object (CVUO) targets (for use in Visual Studio C# or Visual Basic .NET solutions), and PowerBuilder .NET Assembly targets. Additionally, the WPF and Win32 application targets have received enhancements.

- **.NET 4.0 Framework support.** PowerBuilder 12.5 brings all the improvements and power of .NET 4.0 to PowerBuilder developers, such as improved security for .NET applications and improved exception handling.

- **REST client support.** PowerBuilder .NET targets can be clients that consume RESTful (Representational State Transfer) Web services, making it possible for PowerBuilder applications to support the compositional demands of new applications. These features are in addition to the support provided in previous releases for Web services using the .NET or EasySoap Web service engines. The overall effect of these interoperability capabilities is more integration options for PowerBuilder applications, presenting new paths for modernization and application evolution.

- **PowerScript language.** Capabilities for dynamically connecting PowerScript methods to .NET events have been added. The method could be a PowerBuilder global function, an instance function of a PowerBuilder object, or an instance or a static function of a .NET object. Additionally, the .NET Assembly is expanded to expose some PowerBuilder language elements for access in .NET application development. Another useful set of enhancements permits the Objects tab of the Project painter to expose language elements such as functions with generic types or delegates as parameters, parameterized constructors, events, indexers, .NET properties, and instance variables.

- **Multithreading.** The PowerBuilder .NET runtime now supports multithreading, thus addressing issues related to .NET thread locking and synchronization and allowing Shared Objects to be used to implement multithreading in PowerBuilder.NET.

- **Batch command processing.** Enhancements to support a command-driven automated build process are introduced, thus supporting the trend to incorporate the broadening use of continuous integration techniques in enterprise application development shops.

- **ASE 15.5 BIGTIME/BIGDATETIME.** PowerBuilder 12.5 also provides support for the Sybase Adaptive Server Enterprise (ASE) release 15.5 new time data types.

Modernizing PowerBuilder Applications

In evolving PowerBuilder, Sybase has demonstrated an uncommon level of responsiveness to its customer base. The Sybase PowerBuilder customer advisory board 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 course of action. 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. We expect this to continue under the similarly deliberative and customer-centric SAP, which completed its acquisition of Sybase in July of 2010.

Sybase's evolution of PowerBuilder will result in customer applications that are aligned with .NET and Microsoft's development strategy yet also leverage 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. With the release of PowerBuilder 12.5, developers will have the added freedom of integrating their applications with other .NET modules written in C# or other .NET languages and leveraging the full .NET framework.

FUTURE OUTLOOK

The developer landscape is in the midst of significant transformations as a result of changes taking place in mobile devices. Many developers have taken to developing mobile applications to harness the opportunities presented by mobile devices in the consumer space. Enterprises have taken notice, though they have been expectedly slow to embrace the mobilization of their application stack compared with the pace of change outside the firewall. IDC expects enterprises to engage in a massive long-term transition of their application portfolios on both clients and servers to adjust to these shifts in mobility and concomitant changes in back-end systems as they move to cloud architectures.

Industry Transformations

The transformational forces taking place in the industry have already precipitated a new developer strategy by Microsoft to align its Windows PC architecture with the mobile revolution. Microsoft's approach with the next version of Windows, known as Windows 8, is to add a new touch-style interface and to bring Windows to ARM chipsets like those found in smartphones and tablets. To support this new convergence of the PC and mobile devices, Microsoft has made changes to its strategic choices for languages and development environments. A perceptible alignment with Web technologies and HTML5 in particular occurred in the midst of the development of Internet Explorer 9 in the spring of 2010. By the fall of 2011, Microsoft's support for Web technologies in Windows 8 was unveiled and appears to be comprehensive, encompassing the JavaScript language as a first-class citizen for developing applications.

While Microsoft will continue to support existing applications and platforms in its Desktop style in Windows 8, it has chosen a specific set of strategic technologies for its new Metro style applications. In particular, Microsoft has reworked and streamlined many of the Windows APIs. Technologies such as C# and the CLR remain important for Windows 8, but the .NET Framework APIs are not supported in their current form. IDC expects .NET to continue to be strategic on the server side for Windows development and expects .NET applications to be supported for many years to come on the Desktop-style interface for Windows, but the alignment with the Web ecosystem of languages and tools is an important marker for Microsoft partners.

PowerBuilder Road Map

Sybase intends to continue an aggressive evolution of PowerBuilder by aligning it more decisively with the Web ecosystem. Among other investments, Sybase intends to bring Web capabilities to PowerBuilder through a DataWindow for HTML5 in the next major release of the product. IDC believes this alignment will be key to maintaining the viability of PowerBuilder for new applications. HTML5 tools are currently at an immature state in the industry. Today, HTML5 can be considered as a viable mobile target for Web sites targeting smartphones and tablets, most of which sport HTML5-capable browsers. IDC expects HTML5 to gain traction as more devices and PCs become HTML5 capable so that 90% of all mobile devices will be able to run HTML5 applications by 2013. HTML5-capable desktop browsers, which generally experience a much slower pace of upgrade, are not expected to

reach 90% penetration on PCs until 2015. It is expected that HTML5 will begin to be targeted as a front end for enterprise applications starting in the 2013–2014 time frame initially by aggressive organizations that run new browsers. Sybase should target that time frame for an HTML5-capable version of PowerBuilder.

CHALLENGES/OPPORTUNITIES

There are two fundamental challenges facing Sybase today with PowerBuilder:

- **Execution and road map.** Sybase has finally executed on its four-phase .NET strategy. However, the integration agenda can benefit from additional evolution to support HTML5 front ends expected to pick up traction for desktop applications in the 2013–2014 time frame. Sybase must also address the growing number of touch-oriented devices that are gaining some traction in the enterprise. While most existing touch-based devices are consumer oriented and not typical targets of PowerBuilder applications, the urgency to support touch may become palpable after Microsoft's Windows 8 ships in late 2012 and ISVs begin the process of supporting its touch interface. Sybase's road map should address the combined evolution to HTML5 and mobile devices.

- **Perception and marketing.** Despite the progressive approach taken by Sybase in the evolution of PowerBuilder, some might continue to see it as a development tool and runtime from the past. This is a perception problem insofar as it makes it difficult to attract new developers. IDC sees that the best antidote for this perception issue is continued investment and evolution of the platform and a more aggressive marketing strategy in alignment with new trends and technologies that are now pervading the application development space. Aligning PowerBuilder with new trends around mobile, cloud, and social might gain Sybase more adoption from an entirely new and more pragmatic crowd of developers that will find value in its productive approach to application development.

Sybase has been a strong and steady steward of PowerBuilder and in recent days has invested in marketing it more aggressively than at any time in the past decade. Twice as many people attended PowerBuilder sessions at the 2011 TechWave conference than at any TechWave event in the past decade. 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. The Sybase evolution strategy for PowerBuilder provides developers with the salient aspects of the popular Visual Studio IDE while retaining PowerBuilder's characteristic productivity of development. The strategy places PowerBuilder in the large and successful Microsoft ecosystem. PowerBuilder developers have also taken note of the transformations occurring in the industry, such as the explosion of mobile

devices and the shift to HTML5 for Web applications. In recent presentations and user discussions, Sybase has articulated a strategy to take PowerBuilder developers to the mobile and Web promised land. 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 2011 IDC. Reproduction without written permission is completely forbidden.