White Paper

PowerBuilder – Stay or not to Stay....

Why or why not to stay with PowerBuilder?
What to do, when the decision is to move away?

- Re-write or
- Buy a product from the market or
- Migrate
    - o Do it yourself
    - o Engage a competent vendor

Which ecosystem to choose : .Net or Java?
Why engage a vendor who migrates using an automated tool cum process?
What are the vendor selection criteria's?

"Change is inevitable because it is exponentially driven by knowledge and technology; it is accelerating at the speed of light. It is better to be an adopter of change rather than a victim of change."

Written by Ve.Ramesh Babu
Replacement Technology Practice
rameshbabu@zslinc.com

Audience - Individuals who are responsible for modernizing legacy applications or looking for gains in efficiencies and cost savings by moving to current, proven technologies.

The below flow diagram gives the essence of the entire document and this will help you to make a decision, whether to stay or not to stay in PowerBuilder applications and HOW?

PowerBuilder – to Stay or not to Stay… Considerations…

Why stay in PowerBuilder?

- Client/Server delivery model is good enough
- No big change in business requirements or workflow
- Unbeatable productivity for Client/Server applications
- Sufficient number of PowerBuilder resources available internally, because trying to find PB resources in the marketplace today is extremely difficult.
- Existing documentation to carry out required changes and enhancements is adequate

Facing issues like ...

- *Lack of PB resources*
- *No documentation for the existing PB applications*

Have alternate plan handy and trigger the migration task at the appropriate time

Engage suitable vendor to maintain PowerBuilder applications

Why move away from PowerBuilder?

- Needs of the Enterprise
- Change in business requirements
- Need to extend the business functionalities to vendors and clients
- Scale up the number of concurrent users
- Web and mobile integrations
- Cope with dynamic changes
- Provide enhancement and change in workflow
- Minimize the future risk
- Decreasing pool of experienced PowerBuilder developers

What to do next ?
Refer to section 4 for details…

- *Re-write from scratch*
- *Buy a new product from the market*
- *Engage migration experts/ company who can help by using automation tools and processes*

Choose .Net or Java
Refer to section 5 & 11 for details…

- *Depends on IT need*
- *One's development experience*
- *Internal skill sets of resources*
- *Corporate and political issues*

Select Vendor for Migration
Refer to section 13 &14 for details…

Vendor Selection Criteria

- *Garnered experience in PowerBuilder*
- *Good experience in .Net & Java*
- *Specific migration experience*
- *Proven automation tool & processes*
- *Quick effort estimation processes*
- *Vendor's risk management capabilities*
- *Appropriate security policies*
- *Service level agreement*
- *Fixed Bid and Fixed Duration*
- *Flexible engagement model*
- *Vendor background check*
- *Proper infra-structure*

Table of Contents

"One can assess the organization's business quality and performance by analyzing what applications are being used and how they are being put to use. It is inevitable that the enterprise's choice of applications and its amenability to desired changes will determine the performance of the enterprise"

- Ramesh

1. Do enterprises really need to move away from PowerBuilder?

PowerBuilder is one of the earlier tools, which enabled programmers to develop OOPs (Object Oriented Programming) based development with a wonderful IDE (Integrated Development Environment). PowerBuilder introduced to the market the concept of datawindow, which was a great help in the rapid development of data oriented business applications. In fact, datawindow was a great invention of the 90's for the programming world and engraved inside the window helps in handling huge amounts of data. Datawindow as a single control helps to capture data, generate reports in various styles and formats and also allows rendering graphical representation of the data.

PowerBuilder has been a popular choice for the development of Client/Server applications for the last two decades and it is one of the best time tested tools for developers. It is a robust, innovative and proven technology, delivering unbeatable productivity by extensively supporting object oriented features like inheritance, encapsulation and polymorphism; PowerBuilder provides a rich UI in building Client/Server applications. The strength of PowerBuilder lies in its superior development IDE and its support for any highly event driven programming model. PowerBuilder delivers high-quality, robust applications efficiently and more effectively than most other development tools. PowerBuilder adequately meets all the requirements that it was originally designed for. The enormous benefits offered by PowerBuilder include the life it induces into business applications, especially the great end user experience.

When .Net and Java emerged as two major ecosystems and standards for developing enterprise's applications, PowerBuilder started waning out from the developer community due to a lack of a sustained marketing approach by Sybase.

Despite PowerBuilder being an exceptional language, its proprietary nature and non- standard web deployment platform as well as the lack of PowerBuilder resources are forcing many companies to plan for strategic alternatives.

There are hardly any tutorials, training institutes, sufficient learning materials or documentation available for PowerBuilder and as such, there is very little awareness among the IT newcomer for learning PowerBuilder. Many corporate executives are quite perplexed about the future of PowerBuilder; they believe that the language is completely dead. The forces of the market have caused IT executives to contemplate whether to stay with PowerBuilder or move away from it entirely. Enterprises eventually will have to make a decision on whether or not to stay with PowerBuilder depending upon their appetite for risk and what level of jeopardy the current environment puts them in.....will a move to another platform immediately minimize the risk?

To cope with the dynamic changes of ongoing business requirements put upon companies today and to stay abreast the current market trends, it is necessary for enterprises to fully embrace the change and make the necessary shift.

Sybase's current strategy of allowing the PowerBuilder code to be deployed into the .Net platform shows that PowerBuilder is aligning towards .Net. Though, the deployable code is in .Net (compiled form) and the actual code development and modifications still need to be done in PowerScript, none know whether this strategy will stand the test of time; it is only a question of time before PowerBuilder script may be consigned to history.

"Any solution to a problem changes the problem"

- **R. W. Johnson**

The Visual Studio suite already includes three development languages, VB.Net, C#.Net and ASP.Net that more than adequately provide developers with the tools needed to quickly and efficiently write the needed applications desired by the enterprise. Does it make sense to also embed PowerBuilder? This would add additional cost to the enterprise as it relates to added license cost not to mention the maintenance of two different environments, standards and scripts.

A more recent event has now raised additional questions regarding the viability of PowerBuilder, the acquisition of Sybase by SAP. Will PowerBuilder end up the same way that Visual FoxPro has....the stoppage of development and support for this language by its acquirer, Microsoft? It is impossible to predict the future of PowerBuilder and only time will tell whether this tool will continue to be viable into the future.

In my opinion, Sybase should focus only on Datawindow development and enhancements addressing both ecosystems; .Net and Java. This approach could help Sybase to focus on development and marketability of patented Datawindow technology for delivery to various platforms and models. This type of focus most likely would help developers to embrace Datawindow technology, which would then enable Datawindows to stay alive into the distant future.

2. Why enterprises could stay with PowerBuilder?

Though PowerBuilder may be seen as an antiquated tool, it has not reached its end-of-life and it is still alive as a niche player in the marketplace. There is still an astonishingly high demand for PowerBuilder resources. PowerBuilder has established itself as the king of the Client/Server market; it is an incredible tool to work with and an excellent tool for Rapid Application Development as well as for building CRUD applications very quickly. Nothing could beat PowerBuilder when it comes to building rapid and robust Client/Server applications for any business domain. PowerBuilder is a powerful, easy- to-use scripting language thanks to its patented "datawindow technology", but it has failed to sustain as an independent language for the delivery of applications needed by organizations today; requirements such as application functionality on handheld devices the web, mobile, etc.

It still makes sense for an enterprise to stay in PowerBuilder – provided the enterprise:

1. Is not foreseeing any need to change or to extend the business functionalities to the web and the business requirements are confined to the client/server platform.
2. Has enough internal PowerBuilder resources to maintain and manage the existing PowerBuilder applications.
3. Corporate policy is to outsource or hire external resources to maintain and manage the existing applications.
4. There is sufficient documentation on the business functionalities and workflow.

However, it is strongly advisable to chalk out a clear strategy and backup plan to migrate from PowerBuilder applications to one of the standard development platforms. This will bring long term benefits and further insurance to the enterprise.

"Technology that has been deemed to be the best need not be the correct technology choice for a particular organization; the right solution must be aligned to the business requirements and is highly dependent upon the capabilities of the individuals implementing, supporting and maintaining it."

- Ramesh

3. Why enterprises need to move away from PowerBuilder?

With Microsoft's .NET and Java technologies flourishing in the marketplace today, it has directly and in- directly caused languages like PowerBuilder and Delphi to lose their market share. The proprietary scripting language, lacking model driven design and approach (though Sybase tried in various ways to deliver this, which has invariably failed to sustain in the market space) as well as having a steep learning curve makes the whole learning process difficult for new developers; it has been a niche language for a long time now. In order to extend the business functionalities to vendors and clients, or scaling up the number of concurrent users, or in providing web and mobile integration, or to overcome technological obsolescence and to ensure future profitability with low total cost of ownership, many enterprises are looking to move away from PowerBuilder.

The major reasons to consider moving away from PowerBuilder are:

- Scarcity of PowerBuilder professionals - very few new developers are getting the opportunity to learn and most are not willing to spend money and time to learn PowerBuilder. There is little to no awareness among the developer community about PowerBuilder along with the lack of long term opportunities.
- If the enterprise's application functionalities need to be extended to web, mobile and/ or to expose those to external application and users.
- Needs related to application level integration with other enterprise applications; let it be vendor or client applications.
- Sybase itself is advocating and aligning PowerBuilder delivery platform on .Net

4. Various approaches to move away from PowerBuilder

4.1 Comparison – Re-write vs. buy new product vs. tool based migration

	Re write	Buy New Product	Tool Based Conversion
Risk	Riskiest, capturing all the business functionalities would be a • Herculean task • Riskiest, because it required recreation of embedded/ undocumented functionalities	Second riskiest, if the correct product is not chosen, business workflow will be affected; this may result in business loss. Also getting the mind share of the user to use the new system would be another difficult task	All business knowledge and workflow is preserved; user community virtually Unaffected
ROI	Lowest, because it takes maximum development cycle	Medium -depends on the product & vendor and also on the implementation cost	High

"In the context of SOA, Gartner states that, "there is no better environment in which to source services than the installed base of working, time-tested business functions implemented in legacy systems." Vecchio, Dale, "CIO Update: Extend Mainframe Life by Adopting a Service-Oriented Architecture," **Gartner, March 19, 2003."**

Cost Effectiveness	High cost	Medium cost	Very low cost
Time Consumption	High	Choosing the right product and implementation is extremely time consuming	Low
Learning Curve	Long Learning Curve	Very long learning curve as well as need to align enterprise processes towards the product for better results	No learning curve
Project Cost	Very High, need to invest in; • Domain expertise • Developer & Team • Project Management • User Re-Training	Very High • Evaluation Costs/ Time • Product implementation • Data mapping & Conversion • User Re-Training	Low • Most of the code is automatically generated • No need for re-training the user
Application Usability	Excellent, though it takes time to settle down and put in actual use	Good, provided the product meets all business requirements	Excellent and exactly aligns with your business requirements
Maintenance	Normal, after it gets settle down	Only with the product vendor	Normal
Resultant code	Very effective resulting code, which may exactly match the target environment.	No source code, highly depends on the product vendor	The resulting code is generated automatically, which is human readable & manageable.
Performance	Excellent	Depends on the product vendor	Excellent
Documentation	Need a special focus to create all required documents, including system and user manual etc.,	Comes with product manual and user guide.	User manual may not be necessary. However, it is advisable to create required document at the time of migration.

"Composite applications have emerged as an essential architectural pattern for business process fusion and other business strategies... Well-behaved composite applications are based on a classic, logical three-tiered architecture comprising presentation, business and data logic." Pezzini, M., "Composite Applications Help Turn Legacies Into Assets," **Gartner, October 8, 2003."**

5. Where to go .Net or Java?

5.1. Two entirely different ecosystem

It would be very difficult for anyone to say that one technology is better than the other; both the technologies have their own pros and cons. The information that follows is intended to help bring a better understanding of the two technologies to the reader and clarify questions he/she may have.

5.1.1. Benefits of using .NET

.NET framework is an application development platform from Microsoft that provides the easiest, most productive set of languages and tools to rapidly build windows and web applications. .NET comes with enhanced visual designers which increases the application performance and a powerful integrated Visual Studio development environment (IDE). It also supports the creation of applications for wireless, Internet-enabled hand-held devices, etc.

1. .NET is a fully Object-Oriented language
2. Powerful, Flexible, Simplified Data Access
3. Automatic memory management, also known as garbage collection, in .NET runtime, keeps track of the memory that a program allocates and can release that memory when the program no longer needs it
4. .NET security is designed to address some of the vulnerabilities, such as buffer overflows that have been exploited by malicious software. Additionally, .NET provides a common security model for all applications
5. Visual Studio IDE helps developers to write programs faster and more efficiently. With an improved integrated development environment (IDE) and a significantly reduced startup time, .NET offers fast, automatic formatting of code, improved IntelliSense, an enhanced object browser, XML designer and much more
6. A powerful development tool for creating windows and web-based applications
7. Easy and simple deployment

5.1.2. Benefits of using Java

Java is a high-level object-oriented programming language developed by Sun Microsystems for developing web applications. Though developing Client/Server applications are doable using Java, it is not as easy as developing in 4GL's like PowerBuilder, VB .Net or C++, all of which have great user friendly interfaces.

1. Write Once, Run Anywhere (WORA)
2. Lots of open source frameworks are available to choose from
3. The Java Database Connectivity (JDBC) API is the industry standard for database-independent connectivity between the Java programming language and various databases
4. Java has effective virtual memory management
5. Java has a large set of tools and implementations of commonly used API's for commonly used security algorithms, mechanisms, and protocols

"We are prepared to take risks, but intelligent risks. The policy of being too cautious is the greatest risk of all"

- Jawaharlal Nehru

5.2. Comparative Analysis between .NET and Java

	.Net	Java
Development environment	Visual Studio is a much easier and efficient tool for developers to use	Many IDE, for example: JDeveloper, Eclipse, Velocity etc., difficult to choose one over the other
Developer Productivity	Very good	Dependent on the efficiency of the tool and framework being used
3rd party frameworks	Telerik, Infragistics - supported by the respective vendors	100+ frameworks are available, again difficult to choose from, example: Struts, Spring, Stripe Iceface, JSF, Wicket, Tapestry and many more. ADF-Application Development Framework, extension of JSF, developed and supported by Oracle and it is very promising
Maintenance	Fairly easy and more resources are available	Depends on the technology and framework that you use.
Deployment	Only in Microsoft Platform	Multiple Platform
Upgrade path	Dependent on Microsoft	Though it looks like open source, but dependent on the respective open source community and framework developers. If you choose wrong path, leads to disaster
Standards	Defined by Microsoft	Defined by Sun initially, but now by open source community and the respective framework community. What will now happen based upon Oracles acquisition?
Scalability	Good	Good
Interoperability	Good	Very good
Security	Good	Good
Supported Languages	Though it supports VB.Net, C#.net and Asp.net as the language to be used to write code in VS, but you can deploy literally any language in .Net framework using CRL	Only JAVA, though it claim to be supporting other languages.
Training	.Net requires minimum amount of training to be proficient using the tool	Java requires a substantial amount of training in order to be proficient in using the tool for actual developmental needs
Resource Availability	Experienced resources are easily available	Getting experienced resources to develop the applications on a specific tool and framework is a herculean task in Java, and most of the time the client needs to compromise on the quality of the resources; also the developers need additional training on the chosen tool and framework before starting the actual development

"A legacy application has been with the enterprise longer than the programmers who are now maintaining it, lacks good documentation, and has untouchable code"

- Joe Celko, IT Writer

6. Synergy between PowerBuilder and .Net

PowerBuilder is an excellent development environment; in fact it is one of the easiest tools for the development of any application. Its robust IDE, easy-to-read PowerScript and sophisticated debugging makes it a powerful development tool.

The new PowerBuilder version12 is in perfect sync with Visual Studio enabling the delivery of more sophisticated business applications, let it be WPF, plain window application or web applications. It is a perfect marriage between PowerBuilder and the .Net environment, as the PowerScript is delivered in .Net environment. The only apprehension is how the developer community will accept this. The power of datawindow is now available for .Net developers too; it is a single environment that could be used by both PowerBuilder and .Net developers. In fact, .Net developers could use the features of datawindow to develop applications quickly.

Datawindow is one single all-in-one component that delivers various controls like Grid (much better than native .Net Grid), graphs, reports etc.

7. Synergy between PowerBuilder and Java

In fact, a few years prior, Sybase was supporting the deployment of PowerBuilder components (PBNVO) as CORBA in multiple application servers, starting from its own EA Server. Further to this Sybase was positioning to move forward in the direction of JAVA providing support for deployments on WebShere and Weblogic via a plug-in to either of these application servers. Ultimately there was very little support in this direction by the developer community and as a result was not a successful strategy and at this point Sybase realigned itself with .NET as the preferred platform for PowerBuilder deployments.

8. Where is Sybase / SAP steering PowerBuilder

In the beginning Sybase tried to align PowerBuilder with Java's world by adopting various strategies and approaches, but the market did not buy the idea. Now, it is indeed moving PowerBuilder in the direction of .Net. PowerBuilder dramatically simplifies .NET development and is a part of Sybase's plan to deliver robust support for the .NET framework; Sybase provides the flexibility to enable PowerBuilder application deployments in the .NET environment. In support of this direction, Sybase has announced a new version of PowerBuilder 12.0 that completely supports the .NET framework and deploys managed code in a compiled form. It is always feasible for a PowerBuilder user to adapt to .NET technology.

Enterprises need developers with dual skill sets; they are looking for ways to merge the past with the future. The easiest and most cost effective migratory path for companies would be to train PowerBuilder users to be proficient in the .NET environment. Successful businesses are always aspiring to achieve the best value for time and money, and moving off of a legacy application could be achieved more efficiently by providing a migration path to the latest .NET environment for existing customers.

"People associate the term legacy with big iron and Big Blue, but the phrase is increasingly being used to include any and every application in existence before the birth of the Web."

- Sarah L Roberts-Witt, Writer on Internet infrastructure and services

9. If PowerBuilder 12.0 code is deployable in .Net, then where is the catch?

In the new PowerBuilder version 12.0, the PowerBuilder code can be deployable in .Net environments, however, that does not mean that the existing PowerBuilder applications from lower versions can be deployable straight away in the .Net environment.

Applications developed in older versions of PowerBuilder need a great amount of effort and tweaking to fit into the required format for deployment in the .Net environment.

However, the additional catch is that even PowerBuilder 12 does not allow the developer to edit or make modifications directly into any of the .Net languages like VB.Net, C#.Net or Asp.Net. The code has to be re-written in PowerScript and this leaves little to no room to manage the developed codes directly in the .Net environment which makes the developer's life much more complex in the event of multi-tier or model based developments.

It is advisable to spend the effort and cost to migrate PowerBuilder directly to .Net rather spending the same or more effort and cost in tweaking the PowerBuilder code to deploy in .Net environment.

10. Is there a third ecosystem?

Apart from Java and .NET technologies, there is another open source community that advocates strongly the use of PHP. Though many advantages are assigned to this technology, the development of enterprise business applications on this tool is a daunting task and is outside the scope of this document.

11. Decide which ecosystem to fall into...

11.1 Corporate Policies and Political pressures

It is imperative to consider the following items when contemplating the migration of legacy applications.

1. Has your current technology reached the end of its lifecycle and no longer has support from the technology vendor?
2. Will the new technology facilitate your corporate goals and growth?
3. Will the new technology bring substantial ROI?
4. Will this new technology save on AOC - "Application Operating Cost"?

Regardless of whatever Legacy migration solution is chosen, it not only needs assurance from the higher authority in the organization, but also needs to have cooperation and mind share of the team that owns the application. The decision to be made also needs to be consistent with the organization's policies in order for the project to be successful. Some corporate policies support more than one technology for migration while some other organizations are inclined towards specific technology.

"There is a strong correlation between application quality and the cost of maintenance. Hayes, Ian S., "Software Metrics: What Do They Really Mean?"

Aeurbach Publications, Handbook of IS Management

An organization's decision-making might seem to be a rational process, but rather, it is also a political process. An organization's decisions range from autocratic to democratic extremes and decisions not only involve conflicting interests but are also driven by one's self-interest and the comforts of a particular technology that he/she has experience with. Corporate policies and politics as well as one's self interest play decisive roles in the determination of the migration path that an organization ultimately moves forward with.

However, corporations need to look at technology aligning with the business requirements and road map rather than aligning the business with technology.

11.2. Aligning with internal skill sets

1. The skill set of resources become an important factor for migrating applications.
2. What would be the additional investment in manpower in the absence of skill sets / resources on new technology?
3. What type of investment does an organization make in domain consultants?
4. Are the IT staff and end-users fully re-trained on the new environment?
5. The skill set of resources becomes the deciding criteria for legacy migration to the desired target platform.

12. Are there any tools available to migrate PowerBuilder to .Net or Java 100% automatically?

There are no tools which can 100% automate the process of migrating PowerBuilder applications to .Net or Java.

12.1. GUI convertors

There are a couple of companies who offer the services of converting all the GUI's of PowerBuilder to XML/DHTML's. After this process the client needs to write the code behind the GUI's, which may not be of a great help for the migration process.

12.2. Only Business Logic conversion

Some vendors offer to create code snippets based on PBNVO which do not solve the problem for the client. Even for testing these converted codes, clients have no front end and hence have to migrate all the balance portions of the application before testing through the front end. This results in a high amount of risk and it is very difficult to quantify or identify the errors. It would be a matter of accountability and responsibility of who has to fix the errors and within what time frame etc.,

In PowerBuilder code, most of the time the business logic is glued along with the front end; difficulties are faced in the separation of the UI from the business logic, leading to failure in migrating from the legacy system. This leaves the vendor, who offers to migrate only the business logic to the target platform, in vain.

12.3. Tool-cum –Service based migration approach

A tool based migrating service leaves the entire end to end migration task to the vendor, whereby the vendor takes on the entire responsibility and is accountable for delivering the newly migrated application following rigorous testing in an environment that is specified by the client.

"In people's handling of affairs they often ruin things when they are right at the point of completion. Therefore we say "If you're as careful at the end as you were at the beginning you'll have no failures"

- Lao-Tzu, Te-tao Ching

A vendor who has experience in .Net or Java alone would not be able to migrate the PowerBuilder application successfully to the required target platform. It is inevitable that the vendor must also have a great knowledge and expertise in PowerBuilder as well.

In most cases, clients may not have any documentation inclusive of user manuals and in such cases the business logic and workflow are in the form of PowerBuilder code which needs to be extracted with the help of a tool and or with the help of a PowerBuilder expert.

13. Which solution should one choose?

13.1. Tool based service vendor is the best choice

Compared to a re-write, a tool based migration solution should provide a quicker and substantially larger ROI to the enterprise without compromise to the quality or performance of the code; in fact it should ensure that ALL the business logic is retained as well as improve the performance.

The following items are critical criteria to consider when deciding on the proper vendor to work with:

- The tool migrates the PB Client/Server application to the web with a minimal effort.
- It automatically generates the script for GUI and business logic, with the same look and feel of the legacy application. Alternatively, changes to the GUI for a new look and feel could be accommodated without losing the current application workflow.
- Retained business logic from legacy application
- Refactoring
- Flexibility in target architecture
- Negligible learning costs
- Reduced Maintenance, Support and Enhancement Costs
- Faster time to market for new initiatives
- Cost and Time savings over a complete re-write of the application

13.2. End-to-End responsibilities

Many enterprises are increasingly looking for a vendor to take on the responsibility for an end-to-end solution of migrating legacy applications to the standard platforms. End-to- end responsibility should also include the added value service of incremental improvements to the functionality of the application post migration providing numerous additional benefits to the client.

13.3. Post Migration Support

Post Migration, the deployed application needs to function effortlessly without using the automation tool or support from the vendor. If any functional changes are to be carried out in the code, it can be done directly in the .NET code. The programmers from the client's organization should be proficient enough to make these changes themselves without the assistance of the vendors migration experts; the client will have all the required documentation etc. from the vendor.

Enterprises might contemplate engaging the vendor company for the post migration maintenance of the application for a specific period of time. This might provide the client with additional room for negotiations with the vendor as well as provide a comfort level that the migrated application will perform as prescribed.

"Software application assets are unlike other fixed assets (machinery, furniture) which depreciate over a period of time nor like land and buildings which appreciates in most instances. Software applications are the life line of the enterprise which helps to determine how the business is being conducted. The knowledge assets of an organisations are comprehensively captured in a business logic repository otherwise known as "the application."

- Ramesh

14. Vendor Selection Criteria

- Garnered experience in PowerBuilder
- Good experience in .Net and / or Java
- Specific migration experience and good references
- Proven automated tool & processes
- Quick effort estimation processes
- Flexibility in target architecture
- Vendor's risk management capabilities
- Appropriate security policies
- Service level agreement
- Fixed Bid and Fixed Duration
- Flexible engagement model
- Vendor background check
- Proper infra-structure

15. Conclusion

In conclusion, migrating from PowerBuilder to the standard environment as required by today's business needs, an enterprise should enter into an introspection of the business needs, its resources, requirements, time to deploy and future scalability etc.

Choosing the right vendor who can shoulder these responsibilities by providing an end to end solution in the shortest time at the perfect price is the need of the hour today.

The vendor should be versatile in both language environments.

It should be a vendor who continues to provide a meaningful value addition to the legacy systems functionality and who engages in continuity in the form of maintaining the same over the future life span of the system. This should be the enterprise's aim.

About ZSL's Replacement Technologies Practice – *is a formidable practice within ZSL that has been in existence for more than a decade. Year after year the company has invested substantial R&D dollars in tools to help automate the migration process used by companies to convert their old legacy applications to more modern technologies. ZSL continues the development of it's proprietary conversion tool kit, PowerMigrator, which has helped numerous companies automate up to 70% of the effort needed to convert old legacy code to either the .NET or J2EE platform while still retaining the business logic of the old application.*

I would like to thank my colleagues and friends who helped me in making this whitepaper possible. My special thanks to Jim and Dr. Arasu.

Disclaimer - the information and the content in this whitepaper is the perception of the author and arrived at based on a study and understanding of the marketplace. The author is not responsible for any damage or loss based on the decision made by the recommendations or understanding from this document.

rameshbabu@zslinc.com

www.zslinc.com/rtp