

I D C T O P L I N E

4GLs: More Prevalent Than You May Think

January 2007

Information adapted from *IDC's Software Developer Collaborative Study*

Sponsored by Four J's

A 4GL Heritage

As a general category, fourth-generation languages (4GLs) remain an important component of the overall development landscape, garnering a respectable share of use. 4GL programming languages are specifically designed to facilitate the rapid development of business applications by abstracting away from the user the need to know and apply lower-level programming and system commands. These frameworks can help reduce complexity and promote consistency across the development effort, in turn allowing for faster time to market with quality software applications. Many mission-critical business solutions running today have been built and rely on 4GL technology. Some tools that support these languages have persisted for decades, evolving with market demands and advancing to address the various types of technologies that interface with these application environments.

4GL tools provide frameworks that let users enter commands using simpler, more business-oriented terms, with facilities to rapidly design and generate screens and reports. These features, and others, are geared to ease access and use of specific databases and client interfaces, along with query language and integrated database management capabilities. Most of these technologies have expanded to provide richer GUI support, interactive validation and debugging services, and performance enhancement mechanisms. 4GLs, and most rapid application development (RAD) tools, are typically more "declarative" in nature. To be considered a 4GL, a tool must provide a higher level of abstraction than a 3GL, afford dictionary-based application development, and perform semantic consistency checking.

The utility of 4GLs goes beyond their multitier architectural support to allow users to readily define functions, methods, and structures. Some 4GL unified development environments (UDEs) can provide functionality to help integrate and orchestrate an ever-increasing variety of clients (browsers, rich clients, mobile clients), servers (Web servers, application servers, and database servers), and user-defined processes. Consequently, such capabilities can bring 4GLs well beyond their classic client/server roots by providing an infrastructure layer that can address a myriad of legacy applications, databases, and servers.

When the term 4GL is mentioned, most individuals typically think of legacy 4GLs; however, many common development environments have a heritage in 4GL roots, including Visual Basic, Delphi, and PL/SQL. Legacy 4GLs are early abstracted development environments that were marketed well before the advent of visual programming interfaces. Yet, despite the antiquated nature of their original environments, many of these tools remain in use today, having evolved with market demands. Gone are the days when these tools merely provided a compiler and a set of runtime libraries. 4GLs that were originally designed to address a single proprietary hardware and database platform have advanced to incorporate support for multiple deployment scenarios and thus can bridge varied generations of technology and provide more robust infrastructure and provisioning capabilities. Indeed, certain 4GL vendors have systematically embraced a cross-platform philosophy, expanding capabilities to address industry standards such as XML and Web services to assist in interfacing these 4GL solutions with service oriented architecture (SOA) environments. Such advances have made these technologies an invaluable resource to a diverse and loyal customer base.

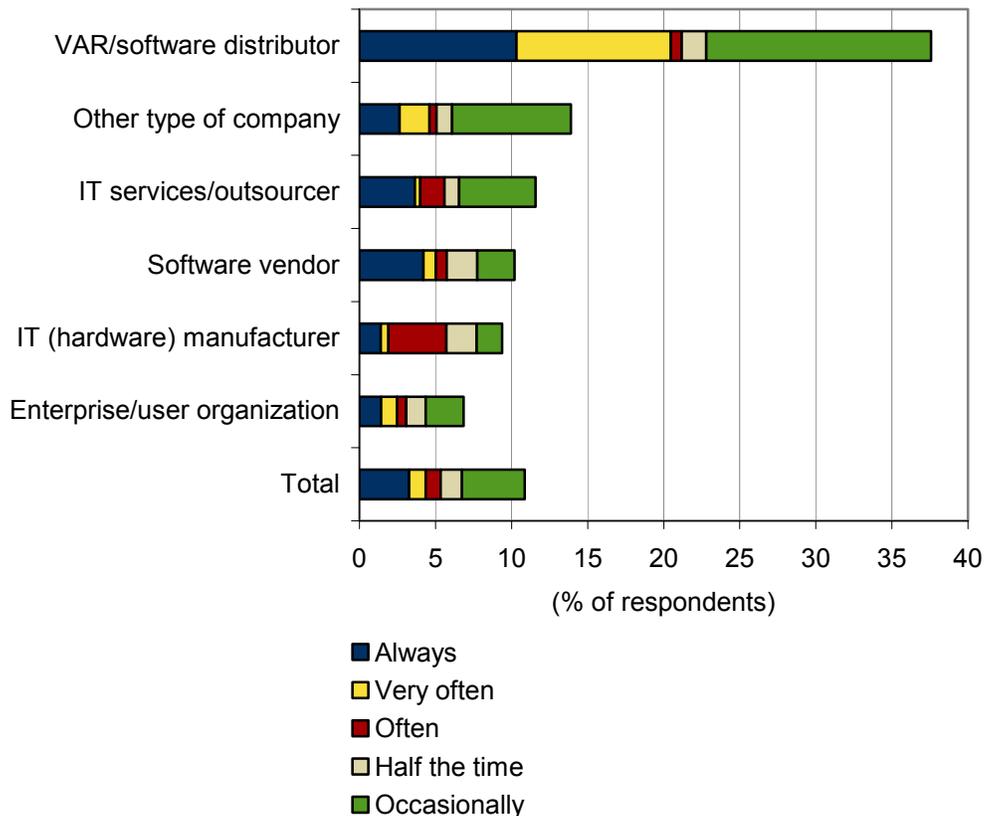
Solid 4GL Footing

A recent IDC study of IT professionals around the globe notes the volume of respondents who are currently engaged in some capacity with a 4GL language. For example, over 10.8% of individuals across varied company types and sizes, industries, and geographies reported using 4GL technology "occasionally" to "always." Even more noteworthy is the 37.6% of respondents from value-added reseller (VAR) and other software distributor organizations that reported being engaged with this technology "occasionally" to "always," and over 20% of this segment of the respondent base reported using this technology "very often" or "always" (see Figure 1). Many other 4GL environments reflected similar findings in this study, with some variation up or down, depending on particular market sectors and organization types.

FIGURE 1

Current Use of IBM Informix 4GL Development Environment

Q. How often do you personally use each of the following rapid application development environments and languages (4GLs): IBM Informix 4GL?



n = 3,240

Note: Data is weighted from 3,240 responses in a survey of Worldwide IT professionals in IT and enterprise firms (including IT managers, architects, developers, data and QA specialists).

Source: IDC's Software Developer Collaborative, 2005

Many VARs are typically focused on addressing very specialized requirements for certain industries or particular use cases, and with unrelenting competitive pressures, they need to be efficient and pass on cost savings to customers. They are also key providers of technology to small and medium-sized businesses (SMBs) that often require foundational application infrastructure and reporting capabilities to be integrated into their solutions. Similarly, organizations that use 4GLs are looking for an environment that can readily evolve with changing business requirements.

Most enterprises leverage many different types of development languages that best suit each individual purpose. Many organizations realized that certain tools were best used for interactive sessions, others perhaps more appropriate for Web-based design, and others that could be advantageous for more scientific endeavors. And as new technologies are introduced, it is neither necessary nor often practical to holistically change all existing applications to the latest form. It is more so the case that certain assessments be made regarding appropriate migrations, modernization steps, integration options, or other options to best leverage existing applications. The key for many is risk mitigation, to change over solutions where most needed as there are many variables and dependencies to consider.

Addressing a Hybrid World

One, if not the top, challenge for enterprises today is being able to address change in a more flexible, dynamic, and rapid fashion. However, with competitive and fiscal constraints, efficiency cannot be compromised. Organizations, therefore, are looking to focus on their differentiating offerings and make technology an integral part, and not the goal, of their businesses. They are also keen on taking away underlying dependencies on specific technology and hardware underpinnings to leverage more flexible sourcing and integration advantages.

The promise of 4GL is to make developers more productive, allowing them to maintain applications and readily add new features and functions. Many users note that 4GLs make for shorter development cycles with notable business benefits. By abstracting the complexities of underlying interfaces and translation routines, a more controlled and efficient product can be created. The trade-off of not allowing detailed manipulation at lower system and infrastructure levels is the choice that most 4GL users make to gain this type of standardized environment.

The old world of 4GL was constrained to specific database and OS environments; however, with support for advances in key open standards being embraced across the software industry, applications built within 4GL environments can be addressed along with the many other siloed solutions across heterogeneous environments. Therefore, organizations can select tools and technologies that are best suited to the need at hand versus attempting to fit all solutions into one platform that may or may not be appropriate for all of an enterprise's requirements.

The key to making this approach successful is that vendors must continue to evolve and develop solutions that adhere to standards. Users should also look for providers that maintain a healthy base of customers because it usually reflects a consistent dedication to advancing their environments to adapt to market demands. Looking at many of the longest-surviving 4GLs and frameworks, we find that these technologies indeed have manifested into hybrids. Therefore, labeling a development environment as a specific "generation" has indeed become a challenge and, as such, is becoming a moot point for today's rapidly changing world. Organizations are more focused on getting results, mitigating risk, and looking for flexible solutions, including 4GLs, that can accommodate varied and changing technology variables and skill sets.

A B O U T T H I S P U B L I C A T I O N

This publication was produced by IDC Go-to-Market Services. The opinion, analysis, and research results presented here-in are drawn from more detailed research and analysis independently conducted and published by IDC, unless specific vendor sponsorship is noted. IDG Go-to-Market Services makes IDC content available in a wide range of formats for distribution by various companies. A license to distribute IDC content does not imply endorsement of or opinion about the licensee.

C O P Y R I G H T A N D R E S T R I C T I O N S

Any IDC information or reference to IDC that is to be used in advertising, press releases, or promotional materials requires prior written approval from IDC. For permission requests, contact the GMS information line at 508-988-7610 or gms@idc.com. Translation and/or localization of this document requires an additional license from IDC.

For more information on IDC, visit www.idc.com. For more information on IDC GMS, visit www.idc.com/gms.

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