



# **Development Tools in 2006: any Room for a 4GL-style Language?**

*An independent study by Jean Georges Perrin, IIUG Board Member.*

## Copyright Notice

This document is copyright © 2006 Jean Georges Perrin. This document is the sole property of Jean Georges Perrin. It reflects its personal views and does not reflect the view of past, current or future employers. It does not reflect the views of the IIUG Board of Directors; despite he is a member of it, at time of writing. No excerpt, quote or elements can be copied, published or distributed prior to his written approval. If you wish to use any part of this document, please send a mail to [jgp@jgp.net](mailto:jgp@jgp.net), and state what part(s) you want to use and in which context you are going to use it.

Informix, IBM, Rational, DB2, WebSphere are trademarks of International Business Machines Corporation in the United States, other countries, or both.

Intel is a trademark of Intel Corporation in the United States, other countries, or both.

Four J's, Business Development Suite, Universal Compiler, Genero, Genero Studio, Genero db, Genero Application Generator and Genero Report Writer and all other Four J's Development Tools products herein referenced are trademarks of Four J's Development Tools Europe, Ltd in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft®, Vista, SQL Server, Word, Excel, Windows, and Windows NT are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Querix Hydra4GL, Querix Phoenix, Querix Chimera, Querix Arachne and all other Querix products herein referenced are trademarks of Querix, Ltd in the United States, other countries, or both.

Other company, product or service names may be trademarks or service marks of others.

The original document and its latest version can be found on <http://www.jgp.net>.

Revision 019.

## Introduction

In March 2006, I started a survey Informix 4GL and its possible replacements.

The survey was run from March to July 2006. 95 people replied. Replies came from all around the world. The idea was to get a general feeling about how people were using 4GL, why they used it, on which platforms and against which databases and what are the trends.

Vendors and product surveyed were:

- Aubit Computing with Aubit4GL,
- Four J's with Business Development Suite (BDS) and Genero.
- IBM with Informix I-4GL and Rational EGL.
- Querix with Hydra4GL.

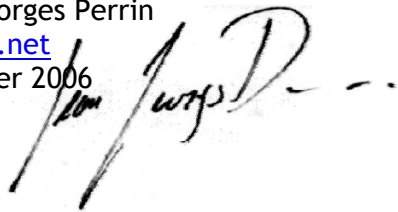
It took me several months to analyze the results, find some correlation, come back to some of the respondents, check with participating vendors & suppliers, etc. On a personal note, from the time I started to now, I had three different employers, so it was not really an easy task! I hope you'll enjoy the result as much as I suffered doing it. Nevertheless here it is. Vendors will probably enjoy it less but I hope to have represented the users' interests and motivation.

I want to thank my "buddies" of the IIUG Board of Directors as well as Mehdi Afshar (Querix), Mike Aubury (Aubit Computing), Venkatesh Gopal (IBM), Hubert Hoelzl (Querix), Jerry Keesee (IBM), Omkar Nimbalkar (IBM), Andrea Reid (IBM), Jonathan Sayles (IBM), Dominic Smith (Querix) and many more who acted under cover or wanted to remain anonymous; who helped me propagate the survey and commented the initial drafts.

**I specially want to thank all the 4GL users that participated in this study. Thank you! You love it, you are passionate about it and I could feel that in every answer you sent.**

Enjoy your reading,

Jean Georges Perrin  
[jgp@jgp.net](mailto:jgp@jgp.net)  
December 2006



|   |    |
|---|----|
| Copyright Notice .....  | 2  |
| Introduction .....  | 3  |
| Survey Results .....  | 5  |
| Population.....   | 5  |
| Development Type .....  | 6  |
| Usage.....  | 6  |
| Distribution Used.....  | 7  |
| Development & Deployment Platforms.....                           | 8  |
| Databases .....   | 9  |
| Front-Ends.....   | 10 |
| Windows .....   | 10 |
| Text User Interface (TUI) .....                                   | 10 |
| Web Interface .....   | 11 |
| Java Client .....   | 11 |
| X Window .....  | 11 |
| Development Perspective .....                                     | 11 |
| Favorite Features .....   | 11 |
| Other Tools .....   | 13 |
| Tools Being Considered .....                                      | 14 |
| Wishes .....  | 14 |
| Graphical User Interface (GUI) .....                              | 15 |
| Web Development .....   | 15 |
| Integration .....   | 15 |
| Reporting .....   | 16 |
| Databases! I want more Databases! .....                           | 16 |
| Tools .....   | 16 |
| Legacy Tools .....  | 16 |
| Language Enhancements .....                                       | 16 |
| Business .....  | 18 |
| Other Ideas .....   | 18 |
| Migration .....   | 19 |
| Migrating? Ok, but How? .....                                     | 19 |
| And Why? .....  | 19 |
| Free Speech.....  | 20 |
| Satisfaction .....  | 21 |
| Currently Available Products.....                                 | 21 |
| Aubit Computing Aubit4GL .....                                    | 21 |
| Four J's Development Tools Business Development Suite (BDS) ..... | 22 |
| Four J's Development Tools Genero .....                           | 22 |
| IBM Informix I-4GL .....  | 22 |
| IBM Rational Enterprise Generation Language (EGL) .....           | 22 |
| Querix Hydra4GL .....   | 23 |
| The Fight of the Windows GUI .....                                | 24 |
| Four J's BDS & Four J's Windows Front End .....                   | 24 |
| Querix Hydra & Querix Phoenix .....                               | 24 |
| Four J's Genero & Four J's Genero Desktop Client (GDC) .....      | 24 |

|                                 |    |
|---------------------------------|----|
| My Dream .....                  | 24 |
| Open Source .....               | 25 |
| Summary .....                   | 25 |
| Third Party Solutions .....     | 27 |
| Trends.....                     | 27 |
| Aubit4GL.....                   | 28 |
| Four J's Development Tools..... | 28 |
| IBM .....                       | 28 |
| Querix.....                     | 28 |
| Conclusion.....                 | 29 |
| Additional Resources.....       | 30 |
| Aubit4GL.....                   | 30 |
| Commercial Support Web Site     | 30 |
| SourceForge Community Web Site  | 30 |
| Four J's Development Tools..... | 30 |
| Web Site                        | 30 |
| IBM .....                       | 30 |
| Informix 4GL                    | 30 |
| EGL on developerWorks           | 30 |
| EGL RedBooks                    | 30 |
| Querix.....                     | 31 |
| Web Site                        | 31 |
| Querix Community Forums         | 31 |
| MoreData .....                  | 31 |
| Web Site                        | 31 |

## Survey Results

### Population

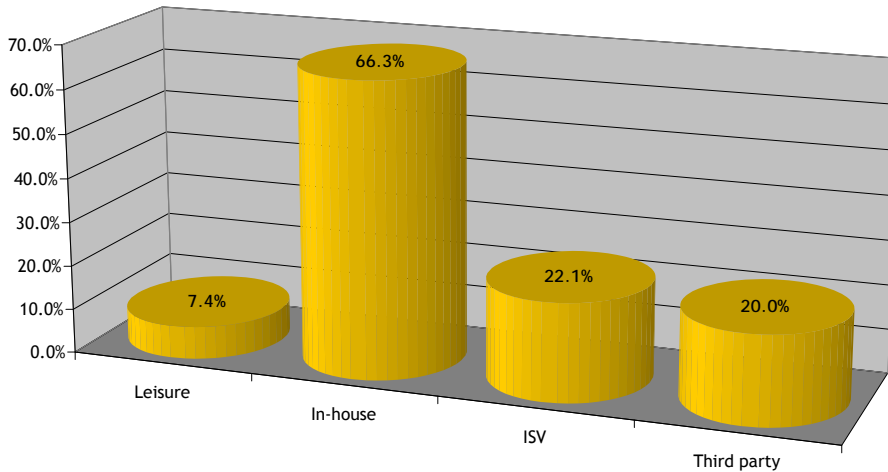
The studied population has been contacted through personal contacts, the IIUG mailing lists and forums, the Usenet newsgroup (comp.databases.infomix - cdi), the Aubit4GL mailing lists, Four J's mailing list and the Querix Forum.

Respondents could remain anonymous if they wished and some used this right. Among the identified population, all respondents were working for different companies.

Employees from IBM (working either on 4GL or EGL or close to it), Aubit, Four J's Development Tools and Querix were not allowed to participate.

The answers came from three major areas of the world: North America, Europe (mainly United Kingdom) and South America.

## Development Type



There are four type of development:

**Leisure development** is what people do on their spare time. Like every geek, I'd love to do more...

**In-house development** is done when you are developing for your company's own usage.

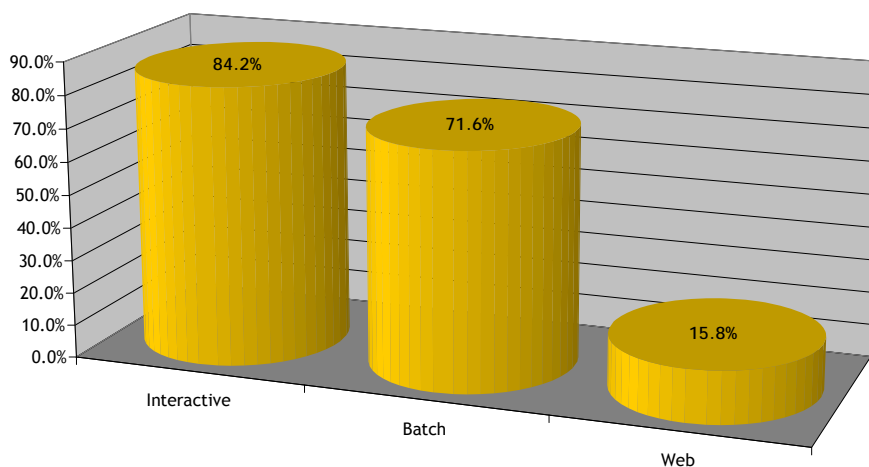
**Independent Software Vendors (ISVs)** are developing solu-

tions they sell to customers, where basically the language and source code is not publicized. Well known ISVs include SAP, Lawson, Tecsys...

**Third party development** is basically outsourced development.

4GL remains very high in in-house development and surprisingly in third party development, where people outsource some 4GL development to other companies. If the 4GL market was flat (or dead), third party development companies would not be out there.

## Usage

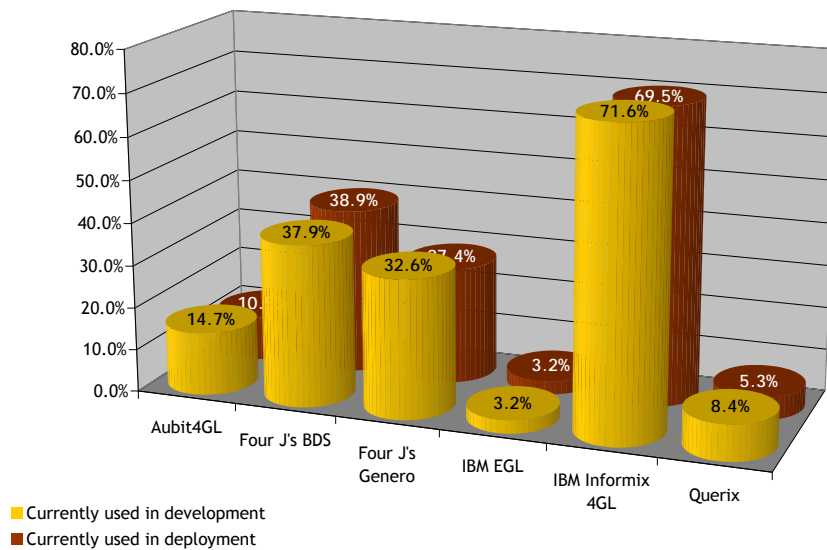


Interactive applications are standard user applications, such as Microsoft Outlook or the famous **Informix 4GL stores demo**. Batch applications are usually unattended programs executing tasks, such as maintenance, reporting, usually during the night.

Due to its nature, 4GL is very often used for interactive applications (all the business applications out there) and batch applications. Often batch applications are embedded within Unix shells.

Web applications are more rarely built with 4GL, although some solutions exist. Querix offers Arachne. Four J's has a BDS solution (Web Front End) and a Genero-based solution, which are not compatible. A very popular request is indeed to be able to build web applications using 4GL. No vendor has a page interpreter module (à la PHP) understanding 4GL.

## Distribution Used



Another interesting repartition is to see what people are using for developing and deploying. It is even more interesting to understand the meaning of the evolution.

Totals are well over 100% as many 4GL shops uses more than one distributions. **51.6% surveyed sites use over two develop-**

**ment tools** and 47.4% deploy using at least three tools.

For Aubit4GL (10.5%), Genero (27.4%) and Querix (5.3%), why deployment figures are lower than development? The wrong conclusion would be that people are developing and throwing the result away! The right conclusion is that the trend is that people are currently investing in Aubit4GL, Genero and Querix.

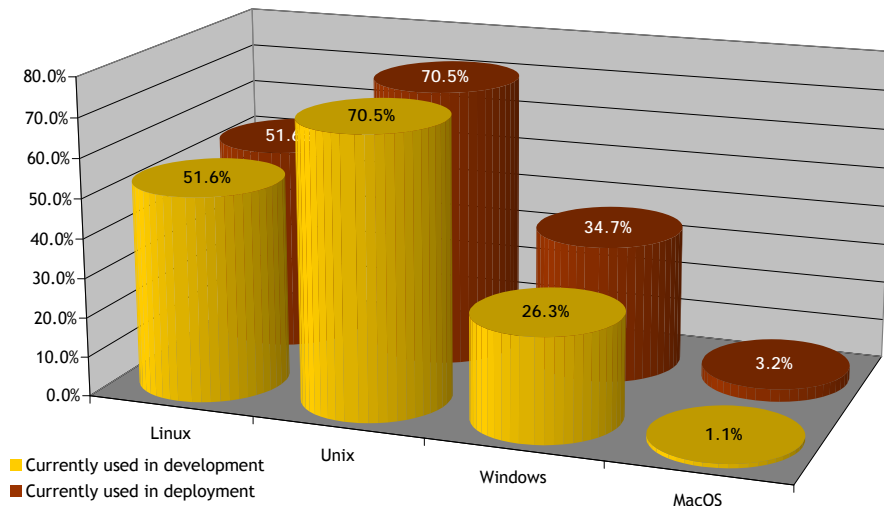
Four J's BDS is still deployed; however, less people are really developing with it, as they tend to go towards Genero. The Four J's situation is interesting to analyze: although Four J's offer two products representing an impressive 37.9% and 32.6% market penetration, the total of Four J's penetration is 51.6% in development and 50.5% in deployment. Within the Four J's market share, the details indicate that 44.9% of developments go from BDS to Genero, 18.4% are using Genero but did not indicate their origin and 36.7% are staying with BDS. The latter figure should not be interpreted as a judgment over Genero's capabilities, but rather as an illustration of the reluctance to change. **Reluctance to change is very high within the 4GL community.**

Why is Informix 4GL development higher than Informix 4GL deployment? Most developers usually keep a copy of Informix 4GL somewhere, as a reference. They still massively develop with it, but deploy less as they tend to switch to another vendor. Many 4GL developers have not adopted the "extra" features offered by some vendors that would make them

lose their compatibility with the Informix 4GL compiler. That's why Four J's and Querix added a special comment<sup>1</sup>. Using Querix Hydra, you can also customize your Graphical User Interface (GUI) using external configuration files.

EGL is still very small, but has growing interest. The positive sign is that some companies are already deploying EGL-based solutions.

## Development & Deployment Platforms



We had a closer look at server platforms. Figure between development and deployment are very similar, except for Windows. The 4GL community is and remains mainly Unix-oriented.

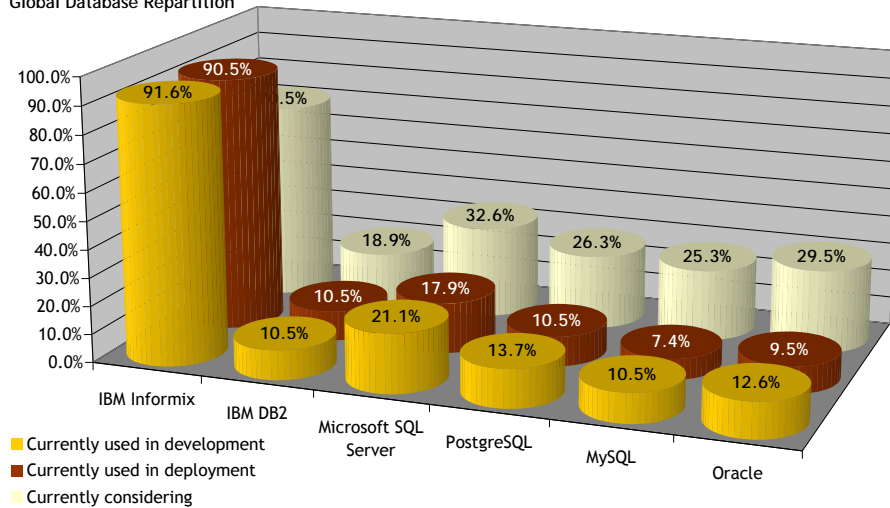
Windows and MacOS are mainly used for deployment.

<sup>1</sup> Four J's BDS and Querix Hydra offer a special comment that is seen as a comment for the traditional Informix 4GL, but not for the compiler, allowing to easily use their extensions while maintaining the compatibility with Informix 4GL. Querix even added a special comment to isolate its extensions from Four J's.



## Databases

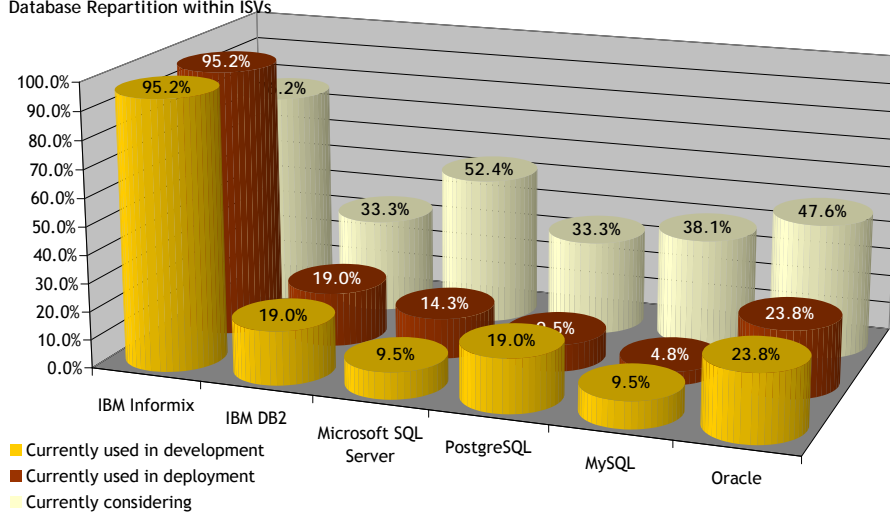
Global Database Repartition



Databases distribution is certainly one of the most interesting topics to analyze: historically Informix 4GL was only running against Informix databases. The first vendor to “break” the barrier was Querix that supported Oracle. Four J’s followed when they signed the distribution agreement with Informix Corp. creat-

ing Dynamic 4GL (D4GL). Informix, at that time, gave blessing and was supportive of the multi-database architecture.

Database Repartition within ISVs



The top graph illustrates general database repartition, whereas the bottom graph illustrates database repartition among ISVs.

The separation was done as ISVs, even if they are a smaller group, are more dependent on integration constraints. Their customers might be running IBM DB2 or Microsoft SQL

Server and will dictate the use of their database.

I would not consider the lower figures in Informix consideration as a negative sign, for several reasons:

- The question, was “what databases are you considering”, which can be interpreted as “what *other* databases are you considering”.

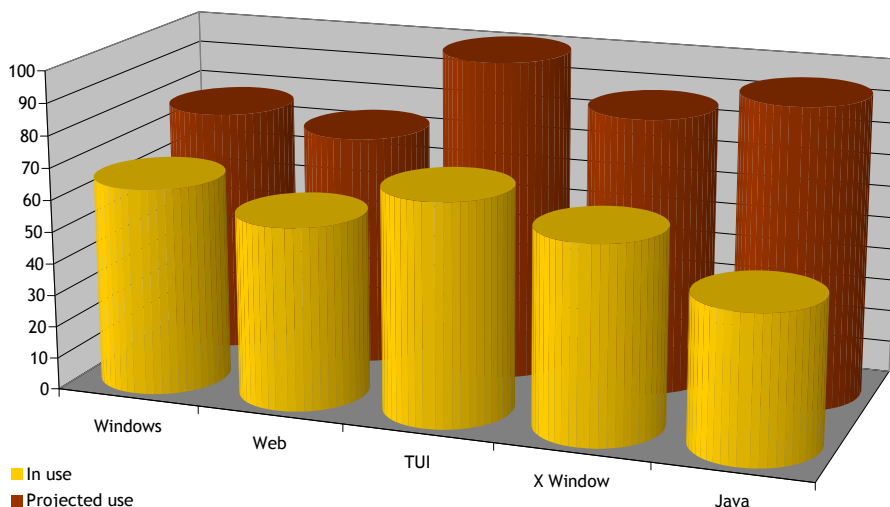
- IBM has put quite some efforts in (re)promoting Informix.
- The survey data is coming prior to the announcement of IBM Informix vNext, code-named Cheetah.

The most interesting fact is that, within commercial databases, Microsoft SQL Server seems to be more attractive than Oracle. On Open Source database, PostgreSQL is more widely used than MySQL, but MySQL has a very high attractiveness ratio within ISVs.

Those figures should be analyzed more carefully over time, creating a trend. However, none of the results seem to be shocking in either direction. I am personally very pleased of seeing Informix staying very strong in all populations.

Some answers mentioned interest in other databases like Derby (aka Cloudscape) and Genero DB (aka ANTs).

## Front-Ends



Respondents needed to say which user interface they are using now and which user interface they wanted to adopt and/or keep in the future.

They needed to give a mark on how important this particular front-end technology was important to them.

The terms “VT 100” mode, text mode, Character User Interface (CUI), Text User Interface (TUI), green screen or dumb terminal all mean the same. I will use TUI.

## Windows

Microsoft Windows remains the “king of the desktop” and therefore everybody having Windows support wants to keep it, but it records the lowest growth.

## Text User Interface (TUI)

TUI remains the most used user interface, followed by Windows. TUI also remains the most wanted user interface in the future.

## Web Interface

An interesting interface is the web. It looks like the web does not have many proponents. And the other end, it is one of the most requested feature request: 4GL (with the exception of EGL) are not very well suited for the web. Lack of support of very interactive Informix 4GL features, like AFTER / BEFORE FIELD, have always harmed those solutions. We can only hope that those tools will quickly support AJAX and bring the expected level of interactivity to the client. On the other hand, EGL already offers very rich and interactive user interface.

## Java Client

Expectations in a Java client / front-end have always been very high and we keep seeing this: zero deployment, embedability in a browser, cross platform, etc.

People are still over-expecting from it and disillusion will probably strike back, once more. A solution would be a Rich Client Platform (RCP) / Simple Widget Toolkit (SWT) based client. RCP is a technology coming from the Eclipse project, bringing **fast and native user interface**.

## X Window

X Window user interface has been gaining momentum in the last year as people tend to install Linux on the desktop to get rid of Microsoft. Credible alternatives are web or Java-based interfaces.

## Development Perspective

### Favorite Features

Every developer has its favorite features. I wanted to understand why 4GL was so appreciated by its developers, and which features they really liked. I have asked developers to list their more popular feature. Entry in the survey was a text form (vs. a pre-defined list of choices) as I did not want to promote or encourage any feature. Each respondent needed to give only three features. Respondents complained it was difficult for them to make a choice!

| Feature                     | %        | Description   |
|-----------------------------|----------|---|
| <b>Interactive debugger</b> | de- 2.7% | I am personally glad this feature is so low... However, I am absolutely certain that most developers really use & need a debugger. It is also a popular "Wishes" (see p. 14).<br><br>In a previous life, I remember using a state-of-the-art debugger for debugging 4GL web applications. |
| <b>Stability</b>            | 5.4%     | It is stable; people don't even bother mentioning it is!  |

| Feature                           | %     | Description   |
|-----------------------------------|-------|---|
| Compatibility with the C language | 5.4%  | <p>All 4GL available (except EGL) are C-based, which means that they rely on the C runtime system of the underlying Operating System to work.</p> <p>It is also the only way to extend the environment in which 4GL evolves. You want to communicate via TCP/IP? Add a C extension. You want to generate PDF? Add a C extension.</p> <p>EGL, relying on Java, can also be extended with C code (using the Java Native Interface or JNI). EGL can also call Java.</p>  |
| Windows support                   | 8.1%  | <p>This feature applies to all using any 4GL besides Informix 4GL. Microsoft Windows is out there, whether you like it or not. It is great that 4GL runs on Windows.</p>  |
| DDE                               | 10.8% | <p>Dynamic Data Exchange (DDE) is a technology Microsoft developed introduced in 1987 to allow processes (and applications) to intercommunicate on Windows. DDE is now considered obsolete and replaced by OLE and OLE Automation, but still ships in most office-oriented products.</p> <p>On typical 4GL architectures, this feature is associated at the client level, allowing to, for example, start Excel and enter some data in it. Four J's and Querix are the only ones providing such features.</p> |
| Database independence             | 10.8% | <p>It's a rather strangely low figure. One would expect this to be more demanded. However, ISVs really appreciate the fact they have it: "We can now tick more boxes in RFPs".</p>  |
| Performance                       | 13.5% | <p>The fact that 4GL is really efficient when it runs is so natural that most forgot it...</p>  |
| Rich Graphical User Interface     | 13.5% | <p>Rich Graphical User Interface is a popular feature, mainly coming from the Four J's Genero users.</p> <p>The fight is really here between Four J's users and Querix users. See "The Fight of the Windows GUI" on p. 24.</p>  |
| Easy Language, RAD                | 24.3% | <p>4GL is indeed an easy language. By definition it is readable and understandable by normal human beings. One respondent mentioned "4GL code is vaguely readable by non-programmers".</p>  |

| Feature                                     | %     | Description  |
|---|-------|--|
| <b>Reporting</b>                            | 37.8% | Reporting is one of the paradox of 4GL. Reporting is easy, efficient; source code of reports is editable, readable, scalable. It's a popular feature!<br><br>It is not enough. Nowadays, people expect PDFs, Excel sheets, e-mail, etc. We will see more of that in the "Wishes" (see p. 14).  |
| <b>Embedded SQL</b>                         | 43.2% | Allowing the developer to directly type SQL and getting results very easily (like <code>SELECT INTO</code> ) is a much appreciated feature. Very few other languages allow that.   |
| <b>Easy User Interface (UI) development</b> | 67.6% | By far, the most popular feature. It's true that designing a form is easy. You take vi (it does not work with emacs <sup>2</sup> ), place your fields, add some attributes and off it goes... No need for a WYSIWYG tool, no need for a screen painter.<br><br>To make your life even easier, you can use tools like <b>sperform</b> to create the sufficient mock-up.<br><br>With modern tools, you need a screen designer to achieve your results, even (or should I say "especially") in web development. However, the GUI result is rather rudimentary and matching it to Windows or modern Window managers is a daily challenge for developers & architects working for Four J's or Querix. |

## Other Tools

An average of 2.4 tools is used on top of 4GL. Most popular being C, C++ and ESQ/L/C, followed by Visual Basic and derivatives... Everyone, except two, have at least one tool, and up to five in the development chain.

Once more, this was an open text question, so I needed to create some groups. Some groups may not seem "natural", such as VB and VBScript, but usage is similar.

| Language & Tool  | Usage                                       |
|--|---|
| C, C++, ESQ/L/C  | 36.6%                                       |
| Visual Basic (VB), Visual Basic for Applications (VBA), VBScript | 29.6%    Mainly used for office automation. |
| Perl   | 25.4%    Including Perl DBI.                |
| Java   | 25.4%                                       |

<sup>2</sup> Just kidding, any text editor will go!

| Language & Tool  | Usage |  |
|------------------|-------|--|
| Batch, shell     | 21.1% | Shell is not limited to batch. A lot of interactive applications are relying on shell / system calls.          |
| .net             | 18.3% | May include some VB.net and ASP.net.   |
| PHP              | 15.5% | Mainly for web.  |
| C#               | 2.8%  |  |
| HTML, JavaScript | 2.8%  |  |
| EGL              | 1.4%  |  |
| OpenOffice       | 1.4%  | Used, in this case for reporting and office automation.  |
| New Era          | 1.4%  | Yes, there is one.   |
| Other            | 60.6% | Including Python, other proprietary tools (Delphi, Crystal Reports, ColdFusion...), Unix tools (awk, flex...). |

The main purposes of using additional tools with 4GL are:

- **Integration.**
- Web development.
- Reporting (mainly PDF, laser printing...).

### Tools Being Considered

Developers are constantly looking for more tools; management brings new needs which 4GLs have difficulty to implement, such as complex web applications, web services, etc. It is not surprising that the tools being considered are for the same needs as the needs being currently fulfilled.

Sounds like a complex sentence? Actually some companies are in advance and have already implemented some additional features in their 4GL applications, like PDF generation.

However, the common needs to be fulfilled by the additional tools are:

- Building more attractive application.
- Integrating the business logic in a web environment.

In this regard, all companies are considering tools such as (in descending order): Java, EGL, PHP, Perl, Python, Ruby on Rails...

Companies using only Informix 4GL are considering Four J's or Querix for their GUI and integration features.

### Wishes

This is my favorite part of the survey. It lists what companies want, what vendors should provide.

As for many items in this survey, this was a “free text” field as I did not want to interfere or create attraction to a particular feature or certain type of feature. I love Integrated Development Environment (IDEs), but I do not share this passion with many 4GL developers. On the other hand, a lot of them wish to have a more modern debugger, which is not something I really miss when I do not have it.

Therefore, let’s enter this science fiction zone, and see what the 4GLs of tomorrow would be, based on users, developers, business owners...

### Graphical User Interface (GUI)

The GUI improvement is by far what respondents want enhanced. They want more than what is currently offered by either Querix or Four J’s.

More of that is found in “The Fight of the Windows GUI”, p. 24.

### Web Development

The most popular request, in term of new development platforms, is the Web. It is not only for migration of existing web applications to the web, but also to develop **new web applications** using 4GL, like PHP.

The advances of technologies such as AJAX can make it actually easier to port 4GL applications to the web and finally brings a solution to handling the infamous BEFORE / AFTER FIELD on the web (that’s a hint for the vendors<sup>3</sup>...).

Generating HTML for reports has also been mentioned several times. Querix offers some reporting functionalities in Arachne.

Strangely enough XML has been requested only three times. My ego will fade...

### Integration

Integration with other languages (or stealing some other languages) in 4GL is a very popular request. In order of importance: Java, Perl and PHP.

Integration goes beyond language integration:

- Integration with office products (OpenOffice, Microsoft Office...), switching from DDE to OLE / COM.
- Use of external devices such as scanners...
- Integration with the Operating System, by providing more OS library, to access files (a recurring need is to read ASCII files), network via TCP/IP...

---

<sup>3</sup> Ok, that’s even the second time I mention this, so if they don’t get it...

## Reporting

As previously stated, reporting is a popular feature and it is also where it collects the more needs for improvements: building more complex reports, including charts, generating Excel files, outputting reports in Rich Text Format (RTF), Microsoft Word, OpenOffice Writer and... ..Portable Document Format (PDF) for distribution and read via Acrobat Reader.

Support for (more) printers is also a frequent need, especially when your application needs to print your bills...

## Databases! I want more Databases!

Developers, managers and sales want more support for more databases, either more transparently, like support for DB2 in Informix 4GL or plain pure ODBC (JDBC?).

SQL is so well embedded in 4GL that it does not follow all the syntax offered by the engine, like `SELECT FIRST`, to get the first rows. Putting 4GL more in sync with the engines would be great. I guess you now all understand why no other language provides such a level of “embeddement” with SQL.

## Tools

Development tools remains quite rudimentary. Querix provides an IDE, Four J’s announced Genero Studio.

Demand for a debugger, integration within an IDE (preferably Eclipse), a screen designer, unit testing tool, a documenter, a stricter compiler, etc. are recurrent requests. The screen painter should be able to generate handling code, and keep it in sync (which implies a screen painter, and from what I have read, preferably in Eclipse).

A graphical code generator, built within the IDE would be a nice feature to have. I am (personally) not too keen on code generators. I think a good design (sometimes at language level) should replace a code generator. However, wizards, used for getting started, could be useful. It’s always good to start with a “Hello World” kind of app, or a complete CRUD (Create, Read, Update and Delete) application.

## Legacy Tools

Support for **ace** and **perform** has been asked for several times.

## Language Enhancements

Developers wish more ability to use their data: an equivalent of the C typedef, constants, more “containers” like the Apache Collections<sup>4</sup>, Perl, etc. Elements like Maps, Hashes, Lists, etc.

---

<sup>4</sup> Part of the Jakarta Commons.



In this regard, Querix has added extensions to the `DEFINE` statement. It can now create standard types, which can be re-used or even derived. For example, if you want to define a standard birthday type, you can use something like `DEFINE Birthday TYPE AS DATE`. Later I can reuse birthday, like `DEFINE myBirthday OF Birthday`. Of course, this makes a lot more sense when you manipulate complex types like `RECORD`.

Let's look at this:

```
DEFINE BasePerson TYPE AS RECORD
  firstName CHAR(40),
  lastName CHAR(40)
END RECORD
```

I can use it:

```
DEFINE aPerson OF BasePerson
```

It can be extended as: an `USPerson` and a `FrenchPerson`:

```
DEFINE USPerson UNDER BasePerson AS RECORD
  middleName CHAR(40),
  ssn CHAR(9) # Social Security Number
END RECORD
```

```
DEFINE FrenchPerson UNDER BasePerson AS RECORD
  ssn CHAR(15)
END RECORD
```

When I “instantiate” a `FrenchPerson`, I can access its Social Security Number, not its middle name, whereas I can access both on an `USPerson`. This is a very interesting (not only to notice that we can have more French citizens than US citizens), but it will make development a lot easier.

Of course, with such new structures coming, the ability to pass elements either by reference or by value becomes a necessity.

Aspect Oriented Programming (AOP) and Object Oriented Programming (OOP) are common requests, enabling features such as classes, simple inheritance (most requests remain basic), polymorphism, or the ability to call a function by its name.

Developers expect more and standard string manipulation functions. As a consequence they wish for more Perl-like functions, or even Perl integration.

Batch users did not wish to remain orphans, but they were not very wordy about what they wished... What can I imagine<sup>5</sup>, a connection pooling facility that will enable batch to connect and perform transactions faster, a multiplexing / multi threaded syntax for multiple SQL statements to execute on the different engines, a mail interface to send reports, status...

---

<sup>5</sup> Hey, as a 4GL user I can fill the survey too...

Various other ideas:

- Multi-threading API in 4GL.
- Timeout interrupt for interactive applications.

## Business

*"Make the language a standard"*. If the language becomes an open standard, third party companies will be less afraid to invest than if it was in a closed environment. Vendors will not compete on the language itself, but on extensions, on infrastructure components, on extensibility...

Companies like Querix and Four J's do not simply offer a compiler and a set of run time libraries. They are infrastructure providers. The wish would be to have this infrastructure more open and documented. Genero's communication between its clients and its server is based on XML. That's cool, but what is the point if XML is closed to Four J's and not shared with partners? Opening and standardizing this communication channel will enable partners to build additional solutions.

*"Too expensive and lack of marketing!"* It is an extreme statement: when digging, business people are complaining about the runtime fee (there is however a CPU licensing model within most companies).

Another request was for commercial vendors to propose a **Home Edition** with a very low to free license so that developers can train at home. Querix already offers such a license.

It's all about making the bride nicer for wedding: make 4GL more attractive.

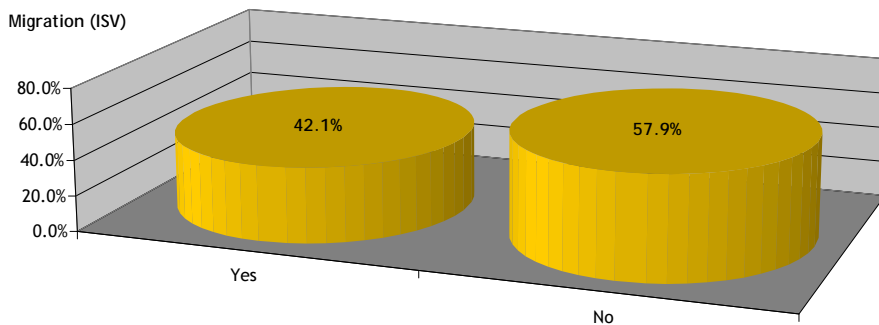
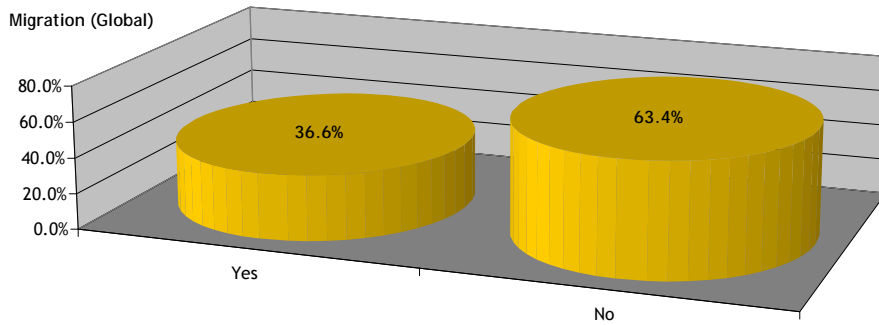
Several respondents even requested a good Java migration toolkit.

## Other Ideas

Here a list of unique ideas I really found interesting, but did not know really where to put in my previous categories:

- 4GL.net would be the equivalent of any .net language, fully integrated in Visual Studio.
- Port 4GL to mainframe.
- Have IBM release their 4GL test suite so that it serves as a standard.

## Migration



This is probably the saddest conclusion of this survey. More than a third of the companies are migrated away from 4GL.

36.6% of the companies interviewed are **migrating**. If we focus on ISV, the ratio climbs to 42.1% of companies.

Those figures are not about companies *planning* to migrate, but about companies in the process of migrating.

### Migrating? Ok, but How?

So what are those companies migrating to? Not everyone answered, but 7 respondents are going to Microsoft .net, 5 to Java / Java EE, 4 to EGL, 5 to another 4GL and 4 to other (Ruby on Rails, PHP, and an ERP system).

.net answers include ASP.net, C# and VB.net. EGL is relying on the same technologies as Java. It really means that the game is tight between a migration to either the Java EE platform or the .net platform.

2 respondents are not migrating but switching to another vendor, while 5 considered switching to another vendor is a **migration**. As a summary, 7 respondents stayed in 4GL with another vendor.

### And Why?

The main reason is the lack of proper GUI and critical features such as PDF or Excel reporting.

Then, it is customer driven. One could argue that the customer does not see what is under the hood, but this is wrong. Customers are more and more demanding, they want to know

what the software is based on. Usually they demand either the .net or the Java EE platform. It limits the opportunities of ISVs and development houses.

The next criterion is getting away from Informix. I don't like the idea, but it is the result of the previous marketing policy from IBM. Clearly, things are changing for the good and, as stated in "Databases", on p. 9, it is a trend to watch.

The least three are tied, but they reflect important lacks: lack of support, lack of features and lack of developers.

Don't understand lack of technical support. All vendors will be pleased to take your Euros or Dollars (or any other currency to be honest), and provide you technical support. All vendors are not alike, according to the respondents. It's more about the impression that 4GL is simply dead. I do not think so, but compared to the media fuss, the impressive network of business partners, it is difficult to see 4GL vs. Java or .net. Worse, 4GL is no standard. Bringing partners on the ship will only happen if the 4 vendors agree to a standard definition of 4GL.

Lack of developers has been a recurring problem for the last 10 years. When I was at University, everybody told me "if you want to make money, go do some COBOL in Switzerland". It is difficult to attract youngsters to 4GL as it is to COBOL or mainframes. Nowadays, every student wants to go Java or .net, just like I wanted to go C++ over a decade ago (and finished doing Visual Basic).

A small minority is willing to leave the Unix world or is witching to an ERP system.

## Free Speech

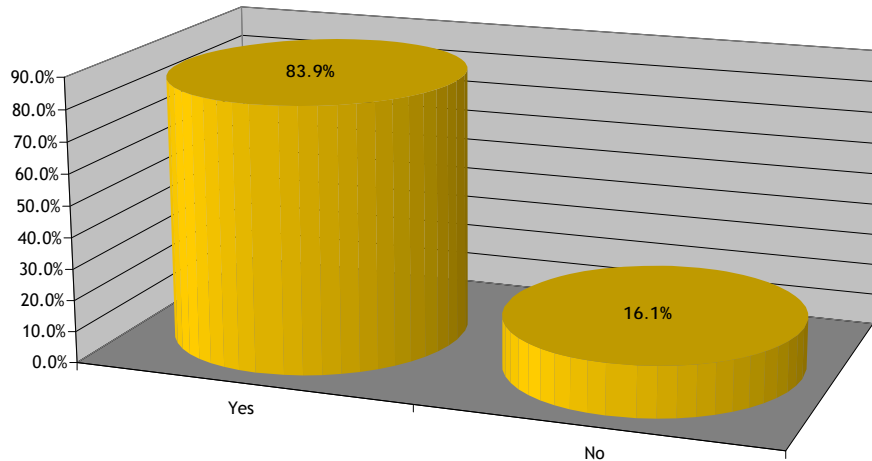
The final questions of the survey were mainly to let respondents empty their chest and share why they would change vendors.

The fact that IBM does not plan to add new features to 4GL is interpreted as a slow disengagement. No other vendors have been able to promote or impose new features. The ideal vendor will provide **more features and top support**.

It's really the area where developers, managers and all respondents expressed their feeling and this is probably an area where this study differs from Gartner, IDC, MetaGroup, etc. The emotional part is really "big" among 4GL users. Nobody wants to see it go, and almost every respondent was bringing its small "brick" to build an idea of renewal and continuity.

The successful vendor will also understand this passion and drives it towards success. I will not name companies & organization but some of them really do care about customers...

## Satisfaction



It becomes legitimate, after all those questions, to ask the only legitimate one. **Are you satisfied with 4GL?**

I must admit I was really astonished, almost shocked by the result. I knew 4GL was popular, but I could not imagine than more than **four users out of**

**five are satisfied**, with a language and platform that did not evolved that much in the last 10 years. Users are committed, passionate about it. They finally find a relief and some of them were happy to have a friend to talk to and thanked me for doing this job (if only I knew the time it took me before I started...).

Thanks to you all! It's an incredible opportunity and chance to speak to passionate people.

## Currently Available Products

### Aubit Computing Aubit4GL

Aubit Computing Ltd is the company sponsoring the Aubit4GL open source compiler. The company provides technical support and services but the core product remains open source and can be downloaded for free from Source Forge or Aubit Computing (see Additional Resources, p. 30).

Users are extremely satisfied by the technical support provided by Aubit Computing Ltd. Aubit4GL managed to create a very strong community around itself, where members are passionate about the product they use. You seldom find such an enthusiasm around a product. However, to my opinion, the Aubit4GL lacks a few developers.

A lot of developers have provided framework and tools to the Aubit4GL project. Those extensions could eventually be used with commercial products such as BDS, Genero, I-4GL and Hydra.

The project is still a bit rough and lacks a few things like Windows binaries, point and click installation<sup>6</sup>, a better documentation, etc. They are looking for volunteers as well.

<sup>6</sup> They are currently working on it.

## **Four J's Development Tools Business Development Suite (BDS)**

BDS is an end-of-life product, as Genero is now the product driving all attention from Four J's. It has barely evolved over the last 4 years but it's considered a bullet proof product.

## **Four J's Development Tools Genero**

Four J's is and remains the leader in the market of pure Informix 4GL clones. They have developed an incredibly rich technology over the years. They have proven to be, in this field, an incredible innovator.

Genero is an interesting product. The graphical user interface is very rich, with most of the widgets you would expect (except tree views).

Genero, like BDS, support Web Services, allowing its usage in a Service Oriented Architecture (SOA) deployment.

Genero offers connectivity to most databases through its proprietary extension, Open Database Interface (ODI). It however lacks a standard JDBC / ODBC connection to access source of information not accessible via ODI.

The very rich set of extensions found in Genero is using a mix of 4GL style (new and extended syntax for forms, 4GL keywords and grammar inherited from BDS...) and Object Oriented (OO) syntax (manipulation of the XML-based Abstract User Interface...).

The soon-to-be-available Genero Studio has been expected for a while. I have unfortunately not worked with it but it looks interesting. The only comment is why releasing yet-another-IDE as Eclipse is ruling in the development environments space. The same comment can be applied to Querix HydraStudio, but their IDE has been out for several years now.

## **IBM Informix I-4GL**

I-4GL is the reference implementation. Informix 4GL, now version 7.32, is very stable. IBM is maintaining it but no new major feature and development are planned.

## **IBM Rational Enterprise Generation Language (EGL)**

EGL remains a big question mark for many 4GL developers. Many have created unnecessary FUD (Fear, Uncertainty and Doubt) around the product. The purpose of this study is not to

advocate EGL, but as I did the study very consistently, I went into details about EGL and found out that many of what's rumored outside is not really true.

The community is available through developerWorks (see Additional Resources, p. 30) and it is very responsive. IBM has set up an eco system team, acting like a SWAT team between the users and engineering to provide first class service to the community.

EGL is not only a language but a wonderful environment to develop web based applications. For the last 12 years, I was looking for a product combining the ease of use of Visual Basic, the stability and scalability of Informix 4GL, a product that allows sharing of tasks between application developers and web designers, while providing an easy way to deploy. EGL is the closest to my dream development environment. It still needs some polishing and some operations could be done in an easier way; however, the learning curve is short.

EGL is also the only one providing an excellent toolset. The IBM Rational Application Developer v6 is an incredibly rich environment. It provides all the tools you need (and even some you think you don't need): web site management, web designing, database browsing... As it is based on Eclipse, you can add 1,000s of plug-ins to it if you wish.

The language is proprietary to IBM. Nowadays, you would expect a language specification to be driven by external standardization organization such as IEEE, ECMA, OSI, OMG...

## **Querix Hydra4GL**

Querix has been Four J's challenger for over the last 10 years. This forced them to be closer to their customers and to provide more efficient and pragmatic solutions. As an example, they have been shipping an IDE for years. They may not appear as innovative as Four J's or IBM, but their toolset is based on mature technology, driven by customer-care.

Querix has a very interesting approach to 4GL. The code itself can stay as it is and the tuning or GULing is done through external "script" files or programmatically. Most companies see it as a big risk to migrate everything in one big project. and for this reason, Querix supports both, language extensions and functions.

One of the main strengths of Hydra is its natural support for databases which really converts the original Informix SQL statements to Oracle, DB2, SQL Server and many others. Hydra also supports the Four J's BDS syntax.

The Windows client being ActiveX compatible offers a very interesting approach to build very rich user interface and extending it by developing your own custom made controls.

Querix provides a very decent, fast IDE with all the features you'd expect: project management, source code editor, screen designer... When will I get all that on Eclipse? It seems to be quite soon now!

Querix technical support is also known for providing excellent help.

Querix has also an excellent level of security, especially with native support for SSL between the front-end and the application.

Hydra, in its current version, does not support generation of P-code. It thus relies on a C compiler to finish the compilation. With the popularization of the Gnu C Compiler (GCC) and HydraStudio, the process is achieved easily and transparently.

## **The Fight of the Windows GUI**

I can't remember who the first released a GUI 4GL. It was before "my time". It does not matter really. Now, there three offerings in the game and each has its features, pros and cons.

### **Four J's BDS & Four J's Windows Front End**

The front end part (Windows Front End) of the Four J's BDS architecture is a server listening to request made by the virtual machine. The client is based on Tcl/Tk and receives such information. Four J's did not follow the Tcl/Tk branch and switched to Genero.

### **Querix Hydra & Querix Phoenix**

Querix Phoenix is the Windows client. Phoenix relies on Microsoft technology and should be seen as an ActiveX component handler, which receives order from the virtual machine. An ActiveX is (usually) a rather small graphical component. The beauty of Phoenix is its capacity to integrate more of those components, making the user experience a lot richer: you can embed tree views, charts and even MP3 players...

### **Four J's Genero & Four J's Genero Desktop Client (GDC)**

By switching to a complete new technology, Four J's generalized the use of the Qt toolkit across its client product, benefiting of new platforms such as MacOS X, ActiveX distribution, etc. Replacing the aging Tcl/Tk by Qt was certainly a smart move, which however, forbids extensibility. To make user interface more attractive, Genero's form definition, based on the original .per, has changed to enable layout managers, etc.

### **My Dream**

None of the solution described above are perfect both Phoenix and GDC provide good user interface, but they miss complex "widgetry" and the ability to have non-modal windows: complex graphical widgets such as tree view or data grid "out of the box", non-modal windows allowing to have multiple windows active at the same time, etc.

I'd love to have an extensible thin client, standard driven, cross platform, running multiple applications, from different vendors, with limited bandwidth consumption, with automatic



updates and rich user experience... I guess I still need to wait to get my Universal Thin Client...

## Open Source

Open Source is certainly more than a trend. This study is however not here to educate anyone about it.

The choice of going Open Source vs. staying or buying a piece of software is not as easy at it appears. The ability to deliver a service is what you should be looking after, more than the permanence of the supplier or the source code.

Source code can be protected via an escrow service. So there is always a possibility to get the source code. So, when your supplier ceases its activity, you can get access to it. But in case of failure, who will be able to provide the service to recover from the source code of a product?

There are many other benefits to Open Source, but the fact you can hack into the code is certainly not one! Your business, even for ISVs, is probably not about maintaining C libraries, hacking into compiler techniques, etc.

Finally don't misunderstand me, I am not against Open Source, I like it, there are different grades of Open Source, and like every choice made for the Enterprise, its adoption should be done wisely.

## Summary

The following summary table describes the product as of December 2006. Many announcements were made or are coming in the following weeks.

|             | Aubit4GL                     | BDS                      | Genero                   | Informix I-4GL           | Rational EGL                               | Hydra4GL                     |
|-------------|------------------------------|--------------------------|--------------------------|--------------------------|--|------------------------------|
| <b>Pros</b> | Excellent technical support. | Very stable.             | Good to very good GUI.   | The reference.           | Backed by IBM as a <b>global</b> language. | Excellent technical support. |
|             | Open Source.                 | Good documentation.      | Complete infrastructure. | Very stable.             | Community around developer-Works.          | Excellent database support.  |
|             | Developer community.         | Complete infrastructure. | Variety of front-ends.   | Excellent documentation. | Rich documentation set.                    | Extensible UI.               |

|  | Aubit4GL   | BDS                                 | Genero                                     | Informix I-4GL                                 | Rational EGL                          | Hydra4GL                      |
|--|--|-------------------------------------|--|--|---------------------------------------|-------------------------------|
| <b>Pros</b>                              | Strong integration features.                                 | Variety of front-ends.              |  |  | Wonderful development environment.    | No alteration to source code. |
|  |  |                                     |  |  | Strong integration features.          | Very good documentation.      |
|  |  |                                     |  |  | High level of security.               | Very high level of security.  |
|  |  |                                     |  |  |                                       | Strong integration features.  |
| <b>Cons</b>                              | Limited number of binaries available.                        | Being phased out.                   | Extensions are mixing 4GL and OO syntax.   | Stalled.                                       | A bit young.                          | P-code not available.         |
|  | Point and click installation not available.                  | No downloadable evaluation version. | Documentation could be better.             | No downloadable evaluation version.            | Proprietary to IBM.                   | No support for web services.  |
|  | Lacks very good GUI.   | No ODBC / JDBC connectivity.        | No ODBC / JDBC connectivity.               | No support for web services.                   |                                       |                               |
|  | No support for web services.                                 |                                     |  | No connectivity to non Informix DB.            |                                       |                               |
|  |  |                                     |  | Really miss a Windows version.                 |                                       |                               |
| <b>If you have no experience with...</b> | Worth having a look if you are comfortable in C compilation. | One of my favorites in the past.    | Interesting tool, definitely worth a look. | Only if you need the reference implementation. | Definitely the environment to follow. | Great all-in-one tool.        |

|  | Aubit4GL   | BDS                              | Genero  | Informix I-4GL        | Rational EGL  | Hydra4GL  |
|--|--|----------------------------------|---|-----------------------|---|---|
| <b>If you want to switch from I-4GL...</b> | Too early for prime time alone, get assistance from Aubit Computing. | Choose another one...            | To benefit from the extensions, you leave the pure I-4GL world. Some call this a "migration". |                       | Provides great value but migration is a one way trip. | Efficient tool, very high level of compatibility. |
| <b>If you want to build batch apps...</b>  | Great.   | Great.                           | Great.  | Great.                | Great.  | Great.  |
| <b>If you want to build web apps...</b>    | NC   | One of my favorites in the past. | NC  | Poor. Not a web tool. | One of the best tools in its category.                | Efficient approach to the web. Not a web tool.    |
| <b>Sales Model</b>                         | Optional (but recommended) support.                                  | Compiler + Run time.             | Compiler + Run time.  | Compiler + Run time.  | Development Environment.                              | Compiler + Run time.                              |

## Third Party Solutions

The fact that third party solutions exist, on top of 4GL and across vendors, make the 4GL market more dynamic. Unfortunately, as of now, I have found only one company.

**MoreData**, a Portuguese company based in Lisbon, offers a very interesting solution that can be plugged on top of any 4GL application and expose Web Services or JCA (Java Connector Architecture). The solution is independent from any compiler / run time environment, has been thoroughly tested and is now in production in a large telecommunication company.

## Trends

The trends I describe here are personal conclusions I have drawn from the tremendous feedback I got from users, from many discussions with many users, vendors, by attending seminars, webcasts, trainings on the different matters and from my experience in the industry (which specifically on 4GL started in 1997).

## **Aubit4GL**

Aubit4GL is maturing slowly but surely. It needs more volunteers to make it a really successful project.

## **Four J's Development Tools**

Four J's is now 100% focused on Genero. Genero users with a 4GL background love the product. They recently announced support for more complex widgets and more complex ergonomics, which include tree widgets, multiple dialogs active simultaneously, etc. Those enhancements will be available as part of Genero version 2. They are really expected by many users.

However, Four J's has an unfortunate reputation of not delivering what has been announced: tree widgets were initially announced in 1998, a lot of users are still waiting for the Genero print client<sup>7</sup> and many remember the recurrent announcements of an IDE. Some are also curious about what are Four J's plans as they started selling their own database, Genero db.

A good side of the company is that it is developing relations with other companies to make its offer more interesting and global like testing with Froglogic or database support with ANTs.

However, their long tradition of innovation has been reduced in the last years.

## **IBM**

IBM will continue to support I-4GL and provide fix packs that will provide enhancements, such as support for thread-safe generated C code, and update platforms and operating systems, as required.

On the EGL front, things are moving in the right direction. The migration tool from I-4GL to EGL is maturing at every release. It is not perfect and does not convert 100% of your code but it is not its goal. Unfortunately, being under Non Disclosure Agreement (NDA) with IBM, I cannot say or write much more. But watch for more great enhancements in the first quarters of 2007!

## **Querix**

Querix is also very discreet and cautious about their announcements.

---

<sup>7</sup> Which a respondent qualified of "vapourware" (note the spelling).

Hydra4GL v4.2 will be updated by v4.3, currently in beta testing. Version 5 should be out by year's end and will include P-code (cross platform portability).

## Conclusion

It started when I was a kid, and until recently, there was a show on the French TV called "*L'école des fans*", the fans' school. Kids were singing an invited artist's songs. At the end, they all got a mark. They always got 10, the best mark. It is difficult to give marks. I did not ask marks from my panel and I do not want to give any. I preferred to give my best bets based on what you are trying to achieve; following the "choose the best tool for the best job" paradigm. So my best bets, as of December 2006, would be:

- If you want to achieve a Service Oriented Architecture (SOA), **Rational EGL** might be the best pick, followed closely by **Genero** and some third party solutions.
- If you want to build web applications, use **Rational EGL**, it provides all you need and even more. Building web services, web applications and reports is very easy with the IBM tool.
- If you have batch applications and want to keep doing batch applications connecting to an Informix database, keep what you have. None of the tool will add significant value to provide a good return on investment.
- If you want to build Windows applications, the fastest and less intrusive way is by using **Hydra4GL**, the choice between **Genero** and **Hydra** will then be on features and extensibility.
- If you want to build Mac applications, you also have the choice between **Genero** (using the Mac client) and **Hydra4GL** (using the Java client).
- If you want to build text applications, keep what you have.
- If you are looking for a cost effective text solution, go either to **Aubit4GL** or to **Rational EGL**.
- If you are looking for the highest database independence, choose **Hydra4GL**.
- If you need support for Web Services, choose **Genero** or **Rational EGL**.
- If you need support for the highest level of security, choose **Hydra4GL** or **Rational EGL**.
- The best support is available on **Aubit4GL** and **Hydra4GL**, from respectively Aubit Computing Ltd and Querix.

- The best “integration with the outside world” (without web services) is really **Hydra4GL**, **Rational EGL** and **Aubit4GL**.

So now, who will be the first to announce support for Microsoft Vista?

## Additional Resources

### Aubit4GL

#### Commercial Support Web Site

See: <http://www.aubit.com>.

#### SourceForge Community Web Site

See: <http://aubit4gl.sourceforge.net>.

### Four J's Development Tools

Products: BDS (Business Development System), Genero.

#### Web Site

See: <http://www.4js.com>.

### IBM

Products: Informix 4GL, Rational EGL.

#### Informix 4GL

See: <http://www.ibm.com/software/data/informix/tools/4gl/>.

#### EGL on developerWorks

See: <http://www.ibm.com/developerworks/rational/products/egl/>

#### EGL RedBooks

See: <http://www.redbooks.ibm.com/cgi-bin/searchsite.cgi?query=EGL>.

## Querix

Products: Hydra4GL.

### Web Site

See: <http://www.querix.com>.

### Querix Community Forums

See: <https://www.querix.com/forums>.

## MoreData

### Web Site

See: <http://www.moredata.pt>.



This study was realized by:

**Jean Georges Perrin**

10 rue Saint Wendelin  
67500 Haguenau  
France

Tel. +33 6 10 99 35 55 (CET)

e-mail [jgp@jgp.net](mailto:jgp@jgp.net)

Copyright © 2006, Jean Georges Perrin.