IBM Information Management software

Converting Informix 4GL Applications to Informix Genero

Why Convert?

Your Informix 4GL applications are fundamental to your business; too important to disrupt, too costly to rewrite and packaged applications simply don't cut it.

With Informix Genero, important applications can be rendered for the 21st century quickly, painlessly and cost effectively.

By upgrading your application to Genero you will:

- Save money on needless re-development
- Protect tens of man years worth of investment in domain specific development
- Protect your domain knowledge resources and competitive edge
- Adapt applications rapidly to changing market needs and business models
- Give users the state-of-the-art user interface they deserve
- Run applications across the desktop, web or smartphone concurrently
- Improve developer productivity
- Support industry standard databases including IBM DB2 and IBM Informix...
- Run apps indifferently between Unix, Linux or Windows without recompilation
- "One binary fits all": virtual machine architecture means compile once, test once, run anywhere
- Reduce the size of your program executables

Conversion Options

The most valuable investment made in the development of any application is human; the intimate knowledge of your company's business and the effort needed to transcribe that on an ongoing basis into software processes is what differentiates you from the crowd.

Depending on the amount of time and resources at your disposal, you may adopt one of two approaches.

1. **Short term** - simply recompile the existing application with Genero and make minor 'tweaks' to the user interface. Adopt this approach if you need a solution in a matter of weeks.

2. **Medium-long term** - unleash the full power of Genero; add drag and drop, web services, business graphics, and browser styles for a minimum of disruption with maximum effect. Expect a solution in several months.

If you have limited resources and seek a short term solution:

Your application will closely resemble the current one with information organized into screen forms of 25 rows by 80 columns. This is normal; your forms layouts respect the structure of IBM® Informix® 4GL. With a few minor modifications, widgets can be added to embellish the look and feel, but the structural field layout will be the same. The mouse is supported transparently - there is no need to program anything.

If you have the resources and want to invest in a long term solution:

You want to rethink the ergonomics of your application. When you first developed your application, graphical objects such as tabs, folders, date, combo boxes, radio boxes, tree widgets, pictureflow, multiple dialogs, the worldwide web, drag and drop didn't exist.

Resize tables, sort/hide columns, perform conditional input and display business graphics over the Internet without the need to write code.

The ability to translate forms and add security templates will also dramatically reduce the number of forms your application calls upon. The end result will be a state-of-the-art graphical application.

Considerations for Converting I-4GL/D-4GL to Informix Genero

Exploit Cross Platform Widget Sets to the Full

Genero introduces a new way to render screen forms by representing them logically as an XML tree. This method enables the logical abstraction of the business logic from the physical implementation of the client technology. As a result, applications may execute across multiple client technologies and fully exploit the local widget set, whether it is Windows, Linux, HTML or Java. User 'A' may be connected to Windows; User 'C' to a browser; while User 'D' is using a Linux or Motif workstation.

IBM® Informix® 4GL compatible

Almost all of the language key words are the same and in many cases a simple recompilation of your I4GL source code with Genero will still work. But the result will lack panache. To unleash Genero's power, form definition files and some parts of the code can and should be modified.

Improve the ergonomics, reduce code length

As you review the design and ergonomics of your application, you will find it possible to reduce the number of forms and the code length - in some cases dramatically. Whereas in the past with IBM Informix 4GL, application presentation was embedded in the business logic, the use of an XML architecture with Genero can reduce code length by anywhere between 50% and 75%. The result will be tighter more reliable code that executes faster than ever before.

Conversion check-list

Are you ready for Genero? The following is a check list of items to be reviewed to execute a successful 4GL conversion. Review your I4GL application to identify each of the items listed below.

'C' functions
If 'C' functions are used in your I4GL application, they will need to be reviewed. Many of them can be linked 'as is' to the Genero run time system, some will need minor adjustments, and others may need to be rewritten. This can be done Genero's I4GL equivalent - Business Development Language (BDL) - given the many system level language extensions we have added.

'C' functions that interact with the user will however need to be rewritten in BDL. The Genero run time system does not support them. This means that 'C' functions using libcurses will need to be rewritten too. Using 'C' functions in an application impairs its portability and we recommend to rewrite as many of them in BDL as possible.

FGL_LASTKEY / FGL_KEYVAL / FGL_GETKEY functions
FGL_LASTKEY, FGL_KEYVAL and FGL_GETKEY functions are supported for compatibility reasons and can be used in BDL applications. It is highly recommended to review parts of code that call these functions.

Using INPUT ARRAY to display arrays
In many I4GL applications, developers have used INPUT ARRAY statements with hidden fields to work around DISPLAY ARRAY statement shortcomings. Such workarounds forced developers to write a lot of unproductive code and emulate all permutations of end-users actions, hiding the INPUT ARRAY features to make it resemble a DISPLAY ARRAY.

Genero corrects this and provides a DISPLAY ARRAY statement to avoid such 'workarounds'. If you use such methods, we recommend you review and modify this code.

Menus
I4GL menus may look good in ASCII mode but they rarely look good graphic mode. Most of the time, menu functions need to be rewritten for this reason. This is purely an aesthetic consideration and if your menus are acceptable to your user community, they can be left until a later date.

FUNCTION definitions
The I4GL compiler allows the developer to define a function with the same name several times. Depending on the I4GL version, the first or the last FUNCTION definition in the stack will be used. The result cannot be guaranteed. Genero imposes a stricter regime and does not allow loose coding of this nature. The compiler generates an error when a FUNCTION is defined more than once.
ERROR(-6203):Module : The function has already been defined in module

FUNCTION arguments definitions
The I4GL compiler does not take into consideration how many arguments are passed to a function and how many returning arguments are defined. Genero does and will generate an error.

ERROR(-6200):Module 'Module Name': The function .(,) will be called as (,).

External file management
I4GL does not include external file management commands. To work around this defect, C functions are used. The BDL language now includes channel commands that can be used to read, write or append data from/to external files. ‘C’ functions can be maintained in the first step of the conversion but the update of these ‘C’ functions to BDL functions need to be scheduled.

P-Code / C-Code
I4GL allows developers to choose between generating P-Code and C-Code. Genero generates P-code only and performance testing shows that it delivers equivalent performance to I4GL 'C' binaries given its more modular and non-monolithic structure.

FORM files (.per files)
To enable the support of heterogeneous client devices from a single source code, FORM file logic will need to be revisited. I4GL FORM formats are still supported, but they have limited graphic capabilities. Choose the most important data entry forms for ergonomic rework first. Less significant FORM files can be compiled ‘as is’ to save time.

FGL_GETKEY
The FGL_GETKEY function is no longer supported in Genero. Code segments calling this function need to be rewritten in a more abstract way.

DISPLAY … AT … statement
The DISPLAY AT statement displays a variable at a given position. This given position is defined on a character-by-character grid. As Genero forms are no longer defined on a character-by-character mode, this command can no longer be used. The command is still supported for compatibility reasons but the position on the display is no longer ensured. DISPLAY AT commands must be replaced by other commands like DISPLAY TO.

ON KEY statements / MENU COMMAND KEY statements
The ON KEY statement defines a shortcut key in a dialog. Genero introduces the MVC (Model/View/Controller) concept in 4GL. The ON KEY command is still supported and can be

kept during the first phase of conversion. But the ON KEY and the COMMAND KEY must be replaced by the ON ACTION statement.

Graphical Menus
'Beauty is in the eye of the beholder' as the saying goes. Genero offers new ways of displaying menus and by using style sheets; changing the look of a menu has never been easier.

- 8005 Error number
The -8005 error number is displayed at compilation time when features that are no longer supported (deprecated) are used. This tip helps you to find out the last deprecated features that are still in the code after the conversion.

Operating system
Check the list of available ports at [link to the list of available ports] to see if your operating system is supported.

40 Mb disk space
According to the type of installation, the required disk space varies. The runtime environment needs less room than the development environment. During the installation, about 15 Mb are needed in the current temporary directory.

A network card
A network card is not absolutely necessary. It is recommended if you want to test your application in an n-tier architecture.

About 2 to 6 Mb memory per user
The amount of memory needed will change according to the Operating System, the database engine and your application. Each DVM process takes 2 to 6 Mb. For example, a typical requirement for a 30-user runtime environment with an Informix database server is a 500 MHz processor with 512 Mb RAM.

A graphical user environment
Either a Unix graphical environment (X11, …) or Windows or Mac is needed to run a graphical client. An Internet browser and a web server may be needed too if the application is expected to run over the Internet.

An Internet connection
During the licensing process, you will have to validate the license keys on Four J's web site. The machine you are installing does not need to be connected to internet. You can use a nearby machine for that.

'C' Linker
A runtime system or 'runner' is needed to execute programs. It is automatically created during the installation phase. Depending on the database version and the operating system, this runner

may need to be created manually. To do so, a ‘C’ linker is needed. The ‘C’ linker is only needed on the development machine and once per operating system.

ANSI ‘C’ Compiler
If your I4GL application calls C functions, these functions will need to be recompiled using an Ansi C compiler.

Database engine
The database you use with I4GL will work with Genero. To simplify the installation, it is highly recommended to install Informix CSDK including ESQL/C. Always prefer the latest version of CSDK when installing it.

Genero documentation
Genero documentation will be very helpful during the installation of the product and during the conversion phase. The documentation includes also a list of new features, which may help you in modifying your application.

Post Conversion

As with any major product conversion, the resulting project needs to be thoroughly tested before deploying. It is recommended that both I4GL and Genero app are tested side by side to make sure all the 4GL screens rendered correctly in Genero. More technical details about the conversion may be found under "Publications for IBM Informix Genero":
http://www.ibm.com/support/docview.wss?uid=swg27020967

IBM Corporation
Software Group
Route 100
Somers, NY 10589
U.S.A.

Produced in the United States of America
March 2011