

# The Business of Moving to Omnis Studio

An Overview for Users of Omnis 7 Products

By Mark Phillips  
August 2002



© TigerLogic Corporation 2009. All Rights Reserved.

## Table of Contents

The Business of Moving to Omnis Studio .....	1
An Overview for Users of Omnis 7 Products.....	1
Preface.....	3
Introduction .....	4
A View of Omnis History .....	4
Omnis Classic .....	4
Success and Evolution.....	4
Extending the Product Design .....	5
Object Oriented Analysis and Design.....	5
Enter Omnis Studio .....	6
Omnis Studio Advantages.....	6
Increased productivity in development .....	6
Flexibility in deployment .....	6
Web Applications .....	7
Linux .....	7
MacOS X.....	7
Graphical User Interface .....	7
Getting new customers.....	8
Reduced learning curve.....	8
Rewrite or Migrate?.....	8
Rewrite .....	8
Omnis Studio vs. Java .....	9
Omnis Studio vs. Visual Basic .....	9
Migrate .....	9
Object Oriented Expertise .....	9
Training .....	9
Ready-made Code Libraries .....	10
Conversion Tools .....	10
The Business Case for Omnis Studio .....	10
Market Research is Key .....	10
Arguments Against Migration to Omnis Studio.....	11
Why would I move if my app is selling well and it is profitable?.....	11
My application is not selling well and I can't afford to make the move.....	12
It is just a mountain of work, I have not thought about some of those routines in years, and I just don't want to do it.....	12
Conclusion.....	12

## **Preface**

I would like to thank everyone at Mitford House and at TigerLogic for assisting me in the creation of this white paper. In particular, I would like to thank Birgit Jaeger and Andreas Pfeiffer for their contributions to the text. I could not have produced this without the invaluable technical insights and examples provided by Bob Whiting, Richard Mortimer, Peter Kelly, Chris Stevens, Thad Bogert, Leon Ventor, Chris Clark, Damon Biddle, Paul Squire and Walter Venable. Finally, I want to thank Andrew Smith for being a patient and potent editor, and David Lewis for creating the forum to present the material at the OzOmnis developer conference in August 2002.

## Introduction

When I attended the AmerOmnis conference in 2002, I was truly stuck by a remark made by a developer concerning the issue of migration to Omnis Studio. Prior to that moment I assumed the reasons for a move to Omnis Studio were obvious. I had yearned for an object-oriented 4GL a la Omnis long before OOP became a popular acronym. This interest in

*“It is the responsibility of Raining Data to provide a business case for Omnis Studio.”*

**A developer at the 2002 AmerOmnis Conference**

Object Oriented Analysis and Design plus my participation in the beta program of Omnis Studio must have eased my move to Omnis Studio. I recall thinking of the first release of Omnis Studio that it could not happen fast enough for me. Yet, as I thought about the developer’s remark, I realized there is something quite interesting in their statement.

## A View of Omnis History

To better grasp where Omnis Studio is today it helps to understand the history of Omnis Studio and its predecessor, Omnis 7. Please forgive any errors in this account. This is not a carefully researched history but a brief overview based on my recollection.

### ***Omnis Classic***

Omnis 7 (aka “Omnis Classic”) was a killer application in its time. It provided an accessible solution to many people experimenting with software and an example to many competitors. It is interesting to note that Microsoft was a licensee of Omnis for some time. Many successful applications have been built, some quite large, using the tool. In the early days, Omnis appealed to amateur computer users who built useful products but were not “programmers” in a strict sense nor studied software engineering at university. Omnis 7 was a “fourth generation language” (4GL) with a unique approach to programming. With the release of Omnis 7, the fifth major product release of the technology, Omnis became a player in the Client/Server market and so gained a new audience among corporate and academic users. It was popular with MIS groups as well as those who needed to apply technology to problems outside the mainstream MIS missions.

### **Success and Evolution**

Success is a wonderful thing but it can have its drawbacks. Over time a staggering number of enhancements were submitted by Omnis users. Mitford House (the main Omnis development center, located in the UK) faced a long list of often-conflicting enhancement requests. The enhancements tested the limits of the product.

For example, each Omnis Classic command has its own set of parameters and, in many cases, several depending on the command options. Modifying the layout of editors and reconciling the variations in behavior became a time-consuming task. Moreover, the command set had been extended and amended over the years. There was redundancy and inconsistencies between commands.

The upshot was code maintenance was preventing aggressive action on new feature development. In short, the simple editors that made Omnis attractive became obstacles to implementing the enhancements.

### **Extending the Product Design**

Omnis 7 significantly extended the product design over Omnis 3 and Omnis 5. The addition of Format Variables, Field Parameters, etc. plus the elaboration of the concepts of “procedures and attributes” and the “dot notation” to manipulate these elements were introduced to accommodate the needs of users. These new features of the product mitigated the need for refactoring the user interface for every enhancement and the Omnis developer was given a host of new ways to enhance their applications.

The initial release of Omnis “dot notation” shifted the Omnis programming environment toward the kind of programmatic agility often associated with C/C++ or Pascal. The interactive debugging and forgiving nature of the 4GL environment made Omnis 7 an interesting and popular product.

Yet, in the broad view the tool remained limited by its own architecture. There were a number of issues: Omnis Classic uses a unique and proprietary program syntax that is powerful yet inconsistent when compared to other widely used programming languages. The command set had become over extended and confusing. Many of the commands used the global memory structures that had been carried forward for historical reasons. Command editors had slight variations of behavior. Moreover, the product contained many code segments that were far too old to be maintained in a reasonable way. Taken together, all of this was pushing the product into a fragile state. It became clear to the Omnis designers that the toolset must be reorganized.

### ***Object Oriented Analysis and Design***

Omnis 7 greatly extended the Omnis product line. At the time of its release there was a major change in the software tools market. This change had a tremendous impact on the product design for many software houses.

At the time of this change (mid 1990s), structured programming was tremendously successful. Client/server was king but the cost of maintaining large software systems was rising, not falling as predicted. Moreover, folks with large brains were writing tales of major disasters in software. The Microsoft Excel project was rumored to be flat on its face, the integration of hotel reservations into the airline system was in tatters and other examples of failed and floundering projects littered the landscape. In a nutshell, the problem was chaos; large software systems demonstrate a chaotic behavior that stymied traditional designs.

The software industry as a whole faced an unsettling epiphany: What if truly large-scale systems could not be created successfully and operated reliably? There was, and remains, ample evidence that the structured programming architecture simply cannot handle complex projects. Now that was a very troubling thought for those who must justify the cost of projects.

After an incredible amount of argument and pontificating, Object Oriented Analysis and Design (OOAD) rose to the top as the way to deal with the chaotic nature of large systems. Today, the object oriented design theory has all but replaced the many structured programming methodologies. However, structured programming is by no means dead.

Rather, it is an integral part of OOAD. It can be found in all OOAD projects albeit at the lower levels of implementation.

### ***Enter Omnis Studio***

The development of Omnis Studio began circa 1993 as a series of papers passed among the denizens of Mitford House. Its feature set was designed to address each of the areas discussed above, with a goal of making a product which would appeal to the existing user base and also compare well with products and tools that support OOAD. The Omnis Studio product has significant advantages over Omnis 7, C++, Java and Visual Basic. It is easier to learn than Java and C/C++, offers a rich environment and allows developers to build applications quickly. It is far more flexible than all but the most recent rendition of Visual Basic; the new Visual Basic has yet to earn its credentials but that is a subject for another discussion. It is quite right to say that Omnis Studio has attracted a brand new group of developers.

***The Omnis Studio product has significant advantages over Omnis 7, C++, Java and Visual Basic. It is easier to learn than Java and C/C++, offers a rich environment and allows developers to build applications quickly.***

## **Omnis Studio Advantages**

The following paragraphs highlight the significant advantages of Omnis Studio when compared to Omnis 7. To some degree these advantages also relate to the Omnis developers' sales strategy, who on many occasions have to "sell" the benefits of Omnis Studio to their customers.

### ***Increased productivity in development***

Time is money and time-to-market is critical nowadays. With object orientation come features such as inheritance, reusable objects, and data encapsulation. Developers who take advantage of these design techniques are much more productive than with procedural development as in Omnis 7. Common code techniques in Omnis 7 simply do not offer the elegance, speed and level of generalization that can be attained in Omnis Studio.

In general, fewer lines of code will be required for any given task using Omnis Studio. This translates into lower cost of maintenance and faster response to user requests. In some of the projects I have reviewed as much as a 30% reduction in the number of lines of code to implement a unit of work was realized.

### ***Flexibility in deployment***

When you implement an object-oriented design and follow the 3-tier model of Model-View-Control, you are able to deploy your solution in four ways without changes to the application implementation. You can deploy stand-alone, client/server, HTML and Web Client with few or no changes to the business logic. In this fashion, you can meet your market with the solutions it demands very quickly. As new interface requirements are defined, you simply implement the interfaces and reuse the business logic.

## ***Web Applications***

There are still a great many Client/Server applications in the market and there will always be a need for this type of application. That said, most new projects are planned for the Internet. At the very least, your customers may want the option to deploy over the web. The old Omnis 7 Web Enabler is no longer a serious “state of the art” solution for the web deployment. The combination of HTML and Web Client technology of Omnis Studio gives the optimum possibilities to developers.

## ***Linux***

Germany is the strongest Linux market at the moment, but Linux is a growing market segment in many countries. It has been accepted as a serious business environment, not only for the web but also in every server role. It is important for developers to offer the option to deploy applications on Linux as well as on Windows and Mac.

This is one of the strongest sales arguments, even if your customers may not be using Linux already. It will give them better protection of their investments and a better business perspective if they have the option to go to Linux without re-designing their applications.

## ***MacOS X***

All new Macs come with OS X installed. Customers will only accept Omnis 7 applications running in Mac OS Classic mode for a short time. For these users, not making much use of OS X will quickly become a major issue. Moreover, Apple's Mac OS X strategy appears to be related to the Linux market, as well as the J2EE and .NET market initiatives. For developers who support the Mac OS, these technologies must be considered in the long term.

Omnis Studio also has the benefit of supporting the Aqua interface. This is a key feature in terms of offering a modern look and feel.

## ***Graphical User Interface***

When selling applications, the Graphical User Interface is quite important. You cannot successfully sell applications with an outdated GUI anymore, i.e. without tab panes, sidebars, icon arrays, dynamic lists. For this you need Omnis Studio. I know that several customers have developed their own sort of tab panes and other GUI elements in Omnis 7. Doing this is hard work and it takes a lot of time, which costs money. You can save all that by just using the ready made GUI elements in Omnis Studio. Even if you have a solution in hand, odds are the cost of maintaining these routines over the evolution of the product will be higher than relying on Omnis Studio.

Omnis Studio has over 30 web components for remote forms including: tree list, slider, roll button, progress bar, icon array, marquee, animated gif, and more. All these and more are available for desktop client applications.

## **Getting new customers**

The cost of acquiring new customers is increasing. Omnis 7 is outdated technology that cannot compete with Java, Delphi, Visual Basic and their ilk anymore. Omnis 7 looks outdated and does not appeal to customers, which in turn drives the cost of acquisition higher. One way to win new customers is to show them Omnis Studio applications, tell them about the flexibility of Omnis Studio as a platform, the better performance than Java, more robust applications compared to those written in Delphi and Visual Basic. These arguments simply do not apply to Omnis 7.

*One way to win new customers is to show them Omnis Studio applications, tell them about the flexibility of Omnis Studio as a platform, the better performance than Java, more robust applications compared to those written in Delphi and Visual Basic.*

## **Reduced learning curve**

Many customers complain that it is difficult to find Omnis developers in the market. These people will correctly point out that there are thousands of Delphi and Visual Basic programmers. We cannot solve this problem quickly, but with Omnis Studio there is a much shorter learning curve for educated developers than with Omnis 7. Omnis 7 is difficult to learn because it has its own architecture and way of programming.

We have anecdotal evidence that programmers who learned object oriented analysis at university and/or who knows Visual Basic can learn Studio in a very short time. Indeed, many computer science students and academics have said they find Omnis Studio to be significantly better than other modern toolsets.

## **Rewrite or Migrate?**

Recasting an existing application in another toolset is not a trivial decision. For this reason many Omnis developers have postponed migration. Yet, each day your customers stay on the Omnis 7 based application increases the danger that they will abandon your product and purchase a competing product that is more modern and technically advanced than you can provide with Omnis 7 these days.

## **Rewrite**

Many developers are faced with a wicked dilemma: Rewrite the application in another language or migrate the application to Omnis Studio. In the case of rewriting with a new tool set, the developer must learn a new programming environment and paradigm, then reconsider every single element of the application workflow and data processing to prepare for the creation of a new application. In some cases, this means redefining your target audience. This can be a tremendous amount of work.

The Omnis 7 developer has a leg up in this area. Omnis Studio has been in the market place for five years. In that time many developers have migrated and are happy to share their experiences. In Europe, for example, Omnis Studio accounts for over 60% of the Omnis projects in production or development today. Moreover, a large percentage of existing Omnis 7 code will work unchanged in Omnis Studio.

## Omnis Studio vs. Java

A comprehensive comparison is beyond the scope of this presentation. However, I think it is true to say that implementing applications in Omnis Studio is much easier than with Java. The GUI is far more advanced than Swing presently offers. Anecdotal evidence suggests that projects rendered in Omnis Studio require approximately 30% less time and resources to reach completion when compared with Java.

*Projects rendered in Omnis Studio require approximately 30% less time and resources to reach completion when compared with Java.*

With the advent of J#, Java programmers on the Windows platform face a similar issue. The applications they write in J# will no longer be truly cross-platform. If they stick with pure Java it is likely the OS vendor will progressively close out access to important system support. While this is not important to some, in a crowded marketplace differentiators are hard to come by and portability is a selling point.

## Omnis Studio vs. Visual Basic

Omnis Studio is an object-oriented environment that supports inheritance, encapsulation and strong data typing. Visual Basic embraces some “object like” concepts but does not support object-oriented designs. With the .NET initiative, the Visual Basic market has been thrown into confusion. Despite the up-beat marketing releases from Microsoft, the skill set of typical Visual Basic programmers does not prepare them for the new Visual Basic, VB.Net. They are faced, in many if not most cases, with a steep learning curve and a complete rewrite of their applications. Once done, the .NET interfaces preclude execution of the application outside the .NET environment.

## Migrate

There are really quite a lot of resources at the disposal of the Omnis developer that make migration a far more palatable option. More important in the migration of an Omnis 7 application is that a very large percentage of your existing code will work in the Omnis Studio environment without modification.

## Object Oriented Expertise

You may be thinking to yourself, “I don’t know anything about OOAD”. First, let me say there is nothing in OOAD that you cannot learn easily. Everything in the world is object oriented, at least from one point of view, and with only a small amount of effort you will master the concepts. That said, many leading authors recommend adding a consultant or member to your team experienced with OOAD. That is, get an experienced person on your team to help you avoid the mistakes and pitfalls of designing your first OO system. There are quite a few options for obtaining this expertise.

## Training

Training is available for Omnis Studio by way of the Omnis Technical Newsletter and teacher-lead seminars. DLA, David Swain, TigerLogic and others provide training and mentoring on a fee basis.

Ute Bevier has written a book that is said to be quite interesting and valuable. Given the high adoption of Omnis Studio in Germany, I assume the book has been a great aid there.

Finally, there are many seminars and books that explain OOA&D. Take some time at your favorite bookstore and see what you can find. If you would like a bibliography, please feel free to contact me.

### **Ready-made Code Libraries**

There are a number of options for obtaining the necessary object-oriented base code. Vencor, Computer Dynamics and others such as Joe Maus and Caroline Wilkins offer application frameworks that are available free or for purchase. These frameworks greatly simplify the migration process by providing ready to run code modules to support your business rules and GUI. The frameworks can be studied to see how the OOAD was implemented and this will prepare you for extending the framework to better suit your needs.

### **Conversion Tools**

The conversion tools have improved tremendously in the five years since the release of Omnis Studio. The tools are open so you can freely modify them to suite your projects. Today, the migration to Omnis Studio is quite straightforward and rewards the developer with some interesting benefits. Somewhere between 50% and 80% of your existing code will convert and work. Modifications to take advantage of Omnis Studio OOAD features can result in a significant reduction of the code base, in terms of lines of code, and a general improvement of program behavior. Furthermore, use of the new GUI controls can reduce training issues and increase the attractiveness of the application.

## **The Business Case for Omnis Studio**

The previous section provided some of the general reasons why you should be using Omnis Studio rather than Omnis 7. We could add many more detailed technical reasons, but the arguments above should remove any doubt whether Omnis Studio is the right choice. Nevertheless the initial question has not been answered directly. Should TigerLogic provide a business case for moving to Omnis Studio?

I propose the answer is "No." TigerLogic is in the business of building and selling database and application development tools. Our research focuses on these specific markets. While our research does have some relevance to any given vertical market, TigerLogic does not provide a market research service nor does it claim to have expertise in the various markets served by its customers. Moreover, even if such data was provided the recipient would still be required to analyze the data and draw his or her own conclusions about the meaning of it.

What TigerLogic must do is describe the value proposition and features of the products it offers. TigerLogic can suggest potential applications and markets, and it can work as your partner by providing expert opinions for specific customer requests about technical issues and proof of concept issues.

### ***Market Research is Key***

The business case for any given application is specific to the application. Let us consider a hypothetical contact management application; the figures that follow are loosely based on my experience with the Contact Ease product from Westware Inc. in the mid 1990s. For this reason, do not rely on this estimate; it is for illustration only.

In today's market it may be that a contact manager application might fetch approximately US\$29 per unit. If the subject application includes multi-user capabilities and sales

automation features, the price point might rise to US\$129 per unit. If you, as the developer, extend this product to fulfill the current dream of CRM applications, then perhaps the price point would rise further to US\$350 per unit.

So, how do we arrive at these figures? In order to provide a rational market value estimate for an application, a proper business study must be completed. There are many books on the subject but here are a few points of study that would need to be elaborated:

- Needs and requirements of the target market
- Target market's willingness or ability to pay
- Features of current products in the market space
- Price models of current products in the market space
- What needs are not met or under-served
- Known enhancements coming from current product vendors

This information is used to craft a project plan. The project plan includes at a minimum these items, not necessarily in this order:

- Complete feature set
- Business process definition
- Target software tools and hardware platforms
- Cost estimate to develop the product including how many people for how many months
- Cost to market the product and acquire customers
- Cost of technical support and service required to retain customers

With all of this information in hand a realistic estimation can be made. This estimate is used to derive the price per unit and number of units that must be sold to recoup development costs and assure a profit. At this point you can determine if the project is a good one or not.

As you can see this is a lot of work but extremely worthwhile. A few weeks of market research and business strategy planning can improve your experience tremendously.

## **Arguments Against Migration to Omnis Studio**

There are a number of objections that I have encountered when discussing the migration to Omnis Studio. Here are three of them.

### ***Why would I move if my app is selling well and it is profitable?***

You may be fine where you are. If your market is stable and unchanging then there is no compelling reason to recast your application in any other language. I must be honest in that I cannot believe that in the age of web services and personal digital assistants that this is a true assertion, but if you believe it then I respect that and respectfully suggest we head off to the pub. Meanwhile, some other developer, perhaps a real tenderfoot who has no idea of how much work this is, will go ahead and produce a new competitor and woo your customers away from you.

***My application is not selling well and I can't afford to make the move.***

This may be so. I have talked with a number of developers whose target market no longer exists or simply will not pay sufficiently well to support the developer at any price point. This is a very tough argument and it may be that you have little choice but to plan an exit strategy.

Another way to look at this is simply that you cannot afford not to migrate. If you want to preserve the business you have presently and expand into new market areas, you must evolve the application. Otherwise, some other developer, perhaps a real tenderfoot who has no idea of how much work this is, will go ahead and produce a new competitor and take your customers away from you.

***It is just a mountain of work, I have not thought about some of those routines in years, and I just don't want to do it.***

Fair enough. My reply is "Then don't". If you are happy right where you are and just the thought of migrating makes you ill, then by all means, stay put. We can meet for lunch somewhere but... Some other developer, perhaps a real tenderfoot who has no idea of how much work this is, will go ahead and produce a new competitor and take your customers away. This is the way of the world.

**Conclusion**

The decision to migrate to Omnis Studio, or any other environment, should not be taken lightly. There are several, if not many, non-trivial issues that need your attention. Nevertheless, the market is waiting for you and TigerLogic as well as other Omnis developers are willing and able to help you along. The first step is the hardest. Take it, for once you get going a world of new opportunities will be in your reach.