

Data-driven Programming

By
Johan Nel

A series of articles explaining the principles

Article 2: Hello world

November 2014

Table of contents

1. FUNDAMENTALS.....	1
2. SCENARIO	1
3. REALITY KICKS IN	2
3.1 PROTOTYPE I: GERMAN VERSION OF OUR SOFTWARE	3
3.2 PROTOTYPE II: LANGUAGE AGNOSTIC SOLUTION	3
3.3 THE EXECUTIVE DECISION	4
4. USERS IDENTIFY NEW REQUIREMENTS	4
5. CONCLUSION	5

Listings

LISTING 1:	CLIPPER - BASIC HELLO WORLD EXAMPLE APPLICATION.....	1
LISTING 2:	SOURCE CODE OF HELLOWORLDWIN	2
LISTING 3:	SOURCE CODE OF HELLOWORLDDE	3

1. Fundamentals

Well you probably looked at the title of the second article in our series about data-driven concepts and wondered what on earth has a Hello world application to do with data-driven applications, or maybe if I am smoking something not too legal... In this article I am going to try and show a couple of fundamentals that are probably a first in the [hello]world (no pun intended). We will be looking at some of the concepts of what is wrong about this 3 line application that many of us were taught in our first programming lecture that we ever had (long time ago for some of us).

Listing 1: Clipper - Basic hello world example application

```
FUNCTION Start()  
    ?"Hello world"  
RETURN NIL
```

Examining the above same application we can see that there is very little difference between the Vulcan.NET implementation, how it would be done in Clipper or in Visual Objects. The result, displaying “Hello world”, can still be achieved with 3 lines of code. The graphical display is however more cosmetic and pleasing to the eye of the user.

2. Scenario

Since we are a software development house, our top executives demand that we need to sell software. We had our Hello world application written in Clipper (the core software of the company) and all of our clients are quite happy that it provides all the functionality as per the user requirement specification.

However, during one of our executive’s social functions, client ABC mentioned that one of our competitors has produced a similar application and it is using the latest technology and they are considering terminating our maintenance contract and rather source from our competitor. Back in the office the next day a meeting is called with the IT department and a furious executive is calling the development team incapable and costing the company money, because we have an inferior product and we will loose our client base and go bankrupt soon.

The head of the IT department immediately states that the development team will go on a strategic session to determine how we can re-engineer our application and regain the competitive edge in the software market.

Strategic session and plenty of research done, the department present their vision of how the company should re-align to the latest trends in technology. It was determined that the best way to leverage our application into the next decade would be to use Visual Objects as our programming language. It was also noted that our application is a bit rigid and we need to make it a bit more flexible. Management approves the budget to purchase the new programming language and within a month after the whole development team have undergone the necessary Visual Objects training, IT presents their first version of the application to top management.

Top management is quite happy with the advancement made by IT to re-align them with the latest technology and marketing the new application (HelloWorldWin) is a great success, with orders rolling in beyond expectations.

The IT department is also quite happy since the re-engineering effort showed some fundamental flaws in the original programme, which they were able to address in the Windows version of the software, some data values inside the application was hardcoded which could lead to bugs if the programmer was not aware that it was also used somewhere else causing inconsistencies in the application (Listing 2). All agreed that a move was made in the right direction and that the benefits gained with the new Windows application will greatly reduce maintenance and support needed, allowing the IT department to focus on new software development.

Listing 2: Source code of HelloWorldWin

```
METHOD Start() CLASS APP
  LOCAL sHello AS STRING
  sHello := "Hello world"
  TextBox:Show(, sHello)
RETURN
```

3. Reality kicks in

The IT department just identified core new markets to penetrate with software in the 1 year planning horizon when the CEO walks in to announce that we are doomed, competitor XYZ have released a German version of their software and IT will have to come up with a solution that we don't miss the opportunity of expanding into this market.

Another strategic session is arranged and the IT department decided on two alternatives to investigate, it will create a new HelloWorldWin application called HelloWorldDe or based on research a viable alternative would be that according to textbooks we externalise the data in

our application and distribute one application with the datafile. After long debates it is decided that a prototype of both systems should be developed and a feasibility study done to ensure the right decision is made.

3.1 *Prototype I: German version of our software*

The prototype I team presented their solution and emphasize that it will be feasible, in context that we need to also supply other languages that the following approach be used.

Listing 3: Source code of HelloWorldDe

```
METHOD Start() CLASS APP
  LOCAL sHello AS STRING
  sHello := "Hallo Welt"
  TextBox:Show(sHello)
RETURN
```

It would be a mere changing of the value of sHello, recompile and we have a solution to our German problem. Adding a French version is also not a problem since we need to only create a programme called HelloWorldFr, copy and paste the original source from HelloWorldWin and change the value of sHello to “Bonjour tout le monde”, compile and we can in no time have a French version or a version in any language that is required.

3.2 *Prototype II: Language agnostic solution*

The prototype II team presented their solution and emphasize that they believe their solution is the answer to the problem at hand. Not only will it not create another burden on the software developers (except for a small interface in the HelloWorldWin application) since they have found a newly developed technology called String Data Management File System (SDMFS) and an administrator tool StringPad specifically designed to manage the complexity in a user friendly way. It is a newly developed methodology and does not yet have a huge follower base.

There is however an initial cost involved in this approach:

- The developers need to extend the current application to incorporate the interface to the String Data Management System;
- An expenditure to purchase the Admin tool StringPad;
- A new role in the IT department to be created, training of an existing staff member or appointment of an externally sourced String File Administrator into the newly created role.

3.3 The executive decision

After listening to the two alternatives presented, top management have numerous meetings to discuss the [dis]advantages of both alternatives. It was decided in the end that prototype I be followed since the company is on a tight budget and cannot afford the expense of a new role in the IT department or the capital outlay of the new String File Data Management System and the administrator tool StringPad. Development time is also a lot shorter than with prototype II and can be done with current available software developer staff.

The prototype I team arrange a celebration party since their exceptional cost saving idea and speed of adapting to the new challenge was selected. Over the next couple of months most of the prototype II team resign or found new roles in the company. The company profitability skyrocket and all are happy.

4. Users identify new requirements

Well all is well in the company and users see huge benefits of using the system and methods that can make them much more productive. Some of the enhancements they require are as follow:

- They require the system to say “Goodbye world” too when it exits;
- Some clients are international and they need the system to allow them to choose which language to say Hello and Goodbye to the world in;
- Clients have the requirement that they be able to interface to other international clients in their own language if different to their own;
- Clients require that they be able to manage their own Hello and Goodbye world data since the delay in adding new languages by our company and new releases takes too long;
- Some require that the system also be able to say “How are you” and allow them to communicate it to international business partners in their preferred or native language;
- A new technology called DOTNET is the new buzzword and customers and company strategy demand that software be developed in this new up to date innovative technology.

IT management is quite happy with their approach, the software is selling and the software developers are kept busy, although there are some that feel that they do not cope due to the stress of enhancing and maintaining the software, but that is part of how the industry work, you love it and grind it out, or hate it and find another job that is less stressfull.

5. Conclusion

Although this article still did not address data-driven applications directly, I believe most of us will be able to relate something about the industry in the tongue-in-the-cheek description of how things are sometimes happening. In the next article I will start with the basics of an application, the user interface and in particular the first interaction users have with a system: The application menu.