

(Inter)facing SAS®

Raymond Ebben, OCS Consulting, Rosmalen, Netherlands

1. INTRODUCTION

As a consulting company OCS Consulting develops many applications that make use of SAS® technology. The majority of these applications require a graphical user interface because the users are not programmers and/or the users need to be guided through the process based on data content, their roles or the options available within the application.

The first steps towards building such an application are to define the user requirements, select the application type, the technology in which to build the application and to write the functional and technical specification.

This paper will focus on the application selection process, the design of the user interface and the communication techniques to communicate between the user interface and the SAS® application. This is illustrated with the help of user requirements for an example application and divided in the following sections.

- Benefits of a graphical user interface;
- Application selection;
 - Application types;
 - Application technologies;
 - Application Decision matrix.
- Application development;
 - Designing the interface;
 - Communication techniques.
- Conclusion.

2. DEFINITIONS

Before going into the details of the benefits of a user interface, the process of application selection and application development, a definition of an application and a user interface as referred to in this paper are provided.

2.1 APPLICATIONS

Applications are computer programs that are designed to assist a user into performing a certain task. This ranges from applications that run with little user interaction, such as repetitive (scheduled) processes to applications which have much interaction with a user.

2.2 USER INTERFACE

The part of the application that interacts with the user is called a user interface.

A user interface can exist in many different formats. The most common formats are a graphical or a textual interface. An example of a graphical user interface would be an application such as SAS® Enterprise Guide. An example of a textual user interface would be the options of a SAS® Macro.

Each user interface type presents information to the user and processes the control sequences the user selects to control the application.

3. USER REQUIREMENTS

In this paper the following example user requirements will be used to illustrate the different steps in the application development process from application technology selection to application design.

The example application will be an application which allows users to report laboratory parameters. For this application the following (high level) application requirements are defined:

The application must:

- provide the ability to read the laboratory data which is stored in CDISC standard;
- provide the option to define report settings per laboratory parameter;
 - exclude values from analysis;
 - select the descriptive statistics to be calculated;
 - select the statistical analysis to be performed on the data;
 - select the decimal presentation of the parameter;
- provide the option to specify the report title and footnote;
- provide the option to specify the page number of the first page;
- provide the option to specify the output type (PDF/RTF/TXT);
- be accessible to the employees of the biometrics department;
- use the existing SAS macros for the statistical analysis.

The application should:

- be easy to implement;
- offer an intuitive and easy to use interface, allowing everybody in the department to use the application;
- provide the option to perform exploratory analysis on the data used for the standard report;
- provide the option to implement new statistical analysis.

The first decision to be made is to decide whether the application can be created using only SAS programs, or that the application to be developed requires a graphical user interface. To make that decision it is important to understand the benefits of a graphical user interface, because the biggest drawback of developing an application with a graphical user interface is that in most cases it requires more development time.

4. BENEFITS OF A GRAPHICAL USER INTERFACE

Before building an application with a graphical user interface it is important to verify that a graphical user interface is of added value to the application. The most common or most valuable benefits of a graphical user interface are listed below:

4.1 BROADER AUDIENCE

A SAS Macro would be no issue to provide to a SAS programmer, but a graphical user interface allows for a broader audience to use the application because users are familiar to working with graphical user interfaces; they know a button can be clicked on, a checkbox can be checked and a dropdown list can be used to select a value. No additional training is required for the general usage of a user interface.

4.2 LOWER LEARNING CURVE

A graphical user interface provides a lower learning curve for the users, because next to the fact that users are familiar with graphical user interfaces the users do not need to memorise the functionality and the settings offered by the application; all functionality is presented to the user via the user interface, and in most cases the options that can be selected are limited.

For example, a SAS macro could be offering users a parameter to specify the descriptive statistics to be presented in a summary table. Surely the macro cannot cope with every statistical presentation the user can think of. The possible input is limited. Within a SAS Macro the user needs to look up or learn the possible values, their spelling and how to present them to the macro. A graphical user interface would offer the user the list of available descriptive statistics parameters to choose from, no need for the user to remember the options or how to present them to the macro.

4.3 LESS ERROR PRONE

When given the previous example with the limited list of descriptive statistics accepted by the macro, it can also be stated that a graphical user interface is less error prone, because a graphical user interface allows control over the input and how it is offered to the macro before it is submitted. Offering a limited list to the user ensures that only correct values are passed into the macro. The application also makes sure that the values are offered to the macro in the expected order and or format. Other examples of controlling input are: allowing only numeric values to be entered, preventing submitting special characters, allowing only date values by using a calendar etc.

4.4 GUIDING USERS THROUGH THE PROCESS

One of the major benefits is that a graphical user interface allows for the users to be guided through the process. This can be based on data content, their roles or the options available within the application. Using the data and the applications' options, the user interface will offer only that which is allowed, relevant and sensible information to a user. Guiding users through the process according to their roles allows for the user interface to show or hide functionality. These options together allow for a decision tree of the complete process to be embedded within the application. This provides for an easier application validation.

4.5 EASIER TO MAINTAIN A VALIDATED STATUS

Because the user interface functions as a layer between the user and the (SAS) code, the (SAS) programs can only be executed by the users, ensuring that the code has not been modified in any way.

4.6 THE DECISION FOR THE EXAMPLE APPLICATION

When considering the benefits of a user interface and comparing them to the user requirements of the example application one requirement in particular could benefit from a graphical user interface. This is the requirement that the application should offer an intuitive and easy to use interface, allowing everybody in the department to use the application.

Because the application should have this functionality it needs to be discussed with the project sponsor if the application should be build with a graphical user interface. The other benefits of course come along for free!

5. APPLICATION SELECTION

Once it has been established whether a graphical user interface is of added value, the next step would be to decide what is going to be used to develop the application. To assist in this selection, the selection process has been broken down into several sections. The first section is the high level selection of the application type.

5.1 APPLICATION TYPE

There are two high level application types to choose from:

1. Desktop application
This type of application is an application that can be executed under the systems operating system. This paper focuses on applications that run under the Windows operating system. Examples of Windows desktop applications are: Outlook, Excel, Notepad, Paint, SAS Enterprise Guide, etc. These applications can either be installed on a clients' machine or on the clients' network as a client/server application. The desktop application includes embedded applications as described in § 5.1.1
2. Web application
This type of application can only be presented from within a (web) browser. Example of browser applications are: Webmail, Search engines, Maps and directions, Translators, etc.

Each application type has its own advantages and disadvantages. The most important ones are listed below.

PhUSE 2009

Accessibility

Desktop applications are typically installed on the users' machine or on the companies' network. Access to the machine or the companies' network is required to access the application.

To access a browser application, a computer with a browser and internet access is required. The application does not need to be installed on the users' machine or on the clients' network. This can therefore allow users from all over the world to use this application.

Security (access)

In general, desktop applications are a lower security risk than a web application as the desktop application is also protected by the security that is provided by the clients' internal network which is not publicly accessible.

Performance

The performance of a desktop application is much better than a web application. Next to the fact that a web application is pre-compiled, the most important reason for the difference is the available bandwidth which is still narrower for a web application.

Look and feel / Functionality

Desktop applications still have a more robust look and feel than a web application, also the functionality that can be provided in a desktop application is more extensive.

Implementation

Most companies have implementation policies for desktop applications, being stand-alone or client server. Therefore a desktop application can be easily implemented within the organisation. However, not all companies have an infrastructure and security in place to facilitate web applications, nor have they outsourced this facility.

These advantages and disadvantages are summarised in the table below.

	Accessibilit y	Security (access)	Performance	Look and feel / Functionality	Implementation
Desktop application	-	+	+	+	+
Web application	+	-	-	-	-

5.1.1 EMBEDDED APPLICATIONS

An embedded application is a special type of desktop application. The difference between a regular desktop application and an embedded application is that an embedded application can only be executed from within a desktop application. Examples of embedded applications are SAS/AF[®] or a SAS[®] Enterprise Guide Add-in.

Such an application type would typically be selected if the functionality of the desktop application in which the application is embedded would be utilised. As an example, when creating an Enterprise Guide add-in, this application would generate the validated result in the form of a PDF file. But the data used in the process to create the report can also be made available to the user. This would allow the user to fully utilise the functionalities available within Enterprise Guide to perform further exploratory analysis on the data.

A reason to develop an embedded application such as SAS/AF would be typically selected if the application is to be built completely using only one technology vendor.

5.2 APPLICATION TECHNOLOGIES

With the different advantages and disadvantages of the application types in mind, the different application technologies can be reviewed more easily. Presented below is a list of application technologies, grouped per application type, in which OCS Consulting has developed applications.

Desktop application;

- Borland Delphi (Legacy);
- Microsoft DotNet (recommended).

Embedded application;

- SAS/AF;
- SAS Enterprise Guide Add-in.

Web application;

- SAS/IntrNet® ;
- SAS® Information Delivery Portal;
- Microsoft SharePoint.

To assist in the decision process to select the best application technology the following high level criteria are defined and detailed. Please note that these criteria represent our high level vision and their purpose is only to provide insight into the application technology selection process. They can be used as guidance, but by no means are they intended as a definite list. Also this list needs to be revisited regularly as new or improved technologies offer new possibilities.

Accessibility

The accessibility for the individual application technologies are in principle the same as defined for the desktop and the web applications.

Security (access)

The security for the individual application technologies are in principle the same as defined for the desktop and the web applications. A minor differentiation can be made between the SAS/AF embedded desktop application and the DotNet and Enterprise Guide application. This is because the last two are developed using Microsoft DotNet and can therefore easily make use of integrated windows security such as Windows authentication.

Performance

The DotNet application and the SAS Enterprise Guide Add-in have an excellent performance. The SAS/AF application is a bit slower because it is not designed to make optimal use of multi threading and multi core processing. Web applications in principle always have a lesser performance than a desktop application. This is usually caused by the fact that the interface presented is not pre-compiled, and that is limited by bandwidth. A further differentiation can be made between SAS/IntrNet compared to the SAS Information Delivery Portal and Microsoft SharePoint. This is because SAS/IntrNet does not offer role based content or security; There is no platform integration.

Look and feel

The look and feel can be fully customised for applications using DotNet technologies (Desktop application and SAS Enterprise Guide Add-in). The Microsoft Foundation Classes are part of this technology; therefore the application can be designed to look and feel like any Windows application and by default the application inherits the Windows theme of the client PC seamlessly blending in with the other Windows applications. Within SAS/AF an interface can be designed having the look and feel of a desktop application, but is not capable of inheriting the Windows theme, therefore the application will look like "Classic Windows". Web applications are very common to the user but the look of the application can only be customised so much, and the feel is still less robust than a desktop application.

Functionality

To the SAS Enterprise Guide Add-in and the DotNet application all user interface elements within Windows are available, therefore they offer a wide range of functionality. Next to this Enterprise Guide additionally offers the built-in functionality of Enterprise Guide. The number of elements available within SAS/AF is less. For the web applications the functionality is also limited as with the SAS/AF application, however the SAS Information Delivery Portal and Microsoft SharePoint are so called "of the shelf" products offering standard functionality which can be utilised without additional development.

PhUSE 2009

Implementation

When considering implementation effort it is assumed that nothing is present other than the Windows OS, the base SAS system (Foundation and SAS Enterprise Guide) and the companies' network infrastructure.

The easiest solution to implement would therefore be a DotNet application. This is because most companies already have a policy on how to install desktop applications, also the DotNet application comes with an installer, allowing it to be installed in a few simple steps. The SAS/AF and SAS Enterprise Guide Add-in are also quite easy to install as they only need to be embedded in their host applications.

To utilise the functionality of SAS/InterNet a SAS server must be setup and the communication between the SAS server and the web server must be defined. Therefore implementation is more complex and labour intensive than installing a (an embedded) desktop application.

Finally, because the SAS Information Delivery Portal and Microsoft SharePoint are interfaces to a complete platform these require a more complex and labour intensive process to be implemented in the organisation.

Independency

(Embedded) desktop applications rely on the OS they are installed on and the version of SAS they are using. Because a desktop application uses only SAS code it is the least depending on different versions of SAS. Web applications also rely on the web browser in which they are presented.

Embedding other functionality

Within Microsoft SharePoint it is easiest to embed other functionality as the portal comes standard with most Microsoft related functionality available within the application. In general a web application allows for embedding virtually all functionality available on the web by simply linking to it.

Within SAS/AF the options to embed other functionality are limited and within a SAS Enterprise Guide Add-in most functionality available within Windows can be embedded. As with the DotNet application virtually all functionality available to Windows can be embedded. Please note that if required a browser component can easily be added using DotNet technology.

SAS platform integration

Within the SAS Information Delivery Portal the SAS platform integration comes standard. For Microsoft SharePoint SAS has developed a web part, allowing to connect to the list of stored processes available to the user logged in. Within SAS/AF and a SAS Enterprise Guide Add-in the SAS platform can be utilised, but this does not come standard. The same goes for a DotNet application and a SAS/InterNet application.

Role based functionality

Both the SAS Information Delivery Portal and Microsoft SharePoint come standard with offering role based functionality. This can be implemented in DotNet and a SAS Enterprise Guide Add-in, for instance by using Windows authentication and Windows user groups. For SAS/Internet and SAS/AF this can be developed, but no existing functionality can be used by default.

Exploratory analysis

In principle, only a SAS Enterprise Guide Add-in offers the functionality to perform exploratory analysis on the data processed by the application.

Version control

A version control mechanism comes standard when developing with Microsoft DotNet (Desktop application and SAS Enterprise Guide Add-in). For all other applications a version control system needs to be defined and used.

Single technology vendor

One of the technology requirements can be that the application can only be build using one technology vendor. In this case only the SAS Information Delivery Portal, SAS/IntrNet and SAS/AF remain an option. However the SAS web applications use other technologies like the web server, but these are supported by SAS as part of the SAS platform.

PhUSE 2009

5.3 APPLICATION DECISION MATRIX

The criteria mentioned above can be best presented in an application decision matrix. When a weighing factor is applied to each of the applications' user requirements, the application decision matrix can be used to derive the application technology best suited for the development of the new application.

	Desktop application	Embedded application		Web application		
	DotNet	SAS/AF	SAS Enterprise Guide Add-in	SAS/Intrnet	SAS Information Delivery Portal	Microsoft SharePoint
Accessibility	--	--	--	++	++	++
Security (access)	++	+	++	-	-	-
Performance	++	+	++	-	--	--
Look and feel	++	+	++	-	-	-
Functionality	+	-	++	--	+	+
Implementation	++	+	+	-	--	--
Independency	++	+	+	-	-	-
Embedding other functionality	+	-	+	+	+	++
SAS platform integration	-	+	+	-	++	+
Role based functionality	+	-	+	-	++	++
Exploratory analysis	--	--	++	--	--	--
Version control	++	-	++	-	-	-
Single technology vendor	--	++	--	+	+	--

Please note that this table is just created to provide insight into the decision process and needs to be updated and extended regularly. As an example, if the most important user requirement is to have to have worldwide accessibility, but the application requires complex user interface functionality, other technologies can be looked at to solve this dilemma. In this case technologies like Microsoft Silverlight or a portable desktop application communicating via SOAP could solve this issue.

5.4 THE DECISION FOR THE EXAMPLE APPLICATION

When reviewing the applications user requirements there are a few user requirements that can help in deciding the best technology for the user interface.

These requirements are:

- be accessible to the employees of the biometrics department;
- be easy to implement;
- provide the option to perform exploratory analysis on the data used for the standard report.

The other requirement to take into account is to determine the functionality required.

When reviewing these criteria the following conclusions can be made:

- Because security is important the application should only be accessible from within the biometrics department, therefore a DotNet solution is preferred over a web application. This provides the end users with a familiar look and feel while providing sufficient accessibility.
- Because it should be easy to implement a desktop application is preferred.
- Because the option to perform exploratory analysis is a requirement a SAS Enterprise Guide Add-in is the best candidate.

When looking at the functionality to be developed in this application, it would be no issue to develop the application in Enterprise Guide. Of course an actual decision process would require a more detailed analysis and needs to be discussed with the project sponsor.

But in this example the best suited technology would be to develop a SAS Enterprise Guide Add-in.

6. APPLICATION DEVELOPMENT

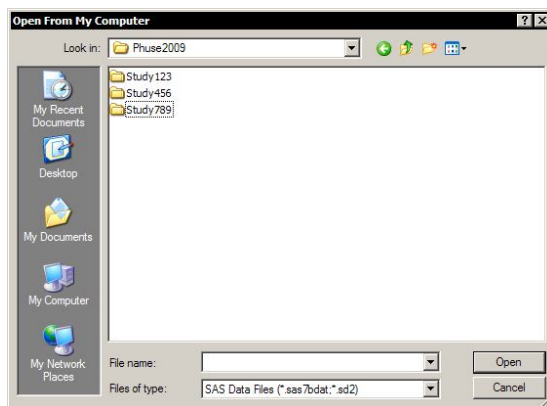
Once the technology in which to develop the application has been decided upon, the interface can be designed, because now the functionality that can be used to design the interface is known.

6.1 DESIGNING THE APPLICATION

The user interface should act as a layer of simplicity over the applications complexity and therefore should be carefully designed. The first step to designing the application is to break down the application in logical components. In the case of the example application the logical components would be: Input and input processing, defining analysis and report parameters, report processing and output, exploratory analysis

6.1.1 INPUT AND INPUT PROCESSING

The example application requires the users to select laboratory data in CDISC standard. Because the application will be developed as a SAS Enterprise Guide Add-in we can easily use the standard Windows Open file dialog to allow the user to select one or more input datasets.



This data is then processed by SAS to ensure that the data is indeed laboratory data in CDISC standard and derive the information required by the user interface, for example the list of different parameters to be analysed.

6.1.2 DEFINING ANALYSIS AND REPORT PARAMETERS

Once the laboratory data has been selected and processed the application should offer the option to define report settings per laboratory parameter. For each laboratory parameter the settings the user must be able to define are:

- exclude values from analysis;
- select the descriptive statistics to be calculated;
- select the statistical analysis to be performed on the data;
- select the decimal presentation of the parameter.

Next to the laboratory parameter specific setting the following general report setting must be available:

- Specify the report title and footnote;
- Specify the page number of the first page;
- Specify the output type (PDF/RTF/LISTING).

Therefore the application can be best divided into two sections, an overall section and a laboratory parameter specific section. The easiest and most intuitive method would be to present a user interface that generates a tab for each laboratory parameter encountered in the input data. In this tab all the parameter options and the individual values for the parameter are presented. Outside the individual tabs would be an area presenting the general report settings. The following image (created with Microsoft Visio) illustrates what the application could look like.

Laboratory parameter reporting

Creatinine | Platelets | Red blood cells | White blood cells | Albumin | Lymphocytes

Imported data

Excl	USUBJID	LBSEQ	LBORRES	LBORRESU	LBORNLO
☐	XXXXX-AU_013-205002	1	65	umol/L	0
☐	XXXXX-AU_013-205002	2	55	umol/L	0
☐	XXXXX-AU_013-205004	1	65	umol/L	0
☐	XXXXX-AU_013-205005	1	60	umol/L	0
☐	XXXXX-AU_013-205005	2	65	umol/L	0
☐	XXXXX-AU_013-205006	1	75	umol/L	0
☐	XXXXX-AU_013-205006	2	75	umol/L	0
☐	XXXXX-AU_013-205007	1	65	umol/L	0
☐	XXXXX-AU_013-205007	2	65	umol/L	0
☐	XXXXX-AU_013-205008	1	50	umol/L	0

Select descriptive statistics to report

☒ N
 ☒ Standard Deviation
 ☐ 1st Percentile
☒ Mean
 ☒ Minimum value
 ☐ 5th Percentile
☒ Median
 ☒ Maximum value
 ☐ 95th Percentile
☒ SUM
 ☒ Most Frequent value
 ☐ 99th Percentile

Select the statistical analysis: <Select the statistical analysis>

Specify the presentation decimals: 1

Define report settings

Specify report title: <Title>

Specify report footnote: <Footnote>

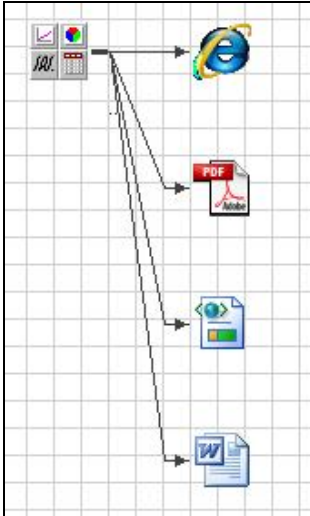
Specify the start page number: 1

Cancel OK

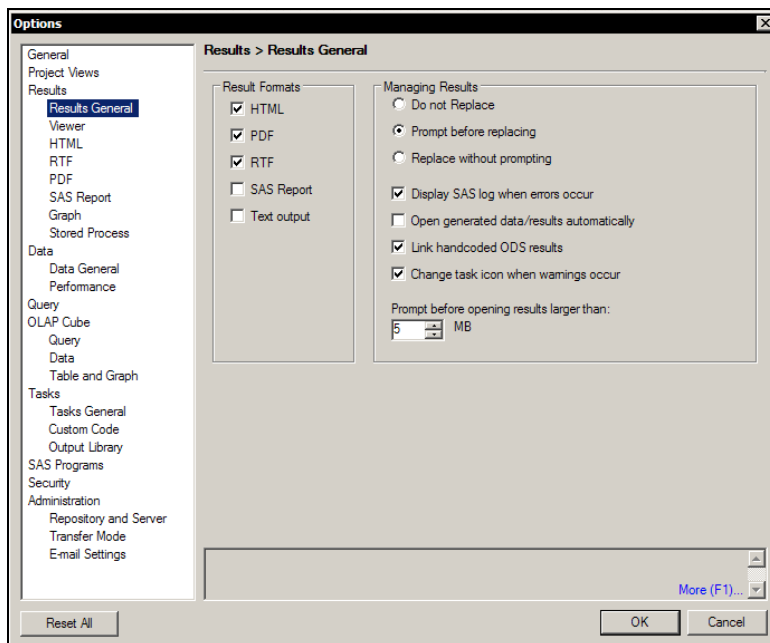
Ready

6.1.3 REPORT PROCESSING AND OUTPUT

Once the report settings have been specified by the user, these parameters can be submitted to the SAS program creating the output. In this case the code is submitted by the application in SAS Enterprise Guide. Therefore the output would be presented as with any SAS Enterprise Guide application in the process explorer Window. An example of this is presented below showing several output options:



One of the user requirements was that the user should have the option to specify the output type. This functionality comes standard within SAS Enterprise Guide. This can be specified in the menu "Tools" -> "Options". This opens up the SAS Enterprise Guide Options screen. In the Options screen "Results" can be selected where the user can specify the required output type.



6.1.4 EXPLORATORY ANALYSIS

The final step would be to allow the user to perform exploratory analysis on the data used by the application for reporting. As the code was submitted via Enterprise Guide, all the data which has not been cleaned up by the application used in the standard analysis program will still be available within Enterprise Guide for further exploratory analysis.

6.2 COMMUNICATION TECHNIQUES

The final step in the design process is to define the communication between the user interface and the SAS code. The key principle is to view the user interface as the layer between the user and the SAS code. In case there would be no user interface a SAS program would be created where the user can specify the applications' parameters and then execute a SAS program or Macro. The communication between the user interface and the SAS code should be treated no different.

6.2.1 INPUT PROCESSING

After the user has selected the files in the Windows File Open dialog, the Add-in submits SAS code to SAS Enterprise Guide:

The code that would be submitted by the Add-in could look something like this:

```
001  /*****
002  **
003  * Input file location and filename(s)
004  *****/
005  */
006  %let InputFolder=C:\PhUSE2009\Study123\;
007  %let InputFiles=SASDataSet1.sas7bdat|SASDataset2.sas7bdat;
008
009  /*****
010  **
```

This is the SAS Code that would be submitted by the user interface to SAS. The output of this program would be one SAS dataset containing the unique names of the available lab parameters. From this the tabs are generated. Next to this a dataset for each individual parameter is generated and this dataset is presented to the user in the data grid presented on the tab for that specific parameter, allowing the user to apply exclusions per parameter.

6.2.2 REPORT PROCESSING

When the user is finished defining the settings per parameter and the general settings for the report, the user interface submits SAS code to Enterprise Guide offering the users settings to the program processing the data.

The code that would be submitted by the Add-in could look something like this:

```
001  /*****
002  **
003  * Report Settings
004  *****/
005  */
006  %let ReportTitle=%str(Lab report PhUSE 2009);
007  %let ReportFootnote=%str(Created by OCS Consulting);
008  %let StartPage=1;
009
010  /*****
011  **
012  * Parameter Settings (For Each Parameter in the user interface)
013  *****/
014  */
015
016  %let Creatinine_exclusions=%str(1 15);
017  %let Creatinine_Statistics=%str(N Mean Median Sum STD Min Max Mode);
018  %let Creatinine_Decimal=3;
019  %let Creatinine_Analysis=TTest;
020
021  <Next parameter ....>
022  .....
```

CONCLUSION

Before starting with the design and development of an application it is important to perform a thorough analysis of the user requirements before deciding on a specific application technology.

ACKNOWLEDGMENTS

I would like to thank OCS Consulting for providing me with the means to write and present this paper, and in particular my colleagues Bas van Bakel, Fred Junier, Yves Poriau and Jules van der Zalm for reviewing my paper or providing useful input.

RECOMMENDED READING

Of each of the main application types that OCS Consulting has developed in the past, a paper has been presented by OCS at PhUSE:

- | | |
|----------------------|--|
| Desktop application | - Migration of a flexible reporting system from SAS 6.12 to SAS 9.1.3
A project experience, Presented at PhUSE 2005 |
| Embedded application | - Using SAS Enterprise Guide Add-In to Enable Guided Statistics,
Presented at PhUSE 2008 |
| Web application | - Using the SAS Information Delivery Portal in the Pharmaceutical Sector,
Presented at PhUSE 2006 |
| | - Web-enabling the Pharmaceutical sector, Presented at PhUSE 2007 |

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the authors at:

Author Name	:	Raymond Ebben
Company	:	OCS Consulting
Address	:	PO BOX 490 5240 AL Rosmalen the Netherlands
Work Phone	:	+31 (0)73 523 6000
Fax	:	+31 (0)73 523 6600
Email	:	sasquestions@ocs-consulting.com
Web	:	www.ocs-consulting.nl

Brand and product names are trademarks of their respective companies.