A Hybrid Modeling Platform to meet Basel II Requirements in Banking

Jeffery Morrision, *SunTrust Bank, Inc.*

Introduction: The Basel Capital Accord, ready for implementation in force around 2006, sets out detailed analytic requirements for risk assessment that will be based on data collected by banks throughout the life cycle of the loan. The aim of Basel II is to introduce a more risk-sensitive capital framework with incentives for good risk management practices. The more sophisticated banks will be able to take advantage of significantly reduced capital requirements, and therefore be more competitive. This paper describes the development of a hybrid modeling platform designed especially for that purpose using VB6 and SAS's OLE Automation capabilities. The discussion will revolve around the motivations for such a platform, its features and benefits, some example code and screen captures, and its outlook for the future.

Motivation: In an effort to grow the analytics capabilities to achieve the Best Practices methods and procedures, it was decided a centralized model generation platform was needed. The purpose of this platform was to standardize statistical approaches and provide a host of integrated tools for the proper construction of the quantitative models necessary for determining potential credit losses. In addition, the platform would:

- Provide an easily understood User Interface for model analytics
- Launch programs across software tools - SAS, S-Plus, SHAZAM
- Accumulate Intellectual Knowledge over time
- Ramp up Training for new associates
- Reduce impact of employee turnover
- Significantly reduce programming mistakes
- Minimize redundancy by integrating new ideas into the system
- Provide huge increase in modeling efficiencies - quality and speed to implementation
- Systematic procedures provide excellent auditing trail of model development and validation

Moreover, this system would assist model management in best meeting regulatory scrutiny as well as ensuring the optimal evolution of statistical models within the organization over time. Finally, it was hoped that the system would be seen as being particularly useful in highlighting methodology for the regulatory agencies as SunTrust begins to collect default and payment data and move more and more to a statistically based risk rating approach.

Development Approach: The modeling system was developed in a two-phased approach. The first and most important phase was the creation of a set of SAS programs that would call SAS Macros containing pre-define parameters. These parameters could include names of the independent variables, the appropriate method for handling missing

values, the types of diagnostics and graphics the modeler wishes to see, and the SAS dataset names desired for model development and validation. Once these programs were properly tested and debugged, then they were moved into phase two – the development of the GUI front-end. VB6 was chosen to construct the GUI because application development knowledge at SunTrust was more widespread than say other approaches such as SAS AF / FRAMES. Once completed, the GUI would dynamically construct the SAS or S-plus code needed for a particular task using information provided by the user. This batch program is then submitted to the appropriate statistical software for execution. Copies of the batch routines are saved sequentially when requested for auditing purposes and in case the user wishes to execute them again outside the GUI platform

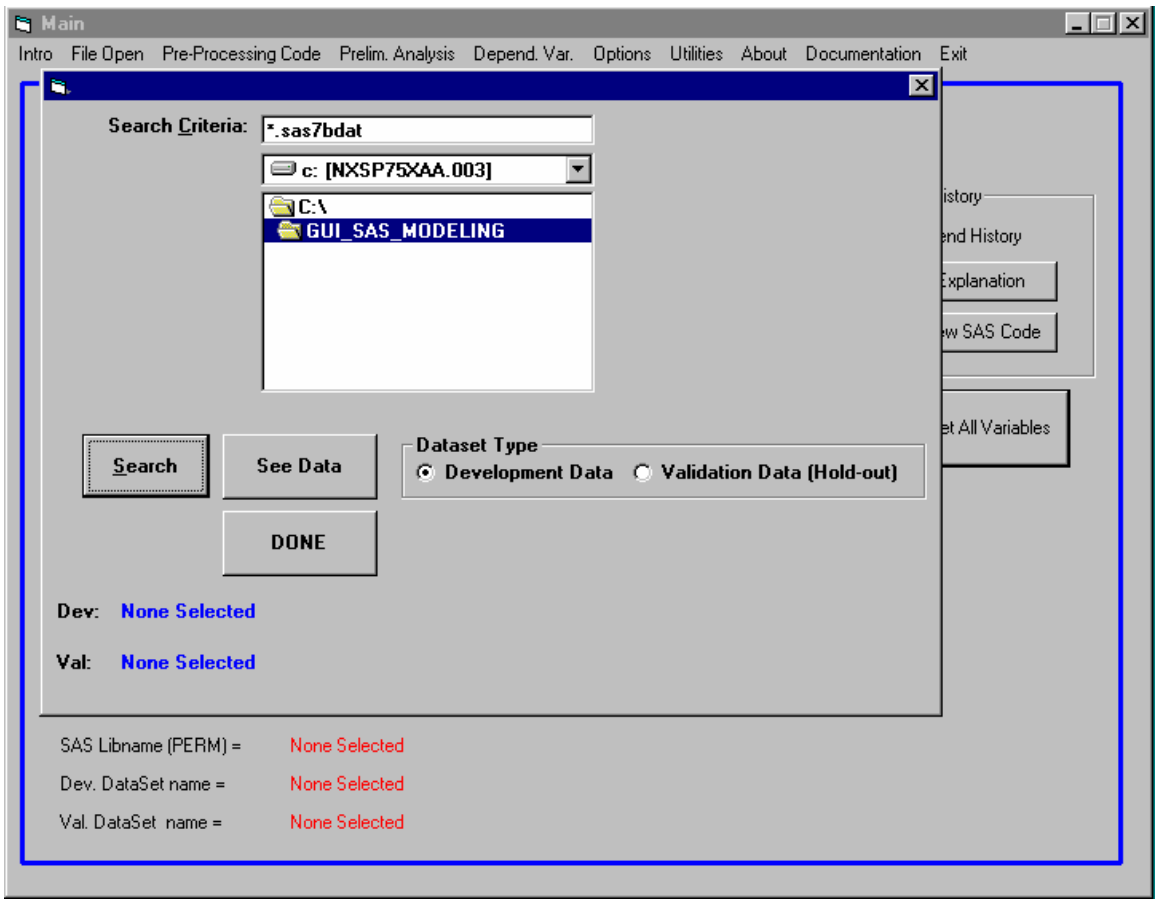
Primary Modeling Procedures: Essential to the design of any modeling system is the awareness of the statistical procedures that will be needed. A minimum of four basic techniques was identified in estimating the models needed under Basel: (1) Logistic Regression (2) Linear Regression and (3) Tobit Regression, and (4) Survival Analysis. Logistic Regression was selected to estimate obligor level default / no default models – an approach specifically suited for data with a binary or dichotomous dependent variable. The output of these models reflects the probability of default, ranging from 0.0 to 1.0. Tobit Regression was chosen as a candidate technique to model loan level Loss Given Default where the data is not binary, but could have numerous values clustered around limits. Time to Event Modeling (Survival Analysis) was also seen as important in profiling the timing of default over and above what could be ascertained from a simple logit specification. Finally, Linear Regression was chosen to estimate the majority of the remaining models that might arise where there was not a clustering of data at upper or lower limits. Neural nets were considered, but because of the complexity of implementation, their use will be restricted downstream in the life of the modeling system.

As modeling is not an exact science and because statisticians come from a wide range of backgrounds and experience, many system designs are possible and could work quite well. However, most would include the following features and therefore were including in SunTrust's platform design:

- Import raw data from in-house databases
- Read data into SAS creating a "SAS dataset"
- Sample & perform data integrity checks
- Create additional variables in SAS if desired
- Create Separate datasets for model development and model validation
- Select approach for handling missing information
- Select modeling technique
- Designate the Dependent Variable
- Specify primary modeling variables
- Specify auto-generating dummies variables
- Specify variables for transformation routines
- Specify variables for interaction effects
- Integrate results of C.A.R.T. node segmentation

- Specify variable selection method
- Perform univariate Analysis
- Generate modeling diagnostics
- Generate implementation code
- Test implementation code on Development & Validation Data
- Accumulate modeling results

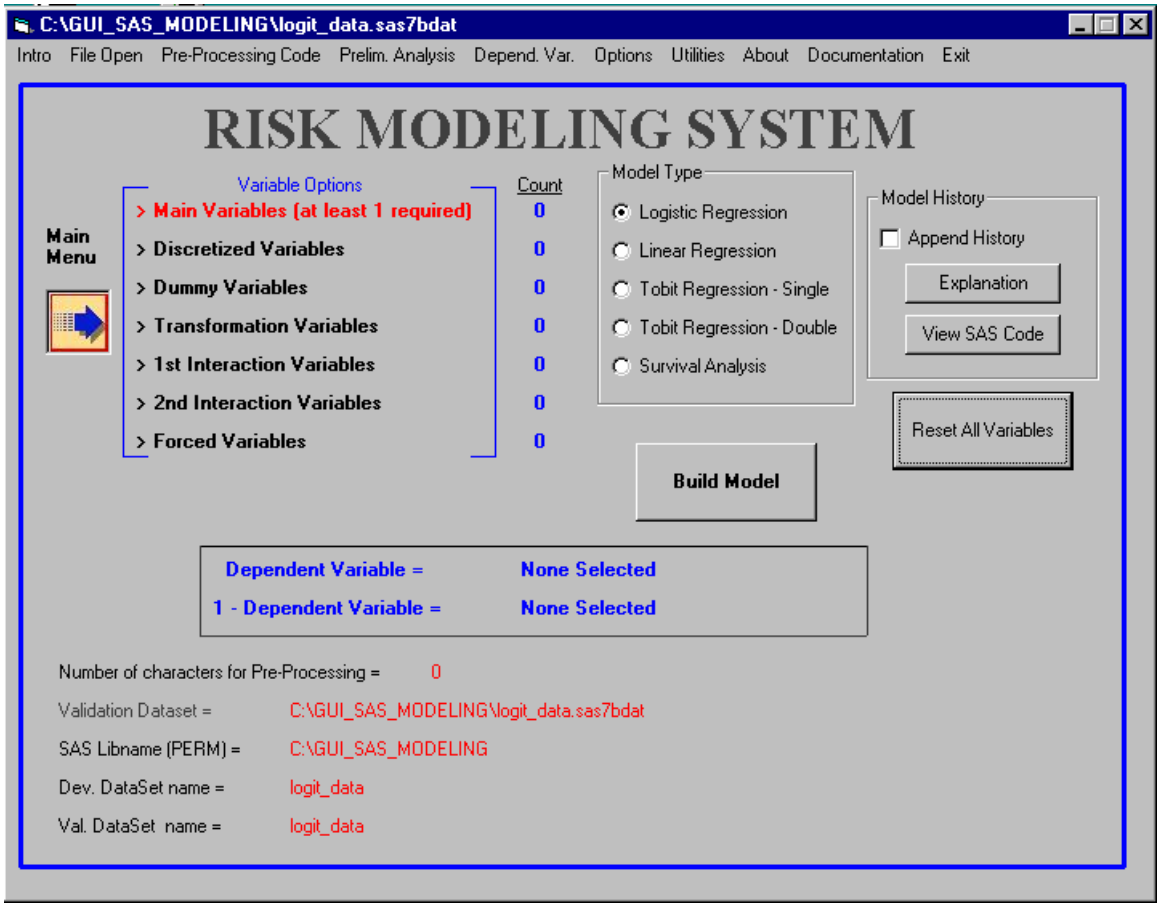
Discussion: The system was designed to look very similar to other programs written in Visual Basic. Below is the main screen that highlights the primary functionality across the top such as FILE OPEN, PRE-PROCESSING, PRELIMINARY ANAYSIS, SPECIFYING DEPENDENT VARIABLE, OPTIONS, UTILITIES, etc. The user begins by specifying the SAS dataset desired for modeling. Through OLE automation, the program



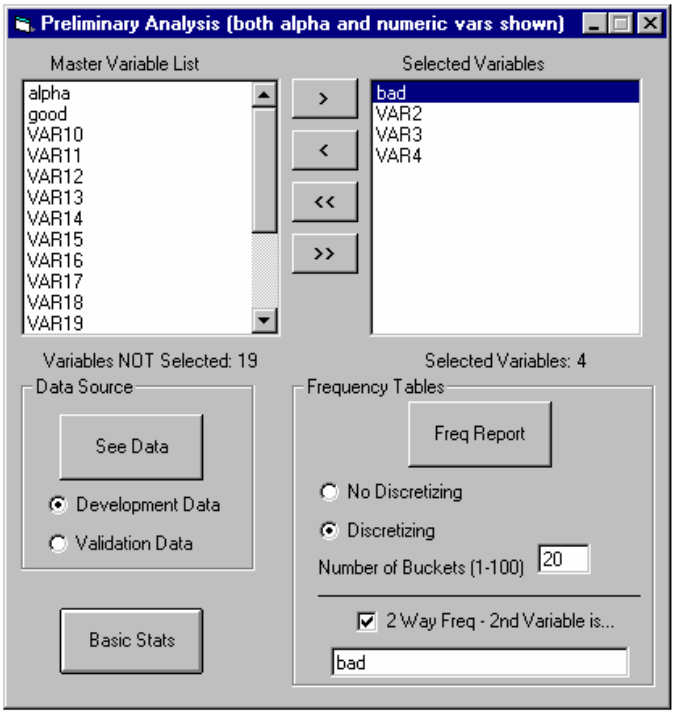
runs a SAS macro in the background and extracts the variable names into a text file which is later imported into Visual Basic list-boxes. Example code for this type of operation looks like this…

```
Set SASObj = CreateObject("SAS.Application")
SASObj.Visible = False
SASObj.Wait = True
mystring1 = "libname perm '" & perm & "';"
SASObj.submit (mystring1)
SASObj.submit ("%include 'c:\GUI_SAS_MODELING\getnamesquick.sas';run;")
mystring2 = "%getnamesquick(perm." & k & ",numeric,'c:\GUI_SAS_MODELING\list.txt');run;"
SASObj.submit (mystring2)
SASObj.submit ("%include 'c:\GUI_SAS_MODELING\getnamesquick.sas';run;")
mystring3 = "%getnamesquick(perm." & k & ",all,'c:\GUI_SAS_MODELING\list_all.txt');run;"
SASObj.submit (mystring3)
SASObj.Quit
Set SASObj = Nothing
```

Note the code that says SASOBJ.VISIBLE=FALSE. This lets its execution be completely invisible to the user. This procedure is in contrast to the majority of the other system tasks where SAS is left open via an interactive window. Next, we see the main menu for the system with certain aspects about the file populated:



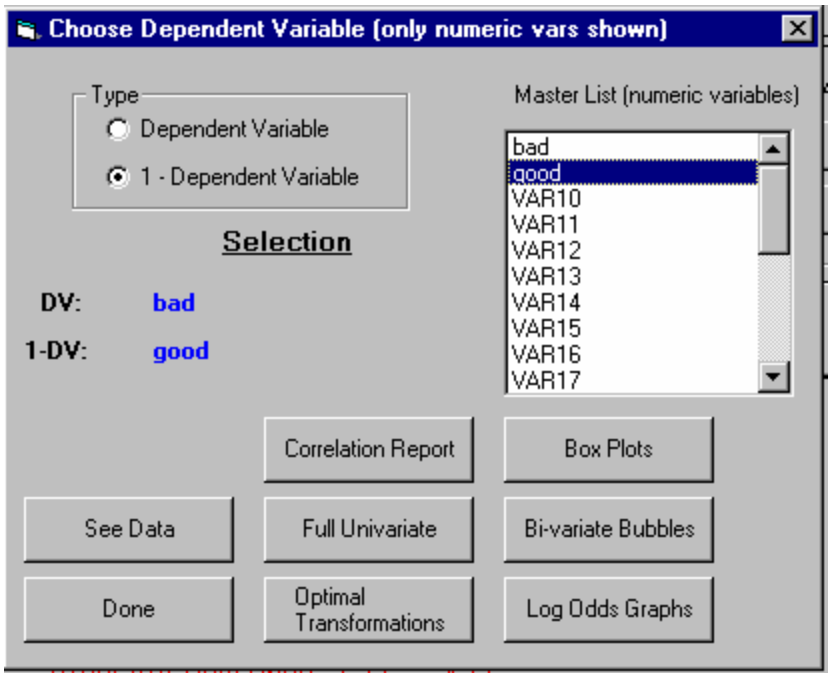
If the user does not wish to create any new variables through the PRE-PROCESSING menu, then a PRELIMINARY ANALYSIS of the data can be developed. Here, basic statistics can be seen and, if desired, the user can see the data via PROC FSVIEW. Finally, frequencies and cross-tabs can be developed where the user may designate a discretizing procedure that buckets the data into groupings.



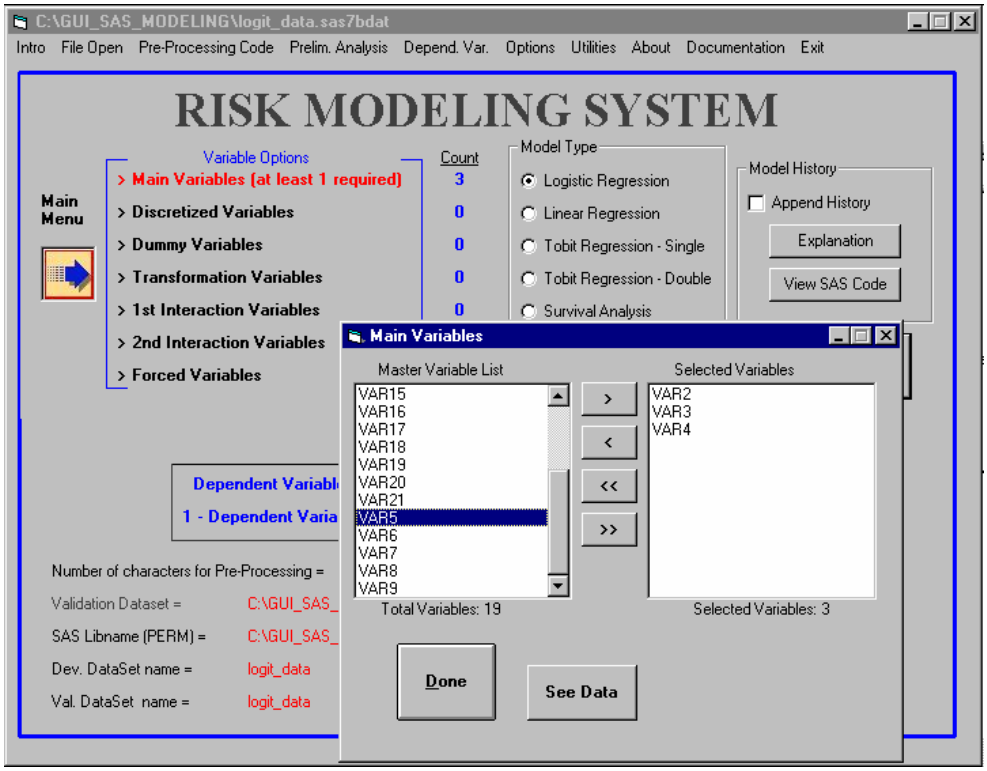
Here, as in many tasks implemented in the system, a SAS batch file is created by Visual Basic and submitted inside the GUI…

```
Open "c:\GUI_SAS_MODELING\gui_corr.sas" For Output As #5
mystring = "libname perm '" & perm & "';"
Print #5, mystring
mystring = "Data work." & k & ";set perm." & k & ";" & preprocess.Text1 & "run;"
Print #5, mystring
mystring = "%include 'c:\GUI_SAS_MODELING\correlation.sas' ;"
Print #5, mystring
mystring = "%let depv= " & depv.Label4.Caption & ";"
Print #5, mystring
mystring = "%include 'c:\GUI_SAS_MODELING\full_varlist.txt' ;"
Print #5, mystring
mystring = "%correlation(work." & k & ",0);run;"
Print #5, mystring
Close #5
Set SASObj = CreateObject("SAS.Application")
SASObj.Visible = True
SASObj.Wait = True
SASObj.submit ("%include 'c:\GUI_SAS_MODELING\gui_corr.sas';run;")
```

Next, more analysis can be done after the user designates which variable is to be considered the DEPENDENT VARIABLE. A number of analysis options are available such as Bivariate and Pairwise Correlations as well as a custom procedure to test for the optional univariate transformations. A variety of graphical plots are also available to the user that are useful in describing the distribution of the data.



With the preliminary analysis behind us, we are ready to begin specifying the explanatory variables in the model. In our menu, we can have a variety of procedures applied that perform discretization of our data, automatic dummy variables, optimal transformations,



and the automatic creation of interaction terms. For example, if we wish to include a set of explanatory variables in the model without anything fancy, then the screen would look like the above. With these explanatory variables specified, then the associated SAS batch

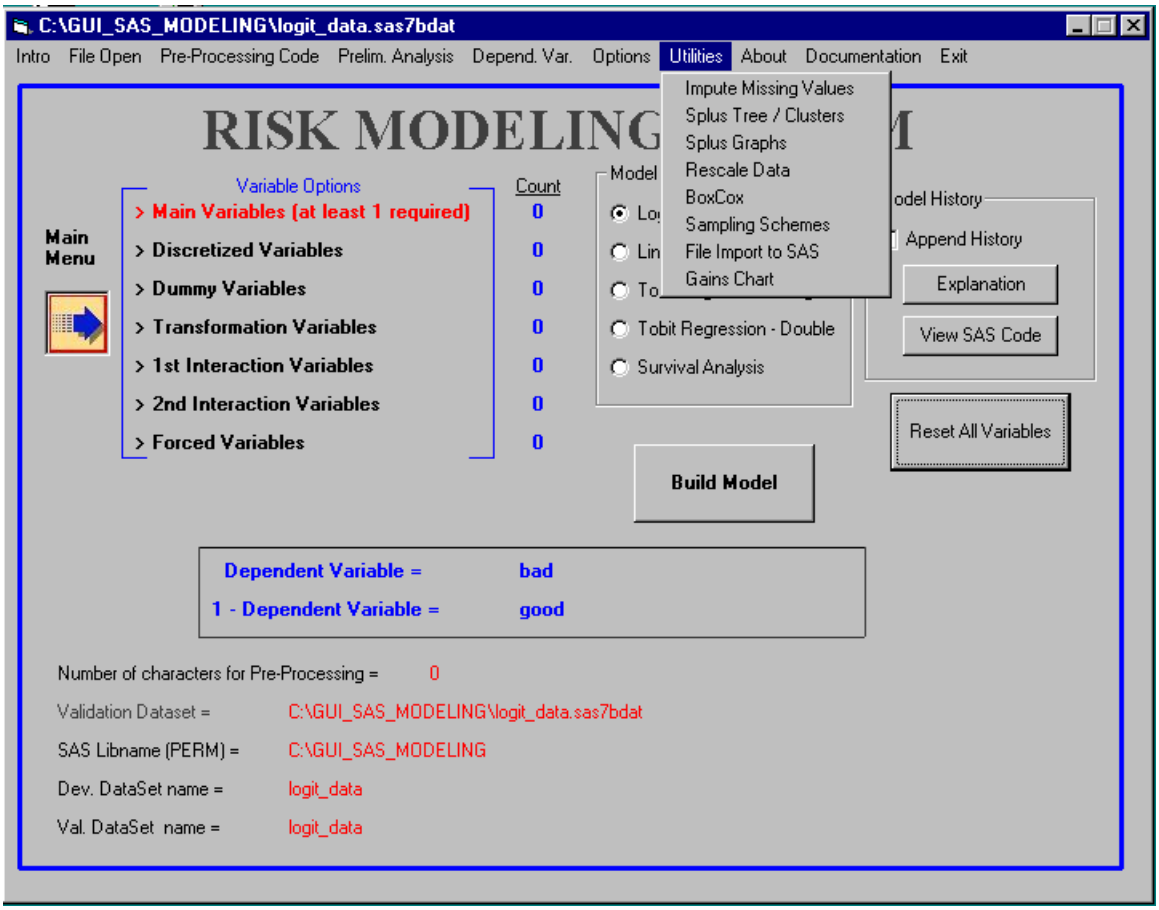
file is executed with the interactive window left open so the user can examine the output, logs, and any graphs generated by the procedure. And if the user had checked the APPEND HISTORY option, the program would automatically accumulate a table of modeling results and rank their accuracy according to some criteria.

VARIABLE	COEFFICIENT	Model	LNLIKE	KS_V	RMSE_V	AIC	SW
Intercept	3.854665916	4	-252.51143	72.3	0.324561	523.0229	564.60351
VAR3	-0.127194964	4	-252.51143	72.3	0.324561	523.0229	564.60351
VAR7	1.017487224	4	-252.51143	72.3	0.324561	523.0229	564.60351
VAR8	-0.336265904	4	-252.51143	72.3	0.324561	523.0229	564.60351
VAR4_XSQ	-0.586245912	4	-252.51143	72.3	0.324561	523.0229	564.60351
VAR5_XSQ	0.000572729	4	-252.51143	72.3	0.324561	523.0229	564.60351
VAR6	1.90334341	4	-252.51143	72.3	0.324561	523.0229	564.60351
var2_2	-1.864024702	4	-252.51143	72.3	0.324561	523.0229	564.60351
I_11	0.064133553	4	-252.51143	72.3	0.324561	523.0229	564.60351
Intercept	0.286047655	3	-266.80762	69.6	0.3348986	549.6152	586.57582
VAR3	0.052146245	3	-266.80762	69.6	0.3348986	549.6152	586.57582
VAR7	1.120932766	3	-266.80762	69.6	0.3348986	549.6152	586.57582
VAR8	-0.331673728	3	-266.80762	69.6	0.3348986	549.6152	586.57582
VAR4_XSQ	-0.149817579	3	-266.80762	69.6	0.3348986	549.6152	586.57582
VAR5_XSQ	0.000475961	3	-266.80762	69.6	0.3348986	549.6152	586.57582
VAR6	1.846343463	3	-266.80762	69.6	0.3348986	549.6152	586.57582
var2_2	-1.589646097	3	-266.80762	69.6	0.3348986	549.6152	586.57582
Intercept	0.017600651	2	-276.135	67.6	0.3423365	566.2700	598.61051
VAR3	0.05958415	2	-276.135	67.6	0.3423365	566.2700	598.61051
VAR7	1.259154057	2	-276.135	67.6	0.3423365	566.2700	598.61051
VAR8	-0.346301703	2	-276.135	67.6	0.3423365	566.2700	598.61051
VAR4_XSQ	-0.168404986	2	-276.135	67.6	0.3423365	566.2700	598.61051
VAR5_XSQ	0.000507674	2	-276.135	67.6	0.3423365	566.2700	598.61051
VAR6	1.899270228	2	-276.135	67.6	0.3423365	566.2700	598.61051
Intercept	0.175296307	1	-384.00198	51.5	0.414296	776.0040	794.48426
VAR3	0.063253639	1	-384.00198	51.5	0.414296	776.0040	794.48426
VAR7	1.240840742	1	-384.00198	51.5	0.414296	776.0040	794.48426
VAR8	-0.370521666	1	-384.00198	51.5	0.414296	776.0040	794.48426

Perhaps the most important feature of the modeling system is its ability to automatically generate the SAS code necessary to reproduce the predicted values. With only minor syntax changes, this code can be translated into a coding algorithm for other languages (VB, etc.) for implementation. Moreover, the modeling system calculates predicted values for the Validation dataset using both the internally generated PROCS as well as the dynamically generated SAS implementation code.

```
* Missing Proxy Code:;
        if var4 = . then var4 = 2.9583333333 ;
        if var5 = . then var5 = 35.976 ;
        if VAR3 = . then VAR3 = 32.396 ;
        if VAR7 = . then VAR7 = 0.308 ;
        if VAR8 = . then VAR8 = 7.572 ;
        if var2_2 = . then var2_2 = 0.1813333333 ;
        if I_11 = . then I_11 = 94.280614667 ;
* Transformation Code:;
        VAR4_XSQ=(VAR4*VAR4);
        VAR5_XSQ=(VAR5*VAR5);
* Logistic Implementation Code:;
        score =
        3.8546659162  +
        VAR3  * -0.127194964  +
        VAR7  * 1.017487224  +
        VAR8  * -0.336265904  +
        VAR4_XSQ  * -0.586245912  +
        VAR5_XSQ  * 0.0005727289  +
        VAR6  * 1.9033434096  +
        var2_2  * -1.864024702  +
        I_11  * 0.0641335527 ;
        score = 1  /(1+exp(-(score)));
```

Finally, modeling system offers a variety of custom written utilities that can be executed without having the specific objective of a model being constructed. These utilities could range from a number of sampling routines, BoxCox transformations, rescaling, and even the development of graphs and C.A.R.T. Trees in S-Plus.



Summary: SunTrust now has an analytics platform that is portable to other PCs under a variety of operating systems including NT and Windows which can be distributed to users through the network or on CDs. It is the official repository of Best Practices reflecting specific statistical techniques for risk rating and validations of loans throughout the company. As its use grows over time, new techniques will be integrated into the user interface to better meet any new Basel standards which are required to support the "advanced" IRB approach to capital requirements.