

## Worcester Polytechnic Institute Digital WPI

---

Major Qualifying Projects (All Years)

Major Qualifying Projects

---

December 2009

# BNP Paribas: Systems Analysis and Concept Development

Aubrey Marie Scarborough  
*Worcester Polytechnic Institute*

Brendan Padraic Dean  
*Worcester Polytechnic Institute*

Timothy Scott Sherrerd  
*Worcester Polytechnic Institute*

Follow this and additional works at: <https://digitalcommons.wpi.edu/mqp-all>

---

### Repository Citation

Scarborough, A. M., Dean, B. P., & Sherrerd, T. S. (2009). *BNP Paribas: Systems Analysis and Concept Development*. Retrieved from <https://digitalcommons.wpi.edu/mqp-all/3497>

This Unrestricted is brought to you for free and open access by the Major Qualifying Projects at Digital WPI. It has been accepted for inclusion in Major Qualifying Projects (All Years) by an authorized administrator of Digital WPI. For more information, please contact [digitalwpi@wpi.edu](mailto:digitalwpi@wpi.edu).



# **BNP Paribas: Business Analysis and Concept Development**

A Major Qualifying Project Report  
For BNP Paribas New York, New York

Submitted to the Faculty of the

WORCESTER POLYTECHNIC INSTITUTE

In partial fulfillment of the requirements for the

Degree of Bachelor of Science

**Submitted By:**

---

Brendan Dean

---

Aubrey Scarborough

---

Tim Sherrerd

Date: 18 December 2009

**Approved By:**

Mr. Philip Coleman, BNP Paribas Liaison  
Professor Arthur Gerstenfeld, Primary Advisor  
Professor Jon Abraham, Co-Advisor  
Professor Dan Dougherty, Co-Advisor

## **Abstract**

This project focuses on the re-engineering of the operations within BNP Paribas' internal investment deals, workflow and risk analysis programs. This includes evaluating their current strategies for data storage and delivery and working to recognize obstacles in the current infrastructure. The goal is to redesign the valuation process for spreadsheet deals with a focus on security and documentation and to develop a proof-of-concept prototype for a new information system using components from the previous analysis.

## **Acknowledgements**

Our team received substantial guidance and support from numerous people during the course of our project who were knowledgeable sources of information and advice. We would like to give special thanks to our advisors; Professors Art Gerstenfeld, Jon Abraham, and Dan Dougherty, for helping our team formulate our ideas and pushing our project to success throughout its duration. From BNP Paribas, our team would especially like to thank Phil Coleman, Andy Clark, and Mike Tanousis for their help and support during our stay at the office.

Our team would also like to acknowledge all those who contributed to the research and development of our project. Special thanks to:

Bruno Braizinha

Dan Ward

Jeremy Ascher

Jordan Prévé

Laurent Frisch

Steve Gillen

## Table of Authorship

<b>Section</b>	<b>Author(s)</b>
Abstract	Aubrey
Acknowledgements	Tim
Executive Summary	Tim
1. Introduction	Aubrey
2. Background	All
3. Methodology	Aubrey
4. Results & Analysis	All
5. Conclusions	Brendan
6. Recommendations	Aubrey
Appendix A	Tim
Appendix B	Aubrey
Appendix C	Tim
Appendix D	All
Appendix E	Tim
Appendix F	Tim
Appendix G	Tim
Appendix H	Brendan

# Table of Contents

Title Page .....	i
Abstract .....	ii
Acknowledgements .....	iii
Table of Authorship .....	iv
Table of Contents .....	2
List of Figures .....	6
List of Tables .....	6
Executive Summary .....	7
1. Introduction .....	10
2. Background .....	12
2.1. Trade Capture Systems .....	12
2.1.1. BNP Paribas' Trade Capture Systems .....	12
2.2. Fixed Income Securities .....	13
2.3. Financial Derivatives .....	14
2.4. Treasury Locks .....	15
2.4.1. Trade Execution .....	16
2.4.2. Hedging Treasury Rate Risk .....	16
2.4.3. Speculation .....	17
3. Methodology .....	18
3.1. Interviews .....	18
3.2. Evaluation of the current process .....	18
3.3. Development of the Prototype .....	19
3.4. Design Choices .....	19
3.4.1. Programming Language .....	19
3.4.2. System Structure .....	20
3.4.2.1. Single Executable .....	20
3.4.2.2. Web Service .....	20
3.4.3. Valuation Calculations .....	20
3.4.3.1. Coded Calculations .....	20
3.4.3.2. Excel Calculations .....	21

3.4.4. Deal Storage .....	21
3.4.4.1. XML .....	22
3.4.4.2. SQL Database .....	22
3.5. System Components .....	22
3.5.1. Bean Stalk .....	22
3.5.2. Excel 2007 .....	23
3.5.3. Microsoft SQL Server .....	23
3.5.4. MadMolly .....	23
3.6. User feedback .....	24
4. Results and Analysis .....	25
4.1. Analysis of Current Program .....	25
4.1.1. Fahrenheit Process .....	25
4.1.2. Concerns with Fahrenheit .....	29
4.1.3. Microsoft Excel .....	30
4.2. Potential Solutions .....	33
4.2.1. Commercial Systems .....	33
4.2.2. Creation of a Custom Application .....	35
4.2.3. Industrialization of Fahrenheit .....	36
4.3. Selection Criteria .....	37
4.3.1. Technical Capabilities .....	38
4.3.1.1. Flexibility .....	39
4.3.1.2. System Security and Deal Integrity .....	39
4.3.1.3. Scalability .....	40
4.3.1.4. Ease of Use .....	41
4.3.2. Logistical Feasibility .....	41
4.3.3. Mathematical Selection Model .....	43
4.4. Model Analysis .....	44
4.4.1. Technical Capability Characteristic Classifications .....	45
4.4.2. Logistical Feasibility Characteristic Classifications .....	48
4.4.3. Model Validation .....	49
4.4.3.1. Adverse Scenario .....	55

4.4.3.2. Worst-Case Scenario.....	56
4.5. The Industrialization of Fahrenheit .....	57
4.5.1. Implementation of Microsoft Excel 2007 .....	57
4.5.2. Dynamic Input Process .....	57
4.5.3. Documentation of Inputs.....	58
4.5.4. Computational Distribution.....	58
4.5.5. New Template Creation and Review .....	58
4.6. Creation of a Prototype.....	59
4.6.1. Program Overview .....	59
4.6.2. Creation of MadMolly .....	60
4.6.3. Bean Stalk Spreadsheet.....	61
4.6.4. Uploading PVs with MadMax .....	61
4.6.5. Generation of Risk Reports.....	61
5. Conclusions.....	62
5.1. Improvements .....	62
5.2. Impact .....	63
6. Recommendations.....	64
6.1. Increased Number of Calculations .....	64
6.2. Scalability of Bean Stalk .....	64
6.3. Discussion.....	65
6.3.1. Lack of Commercial Systems .....	65
6.3.2. Re-evaluation of Tier Requirements .....	65
References.....	66
Appendix A: Sponsor Description.....	70
Appendix B: Acronym Index.....	71
Appendix C: Tier Categorization.....	74
Appendix D: Interviews.....	76
Jeremy Ascher .....	76
Dan Ward.....	78
Nikholas Lopez.....	80
Appendix E: Treasury Lock Valuation.....	81



Appendix F: Financial Risk Measurements .....	84
Appendix G: The Black-Scholes Model .....	86
Appendix H: Bean Stalk White Paper .....	90

## List of Figures

Figure 1: Trade Capture and Valuation Process .....	29
Figure 2: Fahrenheit Process Diagram.....	40
Figure 3: Total Value Comparison with Original Parameters .....	49
Figure 4: Characteristic Weight Sensitivities .....	52
Figure 5: System Tradeoff Opportunities at p=70%.....	54
Figure 6: "Worst Case" $\Omega$ -Value Comparison.....	56

## List of Tables

Table 1: Trade Capture Responsibility Breakdown.....	27
Table 2: Fahrenheit Job Description.....	28
Table 3: Current Trade Limits by Tier.....	30
Table 4: Trade Capture System Parameters.....	34
Table 5: System Selection Criteria .....	37
Table 6: System Characteristics Analysis.....	44
Table 7: System Selection Scorecard.....	44
Table 8: "Best-Fit" $\Omega$ -Value Comparison .....	45
Table 9: Characteristic Sensitivity Formulas.....	51
Table 10: "Adverse Scenario" $\Omega$ -Value Comparison .....	55
Table 11: Risk Sensitivity Formulas.....	85
Table 12: Black-Scholes European Option Formulas.....	88
Table 13: Black-76 Bond Option Formulas.....	89
Table 14: Bean Stalk Parameters .....	93

## **Executive Summary**

BNP Paribas' Fixed Income department uses a variety of in-house systems designed to capture deal information and conduct the valuation, pricing, and risk generation of these trades. One of these systems, known as Multi Asset Derivatives (MAD), is specifically designed for the capture and valuation of complicated, or exotic, derivative trades. While MAD is considered a highly secure system, its complex programming structure prevents the efficient addition of valuation formulas for new trades. To handle these new deal types, BNP adopted a spreadsheet-based solution that employs Excel 2003 as an intermediary for calculations. This program, known as Fahrenheit, stores deal information from MAD in spreadsheets and uses deal-specific templates alongside a library of functions called Westminster to generate PVs and risk sensitivities before storing and repopulating MAD with the output values.

Fahrenheit is not a true trade capture system, but rather a valuation supplement. Designed solely as a calculation medium for certain MAD trades, Fahrenheit stores data from MAD in Excel spreadsheets to run calculation templates provided by the Flow Rates Group (FRG) department of BNP. Upon completion, PVs are uploaded into MAD and the risk figures are saved to a network drive, at which point any Excel files created during the process are deleted. Fahrenheit is run overnight in conjunction with MAD to provide the desired risk and PV values to traders in the morning.

The major concerns regarding Fahrenheit are in its lack of documentation and security mechanisms. Deals contained entirely within the MAD framework are considered secure and well controlled, thus granted "Tier 1" status, which allows for unlimited trading. Since Fahrenheit deals are temporarily stored in Excel spreadsheets which are poorly documented and easily manipulated, leaving room for unintentional or malicious human error, BNP is forced to classify trades valued through this system as Tier 2 or 3 trades. The quantity of these outstanding deals is limited at the discretion of BNP's risk management group, R-CM. To address this issue our group investigated and analyzed three potential solutions including the use of a third-party trade capture system, the designing of a custom application, and the industrialization of Fahrenheit.

Using information collected during interviews with employees and our own analysis of Fahrenheit and the three potential solutions, our group determined a modification of Fahrenheit would be the optimal choice. While there are several flaws associated with the existing Fahrenheit process, small modifications can provide the necessary upgrades while maintaining the desired flexibility of the Excel-based calculations.

We have designed a prototype system, called Bean Stalk, which uses a new C# program to run a stripped version of Fahrenheit in Excel 2007. Bean Stalk was tasked with the handling of a trade type known as a Treasury Lock (TLock), which was previously calculated by Fahrenheit. According to R-CM, moving a trade from Tier 2 to Tier 1 status is unlikely with the continued use of Excel. However, our project provides the security mechanisms and documentation necessary for a potentially greater allowance of Tier 2 and 3 trades.

The calculation spreadsheet in Bean Stalk is essentially the same as Fahrenheit's. Bean Stalk's new features and subsequent advantages center on the flow of information throughout the process. A new program called MadMolly queries MAD and converts trade details into XML format. Bean Stalk then utilizes the XML mapping capabilities of Excel 2007 to dynamically import this XML data into pre-defined templates created for specific trade types. Bean Stalk then stores the output data in the same fashion as the original Fahrenheit system and uploads the data into MAD through an in-house program called MadMax.

The Excel 2007 spreadsheet can never be altered by the user and only exists in a production environment at the time of calculation. By importing data directly from MAD, Bean Stalk circumvents the spreadsheet-dependent data storage process of Fahrenheit and eliminates a large portion of the system's operational risk. Additionally, the XML input data is saved to a SQL database for auditing purposes. Bean Stalk's programming structure also lays the groundwork for distributed computations and increased scalability, although the logic mechanism for this distribution was not developed during our project due to time constraints.

Bean Stalk provides only the foundation for a new valuation system and our group has compiled several recommendations to assist future development. Included in these

recommendations are the expansion of Bean Stalk to new trade types and the potential for increased scalability through the design of a parallel calculation mechanism.

Currently, Bean Stalk only demonstrates the calculation of treasury locks, but should be developed further to include all Fahrenheit trade types effectively phasing out Fahrenheit as a valuation system. In addition to increasing the number of trades handled by Bean Stalk there are ways to increase the volume of trades it can effectively handle. Our final prototype addresses this issue by laying the framework for computational distribution. The development and implementation of the mechanism to run this distribution will allow for the increased scalability of our system along with its potential as a permanent valuation system.

Our group believes a re-evaluation of Tier requirements would prove beneficial. Through our interviews, we noted discrepancies between various BNP departments that utilize Fahrenheit and other trade capture and valuation systems. Opinions about the Tier requirements were varied and led to uncertainty about the definitions for each type of trade. Discussion between R-CM, FRG, and IT could result in a clearer understanding of the goals of BNP as well as dissolve confusion between these departments.

# 1. Introduction

While ranked by Forbes as one of the largest companies in the world, BNP must continually emphasize the improvement of trading systems as well as the development of growth opportunities. The technological infrastructure of BNP must grow to keep up with new industry requirements and safety standards. This report addresses both issues of security and the creation of a system that can allow for more trades. To complete these tasks our focus is kept on an in-house system at BNP known as Fahrenheit. Fahrenheit is a key component in the processing of trades within BNP's larger trade capture system. New trades must be monitored at each stage of their capture, calculation and storage. This project deals with the research into various solutions to security issues as well as fulfillment of BNP requirements. Analysis of the various components of the in-house trade capture system, its functions and shortcomings are explored while the current process for a new trade is explained in detail. Due to the time constraints of our allotted seven-week project period, as much of the solution was created as possible. Further discussion is held later in the report about the future of the project and possible routes for continuation.

Our project was completed in two stages, one focusing on research and analysis and the other part working on the design and implementation of our solution. Throughout the project we concentrated on completing the following three objectives:

1. Analyze the current system by determining areas of weakness to be addressed.
2. Design a solution that best fits the problem after identifying potential courses of action and requirements of the system.
3. Design a proof-of-concept system for testing and implementation.

We were able to accomplish these objectives through a variety of methods. We were glad to have the support of our sponsors, who gave us demonstrations of BNP software and introduced us to members of different departments to aid us in our research. We were able to gain valuable information through a series of interviews which supported our research for solution requirements. We also gained feedback through the development of

the proof-of-concept that enabled us to provide BNP with a working prototype at the completion of the project.

## **2. Background**

To provide a background for the overall scope of this project, this section contains an overview of several important ideas and concepts used throughout the remainder. We first review the concept of a trade capture system in general and the specific systems currently in place at BNP Paribas. Additionally, we provide a financial foundation for the terms and concepts used by these systems and more importantly a detailed explanation of the trade type our eventual prototype uses for testing purposes.

### **2.1. Trade Capture Systems**

As new deals are executed they are captured and documented in trade capture systems categorized using a set of predetermined matching criteria. These systems contain vital information about the life cycle of each trade and are used to calculate the value of trades for booking purposes on a daily basis. Trade capture systems are also often used to generate risk sensitivities (see Appendix F) daily for management. Trade capture systems are tightly-controlled operational systems which often have a flexible design that enables the capture of trades for a number of different financial instruments (Peresys, 2009).

The categorization of trades is also guided by the Committee on Uniform Security Identification Procedures (CUSIP). A CUSIP number is a unique identifier for a financial instrument that identifies the company or issuer it pertains to and the type of security traded. Each CUSIP is generated using a formula that intakes the properties of an instrument and assigns it a series of nine alphanumeric characters (CUSIP, 2009).

#### **2.1.1. BNP Paribas' Trade Capture Systems**

BNP Paribas' Fixed Income division employs two primary trade capture systems. The first of these systems used for “vanilla” trades is known as Star. Star captures trades that are straightforward in both detail information and valuation complexity and uses BNP's library of functions known as Westminster to perform the necessary valuation and risk sensitivity calculations. The second system, used for more complicated “exotic” trades, is known as Multi-Asset Derivatives, or MAD. MAD handles trades that are too



complex for Star and employs another BNP system, known as PhaseTwo, to handle valuation and risk sensitivity calculations.

As a tightly-controlled trade capture system, MAD's coding is very rigid and inflexible. Because of this, some new trades are too complex for MAD to handle. These deals are collectively referred to as "spreadsheet deals" because of the Excel-based approach to their calculations. These deals, while booked in MAD at the time of trade execution, have valuations that fall outside of MAD's computational framework and need to be valued by some other method. One of these methods, known as Fahrenheit, is the centerpiece of this project. Fahrenheit uses deal information stored in MAD to calculate the value and risk sensitivities of certain trade types using pre-created Microsoft Excel templates before repopulating MAD with the new valuation information. This process will be reviewed in depth later in the project.

## **2.2. Fixed Income Securities**

Fixed income refers to any type of investment that yields a regular, known rate of return (Investopedia ULC, 2009a). For example, when a lender loans money to a borrower the borrower must make interest payments at regular intervals. This loan is a form of a fixed income security since these interest payments are predefined at a fixed interest rate. The most common fixed income security is known as a bond. A bond typically makes periodic "coupon payments" calculated as a percent, the coupon rate, of the bond's face value. Although most bonds have coupon payments, some bonds referred to as "zeroes", do not have any coupon payments. All bonds also have a final payment made at maturity known as its redemption value. While a bond's redemption value is usually equal to its face value, this is not necessarily always the case.

The valuation of a bond uses its face value  $F$ , redemption value at maturity  $C$ , periodic coupon rate  $r$ , coupon periods until maturity  $n$ , and effective yield per coupon period  $i$  to calculate its price  $P$ . The formula, equal to the sum of the present value of its coupon payments and the present value of its redemption value, is given below.

$$P = (Fr)a_{n|i} + Cv^n$$

Here we use a symbol from actuarial notation known as an annuity symbol. Spoken “a angle n i”, this symbol refers to the present value of a series of payments of \$1 made at the end of each period for n periods at an effective interest rate per period  $i$ .

Formulaically, we can represent this annuity stream as follows:

$$a_{n|i} = v + v^1 + \dots + v^n = (1 - v^n)/i$$

We use the symbol  $v^n$  to represent the “discount factor” of a \$1 payment at time n. Using any dollar amount  $x$ , we can see that its present value is equal to the cash flow times its discount factor. The formula is given here:

$$xv^n = x[1/(1 + i)^n] = x/(1 + i)^n$$

Having defined all the variables, we can now calculate the price of a bond making periodic coupon payments using a given effective interest rate, or the yield we wish to earn on the investment. It is easy to show that if this rate is equal to the coupon rate of the bond (if  $i = r$ ) then the formula collapses to:

$$P = F - (F - C)v^n$$

Additionally, if the face value is also equal to the bond’s redemption value ( $F = C$ ) then we see that the price is simply equal to its redemption value.

### **2.3. Financial Derivatives**

Derivatives are financial contracts, or financial instruments, whose values are derived from that of an underlying asset. These instruments can include the values from other assets, indexes, events, values or conditions (Hull, 2009). These derivatives can be leveraged such that small adjustments in the underlying value cause large differences in the value of the derivative. This leverage may also be used by investors to speculate on

the value of an underlying asset without needing the capital to physically purchase and sell the asset itself (Lee & Lee, 2006).

Derivatives are broadly categorized by three general characteristics of the instrument: the relationship between the derivative and the underlying asset, the type of underlying asset, and the market the derivative is traded in. Derivatives are split into two broad categories based on the market they are traded in including over-the-counter derivatives, or derivatives that trade between private parties, and exchange-traded derivatives that trade publicly on a recognized exchange like the Chicago Mercantile Exchange. Similarly, the type of underlying asset breaks derivatives down into subsections such as equity derivatives, foreign exchange derivatives, bond derivatives, etc (Saliba, 2005).

Finally, the relationship between a derivative and its underlying asset defines the type of a derivative. A derivative that provides the buyer with the right to buy or sell an asset for a given price at a stated expiration date is referred to as an option (Lee & Lee, 2006). The most common forms of options are stock options, which use stock prices as an underlying asset, and bond options, which use bond prices. There are many means of valuating these types of options including the binomial options model, the Black-Scholes and Black-76 options models (see Appendix G), and the Monte Carlo option model. A swap is a form of derivative in which two counterparties exchange the benefits of financial instruments. Swaps are commonly used to hedge interest rate risk by exchanging a stream of floating interest rate payments for a stream of fixed interest rate ones (Hull, 2009).

Derivatives on fixed income securities have many uses. They are commonly used to hedge bond positions, speculate on fixed income securities, leverage fixed income trades, and eliminate or mitigate a variety of risks including default risk, Treasury rate risk, and credit risk (Hull, 2009). In the next section we review one of these types of securities known as a Treasury Lock.

## **2.4. Treasury Locks**

A Treasury Lock, also known as a TLock, is a simple derivative security and type of interest rate swap which uses the market yield of a Treasury note or bond, frequently

referred to as the “risk-free” rate for a particular maturity, as its underlying “asset” (Investopedia ULC, 2009c). A Treasury Lock is essentially the synthetic forward sale of a Treasury (BMO, 1997) which represents an agreement between a buyer and a seller that fixes the yield, and consequently the price, of a specific Treasury note or bond over the lock period. A TLock is almost always a short-term derivative with a lock period of less than one year, and usually less than three months. It should be noted that a TLock is *not* an option, as there is no exchange of securities, and is typically cash settled (BNY, 2009).

The purchase of a TLock is a means of opening a long position on Treasury yields as the investor is betting on a rise in rates. Similarly, the sale of a TLock opens a short position on Treasury rates. Since Treasury yields are inversely proportional to the price of the underlying bond, a long position on a TLock is analogous to a short position on the Treasury itself while a short position on a TLock is comparable to a long position on the underlying Treasury. The differences between a position on Treasury rates and the underlying Treasury will be discussed later in this section.

#### **2.4.1. Trade Execution**

To enter into a Treasury Lock the buyer first selects an appropriate Treasury bill or note and the total notional amount, sets the transaction maturity date, and agrees upon a lock rate with the seller. The seller then usually hedges the TLock and opens a short position by selling the Treasuries in the bond market. Because the seller must pay the coupon rate on the Treasury while the short position is open, the agreed upon lock rate is usually calculated as the initial market Treasury yield rate plus the cost of carrying the position throughout the duration of the TLock (BMO, 2009). Because the calculation of this cost is not pertinent to the scope of this project, no further descriptions or formulas will be given. Upon the expiration of the TLock the seller closes the short position and settles with the buyer using the formulas in the following section.

#### **2.4.2. Hedging Treasury Rate Risk**

Interest rate risk, or the risk assumed by the ownership of an interest-bearing asset due to the volatility of interest rates, is primarily composed of Treasury rate risk and credit risk (Lee & Lee, 2006). A Treasury Lock can be used to protect the buyer from an increase in Treasury rates, or the Treasury rate risk. In effect, the seller of a TLock is

insuring the buyer against rising financing costs associated with similar interest rate increases by assuming the Treasury rate risk over the lock period. In exchange, the seller takes ownership of the potential savings on the buyer's financing costs resulting from falling interest rates.

TLocks are most commonly used by the issuers and borrowers of fixed-rate debt who depend on Treasury rates to calculate their offering interest rate. Buyers typically use the length of the underlying debt to select an appropriate Treasury maturity for the TLock. Institutional interest rates are typically priced as a spread over Treasuries, where the spread is a function of company specific factors and the widening or narrowing of this spread is known as credit risk. These factors, usually summarized by a company's credit rating, are relatively stable over short periods of time (less than one year). Treasury rates however can be substantially more volatile over a similar period and thus become one of the largest components of interest rate risk. TLocks are designed to transfer this Treasury rate risk from the debt issuer or buyer to the seller of the TLock while the buyer retains possession of the associated credit risk.

### **2.4.3. Speculation**

Treasury Locks may also be used as a form of speculation by creating a bet on the direction of treasury rates. By selling a TLock, consequently opening a short position, an investor hopes for a fall in Treasury rates during the lock period which makes the seller the receiver of payment at the derivative's expiration date. Similarly, an investor can buy a TLock and open a long position on treasury rates. If rates rise the buyer will close the TLock and receive the payoff from the seller. Both short and long positions on TLocks can be hedged by opening an opposite position on the Treasuries themselves.

The advantage TLocks can provide over simply betting on the direction of Treasuries is in leverage. The buyer of a TLock is essentially opening a short position on the underlying Treasury since the price of a Treasury is inversely proportional to its yield. However, an investor does not need to tie up capital in the securities themselves. Similarly, the seller of a TLock is opening a long position on the underlying Treasury without needing to ever own the security (Lee & Lee, 2006).

### **3. Methodology**

Our project resulted in a program that aided in the documentation of an exotic trade. To reach this point, we analyzed the current life cycle of spreadsheet deals. We focused on the capture, storage, and security procedures already implemented by BNP. This analysis included interviews with users, an investigation into commercial capture systems as well as the feasibility of both our solution and outside systems. When requirements for a new system were found, we proceeded with our prototype. Through the development process, focus was moved to technical improvements on the current system and creating a working program to test the key concepts. When the program was completed, user feedback was gathered and further changes were made to improve the prototype for future use.

#### **3.1. Interviews**

To understand the system, we held unstructured interviews with various users, managers and developers. It was important for us to understand how employees use the system, especially the comparison between users at different levels of BNP Paribas. We held interviews with Jeremy Ascher from the Risk-Capital Market group. His group is in charge of validating the various calculations made by the system. We also spoke with Daniel Ward and Bruno Braziha both from the Flow Rates Group. They create and test the templates needed to start each spreadsheet deal. A short interview with Nikholas Lopez who works in the Front Office gave us insight to the end users of the spreadsheets.

Through the interviews, information was collected about the steps in the spreadsheet deal lifecycle, including opinions about the layout and ease of use of the system. We then considered the user's evaluation of weaknesses in the process and their ideas for improvement as well as any general information they shared with us.

#### **3.2. Evaluation of the current process**

The current system had to be analyzed in order for us to set the requirements for a new system. For this analysis we focused on the following factors: the duplication of effort for a new deal, any unnecessary steps within the current system, sources of delay

slowing the turnover rate, the flexibility of the system for various trades and also the data accuracy for each trade.

Through further observational analysis, we gained further knowledge of the multiple parts to the system. The systems were demonstrated to us by Laurent Frisch and Michael Tanousis. After the demonstrations we explored the systems ourselves asking for guidance when needed. Our team noted the roles involved in the creation, storage and retrieval of each spreadsheet deal.

With information on procedures found within BNP's intranet and gained from user interviews, each step in the current process was documented. This information led to the evaluation of the relationships between the inefficiencies and steps in process.

### **3.3. Development of the Prototype**

After the analysis of the current processes, the construction of a new capture system began. Once the requirements were documented for the prototype, potential solutions were investigated. These possible solutions went through consideration under a number of factors. Expenses were noted for the implementation of each system, including an examination into the operation costs such as testing and training, new software costs like installation and licenses, and also future costs such as upgrades or replacements.

Through discussions with our sponsors, it was decided that the group should focus their efforts on creating a full prototype for treasury lock calculations. After the specifications were chosen we concluded that the best solution was to circumvent the spreadsheet-dependant data storage process of Fahrenheit and eliminate a large portion of the system's operational risk.

### **3.4. Design Choices**

Analysis of the current systems led to exploration of various routes to a solution. We researched the advantages for different programming languages, using a choice of software programs and assorted system structures.

#### **3.4.1. Programming Language**

C# was chosen as the programming language for this project. The application requires using the Westminster library which can be called from three programming

languages; C#, C++, and VB. C# was chosen because it is more recent and complete documentation available from Microsoft, specifically the documentation on Excel manipulation which was a vital factor for this project. C# is also a good choice for object oriented design, which the developer was most comfortable with.

### **3.4.2. System Structure**

The following structures were considered for the development of our finished prototype.

#### **3.4.2.1. Single Executable**

The final system consists of a single executable file that can live on the server. The executable has very little requirements, only requiring the correct versions of .NET (at least 3.0) and the Westminster libraries it uses.

#### **3.4.2.2. Web Service**

The first system that we sketched out relied on a web service that consisted of a main application and many processing nodes. The main application would pull the information from the database then send the relevant data in a SOAP message to a processing node. The processing node would then run the calculations and return the appropriate values. Once the main application received back the values it would commit them to the proper locations. This approach added more complexity to the system than was needed and was therefore scrapped in favor of the single executable approach.

### **3.4.3. Valuation Calculations**

BNP requires a multitude of calculations in order to monitor their trades. Our report investigates which calculations were made by the BNP departments involved in the trade capture system and what these calculations required to become secure.

#### **3.4.3.1. Coded Calculations**

After our interview with the Flow Rates Group (FRG) it became apparent that performing the valuation using hard coded methods would require a lot of IT intervention. The spreadsheet process had a learning curve that the Dataflow group had overcome recently and changing the process now would require them to learn this new process and it would require some C# knowledge. Since the FRG is not trained to create



the valuation methods in a programming language this responsibility would fall on IT. Dataflow also uses the spreadsheets to debug and develop their models. This means that while Dataflow would still be producing spreadsheets they would then have to be translated into a programming language by IT, debugged, and then inserted into the system. This process would involve a lot of unnecessary and replicated work, debugging the model twice, leading to a greater lead time for adding new types of products.

#### **3.4.3.2. Excel Calculations**

Using standard Excel on a server requires starting an instance of Excel each time there is a workbook that needs to be calculated. This is not the most efficient process as each new instance requires time to start up and load the required DLLs. Since the workbooks that are calculated require Westminster to be loaded this process can be very time consuming.

Excel Services works in conjunction with Sharepoint Server and allows workbooks to be calculated on a server. The process starts with a workbook being uploaded to the Sharepoint server then a request is sent to Excel Services to calculate the book. Excel Services calculates the book then saves the changes back to the Sharepoint copy. Bean Stalk can then retrieve the calculated values from the Sharepoint and display to the user if it is a test/debug job. The advantage of Excel Services really comes from its distribution architecture. Multiple instances of the Excel Calculation Engine can be configured to provide computational power for the one instance of Sharepoint server. This means that when a new workbook is requested it can be sent to an open calculation engine and the number of calculation engines can be increased along with trade volume. Excel Services is not designed with this type of situation in mind specifically but it is a good solution to the problem of providing a distribution and load balancing solution without having to create it from scratch.

#### **3.4.4. Deal Storage**

Fahrenheit originally stored the deals in spreadsheets. These spreadsheets were generated by the Fahrenheit Sheet Builder that would run before the batch cycle and collect all of the deals and their data out of MAD and insert the appropriate values into a template spreadsheet then save all of the spreadsheets to a network drive. The new

system stores deals as XML files instead of spreadsheets. XML was chosen over a relation database because of the wide variety of products that could be handled by Fahrenheit. Instead of making a new table with specific columns for each type of deal, using XML we are able to define the attributes that a deal consists of on a per deal basis.

#### **3.4.4.1. XML**

Storing the deals as XML allows for each new type of deal to be stored easily with all of its attributes. XML is best suited for storing the deals because of the custom attributes can be assigned to elements stored in XML and the deals handled by Bean Stalk are there in part because of their attributes which vary from the standard deals. The other option for storing the deals would have been in a SQL or other similar relational database, which would require new tables created for each type of deal with columns for its attributes. In XML however, a new XML file can simply be created and specify any attributes for the deal without having to create anything new.

#### **3.4.4.2. SQL Database**

Although the attributes of the deal are stored in XML the deals are stored in a SQL database. The reasoning for storing the deals in the database is to allow for quick retrieval of the XML associated with each deal. The database consists of one table that contains two columns, deal number and XML content. The deal number is used as primary key and allows for the correct XML content to be retrieved. The XML content is stored in a column of the xml data type.

### **3.5. System Components**

Within the first couple of weeks of our project, we started compiling an outline of potential solutions for the required system. As the outline expanded we were able to choose the structure for a final prototype. This prototype involved the use of existing systems as well as our own work.

#### **3.5.1. Bean Stalk**

Through our previous research and development we found our solution to create a single executable program that will take the place of Fahrenheit. This application is called Bean Stalk. The first angle of the problem is how to store the deals. For this we chose

XML because of its custom attribute abilities which allow for new types of deals with varying attributes to be captured. The XML deal descriptions will be stored in a relation database along with columns for quick retrieval of the XML data of the requested deal. This will be quicker than trying to parse through the XML to find the matching data. Stalk models Jobs and Deals. The Deal object is extendable by any type of new Deal, the first type implemented is the TLockDeal. The inherited Deals override methods for loading its specific information into its variables and for the calculation of the PVs/risk. A DealBuilder object is delegated the responsibility of creating Deals and uses a switch to determine which type of Deal to create given the deal product type.

### **3.5.2. Excel 2007**

BNP has plans for a 2010 upgrade to Microsoft Office 2007 software. Keeping this information in mind, Excel 2007 was uploaded on a server for our use in developing Bean Stalk. Excel 2007 allows for a much greater amount of information to be processed. This increase in information also allows for a greater amount of risk to be incurred. Our system was developed to safeguard against this increased risk.

### **3.5.3. Microsoft SQL Server**

In the beginning of our development process, Microsoft SQL Sever was supposed to aid us in puling trade information for our calculations. After further discussions with our sponsors, the SQL Server was deemed unnecessary. Instead, the deal details were taken directly from MAD.

SQL Server was later added back into the Bean Stalk system as a means to store deal information for audit information. SQL Server was a good choice for this because it can hold the deal details directly as XML. Along with the deal details information about which job the deal was processed for and when are also stored to provide an audit trail.

### **3.5.4. MadMolly**

Due to the time constraints of the project, only certain calculations were used as examples for the prototype. MadMolly was created after we found a connection was needed between Bean Stalk and MAD.

### **3.6. User feedback**

We demonstrated the progress of the prototype several times throughout the duration of the project. We gained valuable information and suggestions from our sponsors during these demonstrations, each aiding in the production of the final prototype. We then observed the end user's interactions with the new interface.

In addition to these smaller demonstrations, BNP's London IT department was presented with the prototype. A presentation was made for Steve Gillen, the head of fixed income information technology in BNP Paribas' London office. With his suggestions and comments made by our sponsors, we were able to complete another iteration of the prototype using the information gathered to improve upon our original design.

## **4. Results and Analysis**

Using the methodology previously described we were able to assess the Fahrenheit spreadsheet valuation process in place at BNP. After identifying a series of areas of weaknesses, our group was able to design and create a functional prototype targeted at improving performance over these areas. The following section provides the results and analysis of our findings for the development of our trade capture system modification.

### **4.1. Analysis of Current Program**

The first step in our methodology, and the first objective of our project, was to analyze the current Fahrenheit system looking for areas of weakness or processes that have room for improvement. This analysis included compiling a detailed description of the steps involved in Fahrenheit and collecting the opinions of various employees involved in the process. This section describes the results of this analysis.

#### **4.1.1. Fahrenheit Process**

Securities traded by BNP are stored in one of several trade capture systems. Each capture system houses several groups of security types and handle the storage and valuation of trades in different ways. Each type of security is then classified into tiers by Risk-Capital Market, or R-CM. They evaluate the lifecycle and informational flow of a trade and categorize it into one of three tiers according to the requirements listed in Appendix C.

Tier 1 trades are fully represented within a “secure” system such as Star or MAD at inception. Star is a trade capture and valuation system for vanilla products like swaps, government bonds, foreign exchange securities, futures, and options, and is supported by StarPnL and the excel-based Wasp for risk management and pricing functionality. MAD is designed for the capture, pricing, valuation, and risk management of exotic interest rate, foreign exchange, and credit derivatives (M. Tanousis, personal communication, September 21, 2009).

Star, StarPnL, and Wasp call on Westminster, a BNP Paribas dynamic-link library (DLL) of pricing and risk management functions, for vanilla bonds and other interest rate products. Westminster also calls on several programs to gather the data needed to run

calculations. Ramp contains all official and live market data and distributes this data world-wide for real-time calculations (BNP, 2009c). Risk Vector Generator, or RVG, uses BNP's scenario engine to pre-calculate risk vectors at regular intervals for quick retrieval in Excel spreadsheets and a program called Cedar allows users to edit these scenario engine scripts and control parameters and their values.

Both systems store data in FIR files, which contain a secure snapshot of trade details within a program called Murex. Murex is run at the end of each day to calculate PVs using the PhaseTwo program for Westminster trades and distributes information to all BNP Paribas locations. The PVs are then fed and stored in Plato, BNP's official profit and loss system (BNP, 2009c).

Tier 2 and 3 trades are also fully or partially represented in an established system, but certain attributes are deliberately missing because the system is not equipped to adequately handle the valuation, pricing, or risk management of such deals. Instead, the valuation of these trades is performed on external spreadsheets run by the Front Office. These spreadsheets calculate PVs by calling Westminster functions, similar to the method used by Star in Tier 1 trades, but combine data stored in MAD with information stored in external spreadsheets before uploading the results back into the secure system. These systems also represent trades where the booking is correctly represented in the secure system at inception but the entire valuation of the deal is executed externally (BNP, 2009b).

Tier 3 is typically a classification for new trades whose calculations are not fully validated by R-CM. After R-CM is comfortable with the basic valuation structure of a new trade, it is placed within Fahrenheit similar to existing trade types (personal communication, L. Frisch, November 10, 2009). Over time, R-CM further approves the calibration, configuration, and model parameters input directly into the spreadsheet. If corrections need to be made, the templates return to FO who must inform MO and R-CM via email of any modifications to the spreadsheet template. These emails are archived by MO for auditing purposes. When all concerns have been addressed and all processes involved with its run including market data population and the upload of PVs and risk figures are automated, trades are generally upgraded to Tier 2 (BNP, 2009b).

Each new trade type within these systems needs approval by the Trade Acceptance Committee (TAC) as an Exceptional Transaction (ET). New spreadsheet templates are then created and verified before entering a production environment. Various responsibilities of the control system in which the spreadsheet operates are then shared by R-CM and the Front and Middle Offices (FO, MO) throughout the trade’s lifecycle. The following chart outlines the responsibilities for the New York office (BNP, 2009b).

<b>Chart of Responsibility for New York</b>	<b>Tier 2</b>	<b>Tier 3</b>
Declaration of a new spreadsheet	FO	FO
Monthly approval of spreadsheet PV and risk	NR	R-CM
Populating deal details	FO IT	MO
Checks on contractual details	MO	MO
Populating market data	FO IT	MO
Generating spreadsheet PVs	FO IT	FO IT
Generating spreadsheet risk	FO IT	FO IT
Transferring spreadsheet PVs to External Leg	FO IT	FO IT
Transferring spreadsheet risk	FO IT	FO IT
Transfer of cash relating to spreadsheet deal to MUREX	FO IT	MO
Number of copies running above processes	2	2

**Table 1: Trade Capture Responsibility Breakdown**

One of these valuation spreadsheet processes is known as Fahrenheit, which uses pre-created templates populated with deal details to generate PVs and risk reports. After approval from TAC for a new deal type, valuation templates are created and tested by the Flow Rates Group (FRG) and outputs are validated by R-CM. These templates are single sheet workbooks that use the details of a trade and market data to calculate its PV. If changes need to be made, the deal returns to testing until no discrepancies are found. At the execution of a new trade details are entered into MAD by the trader as an “enhanced external leg”, which tells MAD its valuation will be completed externally. The spreadsheets acts as the link between MAD and Fahrenheit, providing the information

needed to run the calculations in Excel (D. Ward, personal communication, November 12, 2009).

Fahrenheit “jobs” are predefined runs categorized broadly by a product type (i.e. Treasury Lock) and a risk object (i.e. PV, delta, gamma, etc.) and include information about the currency type, portfolio number, group number, and Ramp set. The table below is an example of three job definitions in Fahrenheit. Although not shown, each job also includes a Cedar report group, an output path for the calculation results, and whether or not to upload the results to MAD (which it does only for PVs).

<b>Job</b>	<b>Currency</b>	<b>Portfolio</b>	<b>Product</b>	<b>Group</b>	<b>Ramp Set</b>	<b>Ramp Date Offset</b>
PV_TLock	USD	274	TLock	All	Official	0
Delta_TLock	USD	274	TLock	All	Official	0
Vega_TLock	USD	274	TLock	All	Official	0

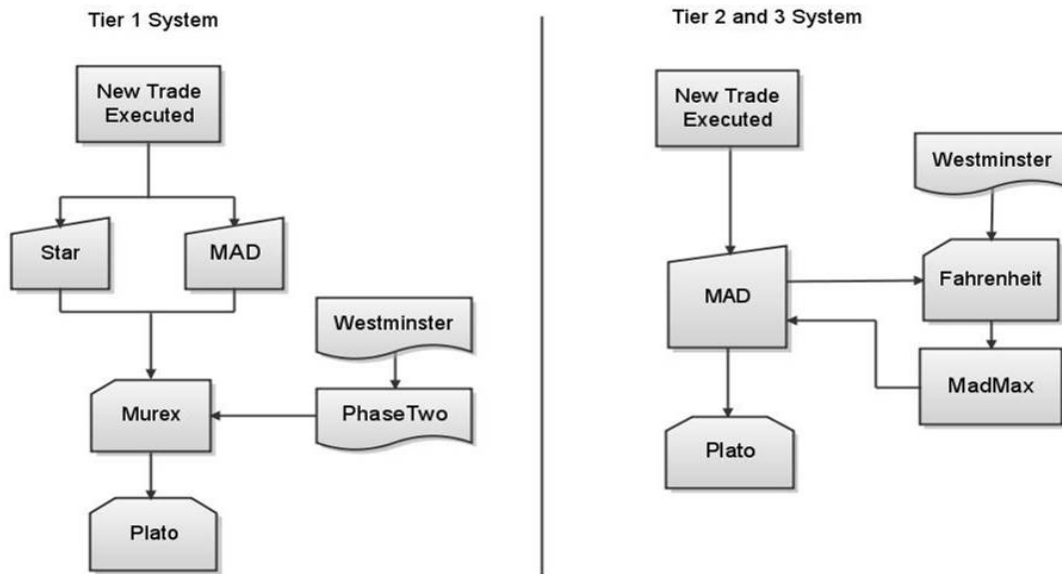
**Table 2: Fahrenheit Job Description**

Fahrenheit jobs are run daily alongside MAD’s overnight valuation batch. Each job matches a report in Cedar which loads risk vectors from RVG to quickly calculate a job’s risk objects. The group number theoretically allows Fahrenheit to split jobs across multiple CPUs by only calculating those in the group number defined by the user. BNP does not currently use this feature, however, because it must be done manually and the job sizes change regularly, making it difficult to assess the benefits of splitting calculations. The Ramp set and date offset tells Westminster which market to load data from and the currency, portfolio, and product columns follow directly from the deals involved in the job.

When launched, a spreadsheet called Fahrenheit Spreadsheet Builder is called to populate the individual deal templates with information from MAD using a program called MadMax. The Fahrenheit valuation workbook then opens and loads the Westminster DLL and the deal templates needed for each specific job. After finishing a job, the outputs are saved to the defined output directory with a DDMM extension and the new PVs are uploaded into MAD by MadMax. Once all jobs have been run, the newly created Fahrenheit valuation workbook is terminated and the process is completed.



The figure below reviews the differences between the capture and valuation processes of a Tier 1 trade and a Tier 2 or 3 trade using Fahrenheit for valuation purposes.



**Figure 1: Trade Capture and Valuation Process**

#### 4.1.2. Concerns with Fahrenheit

Through our interviews we found a disconnect between the various BNP departments in regards to the requirements for spreadsheet trades. The outlined procedures created by R-CM were not completely understood by both IT and FRG. Jeremy Ascher (R-CM) explained a Tier 1 trade should be entirely contained within a bank system such as MAD or Star. In comparison to MAD and Star, Excel is known for high operational risk from both human error and difficulty in validation. R-CM has to check PV calculations every month, which he believes is an unnecessary strain on resources. If Excel is to be used at all, it should be run in the background of programs, with no ability for users to manipulate the spreadsheets themselves.

While comparing the requirements for a Tier 1 deal, Dan Ward of FRG argues that Fahrenheit is identical to any other MAD deal. Despite its characterization as Tier 2 he

believes the Fahrenheit system is more secure than many people will credit it. The final deals that are valued are still ultimately stored in MAD, which according to both R-CM and FRG is a secure system with an established IT support group. FRG does recognize the risk of malicious or fraudulent activity if Excel is used in a program that is not tightly controlled. However there are support systems for Fahrenheit that he feels that fulfill the requirements of R-CM. R-CM's lack of confidence in Fahrenheit's Excel use bogs down the process and creates delays when information is relayed between the groups.

Our interviews also introduced the issue of allowing for more trades to take place. At the moment the system itself is an obstacle in BNP's ability to value a greater number of trades.

Limit	London		New York		Tokyo	
	Tier 2	Tier 3	Tier 2	Tier 3	Tier 2	Tier 3
IFRX	40	15	40	15	25	15
Global Credit	40	15	40	15	25	15

**Table 3: Current Trade Limits by Tier**

These limits are not only a result of the security concerns of the current systems, but also the over-night calculation time for the PVs of the trades. Every trade needs to have real-time market data that is refreshed by the end of the day. Simply increasing the number of trades would require a longer period of time to update the trades, which is not an option.

#### **4.1.3. Microsoft Excel**

Despite its critics, Microsoft's Excel is the leading software for modeling, analyzing and pricing financial instruments on Wall Street. In a survey conducted in the summer of 2009 by the Tabb Group and Microsoft, 83% of participating financial firms were found to be using Excel, with 20-30% of the firm's information stored with the software (Crosman, 2009). Nonetheless, there are two common perspectives when it comes to using Excel. The first is that Excel cannot be trusted. There are flaws in security, the risk

for human error is huge and the effort to monitor systems ties up a lot of company resources. This results in vendors developing applications similar to Excel in efforts to replace the software entirely. The second perspective is that no matter what vendors may develop, the financial industry will always use Excel and vendors must develop applications that work around it (Baker, 2009).

Numerous financial services firms have been attempting to move critical business functions away from Excel for years, citing difficulty in monitoring and maintenance of the software (Levine, 2008). However Excel empowers users with the flexibility to create their own add-ins for varied functionality. In addition to in-house systems, several solution providers do offer software that monitors Excel and addresses auditing concerns (Levine, 2008). Currently, the focus is on the individual market when it comes to trade software and not the major financial companies. TD Ameritrade, Scottrade and Charles Schwab are just a few of the firms contributing to the expectations of online trader customers who expect a high level of service, access to more sophisticated tools and even educational seminars and guidance (Schmerken, 2008). Yet, the ease, flexibility and speed of Excel have not yet been addressed by the market and no such all-inclusive financial system exists for major firms to replace the Microsoft software (Levine, 2008). According to the European Spreadsheet Risks Interest Group, "spreadsheets are integral to the function and operation of the global financial system. It's unrealistic to expect them to be replaced in the short term" (Miller, 2005).

Untested spreadsheets almost certainly contain errors. Given the flaws in many company spreadsheet processes and policies, these errors can often be found in the most fundamental formulae entered in the first place. An interview with a London-based firm disclosed an instance where a "material spreadsheet error" led to the loss of tens of millions of pounds. The main issue lies with the size of the spreadsheets. Evidence from the financial markets shows that spreadsheets are often operating beyond their technological limits. On top of this, firms insist on building in-house IT systems for their own specific calculations. While the sheets are built by employees with strong knowledge of the mathematical models they use, they frequently lack knowledge of basic IT security controls. Financial firms commonly put aside a sizable amount of money to combat the risk of failure while trying to value complex derivative investments (Miller, 2005).

However these fears are combated in-house, by software vendors and even by Microsoft itself. In 2008, Microsoft unveiled a partnership with Bloomberg allowing traders, researchers and middle office employees to streamline their use of Microsoft Excel 2007 in conjunction with a number of Bloomberg databases. This collaboration is the beginning of a trend in collaboration for a safer financial system based on client demand (Microsoft, 2008).

Most commercial systems focus on security and compliance issues, with emphasis on auditing, not streamlining the calculations themselves. Outsourcing security leads to further issues if proper communication and accountability is not addressed. Many vendors may outsource to another party themselves and clients must be aware of protection of personal and sensitive data (Rodier, 2008). While implementing these systems will increase security, human error can still occur. Many of these systems offer ways to view the history of the spreadsheets to combat this issue (Crosman, 2009).

After the interviews with R-CM, FRG and IT, we found that they share a basic concern. BNP needs a secure system to validate more trades with less effort. However it is easy to understand the concerns that R-CM has about the use of Excel.

Jeremy Ascher of R-CM insisted that the best solution would consist of a system that has the new deals fully integrated into one of BNP's official pricing systems. An official pricing system for the new trade would have to go through the necessary research and testing, needing less effort from R-CM for validation. If there is any interaction between the user and Excel, Ascher explains, then the system is deemed unsafe and in need of constant validation.

FRG had slightly differing opinions yet agreed that continuing to write new templates in Excel may not be the ideal solution. The only issue here is that Excel has the flexibility and transparency needed to calculate the trades. This calls for a secure way to use Excel without any manipulation of the spreadsheets themselves.

The concern with Excel, as IT states, is the assumption that the software is fundamentally insecure. For any solution the only obstacle to Tier 2 trades becoming Tier 1 trades is the use of Excel. Even with a solution where the spreadsheet only exists at the time of the calculation, R-CM hesitates to re-label the trade's category. Discussions

should take place between IT, FRG and R-CM about the requirements for tiered trades and ways to improve the clarity and necessity of the restrictions on trade.

## **4.2. Potential Solutions**

When exploring potential solutions for BNP's trade capture system we considered commercial systems, custom-made applications and industrializing the current process. Commercial systems were judged by the services they offer as well as the requirements they met that are needed by BNP's system. Various custom solutions were also discussed, such as replacing Excel with code or even creating a new system. The final course of action included using Excel solely for calculations, industrializing the process of each deal capture and replacing the storage of deals with a new format.

### **4.2.1. Commercial Systems**

When a trade has begun, it is critical to capture the trade details regardless of the market type. This capture must accurately update the position in real time for auditing and risk management purposes. The main goal of the trade capture system is to ensure any trade is up to date to avoid costly corrections.

A trade capture system provides a single solution that integrates all trading activities across multiple asset classes, offering breadth as well as depth of functionality. Most commercial systems claim to deliver a technological advantage in advance financial engineering. This is mainly accomplished by:

- Interface to the electronic platforms that can handle both vanilla and exotic trades.
- Full data mapping between the input data for the trade and market data.
- Integrated front, middle and back-office functionality.
- Complete audit trail for tracking the actions and the values changed.
- Reporting system that provides flexible configuration to automatically schedule reports or creation of reports on request.

A survey of current available systems reveals them to follow these general parameters; however no system found met all of these parameters (Eurobase, 2009; Prospero, 2009; Spectrum, 2009).

Database:	<ul style="list-style-type: none"> <li>• Oracle</li> <li>• MS SQL Server</li> <li>• Sybase</li> </ul>
Language used:	<ul style="list-style-type: none"> <li>• C++</li> <li>• Java</li> <li>• C#</li> </ul>
Operating system:	<ul style="list-style-type: none"> <li>• AIX</li> <li>• HP-Unix</li> <li>• IBM i (i5/OS), z/OS</li> <li>• Linux</li> <li>• Solaris</li> <li>• Windows 2000, 2003, NT4, Vista, XP</li> </ul>
Pricing structure:	<ul style="list-style-type: none"> <li>• Per seat</li> <li>• Fixed</li> <li>• Volume Based</li> </ul>
User interface:	<ul style="list-style-type: none"> <li>• GUI</li> <li>• Web</li> </ul>

**Table 4: Trade Capture System Parameters**

Various commercial trade capture systems distinguish themselves from the competition with different services offered. Some systems stress Sarbanes Oxley, FASB and BASEL II compliance of its system (QuantRisk, 2008). This feature can cut time in the validation of the legality of trades. Ergonomic systems are important, offering services such as filtering, sorting, drag and drop column order and screen views. Peresys created one of the few systems to provide a member to member messaging service for faster communication (Peresys, 2009). Another company recommends their system which is broken up into modules that work with each other to form a fully integrated STP system. Each module can be implemented in phases or in an overhaul of the current system with no coding or scripts required for configuration (Sage, 2009).

While the previously mentioned systems cover needed calculations, the foundation of trade capture relies on recording the details and changes of each trade. Numerous programs compete with Excel in this market. The basic features offered include intuitive data entry windows or wizards with filtering options for various trade views. Like the calculations systems, they offer both FASB-compliant financial statements as well as browser interfaces for anytime, anywhere reporting. Depending on the software

additional features incorporate timelines options for viewing data over any desired period and also elimination of redundant data. These systems have various pricing plans from starting at a base price of around \$2,300 not including any customizations. Renewable licenses are required by most vendors at a price of around \$1,150 with additional users costing \$130-\$1000 each (Knaster, 2004).

The security of trades is a concern of all banks world-wide. There are many companies that focus on making security software for banking firms. Monarch from Datawatch is an exemplary model of business intelligence software that investigates fraud. This software reads reports in any format, parses the information and then exports it to Excel (Crosman, 2009). The most recent version of Monarch takes advantage of Excel 2007's ability to export documents in full-fidelity XML representation. This also enables Monarch to add optional digital signatures embedded within each Excel file. However, it's only in Monarch-generated spreadsheets that users can determine whether the original report has been altered (Baker, 2009). Wall Street banks including J.P. Morgan Chase, Citibank, Scottrade and Wells Fargo use the Monarch software (Crosman, 2009).

Large financial companies remain largely responsible for the creation of add-ins used to aid in the calculations required, BNP included in this group. The majority of outside systems are marketed to smaller financial firms and would often need customization or re-configuration to fully integrate with a company such as BNP. Time and money can be lost depending on the extent of changes needed. This downtime does not even consider the needed integration, training and testing of the potential system. If BNP were to consider a system overhaul, it would be likely that most of IT's functionality would be re-focused on communications with the vendor over the requirements of the system. Future replacements and upgrade costs would also fall under discussion.

#### **4.2.2. Creation of a Custom Application**

With a custom application, a new in-house system would be created for BNP to eventually replace Excel entirely. Calculations once evaluated by Excel would be replaced with extensive code. This code would have to be written and tested by R-CM, FRG and IT. This solution answers R-CM's wishes for a new pricing system to validate

trades without the use of an insecure Excel system. The storage of the deals would likely be placed into a new format such as xml that lessens the risk of fraud and error.

The most plausible custom application would rely on a web service. Information would be input about the trade and the relevant information would be pulled from the database. Calculations would run and return the appropriate values. After receiving these values they would then be distributed to the proper location. This custom application would add further complexity to the system.

Using a newly coded system would require IT intervention, with time, research, testing and training needed to complete the new system. Our seven week time limit would only allow for a sample of the project to be completed, resulting in only a section of a proof-of-concept. The FRG would also be required to learn C# for the new process as well as how to use the system itself. They have already spent a good deal of time learning the best ways to create their pricing templates. Many employees in the FRG are responsible for spreadsheet development and do not know the programming languages necessary for the new system. This would add strain on IT to aid in the creation of calculation models.

Another solution to consider would be the use of Excel Services which works in conjunction with Sharepoint Server. The spreadsheets could be calculated on a server, reducing operational risk. The advantage to using Sharepoint would come from using its distribution architecture to increase the limit of trades that can be calculated at any given time. However Excel Services is not specifically design with our trade process in mind and does not work correctly with BNP's current system. Industrialization of Fahrenheit

#### **4.2.3. Industrialization of Fahrenheit**

Industrializing the current process would allow for Excel to solely be used for calculations. There would be no storage of spreadsheets within Excel and the spreadsheets would only exist at the time of calculation. Within this system the storage of the deals would likely be placed into a new format such as xml that lessens the risk of fraud and error, much like the potential custom application. This system would have few requirements and pull information from the Westminster for the resulting calculations. This solution addresses the main issues brought up in our interviews. Security is ensured



with the spreadsheet only existing as the calculations run. The number of trades that can be captured is increased with the number of CPUs in use.

### 4.3. Selection Criteria

Now that we have identified a set of potential solutions, we need to develop a means of determining the strengths and weaknesses of each to select the optimal one. By collecting and organizing a series of important characteristics and comparing each system with respect to these, we can lay the foundation for the selection of the most appropriate course of action. The two broad categories we will use to analyze these systems are their technical capabilities and their logistical feasibility within BNP’s current information technology framework and our project’s time frame. The table below displays the characteristics we will use to evaluate the effectiveness and plausibility of each solution.

	<b>Outside System</b>	<b>Custom Application</b>	<b>Industrialized Fahrenheit</b>
<b>Technical Capabilities</b>			
Flexibility			
Security			
Scalability			
Ease of Use			
<b>Feasibility</b>			
Cost			
Work Required			
Implementation			

**Table 5: System Selection Criteria**

The three solutions identified earlier will be ranked on a scale of Excellent / Good / Fair / Poor for each characteristic and given a score for both categories based on the results. An Excellent ranking will result in a numerical score of 3, a Good ranking in a score of 2, a Fair ranking in a score of 1, and a Poor ranking in a score of 0. Each characteristic will then weighted by its importance within each category and final score can be computed.

For example, assume we decide that each characteristic within a category should be weighted equally. If our analysis of the creation of a custom application then yields results of “Excellent” for flexibility, security, and cost, “Good” for scalability and work required, and “Fair” for ease of use and implementation, we can calculate the resulting scores of this solution as follows:

$$\text{Technical Capability Score} = (3 + 3 + 2 + 1) / 4 = 9/4 = \mathbf{2.25}$$

$$\text{Logistical Feasibility Score} = (3 + 2 + 1) / 3 = 6/3 = \mathbf{2}$$

Comparing these scores to our other solutions will ultimately aid in the selection of a final solution. This analysis provides a quantitative representation of the “best fit” solution.

#### **4.3.1. Technical Capabilities**

The technical capability of each potential solution is of obvious importance to its eventual overall effectiveness and efficiency. An ideal solution will be one that addresses the concerns of Fahrenheit without sacrificing its beneficial attributes. The technical capability category is composed of four characteristics that highlight several areas of importance for an effective trade valuation system. These broad areas are comprised of the system’s flexibility, scalability, security and integrity, and ease of use.

Our group has determined that maintaining the flexibility of Fahrenheit in our solution is the most important characteristic and accounts for 35% of a system’s technical capability. Similarly, a system’s security, scalability, and ease of use account for 30%, 25%, and 10% respectively. Using these percentages we can formulate the equation for a model’s technical capability score as a function of its individual characteristic scores. We use a system’s numerical flexibility score F, security score S, scalability score V, and ease of use score E to develop the following equation.

$$\text{Technical Capability Score} = .35(F) + .3(S) + .25(V) + .1(E)$$

The overall technical capability score, similar to the individual characteristic scores, will range from zero to three. The following sections will provide detailed explanations on how the weight of each category was developed and justified.

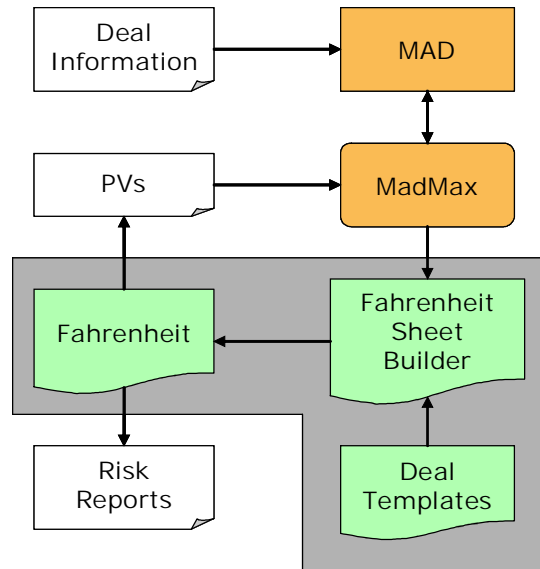
#### **4.3.1.1. Flexibility**

Fahrenheit was originally designed to value trades that MAD was not equipped to properly handle. When a new trade is proposed and it is determined that MAD is not capable of computing its value, two options are presented. Either MAD must be updated to include the valuation of the new trade type or Fahrenheit must be staffed with this responsibility. The time it takes to design, implement, and verify a new trade's calculation coding in MAD is incredibly lengthy and costly. Fahrenheit's primary advantage lies in its flexibility with regards to new deal types through use of Microsoft Excel templates, which allow for relatively quick additions to the model. Thus, trades whose valuation complexity is not beyond the scope of Excel and whose trading volume is not expected to be substantial enough to justify its addition to MAD are delegated to Fahrenheit.

This flexibility is a critical aspect in the underlying foundation of Fahrenheit and an integral component of any new or updated system. This importance must be reflected in the assessment of our potential solutions and as such, flexibility will carry the largest weight for the technical capability category at 35%.

#### **4.3.1.2. System Security and Deal Integrity**

During the analysis of the Fahrenheit process, our group has determined its primary weakness is its lack of information security. Deal details are stored in Excel documents on network servers and lack substantial control mechanisms. Additionally, proper documentation for audit purposes throughout the process is nearly non-existent. The diagram below demonstrates the flow of information throughout the Fahrenheit process.



**Figure 2: Fahrenheit Process Diagram**

The portion of the diagram highlighted in grey represents Fahrenheit’s employment of a spreadsheet called FahrenheitSpreadsheetBuilder, which populates Excel templates with MAD data through MadMax, to load jobs for the Fahrenheit valuation spreadsheet. This exchange is the centerpiece of Fahrenheit’s lack of valuation security or integrity and represents a substantial piece of its operational risk. The Excel files are stored on accessible network drives and have little protection against accidental or malicious changes. Additionally, input data is untraceable and unreliable for auditing purposes, as the information stored in the spreadsheets reflects only the most recent data gathered from MAD.

During the analysis of our three systems, the security of information throughout the valuation process will be a key factor in its selection as a replacement for Fahrenheit. Security will compose 30% of the technical capability category.

#### **4.3.1.3. Scalability**

Our analysis has also identified Fahrenheit’s scalability to be one of its largest liabilities. Its dependence on a single Microsoft Excel workbook for calculations limits the size of runs to the memory and calculation capacities of Excel. This limitation has been established as one of the primary disadvantages of the current system and provides an obvious flag for Risk Management in its Tier classification and security review

process. Without improvement to this limitation, no solution can be truly effective as a permanent alternative to MAD deal valuation.

A new system's response to these scalability concerns lies within its ability to easily distribute computations across multiple servers and CPUs. The development or existence of a mechanism to control this process is an important component of our new deal valuation system. However, the importance of a system's scalability should not take preference over its ability to maintain Fahrenheit's flexible design or the need to bolster the integrity of the entire valuation process. Our assessment will allot 25% of the weight for the technical capability category to scalability.

#### **4.3.1.4. Ease of Use**

While a valuation model's ease of use will not be considered as important as its other technical attributes, it should not be completely discarded either. The users of a new system will likely be unreceptive to an alternative whose operational process is complex and confusing. A system's "ease of use" refers to the amount of effort required by BNP employees to load and run the valuation and risk sensitivity generation process on a daily basis. This includes both the time required to complete any steps and the complexity of the process. A system's ease of use will account for the remaining 10% of its technical capability score.

#### **4.3.2. Logistical Feasibility**

As the second evaluation category, the logistical feasibility of a system concerns its ability to operate efficiently within BNP's existing IT framework. In addition to the ability to meet the technical demands of our new system, the model must also be feasible in its creation, implementation, and maintenance. A model that is difficult to implement and results in high maintenance needs is unlikely to be an effective permanent solution. The individual characteristics of the logistical feasibility category include total initial cost, future maintenance work required, and implementation ease.

The total initial cost of a system refers to the monetary value of creating or licensing software for the model. It does not encompass the salaries of employees needed to design or maintain the system or any long-term monetary costs as these factors will be considered in our other characteristics. Total initial cost is an important characteristic to

consider for fairly obvious reasons. High initial costs can partially or fully negate the technical advantages gained, result in wasted capital investment if a system fails to meet expectations, and provide difficult political hurdles for management to clear to gain support. Total initial cost will account for 40% of the logistical feasibility score.

The maintenance required for a valuation system is a second important contributor to the logistical feasibility of a new model. The work required adding new trade types will generally coincide with a model's technical flexibility, but this characteristic also focuses on the maintenance required for a specific system over the long-term. A valuation system whose monitoring and upkeep require substantial time investments from the IT department will cost BNP higher labor expenditures throughout its lifetime. Additionally, the work required to maintain the system will affect its reliability and consequently its technical capabilities. The work required to maintain a system will be allotted 30% of the logistical feasibility score.

Finally, we must consider the ease with which BNP will be able to implement a new system. Implementation will reflect the necessary labor investment at the installation and immediate outset of a new programs life cycle and its corresponding costs. High implementation costs can result in the waste of financial and human capital and provide political hurdles for management similar to those created by high initial software creation and licensing costs. The cost of implementation will account for the remaining 30% of a system's logistical feasibility score.

From these percentages, we can construct the logistical feasibility score as a function of the three individual characteristics similar to the way we did with the technical capability score. We use a system's numerical cost score  $C$ , work required score  $W$ , and implementation score  $I$  to develop the following equation.

$$\text{Logistical Feasibility Score} = .4(C) + .3(W) + .3(I)$$

The overall logistical feasibility score, similar to the individual characteristic scores, will range from zero to three. This score will represent the ability of BNP's current technology framework to easily assimilate each valuation system.

### 4.3.3. Mathematical Selection Model

With the creation of a numerical score for each category we are able to create a model that uses the parameters specified through our analysis to select the “best fit” solution. While the logistical feasibility and technical capability scores will be meaningful in helping compare models, they do not establish concrete selection criteria. Potential ties within categories and the importance of logistical feasibility versus technical capability are not considered in the formulas given earlier. To address these issues, we have created a model that uses the same inputs we have already specified to identify the strength of each system in a singular numerical value.

In addition to the logistical feasibility score (LFS) and technical capability score (TCS), we must also calculate the average for each category as follows:

$$TCA = (F + S + V + E) / 4$$

$$LFA = (C + W + I) / 3$$

These averages represent the total value of the benefits presented by each model without considering their importance and will range from 0 to 3. By multiplying the scores by the averages we will be able to better distinguish between systems for each category and create a new score with a range from 0 to 9. Using these four values and an arbitrary probability value  $p$ , we now create a model with a single numerical output defined as a system’s total value  $\Omega$  as follows:

$$\Omega = [p \times (TCS \times TCA)] + [(1 - p) \times (LFS \times LFA)]$$

Our arbitrary probability value represents the importance of feasibility versus technical capability. In this model  $p$  defines the weight of the technical capability of a system. By calculating  $\Omega$  for a range of  $p$ -values we can see which system is optimal across various importance values for each category.

#### 4.4. Model Analysis

With the selection criteria in place and a mathematical model constructed, we shift our attention to the application of the analysis to the three systems identified earlier in this section. Using the descriptions of the individual characteristics as a guideline, we ranked each potential solution using the Excellent through Poor rating system. The final results of this analysis are shown in the table below.

	<b>Outside System</b>	<b>Custom Application</b>	<b>Industrialized Fahrenheit</b>
<b>Technical Capabilities</b>			
Flexibility	Poor	Fair	Good
Security	Excellent	Good	Good
Scalability	Excellent	Excellent	Good
Ease of Use	Good	Excellent	Excellent
<b>Feasibility</b>			
Cost	Poor	Excellent	Excellent
Work Required	Poor	Fair	Excellent
Implementation	Poor	Good	Excellent

**Table 6: System Characteristics Analysis**

Applying the numerical formulas described earlier we calculated a technical capability and logistical feasibility score and average for each solution, the results of which are in the table below.

	<b>Outside System</b>	<b>Custom Application</b>	<b>Industrialized Fahrenheit</b>
<b>TC Score</b>	1.85	2.00	2.10
<b>TC Average</b>	2.00	2.25	2.25
<b>LF Score</b>	0.00	2.10	3.00
<b>LF Average</b>	0.00	2.00	3.00

**Table 7: System Selection Scorecard**



This table clearly suggests that the industrialization of Fahrenheit appears to be the optimal choice as it has the highest technical capability score/average and the highest logistical feasibility score/average. Applying these four values to the total value equation defined earlier we can calculate  $\Omega$  for our series of p-values below.

<b>p -Value</b>	<b>Outside System</b>	<b>Custom Application</b>	<b>Industrialized Fahrenheit</b>
<b>0 %</b>	0.00	4.20	9.00
<b>10 %</b>	0.37	4.23	8.57
<b>20 %</b>	0.74	4.26	8.15
<b>30 %</b>	1.11	4.29	7.72
<b>40 %</b>	1.48	4.32	7.29
<b>50 %</b>	1.85	4.35	6.86
<b>60 %</b>	2.22	4.38	6.44
<b>70 %</b>	2.59	4.41	6.01
<b>80 %</b>	2.96	4.44	5.58
<b>90 %</b>	3.33	4.47	5.15
<b>100 %</b>	3.70	4.50	4.73

**Table 8: “Best-Fit”  $\Omega$ -Value Comparison**

The  $\Omega$ -comparison test confirms our initial implication of the industrialization of Fahrenheit as the optimal solution, as we can see that this system returns the highest total value for the entire plausible range of p-values. Before we can accept the results returned from this analysis however, we must be able to prove they are reasonable. First, we must support our classifications of the individual characteristics for each system with the research collected through out the project. Secondly, we must examine the robustness of our model by demonstrating that the results returned remain stable across a potential margin of error in our parameter definitions.

#### **4.4.1. Technical Capability Characteristic Classifications**

The use of a third-party trade capture system is the most difficult of the three options to evaluate. Because of the wide variety of software programs available, choosing the “average” or “most relevant” system to base our classifications upon is incredibly

challenging. To simplify this process, our group has determined to evaluate the “Outside System” option based on software solutions that directly address the weaknesses identified in our Fahrenheit analysis: the unreliability of Excel as a storage mechanism, its lack of documentation procedures, and its inability to handle a high volume of calculations.

The majority of systems that fit this description are software programs that encompass the entire trade capture and valuation process, rather than working simply as an external enhancement like Fahrenheit. While plenty of systems exist that are designed to remain flexible and involve less complications from a logistical standpoint, most of these systems would largely fail to address the concerns identified in our analysis and provide little more than a replacement to Fahrenheit without providing many new beneficial attributes.

The obvious strengths of this type of outside system are its contributions to security and scalability. Similar to MAD, an outside capture and valuation system would be self-containing and easy to control in a production environment. Additionally, we can assume that the design of such a system is geared towards high-volume calculations and not inhibited in the same manner as the current Excel-based approach. Because of this, the outside system option is given a rating of “Excellent” for both security and scalability. Similarly, we assign a rating of “Good” for ease-of-use on the assumption that our selected third-party system is user-friendly and not overly-complex in nature, requiring only minimal new-user training.

Unfortunately, the additional security and scalability of the outside system come at a steep cost to the model’s flexibility. An outside system works primarily in the same manner as MAD, and as such does not lend itself well to the addition of new trade types. Created by third-party vendors, these systems are generally not designed to be tampered with and additions to the model will likely require a heavy amount of labor investment to re-code and validate the calculations of new trade types. The “Outside System” option is assigned a rating of “Poor” for flexibility.

Similar to the use of a third-party system, the creation of a “custom application” is aimed directly at addressing the issues of security and scalability in Fahrenheit. The replacement of Microsoft Excel with a code-based approach to calculations and

information storage adds a layer of protection to the process and creates additional documentation capabilities. The custom application alternative is given a rating of “Good” for security, falling short of “Excellent” because of its continued practice in importing and exporting data from an outside source and its inability to match the higher security levels of a self-containing trade capture system. This option is also assigned a rating of “Excellent” for scalability because of the ease with which calculations can be distributed amongst various servers and processing units. Additionally, assuming the relatively quick and easy creation of a complimentary graphic user interface (GUI), this particular solution is also assigned a rating of “Excellent” for ease-of-use.

While this type of approach to valuation is certainly more flexible than the use of a self-containing trade capture system, it still falls far short of the flexibility offered by Fahrenheit and other Excel-oriented approaches. The need to write new code for each additional trade type and the lack of regularity with which this process can be carried out raise serious concerns in the system’s ability to adapt to new deal types. The custom application option is given a rating of “Fair” for flexibility.

Unlike our first two options, the industrialization of Fahrenheit does not employ the creation or installation of a new system to address the concerns raised in our analysis. Rather, this option bolsters the current Fahrenheit process in its weakest areas and transitions the system from a temporary alternative to a permanent valuation system. This continued use allows for a rating of “Excellent” for ease-of use. Much like the creation of a custom application, this type of solution will also considerably strengthen security by removing the process’s dependency on Excel as a storage mechanism, and as such is given a rating of “Good” for security.

Perhaps the weakest alternative with regards to scalability however, this solution still requires the use of Excel spreadsheets to perform calculations. While the distribution of these calculations across multiple servers is possible, the design and implementation of a distribution mechanism is considerably more involved than using a third-party or in-house code-based system. Additionally, the need for another program to import data from MAD and prepare it for use in Excel will require small code-changes for new trade types. Because of these minor setbacks, the industrialization of Fahrenheit is assigned a rating of “Good” for scalability and flexibility.

#### **4.4.2. Logistical Feasibility Characteristic Classifications**

Having assessed the technical capabilities of each system we repeat the process for the logistical feasibility characteristics: initial cost, maintenance work required, and implementation ease. The initial cost refers to the licensing and purchasing costs associated with the software needed for the system and the third-party option is glaringly weak in this regards. Requiring the purchase of an entirely new trade capture system and the licensing fees associated with it, the outside system alternative is given a rating of “Poor” for cost. Similarly, the implementation of an entirely new trade capture system will require enormous labor investments through out BNP and is also assigned a rating of “Poor”. Created by a third-party vendor, expert knowledge of the new system will likely have to come from outside the company. This lack of inside knowledge may cause difficulties when resolving day-to-day issues and leads to a third rating of “Poor” for work required.

The second potential solution, the creation of a custom application, does not require the purchase of any additional software as it would be created from within the company. Despite this lack of cost, its implementation would be slightly more complex because the validation of its calculations will be slower than it would be for a system that uses the current valuation spreadsheets as a calculation platform. With strong ratings of “Excellent” and “Good” for cost and implementation respectively, however, this system still far surpasses the feasibility of its third-party counterpart. The custom application’s foremost weakness lies in the maintenance work required. Unfortunately, similar to the third-party system, the use of code for calculations prevents the easy and routine addition of new trade types and requires the design, creation, and validation of new formulas for each new deal type. Debugging large quantities of calculation code is also substantially more complicated than debugging errors in Excel. Slightly easier than the work required for the outside system due to the inside knowledge gained from creating the application within the company, this system receives a rating of “Fair” for work required.

Our final alternative, the industrialization of the Fahrenheit process, is the least invasive to the current BNP infrastructure. Only requiring the transition from Excel 2003 to Excel 2007, this process will involve minimal initial and long-term maintenance costs. Similarly, the addition of new trade types will follow the same process already involved

with Fahrenheit and save BNP the hassle of adding new code as needed in the other two alternatives. Other maintenance work will primarily focus around the macros that run the calculation spreadsheets and will be substantially quicker than the maintenance required for our other two systems. For these reasons, the industrialization of Fahrenheit receives a rating of “Excellent” for all three feasibility categories.

#### 4.4.3. Model Validation

Now that we have established the rankings of each characteristic for our three potential solutions our focus shifts to the verification of using the total value equation to choose a “best-fit” solution. To instill confidence in our model’s selection of the industrialization of Fahrenheit, we will show that this selection remains stable not only for the “true” parameters chosen in our original equation but over a range of adverse situations as well. The figure below displays the  $\Omega$  values for each solution from Table 8.

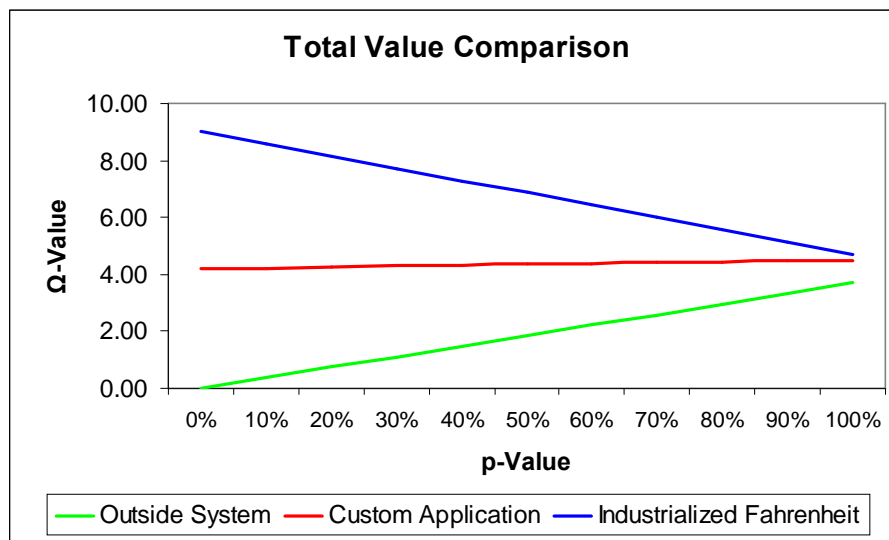


Figure 3: Total Value Comparison with Original Parameters

The convergence of  $\Omega$  at high p-values for our three solutions is caused by the wide gap between each solution’s logistical feasibility score and the comparatively shorter distance between each technical capability score. While the averages for each system are fixed by the ratings for the individual characteristics, the scores are a function of the weight assigned to each parameter and free to vary accordingly. This dependency creates

the possibility that the endpoints of each solution are capable of changing with shifts in the characteristic weights.

From this graph we can also see that the industrialization of Fahrenheit is the only alternative with a negative slope. To be precise, for every 10% increase in  $p$  its  $\Omega$  value drops 0.43, while the  $\Omega$  values for the outside system and a custom application increase 0.37 and 0.03 respectively. These slopes are again dependent on the weights of each characteristic and suggest that under certain scenarios the industrialization of Fahrenheit will no longer be the optimal solution.

To better understand omega's dependency on the characteristic weights we need to first revisit the original formula found below.

$$\Omega = (p \times TCA \times [TCS]) + ((1 - p) \times LFA \times [LFS])$$

It is easy to show that the technical capability and logistical feasibility scores are the only segments of the function dependent on the weights of the individual characteristics. While we fixed these values in our original equation, we now need to replace these numerical values with free variables. We define  $f$ ,  $s$ ,  $v$ , and  $e$  to be the importance assigned to flexibility, security, scalability, and ease of use respectively noting that they must sum to one. Similarly, we define  $c$ ,  $w$ , and  $i$  to equal the weights of cost, work required, and implementation ease respectively, again noting they must sum to one. Replacing the original values with these variables our total value equation becomes:

$$\Omega = (p \times TCA \times [(f \times F) + (s \times S) + (v \times V) + (e \times E)]) + ((1 - p) \times LFA \times [(c \times C) + (w \times W) + (i \times I)])$$

With the total value equation now defined in terms of each characteristic's weight, we can formulate the sensitivity of omega to each by taking its respective partial derivative. This method is similar to the way we evaluate the various risk sensitivities of a portfolio of derivatives (see Appendix F). The table below displays the formula for each sensitivity value.

<u>Sensitivity</u>	<u>Formula</u>
Flexibility	$\delta\Omega / \delta f = F \times p \times TCA$
Security	$\delta\Omega / \delta s = S \times p \times TCA$
Scalability	$\delta\Omega / \delta v = V \times p \times TCA$
Ease of Use	$\delta\Omega / \delta e = E \times p \times TCA$
Cost	$\delta\Omega / \delta c = C \times (1 - p) \times LFA$
Work	$\delta\Omega / \delta w = W \times (1 - p) \times LFA$
Implementation	$\delta\Omega / \delta i = I \times (1 - p) \times LFA$

**Table 9: Characteristic Sensitivity Formulas**

These equations represent the sensitivity of each system's total value with respect to changes in the underlying characteristic weights. We see that each partial derivative is a function of the characteristic's rating, the category average, and the p-value. While the rating and average are fixed, we have allowed p to vary and the sensitivity of each weight will depend on the p-value chosen.

Numerically, the sensitivity for a characteristic represents the change in omega for a change of one in its weight at a given p-value. Since our weight variables are percents, a change of "one" is actually a change of 100% so we divide each partial derivative by 10 for a more relevant statistic. This new value represents the change in omega for a change of 10% in the characteristic's weight and the figure below shows these sensitivities for all three systems across our feasible range of p-values.

Outside System Weight Sensitivities												
P-Value		0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Technical Capability	f	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	s	0.00	0.06	0.12	0.18	0.24	0.30	0.36	0.42	0.48	0.54	0.60
	v	0.00	0.06	0.12	0.18	0.24	0.30	0.36	0.42	0.48	0.54	0.60
	e	0.00	0.04	0.08	0.12	0.16	0.20	0.24	0.28	0.32	0.36	0.40
Feasibility	c	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	w	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	i	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Custom Application Weight Sensitivities												
P-Value		0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Technical Capability	f	0.00	0.02	0.05	0.07	0.09	0.11	0.14	0.16	0.18	0.20	0.23
	s	0.00	0.05	0.09	0.14	0.18	0.23	0.27	0.32	0.36	0.41	0.45
	v	0.00	0.07	0.14	0.20	0.27	0.34	0.41	0.47	0.54	0.61	0.68
	e	0.00	0.07	0.14	0.20	0.27	0.34	0.41	0.47	0.54	0.61	0.68
Feasibility	c	0.60	0.54	0.48	0.42	0.36	0.30	0.24	0.18	0.12	0.06	0.00
	w	0.20	0.18	0.16	0.14	0.12	0.10	0.08	0.06	0.04	0.02	0.00
	i	0.40	0.36	0.32	0.28	0.24	0.20	0.16	0.12	0.08	0.04	0.00

Industrialized Fahrenheit Weight Sensitivities												
P-Value		0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Technical Capability	f	0.00	0.05	0.09	0.14	0.18	0.23	0.27	0.32	0.36	0.41	0.45
	s	0.00	0.05	0.09	0.14	0.18	0.23	0.27	0.32	0.36	0.41	0.45
	v	0.00	0.05	0.09	0.14	0.18	0.23	0.27	0.32	0.36	0.41	0.45
	e	0.00	0.07	0.14	0.20	0.27	0.34	0.41	0.47	0.54	0.61	0.68
Feasibility	c	0.90	0.81	0.72	0.63	0.54	0.45	0.36	0.27	0.18	0.09	0.00
	w	0.90	0.81	0.72	0.63	0.54	0.45	0.36	0.27	0.18	0.09	0.00
	i	0.90	0.81	0.72	0.63	0.54	0.45	0.36	0.27	0.18	0.09	0.00

**Figure 4: Characteristic Weight Sensitivities**

This figure demonstrates that our systems are highly sensitive to changes in characteristic weights at extreme p-values. In general, technical capability sensitivities are higher at high p-values and feasibility sensitivities are higher at low p-values. For example, increasing the weight of the initial cost  $c$  by 10% will result in a 0.72 increase in  $\Omega$  for the industrialized Fahrenheit system at a p-value of 20%. Similarly, a 10% increase in the weight of scalability  $v$  will result in an increase of 0.61 in  $\Omega$  for the custom application at a p-value of 90%.



While this figure suggests we may be able to increase the  $\Omega$  values by changing characteristic weights, it does not well represent the tradeoffs necessary when shifting these percentages. Since the weights of each category must sum to one, adding 10% to one characteristic means we must also remove 10% from another and the corresponding decrease may offset the original increase in  $\Omega$ . For example, when we increase the weight of the initial cost  $c$  by 10% at a  $p$ -value of 20%, we must also decrease the weight of either the work required  $w$  or the implementation ease  $i$  by 10%. This decrease will cause a drop of 0.72 in the value of  $\Omega$  for an industrialized Fahrenheit, offsetting the gain from the increase in  $c$  and causing no net change in the final  $\Omega$  value.

The figure below demonstrates the potential gain or loss from each tradeoff at a  $p$ -value of 70%. The values represent the change in omega for a 10% decrease in the horizontal characteristic and a 10% increase in the vertical one.

Outside System Tradeoff Opportunities					
		f	s	v	e
Technical Capability	f	0.00	-0.42	-0.42	-0.28
	s	0.42	0.00	0.00	0.14
	v	0.42	0.00	0.00	0.14
	e	0.28	-0.14	-0.14	0.00
		c	w	i	
Feasibility	c	0.00	0.00	0.00	
	w	0.00	0.00	0.00	
	i	0.00	0.00	0.00	

Custom Application Tradeoff Opportunities					
		f	s	v	e
Technical Capability	f	0.00	-0.16	-0.32	-0.32
	s	0.16	0.00	-0.16	-0.16
	v	0.32	0.16	0.00	0.00
	e	0.32	0.16	0.00	0.00
		c	w	i	
Feasibility	c	0.00	0.12	0.06	
	w	-0.12	0.00	-0.06	
	i	-0.06	0.06	0.00	

Industrialized Fahrenheit Tradeoff Opportunities					
		f	s	v	e
Technical Capability	f	0.00	0.00	0.00	-0.16
	s	0.00	0.00	0.00	-0.16
	v	0.00	0.00	0.00	-0.16
	e	0.16	0.16	0.16	0.00
		c	w	i	
Feasibility	c	0.00	0.00	0.00	
	w	0.00	0.00	0.00	
	i	0.00	0.00	0.00	

Figure 5: System Tradeoff Opportunities at p=70%

The next step in our validation process is to use this information to construct two scenarios, an “adverse scenario” and a “worst-case” scenario, to test the stability of our original outcome. These scenarios will be created by shifting the characteristic weights and observing the corresponding change to the total value curves for each system.

It is important to note that since the total value equation is a linear function, the sensitivities and corresponding tradeoff opportunities will follow a linear pattern. This means we can apply the tradeoffs from Figure 5 to our three scenarios without concern for which p-values will be affected. Changes to a system at one p-value will have a similar directional impact at all p-values, just on a different scale.

Since we are interested only in scenarios for which the industrialization of Fahrenheit is no longer the optimal solution, we examine the sensitivities constructed earlier for opportunities to increase the total value of the other two alternatives. At the same time we look for ways to minimize the total value of the industrialization of Fahrenheit. For our first scenario we aim to achieve this without leaving a somewhat ambiguous “reasonable” boundary on our parameters. For example, assuming that scalability accounts for 100% of technical capability is plausible but likely not reasonable. For our second scenario, we do not limit ourselves to these boundaries

Figure 5 clearly demonstrates that removing weight from flexibility and adding it to scalability will positively benefit both the outside system and the custom application without having an effect on the industrialization of Fahrenheit. We also can see that shifting importance from ease-of-use to scalability will positively impact the outside system while negatively impacting the industrialization of Fahrenheit and that increasing

the importance of the initial cost at the expense of maintenance work required will benefit the custom application option. With these transitions in mind, the important resulting question is whether or not we can construct actual scenarios for which these shifts would be justified.

#### 4.4.3.1. Adverse Scenario

Assume BNP receives information that a highly scalable and reasonably secure system would be granted Tier 1 status. Next we assume that the developers are willing to sacrifice ease of use and some flexibility to achieve this and that a low cost Tier 1 system will allow for a large support team to run the valuation process and handle maintenance work. We reduce the weight of flexibility to 20% and completely remove the importance of ease of use, increasing the weight of scalability to 50%. We also reduce the weights of work required and implementation to 10% and increase the initial cost importance to 80%. The table below displays the resulting  $\Omega$  values for each system.

<b>p -Value</b>	<b>Outside System</b>	<b>Custom Application</b>	<b>Industrialized Fahrenheit</b>
<b>0 %</b>	0.00	5.40	9.00
<b>10 %</b>	0.48	5.38	8.55
<b>20 %</b>	0.96	5.36	8.10
<b>30 %</b>	1.44	5.33	7.65
<b>40 %</b>	1.92	5.31	7.20
<b>50 %</b>	2.40	5.29	6.75
<b>60 %</b>	2.88	5.27	6.30
<b>70 %</b>	3.36	5.24	5.85
<b>80 %</b>	3.84	5.22	5.40
<b>90 %</b>	4.32	5.20	4.95
<b>100 %</b>	4.80	5.18	4.50

**Table 10: “Adverse Scenario”  $\Omega$ -Value Comparison**

This table shows us that the industrialization of Fahrenheit remains the optimal solution through this scenario for most p-values. In fact, we can interpolate that it remains optimal for a p-value as high as 84.21%, at which point the creation of a custom

application would become the better alternative. The probability that the emphasis on technical capability will surpass 84% is highly unlikely, however. The original importance of Fahrenheit was to create a system that would not become bogged down in the complexities of highly technical systems, and we will assume that the stress placed on technical capability will remain below 80%. As such, the industrialization of Fahrenheit remains the optimal solution even under the adverse characteristic weights presented by this scenario.

#### 4.4.3.2. Worst-Case Scenario

Having demonstrated the industrialization of Fahrenheit remains optimal under even the most adverse, yet reasonable, characteristic weights, we seek to analyze the total value of each system under the most extreme conditions. We can use linear programming to show that shifting all technical capability weight to scalability and all feasibility importance to the initial cost will result in the highest possible  $\Omega$  values for the outside system and custom application options and the lowest possible value for the industrialization of Fahrenheit. The figure below shows the results of these shifts.

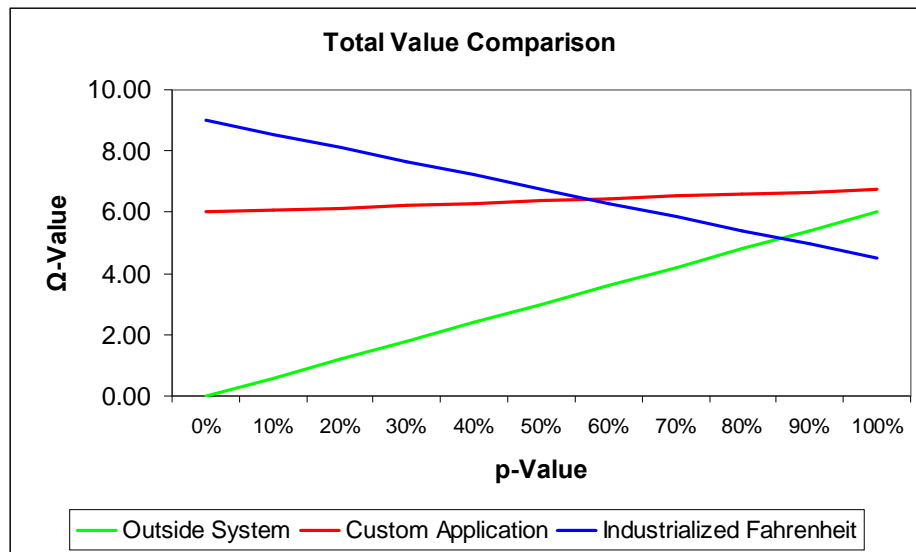


Figure 6: "Worst Case"  $\Omega$ -Value Comparison

This graph shows that an industrialized Fahrenheit program remains the optimal solution for p-values below 57.14%, at which point the creation of a custom application

become the optimal solution. In fact, for p-values higher than 85.71% the industrialization of Fahrenheit is actually the alternative with the lowest total value under these parameters. However, we must consider the conditions under which these results were achieved. Even under the most adverse scenario available, the industrialization of Fahrenheit remains the optimal solution when the stress placed on logistical feasibility is greater than 42.86%. In reality, feasibility is an integral component of a new valuation model because of Fahrenheit's original design as a flexible system and as such, we will confirm its industrialization as the optimal solution.

## **4.5. The Industrialization of Fahrenheit**

Industrializing the current process would allow for Excel to solely be used for calculations and eliminates unwanted risks. The following sections outline the components of our developed process.

### **4.5.1. Implementation of Microsoft Excel 2007**

Excel 2007 has many advantages over Excel 2003 but only one is actively used by Bean Stalk. The new XML functionality from Excel 2007 allows for deals not being stored in Excel files. Originally only the template spreadsheets were to be converted to Excel 2007 format (.xlsm) but the change in spreadsheet size between 2007 and 2003 made this impossible as Excel does not allow a spreadsheet to be imported into a workbook if it is larger than the spreadsheets contained in the workbook. This meant that the Fahrenheit spreadsheet also had to be resaved in the new Excel format.

### **4.5.2. Dynamic Input Process**

One of the greatest deficiencies of Fahrenheit was the storage of deals as individual Excel files. With Excel 2007 being used the XML mapping allows for the dynamic input of deal details into a template. The process of mapping a template is simple for a user with moderate experience. From the developers tab a user is able to select that they want to map an XML file and simply have to point at either an XML file or a schema file. Excel then parses the XML and presents the user with the attributes contained within and the user is able to drag and drop these attributes to the appropriate cell.

#### **4.5.3. Documentation of Inputs**

Providing a trail for auditing purposes was also required for deals processed by Bean Stalk. The solution was to save information about each deal to a Microsoft SQL server for each deal calculation. Right before the XML is inserted into the templates the XML is also saved to the database. Along with the XML the deal number, job name, and calculation date is saved to allow for an auditor to track down what exact values were fed in for the calculations. This provides a means for an auditor to recreate a deal calculation and compare the results with results that have been stored in MAD.

#### **4.5.4. Computational Distribution**

The distribution of Fahrenheit was never really an option unless it was controlled manually with several copies of Fahrenheit on multiple machines and jobs listed individually for each instance to process. With Bean Stalk once a spreadsheet has been created it can be saved to a given location. This location can be monitored by instances of processing nodes that when they detect a new spreadsheet can open the spreadsheet and process it. This approach is well suited for Bean Stalk because the spreadsheet contains all the information on the job and what to do with the results meaning that once a spreadsheet is saved to this location Bean Stalk does not need to monitor it. This could lead to problems if a spreadsheet calculation fails and Bean Stalk believes that job to have completed. To solve this issue the processing nodes can reply to Bean Stalk when they have completed a job or if it has failed. This layer of communication should be minimal and using C# web services the implementation should take minimal work.

#### **4.5.5. New Template Creation and Review**

Creation of new templates is something that is really starting to hit its stride so it was important to not affect this process to much, fortunately Bean Stalk only makes minimal changes. The primary change is the required use of Excel 2007 for XML mapping. This can be done after the template has been developed in Excel 2003, since the newer version is not readily available to most users. After the templates have been developed and tested they can be saved to a server where Excel 2007 is installed and then saved in the new proper format. All new templates will also require a macro and range to be added for XML. The field is used by Bean Stalk to hold the path to the XML file to be imported.

The macro reads this field and calls the import. The macro is simple and will be able to be directly copied from any Bean Stalk template.

## **4.6. Creation of a Prototype**

Our group focused on the creation of an example prototype for treasury lock calculations. This section includes a closer look into the Bean Stalk solution and how it circumvents the spreadsheet-dependant data storage process of Fahrenheit.

### **4.6.1. Program Overview**

The Bean Stalk system consists of an application and an Excel spreadsheet. Bean Stalk is a Windows Console Application written in C# using Visual Studio 2008 and contains the majority of the logic that runs the system. The responsibilities of Bean Stalk are to obtain the list of jobs and deals, the details of the deals, then process the jobs. The list of jobs and deals is obtained by connecting to the Configuration Manager database. To maintain flexibility, when running Bean Stalk the user is able to input arguments that will override job specifications. This is particularly useful for testing and valuation because the user can specify to save the risk objects to any location they want while not uploading any results to MAD. The command line arguments are parsed by leading the value with the name of the parameter it is for and a colon. For example passing “JobName:PV\_Delta” would execute only the PV\_Delta job. If the user is interested in the results of this job from 3 business days ago they could pass “JobName:PV\_Delta CalcDateOffset:-3”.

Bean Stalk is intended to be a prototype only and to provide groundwork to be expanded on in order to create a full functional system. The main limitation of Bean Stalk is that it currently only supports TLock trade types. TLocks are the most simplistic trades handled by Fahrenheit in terms of number of variables and therefore were chosen considering the limited time frame for the project. New types of trades can be added easily by creating a new template mapped to an XML deal and making some minor code changes. Bean Stalk needs new code for each new deal type primarily because of the MAD database schema. The table names follow a naming convention but are not predictable based on the information available to Bean Stalk.

Bean Stalk begins by creating objects to represent all of the jobs and deals. Creating jobs is fairly straightforward and uses a constructor. Deals however are more complicated to create so this responsibility has been delegated to a builder object which takes care of the creation process including querying databases, reading from multidimensional arrays, and massaging data to ensure it will be in the proper format to be used by Excel in calculations. The deal builder is also responsible for calling MadMolly and passing it the component number of the deal so that the deal details XML document can be created.

Once Bean Stalk has modeled the jobs and deals it begins to iterate through the list of jobs. The first step for each job is to construct a list of deals in that job. This is accomplished by matching certain criteria (currency, product, portfolio, and group) of the deal with the same criteria of the job. At this point Bean Stalk has the current job and list of deals associated with the job, it now imports this data into Excel. The job information is copied into the appropriate ranges on the Excel spreadsheet. Loading of deals is a more complicated process because it means adding a new worksheet for each deal.

Importing a deal into the spreadsheet begins by opening the Excel template required by this deal, the path to the template is a field of the deal object. The templates are special prepared Excel spreadsheets that have been mapped with an XML representation of the deal details. The mapping allows for the template to be opened then by running a macro an XML file is imported and the XML attributes are inserted into their corresponding mapped fields. The XML details for the deal are saved to a temporary file and then the macro is run to import this file. Once the template has been filled in it is copied to the Bean Stalk workbook. Bean Stalk limits the number of deals that can be added to a workbook to ensure that Excel does not crash because of lack of resources. If a job contains too many deals the job is divided up dynamically. With all of the deals loaded into the Bean Stalk spreadsheet the sheet is calculated. In the prototype stage this occurs on the same machine as Bean Stalk but this spreadsheet can be distributed in the future to allow for parallel processing. Following the calculation Excel is closed and Bean Stalk continues on to the next job in the queue.

#### **4.6.2. Creation of MadMolly**

In order for Bean Stalk to valueate the deals it must obtain the details for the deal that are used in the calculations. This requires a system that can connect to the MAD database



where the deals Bean Stalk processes are stored. Fahrenheit Sheet Builder was responsible for this in the previous system and utilized MadMax which is an Excel only means of connecting to MAD. Since Bean Stalk uses Excel for calculations only a new program was required to be written to access MAD. The program developed is called MadMolly, like Bean Stalk it is written in C#. MadMolly is responsible for connecting to MAD, querying the database for deal details, and creating and returning an XML document representing a deal and its details. The MAD database is large and consists of approximately a thousand tables. Each deal requires information from at least four tables so inner joins were used to limit the number of queries required.

#### **4.6.3. Bean Stalk Spreadsheet**

Bean Stalk uses a modified version of the Fahrenheit spreadsheet to handle the calculations. This spreadsheet has been stripped of the user interface previously present and the majority of its logic removed. Instead this spreadsheet is populated with the correct information by Bean Stalk at runtime. The spreadsheet still does contain the logic for distributing the results because placing this responsibility on Bean Stalk would require it to receive the results back from the spreadsheet then distribute them. In the future when these spreadsheets may be distributed for processing it will be more practical for the spreadsheets to handle the output.

#### **4.6.4. Uploading PVs with MadMax**

MadMax, as mentioned before, is an Excel based solution for interacting with the MAD database. Unlike MadMolly which is read only MadMax can write to the database. This functionality is used by the Bean Stalk spreadsheet to commit the calculated values back to MAD so that they can be used by interested parties.

#### **4.6.5. Generation of Risk Reports**

The Bean Stalk spreadsheet provides the desired values by creating risk reports which are a structure supplied by the Westminster library. The risk reports are passed values and create an object which is then queried to display the relevant data.

## **5. Conclusions**

Our project improved BNP's spreadsheet deal process by lowering risk and creating a system that can handle a greater number of trades. During the seven week period, various solutions were analyzed and evaluated on the components that met our goals. A prototype was created as an example of industrializing Fahrenheit. This solution provides BNP with the means to capture exotic trades in a secure and flexible way.

### **5.1. Improvements**

Bean Stalk being a prototype system has many improvements that can be made to it before becoming a production system. The first such improvement is the distribution of the calculations across multiple machines. This would allow for parallel processing which would cut down on calculation time which will be necessary if the trade volume increases. The architecture for this distribution should keep a single instance of Bean Stalk that is responsible for creating jobs, assigning them to a processor, and monitoring job status. The system begins with Bean Stalk loading deals into the spreadsheet as usual but then sending this spreadsheet to an external processing node. This can be accomplished by either saving to a shared drive and messaging the node or by sending the spreadsheet in a SOAP message. The processing nodes will then open the spreadsheet and calculate it and send a message back to Bean Stalk indicating if the spreadsheet calculated without error. Along with creating web service processing nodes, Bean Stalk could be converted to a web service to allow a portal for job creation, editing, deletion, and scheduling.

Bean Stalk is designed to be highly flexible without requiring code changes but in its prototype form most configuration options are hard coded in. Adding a configuration file and a configuration reader would allow for changes to be made to the systems behavior without requiring a recompile. An example where this would be useful is the number of deals it takes to require a split for a job; this could even be adjusted to have different values for different job types.

Another area of Bean Stalk that can have improvements made to increase flexibility is MadMolly. As it stands MadMolly can only query the tables that contain treasury lock

deals. This is due to the naming schema used in the MAD database not being able to be accurately predicted by the program. The solution to this would again be a configuration file which would contain the correct tables to use based on the type number which can be retrieved by querying a table that contains all deals with information on the deal that MadMolly has access to.

## **5.2. Impact**

With a non-prototype Bean Stalk in place of Fahrenheit BNP will be able to handle a greater volume of exotic trades efficiently and securely. The magnitude of the impact however will depend on other factors outside of the ability of the system. Primarily the volume of trades will be limited by the tier system in place which even though Bean Stalk may meet the defined requirements for an unlimited tier will not be treated as such due to lack of confidence in the underlying models. Since Bean Stalk will most likely not make trades Tier 1 for non-system reasons the cap on Tier 2 and 3 trades may be increased because of the improvements made with Bean Stalk. This will allow for traders to branch out into new exciting financial products that they previously would have not been able to trade with an acceptable level of risk.

Adding these new types of deals will also require the minimal IT intervention. With minimal training FRG will be able to not only produce the templates but also map them for XML. Once the templates are mapped and stored in an area where Bean Stalk can access them all that IT is responsible for is to modify MadMolly to be able to read the deal details from the corresponding tables for the deal type. All of this means that new products can be made available quickly and in a greater volume than previously and by doing so help BNP's traders earn more money.

## **6. Recommendations**

The goal of our project was to develop a solution for the increased security of a greater number of BNP deals. In the duration of seven weeks, various solutions were evaluated based on required criteria found for a new system. Bean Stalk was created to fulfill those criteria. Our project team was able to complete a working prototype to demonstrate the results of our findings. With continued research, our solution can evolve with the needs of BNP.

In the following sections we will discuss the resulting prototype and potential directions to continue further research and implementation. All of these points can be tackled by either BNP research teams or future MQP projects.

### **6.1. Increased Number of Calculations**

Given the project time constraints, only an example system was created with Bean Stalk and MadMolly that demonstrates the calculations for treasury locks. BNP uses a multitude of other calculations to track their trades including risk and pricing models. Changes can be made to MadMolly, expanding the program to include all of the needed calculations. This could also be accomplished by replacing MadMolly with a new all-inclusive system. Further research can be made to explore the use of Bean Stalk with the other required BNP calculation templates.

### **6.2. Scalability of Bean Stalk**

The research into the expansion of calculations can also lead to the increase in the scalability of calculations. Our instance of Bean Stalk was able to technically fulfill the project objective of increased number of allowable trades with the use of multiple CPUs. A mechanism can be developed for the distribution of additional computations on top of the improvements made by Bean Stalk which will only add to the number of possible trades BNP can value.

### **6.3. Discussion**

Throughout the process of completing our project, various discussions were held about the future of our research. There were a few areas not fully explored due to the relevance to the project and also time restraints. These areas can open routes into additional MQP projects in the future or internal BNP discussions.

#### **6.3.1. Lack of Commercial Systems**

This project explored the possible use of a commercial system to replace BNP's current trade capture system. This potential solution could not meet our specific project objectives and was discarded. A replacement system simply does not exist to fulfill the trade capture needs of any financial firm. With the perceived distrust of Excel-based systems, one would assume that software vendors would make greater efforts to develop a system to meet the needs of large financial firms. However, current software suites only focus on security issues investment banks face and the prevention of fraud. If a program like this were to exist, the implications on the software and financial industries would be immense. It would be worth interviewing software vendors and suggesting they join the MQP program at WPI in order to explore this topic.

#### **6.3.2. Re-evaluation of Tier Requirements**

The re-evaluation of Tiers topic leaves a lot of room for process discussion. Through a well-documented, consistent process, ensuring security can be achieved without isolated rules. With the application of Bean Stalk to BNP system, R-CM should look to review the requirements for the Tiered categories. Discussion between R-CM, FRG and IT can result in a clear understanding of the needs of BNP as well as dissolve confusion between these departments. The improvements of Bean Stalk satisfy Tier 1 requirements according to IT, where FRG was of the opinion that Fahrenheit alone satisfied R-CM's security checks. A discussion should help to clear up these issues and result in a system that satisfies all parties. With the ideal system, tier limits would not be a function of technical limitations and internal risk management but position risk management. The whole Tiered system could potentially be eliminated, freeing up BNP resources for improved business development.

## References

1. Associated Press (AP). (2009). *BNP Paribas Q2 Net Profit Up 6.6%*. Retrieved from <http://www.livemint.com/2009/08/BNP-Paribas-Q2-net-profit-up-6.html>
2. Baker, A. (2009, October 27). *Monarch 10.5 embeds source data in excel spreadsheets*. Retrieved from [http://www.information-management.com/news/grc\\_compliance\\_spreadsheet\\_excel\\_monarch-10016407-1.html](http://www.information-management.com/news/grc_compliance_spreadsheet_excel_monarch-10016407-1.html)
3. Black, F. (1976). The Pricing of Commodity Contracts. *Journal of Financial Economics*, 3, 167-179.
4. Black, F. & Scholes, M. (1973). The Pricing of Options and Corporate Liabilities. *Journal of Political Economy*, 81(3), 167-179.
5. Bank of Montreal (BMO). (1997). *Locking in Treasury Rates with Treasury Locks*. Retrieved from <http://www.interestrateswaps.info>
6. Bank of Montreal (BMO). (2009). *Treasury Locks, Caps, and Collars*. Retrieved from <http://www.bmocm.com/products/marketrisk/intrderiv/treasury/default.aspx>
7. The Bank of New York Mellon (BNY). (2009). *Treasury Lock*. Retrieved from <https://gm.bankofny.com/Derivatives/ProductNotes/InterestRate>
8. BNP Paribas (BNP). (2009a). *Fahrenheit NY*. Provided by BNP Financial Control.
9. BNP. (2009b). *Spreadsheet Deals Control Setup*. Provided by BNP Financial Control.
10. BNP. (2009c). *Westminster Library Function Help and Support*. Retrieved from BNP Intranet.
11. Copp, C. (2009). *Euromarket day finder*. Retrieved from [http://www.bobsguide.com/guide/prod/5-8961\\_Euromarket\\_Day\\_Finder%c2%ae.html](http://www.bobsguide.com/guide/prod/5-8961_Euromarket_Day_Finder%c2%ae.html)
12. Crosman, P. (2009, October 28). *Datawatch upgrade to ease fraud investigations, expert says*. *Wall Street & Technology*, Retrieved from [http://www.wallstreetandtech.com/regulatory-compliance/showArticle.jhtml;jsessionid=4011TUHGK5LJ5QE1GHOSKH4ATMY32JVN?articleID=220900743&\\_requestid=242732](http://www.wallstreetandtech.com/regulatory-compliance/showArticle.jhtml;jsessionid=4011TUHGK5LJ5QE1GHOSKH4ATMY32JVN?articleID=220900743&_requestid=242732)

13. Crosman, P. (2009, October 29). Finding hidden dangers in spreadsheets. *Wall Street & Technology*, Retrieved from [http://www.wallstreetandtech.com/regulatory-compliance/showArticle.jhtml;jsessionid=4011TUHGK5LJ5QE1GHOSKH4ATMY32JVN?articleID=221100139&\\_requestid=242697](http://www.wallstreetandtech.com/regulatory-compliance/showArticle.jhtml;jsessionid=4011TUHGK5LJ5QE1GHOSKH4ATMY32JVN?articleID=221100139&_requestid=242697)
14. CUSIP Global Services. (2009). CUSIP-Based Identifiers. Retrieved from [https://www.cusip.com/static/html/webpage/cusip\\_based\\_iden.html](https://www.cusip.com/static/html/webpage/cusip_based_iden.html)
15. Eurobase. (2009). *Eurobase International Group: Siena eTrader*. Retrieved from [http://www.bobsguide.com/guide/prod/5-4443\\_Siena\\_eTrader.html](http://www.bobsguide.com/guide/prod/5-4443_Siena_eTrader.html)
16. Harvey, C. (2004). *Short-term interest rates*. Retrieved from <http://financial-dictionary.thefreedictionary.com/Short-term+interest+rates>
17. Henderson, D. (2008). *Present Value*. Retrieved from <http://www.econlib.org/library/Enc/PresentValue.html>
18. Hull, J. (2009). *Options, Futures, and Other Derivatives*. Upper Saddle River, NJ: Prentice Hall, Inc.
19. Investopedia ULC. (2009a). *Fixed Income*. Retrieved from <http://www.investopedia.com/terms/f/fixedincome.asp?&viewed=1>
20. Investopedia ULC. (2009b). *Mortgage-Backed Security (MBS)*. Retrieved from <http://www.investopedia.com/terms/m/mbs.asp>
21. Investopedia ULC. (2009c). *Treasury Lock*. Retrieved from <http://www.investopedia.com/terms/t/treasurylock.asp>
22. Knaster, B. (2004, November 29). Fixed Assets: An Epic Story. *Accounting Today*, p60A.
23. Lee, C. & Lee, A. (2006). *Encyclopedia of Finance*. New York, NY: Springer Publishing Co.
24. Levine, C. (2008, March 14). Excel Still Fastest and Most Flexible Trade Capture System. *Wall Street & Technology*, Retrieved from [http://www.wallstreetandtech.com/operations/showArticle.jhtml;jsessionid=4ZXZTHVWB4VZ3QE1GHOSKH4ATMY32JVN?articleID=206903766&\\_requestid=231787](http://www.wallstreetandtech.com/operations/showArticle.jhtml;jsessionid=4ZXZTHVWB4VZ3QE1GHOSKH4ATMY32JVN?articleID=206903766&_requestid=231787)
25. Li, T. (2004, February 6). *What is Application program interface?* Retrieved from [http://searchexchange.techtarget.com/sDefinition/0,,sid43\\_gci213778,00.html](http://searchexchange.techtarget.com/sDefinition/0,,sid43_gci213778,00.html)

26. Microsoft. (2008). *Microsoft Announces Availability of Bloomberg on Windows Mobile Phones*. Retrieved from <http://www.microsoft.com/presspass/press/2008/jun08/06-10bloombergmobilepr.msp>
27. Miller, R. (2005, June 30). Only a Matter of Time before the Spreadsheets Hit the Fan: Inaccurate Financial Information is Wide-Spread and is Putting Businesses at Risk. *The Daily Telegraph*, p003.
28. NVIDIA. (2009). *What is CUDA?* Retrieved from [http://www.nvidia.com/object/cuda\\_what\\_is.html](http://www.nvidia.com/object/cuda_what_is.html)
29. Peresys Ltd. (2009). *Peresys Financial Software Solutions*. Retrieved from <http://www.peresys.com/content/656>
30. Preve, J. (2008). *Business Intelligence for Financial Risk Management*. Retrieved from Worcester Polytechnic Institute Major Qualifying Project Reports. (AG WS09)
31. Prospero. (2009). *Banking and Wealth Management System*. Retrieved from [http://www.bobsguide.com/guide/prod/5-3484\\_Prosporo\\_%2d\\_the\\_Banking\\_and\\_Wealth\\_Management\\_System.html](http://www.bobsguide.com/guide/prod/5-3484_Prosporo_%2d_the_Banking_and_Wealth_Management_System.html)
32. QuantRisk. (2008). Trade Deal Capture. Retrieved from <http://www.quantrisk.com/trade-capture.html>
33. Rodier, M. (2008, January 22). Enhanced Shared Assessments Program can Make Audits Speedier. *Wall Street & Technology*, Retrieved from [http://www.wallstreetandtech.com/regulatory-compliance/showArticle.jhtml;jsessionid=4011TUHGK5LJ5QE1GHOSKH4ATMY32JVN?articleID=205907836&\\_requestid=242843](http://www.wallstreetandtech.com/regulatory-compliance/showArticle.jhtml;jsessionid=4011TUHGK5LJ5QE1GHOSKH4ATMY32JVN?articleID=205907836&_requestid=242843)
34. Sage. (2009). *Wealth Management*. Retrieved from [http://www.sage.ch/index.php?id=wealth\\_management](http://www.sage.ch/index.php?id=wealth_management)
35. Saliba, A.J. (2005). *The Options Workbook: Fundamental Spread Concepts and Strategies for Investors and Traders*. New York, NY: Kaplan Publishing, Inc.
36. Schmerken, I. (2008, September 1). Luring Active Traders. *Wall Street & Technology*, p17.



37. Spectrum. (2009). *Spectrum Treasury*. Retrieved from [http://www.bobsguide.com/guide/prod/5-1339\\_Spectrum\\_Treasury.html](http://www.bobsguide.com/guide/prod/5-1339_Spectrum_Treasury.html)
38. Stevenson, R. (2009). Fortis Asset Sale to BNP Cleared with Dutch OK. *Reuters UK*. Retrieved from <http://uk.reuters.com/article/idUKLT42693120090429>
39. Taleb, N. N. (2007). *The Black Swan: The Impact of the Highly Improbable*. New York, NY: Random House, Inc.
40. Tatum, M. (2009). *What is a Total return swap?* Retrieved from <http://www.wisegeek.com/what-is-a-total-return-swap.htm>

## **Appendix A: Sponsor Description**

The Banque Nationale de Paris (BNP) was formed in 1966 through the merger of two nationalized French retail banks, BNCI and CNEP. Although the French government remains its majority shareholder at 17%, BNP was wholly owned by the French government until its privatization in 1993 (Daneshkhu, 2009). In 2000, BNP successfully bid for the popular retail bank Paribas, formerly known as the Finance Corporation of Paris and the Netherlands, and merged to become BNP Paribas. According to its corporate website, BNP Paribas is “determined to be a benchmark banking group” with its focus on customer satisfaction and the “active pursuit” of improvement in its profitability and earnings capacity. Its four core values: responsiveness, creativity, ambition, and commitment, “encapsulate the essence of [BNP Paribas’] actions” as it strives to become “the bank for a changing world” (BNP, 2009a).

In 1970, BNP entered the United States banking market as the sole stakeholder of the California-based Bank of the West and its investment banking subsidiary BancWest. BNP Paribas (commonly referred to simply as BNP) is currently the largest bank in the Euro zone by total assets and the second largest by market capitalization, employing over 160,000 employees in 83 countries. In 2008 it was ranked by Forbes magazine as the 13<sup>th</sup> largest company in the world, the largest in France (Forbes, 2008), and in 2009 BNP purchased a 75% stake in the Belgian-headquartered Fortis Bank, also making it the largest bank in the Euro zone by deposits held (Stevenson, 2009).

Despite reported losses as high as 350 million Euros from the infamous 2008 Madoff fraud, Standard and Poor’s recently ranked BNP as one of the six strongest banks in the world with a AA corporate rating, a sentiment backed by rival ratings agencies Moody’s and Fitch (BNP, 2009b). BNP’s strong corporate rating is largely due to its avoidance of the recent credit crisis, which it escaped relatively unscathed. BNP Paribas posted a net profit of 3.1 billion Euros in the first half of 2009, boosted by a strong performance from fixed income trading in its Corporate and Investment Banking division (AP, 2009).

## **Appendix B: Acronym Index**

**API** – Application Programming Interface. API is the method used by a computer operating system or by an application program by which a programmer writing an application program can make requests of the operating system or another application.

**BO** – Back Office. The back office of a financial services firm provides administrative and support services.

**CDS** – Credit Default Swap. The buyer of the CDS makes a series of payments to the seller and receives credit protection, while the seller guarantees the credit worthiness of the product. This transfers the risk from the buyer to the seller of the swap.

**CIB** – Corporate and Investment Banking activities of the BNP Paribas Group

**CLN** – Credit Linked Note. A CLN is a security that allows the issuer to transfer a certain credit risk to the buyer.

**ET** – Exceptional Transaction or Trade. A trade labeled ET requires more monitoring throughout the trade lifecycle. The Trade Acceptance Committee determines if a trade is ET.

**FO** – Front Office. The front office of a financial services company typically includes sales personnel and corporate finance.

**FOCUS** – Focus Analytics. A BNP Paribas library of pricing and risk management functions for credit derivatives.

**FRG** – Flow Rates Group. The FRG creates the templates which are used to calculate the pricing and risk for new deals.

**IRG** – Interests Rate Group. The IRG is part of the Fixed Income area of BNP. They deal with research, sales, trading, origination and distribution of the interest rate product line.

**MAD** – Multi-Asset Derivatives. A BNP Paribas system for the pricing, valuation, trade-entry, and risk management of complex interest-rate, FX and credit derivatives.

**MadMax** – MadMax uploads PVs from Fahrenheit into MAD.

**MO** – Middle Office. The middle office in a financial services company generally manages risk and calculates profits and losses.

**PFC** – Product Finance Control. BNP has a PFC Committee that regulates the number of trades allowed.

**Plato** – Plato is the official P&L system used at BNP Paribas.

**PnL or P&L** – Profit and Loss. Profit and loss is often represented in a financial document summarizing the revenues, costs and expenses incurred in a period. Also known as the Income Statement.

**PV** – Present Value. The present value is determined by discounting the future value at the interest rate that the money could earn over the period.

**R-CM** – Risk-Capital Market. R-CM evaluates new trades and categorizes them into the Tier 1, Tier 2 or Tier 3 systems according to company policy. They then validate the calculations of Tier 2 and 3 trades.

**RVG** – Risk Vector Generator. RVG is a server-based risk engine that you can use to perform risk scenarios in real-time as well as scheduling overnight runs.

**SOHO** – Soho consists of a suite of applications that assist front and middle office FX options staff.

**STAR** – A trading system for swaps, government bonds, foreign exchange, futures and options.

**StarPnL** – A system that provides P&L and risk reports for STIR.

**STIR** – Short Term Interest Rates. STIR are interest rates on loan contracts or debt instruments that have maturities of less than one year. Also known as Money Market Rates.

**TAC** – Trade Acceptance Committee. The TAC's approval is needed before any new spreadsheet is prepared for a trade.

**TRS** – Total Return Swap. With a TRS one party makes payments based on a set rate, while the other party makes payments based on the return of an underlying asset. The

underlying asset is referred to as the reference asset. The party receiving the total return benefits from exposure from a reference asset without actually having to own it.

**WASP** – An Excel-spreadsheet-based system for pricing swaps and related instruments.

## **Appendix C: Tier Categorization**

All deals traded by BNP are grouped into one of three tiers which are classified by the characteristics of the deal booking or capture system used to represent the full or partial valuation of the trade. These characteristics are described below in a memo generated by BNP's Financial Control department in February of 2009 (BFI, 2009). While these characteristics provide a good outline for the qualities a trade capture system should adhere to, R-CM has the final say in the tier classification of any trade capture system and may choose to consider or accommodate factors outside the scope of this memo.

### **Tier 1**

- Deal is fully represented within a secure system at inception.
- Audit trail logs exist.
- Established IT support and development exists within system framework.
- The mechanisms of deal and market data population, PV and risk generation as well as the warehousing of such information are well controlled by IT.

### **Tier 2**

- Deal is either fully or partially represented within a secure system with valuation being performed on a spreadsheet, external to that secure system.
- FO cannot physically modify the live official version of spreadsheet (one that produces the official PV and risk).
- Existence of audit trail is dependent on the extent of deal representation within the secure system.
- The environment in which the spreadsheet operates is developed and supported by IT.
- The mechanisms of deal and market data population, PV and risk generation as well as the uploading of such information to secure/back office systems is well controlled by IT/MO/RCM PAT.

### **Tier 3**

- Deal is either fully or partially represented within a secure system with valuation being performed on a spreadsheet, external to that secure system.
- Little/no audit trail logging exists.
- The environment in which the spreadsheet operates is not well developed and/or supported by IT.
- The mechanisms of deal and market data population, PV and risk generation as well as the uploading of such information to secure/back office systems is not well controlled by IT and/or MO.
- Data can be modified by FO.
- Any deal outside the scope of Tier 1 or Tier 2.

## **Appendix D: Interviews**

### **Jeremy Ascher**

Head of New York IRG Portfolio Analysis Team, Risk – Capital Markets

Location: 11/3/2009 at BNP Paribas, New York, New York

Interviewers: Brendan Dean, Aubrey Scarborough, Tim Sherrerd

Responses are paraphrased and have *not* been reviewed by Jeremy Ascher

#### **What do you look for when classifying a trade system as Tier 1?**

If a trade is Tier 1 it should be entirely contained and valued within a bank system like Star or MAD.

#### **Which aspects of the current spreadsheet system are you most uncomfortable with?**

Excel is well known to most employees, including traders, and the potential for incorrectly granted write access increases the risk of human error and the opportunity for fraudulent activity. Second, the legitimacy of the calculations in the spreadsheets has not undergone the scrutiny that the Star and MAD systems have, and the risk of operational error is elevated. R-CM then must validate the templates and check the PVs monthly, which is an unnecessary strain on resources. Additionally, small changes to the calculation of a deal or the addition of a new piece must be manually adjusted for, again raising concerns of human and operational error and requiring the creation and review of new templates.

#### **Would a non-Excel valuation system outside of Star or MAD be useful?**

A C++, VB, or similar program, either self-contained or running Excel in the background, would definitely reduce the risk of human error considering traders and other users would be much less likely to be capable of interfering with or manipulating the calculations.

#### **Would you have any reservations about this type of system?**



One of the largest problems with any system that recalculates PVs and risk reports outside of the current Star and MAD systems is that it will be unable to run the various testing scenarios the bank runs. For example, BNP occasionally runs stress tests and the impact of any trades whose values are calculated outside of Star or MAD will not show through in the testing, hence we are “stuck” using the greeks and PVs generated by Fahrenheit.

**Do you have any suggestions or final thoughts?**

The ideal solution for R-CM is to have new deals fully integrated into one of the bank's official pricing systems. If the products are in an official bank system then there is confidence that they had received the necessary attention from research, hence less effort needed by R-CM for validation.

## **Dan Ward**

Head of Flow Research Group – Americas

Location: 11/12/2009 at BNP Paribas, New York, New York

Interviewers: Brendan Dean, Aubrey Scarborough, Tim Sherrerd

Responses are paraphrased and have *not* been reviewed by Dan Ward

### **What is your role within the Fahrenheit process?**

The Flow Research Group (FRG) is largely an end user of the spreadsheet process. When Fahrenheit needs to be equipped to handle a new deal type the FRG creates a template for the PV and risk calculations as well as creates the pricing tools that are then used by the traders. This template is tested within the Fahrenheit framework and then validated by R-CM. Fahrenheit is a staging ground used for creating more industrial code.

### **How labor-intensive is the creation of a new template?**

Originally, creating a new template was an incredibly painful process. The templates were produced from scratch with many components. With the increased use of Fahrenheit and more practice with building the spreadsheets, the creation of new templates is considerably smoother and more comfortable than it was in the past.

### **What is your interaction with the R-CM group?**

The templates created by FRG must be evaluated and reviewed by Risk Management. This includes the initial review of a new template and the monthly review of the generated PVs. Any issues with the calculations or outputs are relayed back to FRG where they are addressed and hopefully resolved.

### **What are your opinions or suggestions about the current Fahrenheit system?**

While writing new templates in an Excel based system may not be an ideal solution, it does provide a quick solution to new deal types and increases flexibility since the calculations are far more transparent. Unfortunately, when changes are needed the

process can become bogged down as information is relayed between groups. These types of delays eventually need to be addressed if Fahrenheit is going to establish itself as a more permanent solution. A function that compiles spreadsheets could allow for continual debugging and increased calculation speed, but will require a substantial amount of work before it can be implemented.

Despite its characterization as Tier 2 the Fahrenheit system is probably more secure than many people will credit it. The deals it values are still stored within MAD and there is definitely a well established IT support group. Obviously the storage of Excel data in directories carries with it the potential for malicious or fraudulent activity if the operating program is not controlled tightly enough though.

## **Nikholas Lopez**

IRFX Risk Support, Front Office – Americas

Location: 11/11/2009 at BNP Paribas, New York, New York

Interviewers: Aubrey Scarborough

Responses are paraphrased and have *not* been reviewed by Nikholas Lopez

### **What is your role within the trade capture system?**

I calculate the risk and pricing that the traders needs with the spreadsheet systems.

### **Do you have any suggestions for the current spreadsheet system?**

I am not one of the main coders, so I don't know how I would fix something like STAR. However, an Excel spreadsheet will never compete with a faster C++ application in processing 25,000 trades. The traders and risk managers always want a spreadsheet for the flexibility and ease of their calculations and also for their pricing and risk modeling. So right now, that's life, we can't get away from spreadsheets, even if others systems maybe easier for the traders to learn and use.

## Appendix E: Treasury Lock Valuation

As discussed earlier, a Treasury Lock is a form of interest rate swap whose valuation is primarily dependent upon the sensitivity of the underlying Treasury bill. The formula for the present value of a TLock can be found below (BNP, 2009c).

$$PV = \text{Notional} \times (\text{CTY} - \text{SY}) \times \text{Duration} \times \text{DF}$$

The strike yield SY is defined by the buyer and seller at the time of the trade and the current treasury yield CTY is defined by current market data. The notional amount almost always represents the total face value of the underlying debt to be issued by the buyer of the TLock, although this may not necessarily be true in purely speculative trades. The notional amount is always positive for the buyer of a TLock and negative for the seller. For clarity we will refer to the notional amount for the buyer as  $NA_B$  and for the seller as  $NA_S$ , identifying that  $NA_B = -NA_S$  where  $NA_S \leq 0 \leq NA_B$ .

Bond sensitivity is a measurement of the sensitivity of an asset's price to interest rate movements. For a Treasury Lock, this sensitivity represents the change in the underlying Treasury's price for small changes in interest rates and mathematically is represented by its modified duration, often interchangeably referred to simply as its "duration". To calculate this duration we must first calculate the standard Macaulay duration of the bond, the formula for which is given below.

$$\text{Macaulay Duration} = \frac{\sum (CF_t \times v^t \times t)}{\sum (CF_t \times v^t)}$$

Derived by Frederick Macaulay in 1938, the Macaulay duration is calculated by summing the results of multiplying the present value of each cash flow ( $CF_t \times v^t$ ) by the time  $t$  at which it is received, then dividing by the present value of all cash flows (the price of the bond). The Macaulay duration is a measurement of the length of time in years it takes for the price of a bond to be repaid by its internal cash flows and will not change by changing the face value of the bond, assuming its redemption value is equivalent.

Dealing strictly with positive cash flows, the duration must be greater than or equal to 0 and less than or equal to the time in years until the maturity of the bond.

While the Macaulay duration is a useful statistic, it does not provide a good measure of a bond's sensitivity to changes in interest rates. To account for this sensitivity we use a transformation of the Macaulay duration known as the modified duration. The modified duration is equal to the ratio of the percentage decrease in a bond's price to the percentage increase in the yield of the bond. Illustratively, the modified duration is equal to the percentage decrease in the price of a bond from a corresponding 1% increase in interest rates. The equation for finding the modified duration is given below.

$$\text{Modified Duration} = \text{Macaulay Duration} / (1 + (\text{YTM}/n))$$

The modified duration is equal to the Macaulay duration divided by the effective interest rate for the period between each coupon payment. This interest rate is equal to one plus the nominal yield to maturity YTM of the bond divided by the number of coupon payments per year n. It should be noted that because the modified duration is only a first order derivative it does not account for the convexity of yield curves and is only accurate for small changes in interest rates.

With these parameters defined it is easier to conceptually understand the value of a Treasury Lock at its expiration date. We have defined the duration as the ratio of the percentage decrease in a bond's price to the percentage increase in the yield of the bond. If we multiply this by the difference between the strike yield and the prevailing Treasury yield we obtain the percent change in the value of the Treasury for the change in rates over the TLock's lifetime. Multiplying this by the notional amount of the bond, the resulting value is equivalent to the dollar effect the change in interest rates had on the price of the underlying Treasury.

The buyer generally thinks of this value as the change in the cost of issuing debt. If Treasury yields have increased over the period, the value of the TLock will be positive and the amount paid to the buyer will exactly offset the corresponding increase in the price of issuing the debt. Similarly, if yields have fallen the value will be negative and the amount paid by the buyer will offset the decrease in the price of issuing the debt.

Effectively, the buyer has locked in the rate of purchasing Treasuries totaling the notional amount at the original strike yield.

Multiplying by a discount factor DF a TLock can be valued at any point in time prior to its expiration. The discount factor simply uses the current market yield curve to discount the current value of the TLock at its expiration date to today.

As an example, the ABT Company plans to borrow \$1,000,000 on June 30, 2010, for three years. The interest rate ABT will be required to pay on this debt will be equal to the three year Treasury yield at the time of the loan plus a spread dependent on ABT's credit rating. Worried about changes in Treasury yields over the next year but not its own credit rating, ABT desires to lock in the rate of the Treasury component at the current market yield of 2.95%. The ABT Company approaches BNP Paribas to purchase a Treasury Lock on January 1, 2010. Due to the cost of transactions to hedge the TLock, BNP agrees to sell a TLock with a notional amount of \$1,000,000 at a strike yield of 3.00% and an expiration date of June 30, 2010.

On March 31, ABT and BNP close their first quarters and must each report the present value of the TLock. The closing yield of a three year Treasury on March 31 has risen to 3.11% and using the equation provided above, ABT calculates the present value of the Treasury Lock to be \$3,103.47. Had the TLock expired on March 31, BNP would owe ABT this amount, which ABT would use of offset the increased price of borrowing the loan, keeping the effective interest rate as a spread over the Treasury yield of 3%.

However, over the next three months Treasury yields drop and the closing yield of a three year Treasury on the TLocks expiration date of June 30 is 2.87%. Recalculating the present value at (\$3,684.93), ABT finds it must pay BNP this amount. ABT's interest rates on the loan will now have decreased, as a spread over the Treasury yield of 2.87%, but this savings will instead be paid to BNP, keeping the effective interest rate of the loan as a spread over the strike yield of 3%.

## Appendix F: Financial Risk Measurements

To measure the basic risks of financial derivatives many analysts and traders use a variety of risk measurements collectively referred to as the “Greeks”, or risk sensitivities, which represent the sensitivities of derivatives to changes in underlying parameters the value of the derivative is dependent upon. The most common of these are delta ( $\Delta$ ), gamma ( $\Gamma$ ), and vega ( $v$ ). These measurements are vital tools in risk management as they isolate various risk components and allow a portfolio to be rebalanced accordingly to achieve a desired level of exposure. Used in the Black-Scholes model (see Appendix G), these Greeks are also very useful for derivative trading as a means to hedge against adverse market conditions.

Delta denotes the relationship between the price of a derivative and the price of its underlying asset. Most commonly used hedge derivative portfolios from changes in an underlying asset’s price, delta represents the rate of change of a derivative’s value with respect to changes in the underlying security’s price. Known as “delta-hedging,” a trader may seek to minimize the delta value of a portfolio of derivatives, insulating its value from the value of the underlying asset. Formulaically delta is the partial derivative of a portfolio of derivative securities  $V$  on a single underlying asset with respect to the asset’s price  $S$ . As a first-order derivative, delta does not account for the convexity of yield curves and is generally accurate only for small movements in the underlying asset’s price (Hull, 2009).

Gamma illustrates the rate of change of a portfolio’s delta as the underlying asset’s price changes and is the second derivative of a derivative portfolio’s value  $V$  with respect to the underlying asset’s price  $S$ . Gamma may also be represented as the partial derivative of a portfolio’s delta with respect to the underlying asset’s price  $S$ . Gamma’s importance as a second-order derivative is that it corrects for convexity. Traders who wish to establish an effective delta-hedge may attempt to also neutralize a portfolio’s gamma, making the hedge effective over a wider range of interest rate movements.

Although not an actual Greek letter, vega (represented by the Greek letter nu,  $v$ , and often replaced in academics by kappa,  $\kappa$ ) is a first-order derivative which measures a portfolio’s sensitivity to changes in volatility. Similar to delta, vega is the partial



derivative of a portfolio of derivative securities  $V$  with respect to the underlying asset's volatility  $\sigma$  and is accurate generally only for small changes in volatility (Hull, 2009).

	<u>Formula</u>
Delta	$\Delta = \delta V / \delta S$
Gamma	$\Gamma = \delta \Delta / \delta S = \delta^2 V / \delta S^2$
Vega	$v = \delta V / \delta \sigma$

**Table 11: Risk Sensitivity Formulas**

## Appendix G: The Black-Scholes Model

As a common application of the partial derivatives shown in Appendix H, we can represent our risk measurements using a transformation of Black-Scholes option-pricing notation. Developed by Fischer Black and Myron Scholes in their 1973 paper titled “The Pricing of Options and Corporate Liabilities”, the Black-Scholes model is based on the insight that options are implicitly priced if their underlying stock is traded. The parameters used in this model are defined below (Black & Scholes, 1973).

$\Phi$ : Standard Normal Probability Density Function

$\Phi$ : Standard Normal Cumulative Distribution Function

$\tau$ : Time to Maturity

$S$ : Spot Price

$K$ : Strike Price

$r$ : Risk-Free Rate

$\sigma$ : Implied Volatility

Here, the implied volatility of an option is the standard deviation of the underlying security’s logarithmic returns over a period of time  $P$ . The implied volatility can be calculated using the logarithmic returns of any time period and its standard deviation over the period  $\sigma_p$  using the following formula:

$$\sigma = \sigma_p / P^{1/2}$$

Additionally, the risk-free rate is often difficult to accurately define. In their paper, Black and Scholes explicitly assume the existence of a risk-free rate, and further assume this rate is known and constant (Black & Scholes, 1973). While the technical definition of “risk-free” is one of intense debate, US Treasury yield curves are commonly used in practice as a close approximation.

The Black-Scholes model relies on the equality of a partial differential equation (PDE) used to determine the price of a derivative. In the derivation of the Black-Scholes PDE, we create the variables  $d_1$  and  $d_2$  to simplify notation in the final risk sensitivity formulas. These variables are defined below.

$$d_1 = (\ln(S/K) + (r + \sigma^2/2)\tau) / (\sigma\tau^{1/2})$$

$$d_2 = (\ln(S/K) + (r - \sigma^2/2)\tau) / (\sigma\tau^{1/2})$$

Finally, Black and Scholes state several explicit assumptions their model relies on for a particular equity available on the market. These assumptions are listed below (Black & Scholes, 1973).

- It is possible to borrow and lend cash at a known, constant risk-free interest rate.
- The price follows a geometric Brownian motion with constant volatility and drift.
- There are no transaction costs.
- The stock does not pay a dividend.
- It is possible to buy any fraction of a share of a security.
- There are no restrictions on short selling.
- There is no opportunity for arbitrage.

While these assumptions vary from reality in a number of ways, only a few are significant. First, the Black-Scholes model does not properly account for extreme movements in stock prices and produces tail-risk, or the risk associated with highly improbable movements in stock prices. This tail risk can be partially hedged with out-of-the-money options. Examined by Nassim Taleb in his 2007 book *The Black Swan*, these highly improbable occurrences, referred to as “black swans”, can produce significant impacts if not accounted for (Taleb, 2007).

Additionally, the assumptions of cost-less trading, constant volatility and drift, and instantaneous and continuous trading can cause a variety of risks, including liquidity risk, volatility risk, and gap risk. While volatility and gap risk can be hedged with the

appropriate use of delta and vega, liquidity risk is difficult to hedge and traders should be cautious with relying too heavily on this assumption. Because of these variances, the Black-Scholes model is widely used as a useful approximation instead of an exact calculation, and should be treated as such. The table below shows the valuation and risk sensitivity formulas for a European call or put option (Black & Scholes, 1973).

<u>Risk Sensitivity</u>		<u>Call Option</u>	<u>Put Option</u>
Value	V	$S \Phi(d_1) - e^{-r\tau} K \Phi(d_2)$	$e^{-r\tau} K \Phi(-d_2) - S \Phi(-d_1)$
Delta	$\delta V / \delta S$	$\Phi(d_1)$	$\Phi(-d_1)$
Gamma	$\delta^2 V / \delta S^2$	$\varphi(d_1) / (S\sigma\tau^{1/2})$	
Vega	$\delta V / \delta \sigma$	$S\varphi(d_1)\tau^{1/2}$	

**Table 12: Black-Scholes European Option Formulas**

This model does not apply well to bond and commodity securities, however. Since the prices of many commodities are non-random, especially seasonal commodities such as agricultural products, their prices do not exhibit a geometric Brownian motion and the Black-Scholes model does not hold. Similarly, bonds have a definitive maturity date and the prices of these bonds drift towards par value as maturity approaches. Volatility then decreases accordingly and the volatility assumption fails. While several extensions of the model have been developed to deal with these “non-randomness” and “pull-to-par” problems, most rely on the Black model, often referred to as the Black-76 model, presented by Fischer Black in 1976.

The Black formula is similar to the Black-Scholes one in almost every way except the spot price  $S$  of the underlying stock is replaced by the undiscounted forward price  $F$  of the underlying bond or commodity. Instead of assuming that the spot price follows a log-normal process, we assume that the forward price at the maturity of the option is log-normally distributed (Black, 1976). We now can derive our formulas the same as we did in the Black-Scholes model by replacing the spot price with the forward price. During the

derivation of our Black-76 PDE we again create the variables  $d_1$  and  $d_2$  for notational purposes as shown below.

$$d_1 = (\ln(F/K) + (\sigma^2/2)\tau) / (\sigma\tau^{1/2})$$

$$d_2 = (\ln(F/K) - (\sigma^2/2)\tau) / (\sigma\tau^{1/2})$$

Noting that the same limitations that applied to the Black-Scholes model will apply to the Black-76 model, the value and risk sensitivity formulas are shown below (Black, 1976).

<u>Risk Sensitivity</u>		<u>Call Option</u>	<u>Put Option</u>
Value	V	$e^{-r\tau} (F\Phi(d_1) - K\Phi(d_2))$	$e^{-r\tau} (K\Phi(-d_2) - F\Phi(-d_1))$
Delta	$\delta V / \delta S$	$e^{-r\tau} \Phi(d_1)$	$e^{-r\tau} \Phi(-d_1)$
Gamma	$\delta^2 V / \delta S^2$	$e^{-r\tau} (\varphi(d_1) / (F\sigma\tau^{1/2}))$	
Vega	$\delta V / \delta \sigma$	$F e^{-r\tau} \varphi(d_1) \tau^{1/2}$	

**Table 13: Black-76 Bond Option Formulas**

## **Appendix H: Bean Stalk White Paper**

The Bean Stalk system consists of an application and an Excel spreadsheet. Bean Stalk is a Windows Console Application written in C# using Visual Studio 2008 and contains the majority of the logic that runs the system. The responsibilities of Bean Stalk are to obtain the list of jobs and deals, the details of the deals, then process the jobs. The list of jobs and deals is obtained by connecting to the Configuration Manager database. To maintain flexibility, when running Bean Stalk the user is able to input arguments that will override job specifications. This is particularly useful for testing and valuation because the user can specify to save the risk objects to any location they want while not uploading any results to MAD. The command line arguments are parsed by leading the value with the name of the parameter it is for and a colon. For example passing “JobName:PV\_Delta” would execute only the PV\_Delta job. If the user is interested in the results of this job from 3 days ago they could pass “JobName:PV\_Delta CalcDateOffset:-3”.

Bean Stalk is intended to be a prototype only and to provide groundwork to be expanded on in order to create a full functional system. The main limitation of Bean Stalk is that it currently only supports TLock trade types. TLocks are the most simplistic trades handled by Fahrenheit in terms of number of variables and therefore were chosen considering the limited time frame for the project. New types of trades can be added easily by creating a new template mapped to an XML deal and making some minor code changes. Bean Stalk needs new code for each new deal type primarily because of the MAD database schema. The table names follow a naming convention but are not predictable based on the information available to Bean Stalk.

Bean Stalk begins by creating objects to represent all of the jobs and deals. Creating jobs is fairly straightforward and uses a constructor. Deals however are more complicated to create so this responsibility has been delegated to a builder object which takes care of the creation process including querying databases, reading from multidimensional arrays, and massaging data to ensure it will be in the proper format to be used by Excel in calculations. The deal builder is also responsible for calling MadMolly and passing it the component number of the deal so that the deal details XML document can be created.

Once Bean Stalk has modeled the jobs and deals it begins to iterate through the list of jobs. The first step for each job is to construct a list of deals in that job. This is accomplished by matching certain criteria (currency, product, portfolio, and group) of the deal with the same criteria of the job. At this point Bean Stalk has the current job and list of deals associated with the job, it now imports this data into Excel. The job information is copied into the appropriate ranges on the Excel spreadsheet. Loading of deals is a more complicated process because it means adding a new worksheet for each deal.

Importing a deal into the spreadsheet begins by opening the Excel template required by this deal, the path to the template is a field of the deal object. The templates are special prepared Excel spreadsheets that have been mapped with an XML representation of the deal details. The mapping allows for the template to be opened then by running a macro an XML file is imported and the XML attributes are inserted into their corresponding mapped fields. The XML details for the deal are saved to a temporary file and then the macro is run to import this file. Once the template has been filled in it is copied to the Bean Stalk workbook. Bean Stalk limits the number of deals that can be added to a workbook to ensure that Excel does not crash because of lack of resources. If a job contains too many deals the job is divided up dynamically. With all of the deals loaded into the Bean Stalk spreadsheet the sheet is calculated. In the prototype stage this occurs on the same machine as Bean Stalk but this spreadsheet can be distributed in the future to allow for parallel processing. Following the calculation Excel is closed and Bean Stalk continues on to the next job in the queue.

### **MadMolly**

In order for Bean Stalk to value the deals it must obtain the details for the deal that are used in the calculations. This requires a system that can connect to the MAD database where the deals Bean Stalk processes are stored. Fahrenheit Sheet Builder was responsible for this in the previous system and utilized MadMax which is an Excel only means of connecting to MAD. Since Bean Stalk uses Excel for calculations only a new program was required to be written to access MAD. The program developed is called MadMolly, like Bean Stalk it is written in C#. MadMolly is responsible for connecting to MAD, querying the database for deal details, and creating and returning an XML

document representing a deal and its details. The MAD database is large and consists of approximately a thousand tables. Each deal requires information from at least four tables so inner joins were used to limit the number of queries required.

### **Bean Stalk Spreadsheet**

Bean Stalk uses a modified version of the Fahrenheit spreadsheet to handle the calculations. This spreadsheet has been stripped of the user interface previously present and the majority of its logic removed. Instead this spreadsheet is populated with the correct information by Bean Stalk at runtime. The spreadsheet still does contain the logic for distributing the results because placing this responsibility on Bean Stalk would require it to receive the results back from the spreadsheet then distribute them. In the future when these spreadsheets may be distributed for processing it will be more practical for the spreadsheets to handle the output.

### **MadMax**

MadMax, as mentioned before, is an Excel based solution for interacting with the MAD database. Unlike MadMolly which is read only MadMax can write to the database. This functionality is used by the Bean Stalk spreadsheet to commit the calculated values back to MAD so that they can be used by interested parties.

### **Microsoft SQL Server**

Microsoft SQL Server is used by Bean Stalk to store details of each deal that has been calculated. The database Bean contains a single table 'dealAudit' that contains the deal number, job name, calculation date, and the XML details used in the calculation. Storing this information allows for an auditor to track down what inputs were used to produce an output and use these details to recalculate values if necessary.

The address of the SQL server is NYCS00056046.

### **Generation and Validation of Risk Reports**



The Bean Stalk spreadsheet provides the desired values by creating risk reports which are a structure supplied by the Westminster library. The risk reports are passed values and create an object which is then queried to display the relevant data.

Validation of the risk reports generated by Bean Stalked can be performed using the spreadsheet RiskReportQueries which has a worksheet for each TLock job. The spreadsheet has queries of official risk reports generated by Fahrenheit and a matching query that can be pointed to risk reports generated by Bean Stalk.

Parameter	Function (Cast to)
JobName:	Overrides the JobName range (String)
IsTest:	Overrides the IsTest range (bool)
CalcDateOffset:	Overrides the Calc Date Offset value for the job (int)
RampSet:	Overrides the RampSet range (String)
RampDateOffset:	Overrides the Ramp Date Offset value for the job (int)
ExportToMAD:	Overrides the ExportPVsToMAD range (bool)
OutputPath:	Overrides the OutputPath1 value for the job and sets all other OutputPath ranges to blank (String)

**Table 14: Bean Stalk Parameters**